Michael Janczyk
Dirk von Suchodoletz
Bernd Wiebelt
(Hrsg.)

Proceedings of the 5th bwHPC Symposium

HPC Activities in Baden-Württemberg
Freiburg – September 2018

PROCEEDINGS OF THE 5TH BWHPC SYMPOSIUM

Michael Janczyk
Dirk von Suchodoletz
Bernd Wiebelt
(Hrsg.)

PROCEEDINGS OF THE 5TH BWHPC SYMPOSIUM

HPC Activities in Baden-Württemberg Freiburg – September 2018

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

TÜBINGEN
LIBRARY PUBLISHING

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie, detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Die Online-Version dieser Publikation ist auf den Verlagswebseiten von Tübingen Library Publishing frei verfügbar (Open Access).
http://hdl.handle.net/10900/87676
http://nbn-resolving.de/urn:nbn:de:bsz:21-dspace-876765
http://dx.doi.org/10.15496/publikation-29062

1. Auflage 2019 Tübingen Library Publishing
Universitätsbibliothek Tübingen
Wilhelmstraße 32
72074 Tübingen
druckdienste@ub.uni-tuebingen.de
https://tlp.uni-tuebingen.de

ISBN (Softcover): 978-3-946552-26-0
ISBN (PDF): 978-3-946552-25-3

Umschlaggestaltung: Sandra Binder, Universität Tübingen
Coverfoto: pixabay
Satz: Michael Janczyk
Druck und Bindung: Pro BUSINESS digital printing Deutschland GmbH
Printed in Germany

DOI: 10.15496/publikation-29062

Vorwort

Die nun fünfte Ausgabe des bwHPC-Symposiums zeigt, dass sich die Veranstaltung fest etabliert hat. Sie ist Plattform für den Austausch von Wissenschaftlerinnen und Wissenschaftlern mit den Betreibern der großen, föderierten Forschungsinfrastrukturen. Das Symposium versucht, eine Brücke zwischen den verschiedenen Fachdisziplinen und den technisch-organisatorischen Aspekten zu schlagen. Es ergänzt die etablierte Governance aus Landesnutzerausschuss und den lokalen Gremien. Dem diesjährigen Symposium fiel die zusätzliche Rolle zu, die erste Projektphase von bwHPC-S5 einzuleiten, nachdem das fünfjährige Vorgängerprojekt bwHPC-C5 abgeschlossen wurde.[1]

Erfolgreiche Wissenschaft benötigt leistungsfähige Infrastrukturen: Forschung bildet eine zentrale Säule im Selbstverständnis moderner Gesellschaften. Digitalisierte Arbeitsprozesse prägen alle Wissenschaftsdisziplinen. Die Stärke eines Wissenschaftsstandorts leitet sich wesentlich von der Verfügbarkeit attraktiver, integrierter und skalierender Forschungsinfrastrukturen ab. Die Computational Science und damit HPC-Systeme als deren technisches Fundament gewinnen unablässig an Bedeutung. Deshalb besteht eines der Ziele der im Juli 2018 gestarteten bwHPC-S5 Begleitaktivitäten darin, die Forscher*innen in ihrer Arbeit mit den leistungsfähigen Infrastrukturen umfassend zu unterstützen, ohne sie jedoch aus der individuellen Verantwortung für ihre Daten und wissenschaftlichen Workflows zu entlassen.

Der Erkenntnis folgend, dass heutige Anforderungen nicht mehr sinnvoll von einzelnen Universitäten oder Forschungsinstitutionen bedient werden können, koordinieren die wissenschaftlichen Rechenzentren des Landes Baden-Württemberg ihre Aktivitäten im Bereich HPC und der Speicherung großer Datenmengen unter

[1] Vgl. hierzu `http://www.bwhpc.de/projektaufgaben.php`

dem gemeinsamen Dach des »Umsetzungskonzept(es) der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM)«. Damit baut das Land ein wesentliches Alleinstellungsmerkmal bei der Unterstützung der Wissenschaften aus und strebt eine Vorreiterrolle bei der Etablierung von Nationalen Forschungsdateninfrastrukturen (NFDI) und Science Data Centern an.

Der föderierte Betrieb großer Forschungsinfrastrukturen verändert das Umfeld und die Anforderungen der beteiligten Universitätsrechenzentren. Sie müssen sich verstärkt abstimmen und gemeinsame Lösungen finden. Nutzer der Infrastrukturen kommen nun nicht mehr allein aus der eigenen Einrichtung, sondern aus dem ganzen Land. Umgekehrt müssen Forscher*innen der eigenen Universität dabei unterstützt werden, Ressourcen an anderen Standorten zu verwenden. Leistungserbringung, Kosten und Zugehörigkeit zu einer Forschungseinrichtung fallen nicht mehr zwangsläufig zusammen, was neue Herausforderungen in der Leistungsverrechnung und Finanzierung mit sich bringt.

Mit der bereits vor mehr als zehn Jahren mit dem bwGRiD gestarteten HPC-Initiative des Landes wurde eine neue Kultur der Bereitstellung und Nutzung großer Forschungsinfrastrukturen eingeleitet. Durch die Spezialisierung der HPC-Systeme und die Bündelung gemeinsamer Anforderungen der verschiedenen Forschungsgruppen wurden Anstrengungen und Anschaffungen zusammengeführt. Diese nutzt nicht nur Economies-of-Scale, die Erkenntnis, dass der Aufwand für das Design, die Beschaffung und den Betrieb eines Clusters mit zunehmender Größe fast nicht mehr ansteigt. Es erlaubt ebenfalls eine deutlich effizientere Auslastung der Ressourcen, da üblicherweise in Einzelsystemen anfallende Leerlaufzeiten durch andere Nutzer gefüllt werden. Damit entstehen klare Mehrwerte, die es erlauben, einige deutlich größere Systeme mit hoher Leistung zu gleichen Kosten wie viele kleine mit deutlich geringerer Gesamtleistung zu beschaffen. Dabei erlauben es Mechanismen wie »Fairshare« Nutzungsanteile am Gesamtsystem in beliebiger Stückelung herauszugeben. Dieses senkt die Einstiegsschwelle insbesondere für kleinere Gruppen und Nachwuchsforscher*innen. Für große Gruppen und traditionelle Großantragsteller besteht der Mehrwert in einem zügigen Start ihrer Forschung, einer potenziell noch größeren Ressource als in einer individuellen Beschaffung ohne die Nachteile der Aufwendungen für eine eigene Betriebsmannschaft. Durch eine gestaffelte Beschaffung und regelmäßige Erneuerungszyklen bleibt die Attraktivität der Ressourcen langfristig erhalten, und die Beteiligung neuer Nutzergruppen kann sichergestellt werden. Das verändert die Landschaft großer Forschungsinfrastrukturen nachhaltig.

Die im »High Performance Computing« und »High Throughput Computing« zunehmend genutzten großen Datenmengen in stärker föderierten Forschungsinfrastrukturen erfordern geeignete Strategien für ihre Haltung und ihren Zugriff, da selbst bei hochperformanten Netzwerken das Kopieren von Datensätzen immer weniger eine effiziente Lösung darstellt. Im Zuge der Entstehung, Verarbeitung und nachhaltiger Langzeitlagerung und Nachnutzung ändern sich die Anforderungen an das Datenmanagement bezüglich Zugriffsrechten, Leistungserwartungen und Absicherungen im Zeitablauf. Hierzu zählt die zunehmend komplexe Datenhaltung, die von der Generierung der Daten in Messinstrumenten, Simulationen und vielfältigen weiteren Verfahren über das Pre- und Postprocessing bis zur Verarbeitung und Visualisierung in Spezialsystemen und der langfristigen Haltung und Publikation reicht. Im Lebenszyklus von Forschungsdaten entstehen bei den Forschenden verschiedenartige Anforderungen, die von der schnellen Speicherung bei der Datenerhebung, über die Verarbeitung in HPC- und Cloudsystemen bis hin zur Visualisierung reichen. Hinzu kommt die notwendige Aufarbeitung der Daten im Sinne »guter wissenschaftlicher Praxis«, deren Langzeitspeicherung und -verfügbarkeit für Publikation und Nachnutzung. Das Forschungsdatenmanagement (FDM), die nachhaltige und zukunftsorientierte Speicherung von Forschungsdaten, ihre Verfügbarmachung und optimalerweise Publikation, rückt damit in den Fokus moderner Forschungsprozesse. Nachnutzung und Reproduzierbarkeit von Forschungsergebnissen benötigen sowohl Verständnis und Bereitschaft der Forscher*innen, als auch neuartige Dienstleistungen seitens der zentralen Infrastrukturanbieter.

Langfristig müssen geeignete Maßnahmen und Workflows umgesetzt werden, um den Wissenschaftler*innen nicht nur einen Großteil der Datenmanagementaufgaben abzunehmen, sondern auch die Effektivität der Ressourcennutzung zu steigern. Dies ist insbesondere bei der Nutzung von HPC-Ressourcen wünschenswert, indem der dort stattfindende Batch-Betrieb durch die automatische Versorgung mit Daten optimiert wird. Neben weiteren »lokalen« Softwarekomponenten, wie beispielsweise Versionsmanagement, ist auch der Anschluss an, beziehungsweise die einfache Nutzbarkeit bestehender externer Forschungsdateninfrastrukturen und der zukünftigen Nationalen Forschungsdateninfrastruktur notwendig, um Wissenschaftlern die Nutzung der fachspezifischen Repositorien zu ermöglichen. Die Aufgabe der Betreiber und des bwHPC-S5-Projekts besteht demnach zusätzlich darin, beständig neue Lösungen zu suchen und die föderativ angelegten Forschungsinfrastrukturen im Sinne der Forschenden beständig weiterzuentwickeln. Die Anwendung von Virtualisierungs- und Containerisierungsstrategien ist nur ein Beispiel hierfür.

Der Tagungsband besteht aus drei Teilen, wobei im ersten Teil die konzeptionellen und projektbezogenen Darstellungen im Vordergrund stehen. Der zweite Teil beinhaltet die durch ein Peer-Review begutachteten Beiträge der Forscher*innen, die eine Verbindung von Forschungsfragestellungen mit konkreten Berechnungen auf den jeweiligen bwHPC-Systemen herstellen. Im dritten Teil finden sich die durch die Projektpartner begutachteten Texte zu administrativen und betrieblichen Fragestellungen und Überlegungen zur technischen Weiterentwicklung der Betriebsmodelle.

Die in diesem Tagungsband zusammengefassten Konzepte und Aktivitäten beziehen zu großen Teilen ihre Kraft aus der langfristig angelegten Unterstützung des Ministeriums für Wissenschaft, Forschung und Kultur in Baden-Württemberg. Hierzu zählen zuvorderst die finanziellen Beiträge für die Beschaffung und regelmäßige Erneuerung der HPC-Systeme als auch die Förderung der bwHPC-C5- und bwHPC-S5-Begleitprojekte. Weiterhin profitieren die beteiligten Forschungseinrichtungen durch die eScience-Initiative des Landes, durch die beispielsweise das ViCE-Projekt seine Unterstützung erfuhr.

Die Herausgeber
Freiburg, April 2019

Contents

doi: 10.15496/publikation-29062

Grußwort der Ministerin für Wissenschaft, Forschung und Kunst Baden-Württemberg

Der digitale Wandel hat die Art und Weise, wie geforscht und wissenschaftlich gearbeitet wird, grundlegend verändert. Besonders deutlich wird dies bei heutigen Hochleistungsrechnern. Mittels Simulationsverfahren ermöglichen diese Supercomputer Forschung an den Grenzen der Erkenntnis und setzen entscheidende Impulse für Entwicklungen in vielen verschiedenen Fachdisziplinen. Heutzutage ist Spitzenforschung, speziell in den Natur- und Ingenieurwissenschaften, ohne Simulationsverfahren auf Hochleistungsrechnern schwer vorstellbar. Der Bedarf an immer schnelleren Rechnern scheint unaufhaltsam. Aber nicht nur die Nachfrage nach hoher Rechenleistung steigt: Auch wegen der Digitalisierung und durch die Entwicklung neuer wissenschaftlicher Methoden werden immer mehr Forschungsdaten generiert. Denn im Grunde ist jede wissenschaftliche Disziplin datengetrieben und trägt zu einem exponentiellen Wachstum digital zu speichernder Datenmengen bei, was wiederum eine entsprechende Speicherinfrastruktur erfordert.

Im Forschungsalltag werden nicht selten im selben Workflow sowohl Rechen- als auch Speichersysteme für die Verarbeitung großer Datenmengen benötigt. Diesem Umstand trägt das Land Baden-Württemberg mittels der Landesstrategie zu High Performance Computing (HPC) und Data Intensive Computing (DIC) Rechnung. Durch die Verzahnung von rechenintensiver mit datenintensiver Forschungsinfrastruktur entstehen Synergien, die eine neue Qualität von wissenschaftlichem Arbeiten auf höchstem Niveau ermöglichen.

Die weiter voranzutreibende Verzahnung bewertet die renommierte Deutsche Forschungsgemeinschaft (DFG) als folgerichtigen Schritt, um das hohe Niveau der digitalen Forschungsinfrastruktur an den baden-württembergischen Hochschulen aufrecht zu erhalten und ausbauen zu können. Die DFG bescheinigt der baden-württembergischen Landesstrategie »einen wegweisenden Modellcharakter« mit Vorbildcharakter für andere Länder. Diese Einschätzung bestätigt eindrucksvoll den Weg, den das Land mit der Landesstrategie eingeschlagen hat.

Der Wissenschafts- und Wirtschaftsstandort Baden-Württemberg ist im nationalen und internationalen Wettbewerb hervorragend aufgestellt. Zugleich ist es Anspruch des Landes, sich in der Wissenschaft und Wirtschaft auch zukünftig bestmöglich zu positionieren. Dies bedeutet, auch weiterhin optimale Rahmenbedingungen für eine exzellente, international konkurrenzfähige Hochschul- und Forschungslandschaft sowie eine innovationskräftige Wirtschaft zu gewährleisten. Hierfür bedarf es einer fortlaufenden Investition in eine erstklassige digitale Forschungsinfrastruktur, um bei der rasanten Entwicklung führend zu bleiben. Mit der HPC/DIC-Landesstrategie für die Jahre 2017–2024 und der damit verbundenen Investition von einer halben Milliarde Euro in Supercomputer und die digitale Infrastruktur hat das Land Baden-Württemberg hierfür die Weichen gestellt.

Zu betonen ist, dass die Landesstrategie allen Wissenschaften dient. Die Wissenschaftlerinnen und Wissenschaftler aller Fächer im Land sollen bestmöglich unterstützt werden, um innovativ und exzellent forschen zu können. Die digitale Forschungsinfrastruktur ist ein Werkzeug, um als oberstes Ziel bahnbrechende Forschungsergebnisse zu ermöglichen und hierdurch Antworten auf wichtige gesellschaftliche Herausforderungen zu finden. Unter den Themen wie Energiewende, neuer Mobilität, Gesundheitsforschung oder gesellschaftlichem Zusammenhalt gibt es große Fragen, denen wir uns für eine gute Zukunft stellen müssen.

Großes zu erreichen gelingt selten alleine. Bei den heutigen Anforderungen in der Wissenschaft ist es für ein einzelnes Rechenzentrum schwer möglich, allen Nutzerbedarfen erstklassig gerecht zu werden. Die bestmögliche Nutzerunterstützung von Wissenschaftlerinnen und Wissenschaftlern im Land legt eine kooperative Erbringung von Dienstleistungen nahe. Die vielen bwProjekte und Landesdienste sind Ausdruck davon, dass die Hochschulrechenzentren bereits seit vielen Jahren einen erfolgreichen kooperativen Weg eingeschlagen haben. So lassen sich beispielsweise bei begrenzten Ressourcen durch Kooperationen Skaleneffekte nutzen, kosteneffizienter vorhandene Dienste verbessern und neue Dienste entwickeln. Die zwischen den wissenschaftlichen Rechenzentren des Landes entstandene Kooperationskultur ist eine Erfolgsgeschichte. Eine solche Kooperationskultur entwickelt sich nicht von allein, sondern ist dem initiativen, tatkräftigen Handeln vieler Menschen zu verdanken.

An dieser Stelle möchte ich die Gelegenheit nutzen, allen Beteiligten für ihr Engagement zu danken. Und in Kenntnis der Vorerfahrung bin ich mir sicher, dass durch gemeinschaftliches Agieren in und im Umfeld der Hochschulrechenzentren des Landes auch zukünftige Herausforderungen gemeistert, neue Visionen entwickelt und umgesetzt werden. Schließlich wird ein technologischer Wandel auch in Zukunft unser ständiger Begleiter sein.

Theresia Bauer MdL
Ministerin für Wissenschaft, Forschung
und Kunst des Landes Baden-Württemberg

doi: 10.15496/publikation-29062

Grußwort des Prorektors für Forschung der Universität Freiburg anlässlich des 5. bwHPC-Symposiums

Sehr geehrter Herr Ministerialrat Castellaz,
liebe Kolleginnen und Kollegen,
meine sehr verehrten Damen und Herren,

als Vizerektor der Albert-Ludwigs-Universität möchte ich Sie herzlich zum fünften bwHPC-Symposium hier in Freiburg begrüßen.

Dass sich bei uns – als altehrwürdige Volluniversität – Tradition und Moderne nicht ausschließen, können Sie bereits an der Ausstattung dieses Hörsaals sehen. Die Universität ist sehr stolz darauf, ein lebendiger Teil – und mit unserem bwForCluster NEMO sogar ein Standort – der bwHPC-Initiative zu sein.

Wir sind ein überzeugter Verfechter der Landesinitiative bwHPC, welche auf Kooperation und verteilten Kompetenzen beruht. So beteiligen sich Forschende aus Freiburg nicht nur an der Erweiterung des bwForCluster NEMO, sondern rechnen und investieren auch an anderen, ihren jeweiligen Fachrichtungen dedizierten Standorten. Ein Vorzeigebeispiel hierfür im Fall von Freiburg ist die Beteiligung von Prof. Michael Thoss am bwForCluster JUSTUS 2 in Ulm.

Das Kabinett hat nun entschieden, dieses Landeskonzept bis 2024 fortgeschrieben und damit nicht nur die Richtigkeit der Ausrichtung von Kompetenz und Investition bestätigt, sondern auch Planungssicherheit gewährleistet. Andere Bundesländer schauen neidvoll auf diese Konstellation.

Die Universität profitiert natürlich von der Existenz einer Forschungsinfrastruktur wie NEMO. Sie hat eine konsolidierende Wirkung, indem sie einen »Nukleus«, beziehungsweise eine kritische Masse darstellt, an die lokale Erweiterungen »andocken« können. Es muss nicht jedes Mal neu in die Erfindung des Rades investiert werden und es muss nicht in jeder Besenkammer ein Serverraum eingerichtet werden. Für die Forschenden entfällt die Notwendigkeit eines langen Beschaffungsprozesses. Sie können sofort loslegen und sich im Ausgleich an einer Erweiterung oder Erneuerung der Forschungsinfrastruktur beteiligen.

Eine solche Forschungsinfrastruktur hat auch Folgen. Neben der reinen Rechenzeit, was bisher für die Wissenschaft das Non-plus-ultra war, treten die Analyse von großen Datenbeständen und die Nachhaltigkeit bei der Datenaufbewahrung in den Fokus. Die Forderung der DFG nach zehn Jahren Verfügbarkeit ist hier nur der Anfang.

In Zukunft werden wir nicht nur die Daten, sondern auch die Methoden in funktionaler Form, beispielsweise als »Virtualisierte Forschungsumgebungen« aufbewahren müssen. Dahinter steckt die Erkenntnis, dass nicht nur Daten, sondern auch Software und Laufzeitumgebungen dem technischen Wandel unterworfen sind und somit verloren gehen können. Damit ist die wissenschaftliche Überprüfbarkeit gefährdet.

Bereits in der Vergangenheit wurden neue Erkenntnisse aus alten, längst abgeschriebenen Daten gewonnen, wie beispielsweise in der Klimaforschung oder der Astronomie.

Insofern ist es nicht überraschend, dass in Freiburg, ergänzend zum bwForCluster NEMO auch eine weitere große Forschungsinfrastruktur in Form eines »Datengrabes« entsteht.

Damit das zukünftige »Storage for Science« nicht das sprichwörtliche »Datengrab« wird, muss in enger Absprache zwischen den beteiligten Forschenden und den Bereitstellern der Infrastruktur ein tragfähiges und nachhaltiges Konzept zum Forschungsdatenmanagement entwickelt werden. Dies ist nicht alleine eine technische, sondern auch eine wissenschaftliche und organisatorische Herausforderung.

Nachdem dieses Zusammenspiel bereits für den Bereich des wissenschaftlichen Hochleistungsrechnens auf dem bwForCluster NEMO ausgezeichnet funktioniert hat, sind wir zuversichtlich, diese Herausforderung auch im Forschungsdatenmanagement anzunehmen und eine vorbildhafte Lösung zu entwickeln.

Sie sehen, es geht hier nicht nur ums kalte Blech. Ohne das entsprechende Personal, das dieses Blech zum Leben erweckt und mit wegweisenden Konzepten füllt, wäre das eine tote Investition. Diese Konzepte zu entwickeln und die Nutzer auf die aufregende Reise mitzunehmen wäre von den Universitäten aufgrund des damit verbundenen hohen Aufwandes nicht alleine zu stemmen. Aus diesem Grund sind wir sehr froh, dass das MWK dieses Vorhaben mit Begleitprojekten unterstützt.

Dazu zählt an vorderster Stelle das kürzlich beendete Projekt bwHPC-C5 zur Unterstützung der Nutzer von Hochleistungsrechenressourcen, über dessen wissenschaftliche Ergebnisse heute im Rahmen des Symposiums berichtet wird. Das neue Projekt bwHPC-S5, dessen Startschuss auf diesem Symposium offiziell fällt, setzt das abgeschlossene Projekt inhaltlich fort und erweitert dabei den Fokus auf die vorher angesprochene Problematik der nachhaltigen Datenhaltung.

Es bedarf nicht großer Weitsicht anzunehmen, dass die Verarbeitung der in Zukunft anfallenden enormen Datenmengen – nicht nur auf HPC-Systemen – weitere spannende wissenschaftliche Fragestellungen generieren wird.

So steckt beispielsweise die »richtige« Governance der Datenaufbewahrung noch in den Kinderschuhen. Es ist weder zufriedenstellend geklärt, wem die Daten langfristig gehören, noch welche Metadaten bereits bei der Erhebung verpflichtend gemacht werden müssen. Die Aufbereitung, das Verwalten und Vorhalten von Forschungsergebnissen sowie qualitätssichernde Maßnahmen in der Behandlung von Forschungsdaten sind eine nicht zu unterschätzende Herausforderung der einzelnen Wissenschaftlerinnen und Wissenschaftler wie auch der Forschungseinrichtungen.

Das Beispiel »Elsevier« zeigt, dass wir die Fehler aus der Vergangenheit nicht wiederholen dürfen. Denn dieser Konzern hat bereits vor der Wissenschaft erkannt, dass der Besitz von Daten und Information einen erheblichen Wert darstellt und eine Monopolstellung begründen kann. Open Science und Open Data sind uns daher eine Verpflichtung. Das Symposium ist hier eine ideale Plattform, um diese Themen zwischen den Forschenden und den Betreibern zu diskutieren.

Als Ergänzung zu den klassischen Strukturen (Universitätsbibliothek und Wissenschaftliches Rechenzentrum) wird in zukünftigen »Science Data Centers« nicht nur die eigentliche Datenhaltung, sondern insbesondere auch die Sicherung des langfristigen funktionalen Zugriffs eine wesentliche Rolle spielen.

Das MWK hat diese Entwicklung frühzeitig erkannt und entsprechend mit Projekten gefördert. Hier zu nennen sind an erster Stelle ViCE (»Virtual Open Science Collaboration Environment«), CiTAR (»Citing and Archiving Research«) und

SARA (»Software Archiving of Research Artefacts«). Die Cloud-Computing-Aktivitäten des Landes komplettieren diese Sichtweise, indem sie die Verarbeitung von großen wissenschaftlichen Datenmengen bereits jetzt in flexiblerer Weise auch für Nicht-HPC-Nutzer ermöglichen. Damit kann zukünftig auch ein langfristiger funktionaler Zugriff auf Forschungsdaten gewährleistet werden.

Es ist klar, dass solche Herkulesaufgaben nicht von einer Universität alleine bewältigt werden können. Kooperationen sind auch hier die richtige Antwort. Keines der vom MWK geförderten bwProjekte wird von einer Hochschule alleine durchgeführt. Dieses Zusammenspiel zeigt sich ganz konkret inzwischen auch bei Beschaffung und Betrieb von großen Forschungsdateninfrastrukturen. So wird das zukünftige Storagesystem bwSfS (»Storage for Science«) gemeinsam von Tübingen und Freiburg beschafft und geo-redundant betrieben.

Ich wünsche Ihnen allen eine spannende und arbeitsintensive Tagung und bedanke mich für die Aufmerksamkeit.

Gunther Neuhaus
Vizerektor/Prorektor für Forschung
Albert-Ludwigs-Universität Freiburg

I Konzepte und Projekte/Concepts and Projects

Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM)

Gerhard Schneider Vincent Heuveline Karl-Wilhelm Horstmann
Bernhard Neumair Petra Hätscher Josef Kolbitsch Simone Rehm
Michael Resch Thomas Walter Stefan Wesner Peter Castellaz

Das erneuerte Umsetzungskonzept für High Performance Computing wurde um die Bereiche Data Intensive Computing und Large Scale Scientific Data Management erweitert. Hiermit wird eine Erfolgsgeschichte der föderativen Kollaboration der Universitäten des Landes Baden-Württemberg fortgeschrieben. Das neue Konzept adressiert Herausforderungen die sich aus der Digitalisierung der Wissenschaft ergeben. Besonderer Fokus wird auf das nachhaltige Forschungsdatenmanagement gelegt, um den Umgang mit immer größeren Datenmengen zu unterstützen und ihre Nachnutzbarkeit dauerhaft sicherzustellen. Erklärtes Ziel ist die Etablierung einer landesweiten Datenföderation, um die anstehenden Herausforderungen gemeinsam und kollaborativ adressieren zu können. Komplementiert werden die geplanten Aktivitäten durch einen Erweiterung der Support-Strukturen zur Unterstützung bei Datenhaltungsfragen und einer angepassten Governance, unter Beteiligung von Betreibern und Nutzern.

DOI: 10.15496/publikation-29040

1 Motivation

Computational Sciences (Wissenschaftsrat, 2014) und damit die HPC-Systeme als ihr technisches Fundament gewinnen unablässig an Bedeutung, wie auch der Wissenschaftsrat in seinen jüngsten Empfehlungen zur »Finanzierung des Nationalen Hoch- und Höchstleistungsrechnens in Deutschland« (Wissenschaftsrat, 2015) betont. Die fortschreitende Digitalisierung der Wissenschaft generiert auf Basis verschiedener Forschungsinfrastrukturen Forschungsdaten und damit Anforderungen, die von der schnellen Speicherung bei der Datenerhebung, über die Verarbeitung in HPC- und Cloudsystemen bis hin zur notwendigen Aufarbeitung der Daten im Sinne »guter wissenschaftlicher Praxis« reichen. Die Analyse dieser großen Datenmengen zur Gewinnung von neuen Erkenntnissen wird Data Intensive Computing (DIC) genannt – sie wird heute neben Theorie, Experiment und Simulation als vierte Säule der Wissenschaft bezeichnet. Hinzu kommen die notwendigen technischen und organisatorischen Maßnahmen für eine nachhaltige Nutzung der Daten, die eine langfristige Speicherung und eine nach Möglichkeit öffentliche Zugänglichkeit garantieren.

Der Erkenntnis folgend, dass diese neuen Anforderungen nicht mehr allein und von unabhängig agierenden Akteuren bedient werden können, koordinieren die wissenschaftlichen Rechenzentren des Landes Baden-Württemberg ihre Aktivitäten diesbezüglich. Gleichzeitig wollen die Landesuniversitäten den Empfehlungen des Rats für Informationsinfrastrukturen (RfII) folgen und ihre Infrastrukturentwicklungen mit dem Aufbau einer Infrastruktur für Forschungsdatenmanagement (FDM) auf Basis ihrer HPC- und DATA-Konzepte verschränken. Kooperative Lösungen helfen, die beschriebenen Herausforderungen zu bewältigen, und versprechen einen institutionen- und disziplinübergreifenden Mehrwert.

Für den Zeitraum von 2018 bis 2024 ist es das Ziel aller beteiligten Akteure, den beschrittenen Weg der Kooperation gemäß der HPC-Landesstrategie (MWK-BW, 2017) weiter zu verfolgen. Damit baut das Land Baden-Württemberg ein wesentliches Alleinstellungsmerkmal bei der Unterstützung der Wissenschaften aus und bekundet ausdrücklich das Interesse und die Bereitschaft, in einer frühen Phase beim Aufbau und der Entwicklung der Nationalen Forschungsdateninfrastruktur (NFDI) mitzuwirken. Im Sinne eines integrierten Ansatzes werden die bestehenden Konzepte für HPC, DIC und LS^{2}DM weiterentwickelt und in einer gemeinsamen Strategie zusammengeführt. Gleichzeitig werden die Grundlagen für eine frühe Beteiligung am Aufbau einer NFDI geschaffen und erforderliche Infrastrukturen bereitgestellt.

2 Ziele

Die wesentlichen Ziele der geplanten Weiterentwicklung sind die Festigung der für die Forschung und Wissenschaft in Baden-Württemberg bereitgestellten Dienste und Infrastrukturen in den Bereichen HPC, DIC und LS^{2}DM und aktive Unterstützung bei Entwicklung und Standardisierung einer NFDI. Weiterhin soll die Einbettung der Landesstrategie HPC, DIC und LS^{2}DM in den bundesweiten und europäischen Kontext erfolgen, sowie die Bündelung und Integration der individuellen Aktivitäten in diesem Themenbereich aus den einzelnen wissenschaftlichen Communitys vorangetrieben werden. Ebenfalls geplant sind eine Anpassung der Kooperations-, Governance- und Support-Strukturen sowie der Ausbau und die Schärfung der fachlichen Ausdifferenzierung für qualitativ hochwertige und fundierte Nutzerunterstützung. Das Konzept konkretisiert gleichzeitig das Rahmenkonzept bwDATA (Schneider, Heuveline, Horstmann, Neumair, Waldvogel u. a., 2018) bei der Umsetzung der darin unter »Große wissenschaftliche Daten und Kopplung bwDATA und bwHPC« beschriebenen Vorhaben mit der Erarbeitung der notwendigen konkreten Schritte zurVerknüpfung von HPC- und Forschungsdatenzentren im Land.

Daraus ergeben sich Maßnahmen, die von der Erweiterung der Aufgabenbereiche der Kompetenzzentren um Fragen der Datenhaltung, der Einrichtung weiterer Kompetenzzentren, den bedarfsgerechter Ausbau und Erneuerung der HPC-Systeme aller drei Ebenen und Datenmanagementsysteme, bis hin zur Entwicklung einer landesweiten Datenföderation reichen. Gleichzeitig erfolgt die Fortführung des Begleitprojektes (Barthel u. a., 2019) mit angepassten Schwerpunkten und Ausstattung im Hinblick auf die neuen Anforderungen einer Integration von HPC, DIC und LS^{2}DM.

3 Entwicklung der HPC-Systeme der Ebenen 1 und 2

Das HLRS wird die Weiterführung des baden-württembergischen Engagements für Tier-0 und Tier-1 in GCS und PRACE verfolgen und daher mit seiner Fokussierung auf Ingenieurwissenschaften eine im Rahmen der deutschen Profilbildung optimale

Architektur für die Forschungsgebiete Mobilität, Energie, Umwelt und Gesundheit umsetzen, um in diesem Bereich seine europaweit führende Stellung auszubauen. Die fortschreitende Einbindung der Simulation in den wissenschaftlichen Forschungsprozess wird darüber hinaus eine entsprechende Integration des eigentlichen Supercomputers in unterstützende Systeme der Datenhaltung und Visualisierung erfordern.

Das SCC übernimmt die Weiterentwicklung der Tier-2-Versorgung und die Unterstützung von Data Analytics. Die anwendungswissenschaftlichen und methodenwissenschaftlichen Schwerpunkte (Materialwissenschaften und Werkstofftechnik, Erdsystem- und Umweltwissenschaften sowie Energie- und Mobilitätsforschung, datenintensives Rechnen) orientieren sich eng an der Profilierung des SCC innerhalb der Gauß-Allianz.

4 Entwicklung der HPC-Systeme der Ebene 3

Die Weiterentwicklung der bwForCluster bedingt bei einer Erneuerung der Systeme die disziplinspezifische Ausrichtung sowohl auf Tier-3 Capacity Computing als auch auf Data Intensive Computing. Damit verbunden ist die Integration der Tier-3-HPC-Systeme mit den angeschlossenen Datenmanagement-Systemen für ihre jeweilige Community und die Anbindung an die HPC-Ressourcen auf Tier-2 und Tier-0/1.

Die bereits praktizierte enge Abstimmung mit den Fachwissenschaften und deren Vertretungen, insbesondere dem Landesnutzerausschuss (LNA-BW) sowie dem Arbeitskreis der Leiter der wissenschaftlichen Rechenzentren in Baden-Württemberg (ALWR-BW) wird dazu fortgeführt.

Die Zuordnung der Fachdisziplinen zu den bwForClustern auf Tier-3 inklusive der zugeordneten speziell ausgeprägten Datenmanagementsysteme soll kontinuierlich weiterentwickelt werden, um weitere Fachdisziplinen und Wissenschaftler und Wissenschaftlerinnen an die Möglichkeiten des Capacity Computing und Data Intensive Computing heranzuführen und bei der Nutzung unterstützen zu können.

Auf Tier-3 soll die fachliche Zuordnung künftig im Detail auf Basis der DFG-Fachsystematik erfolgen (Abbildung 1). Insbesondere bei Erweiterung und Erneuerung der bwForCluster erfolgt eine die bisherige Nutzung der Systeme berücksichtigende Anpassung der fachlichen Widmung. Die Ingenieurwissenschaften nehmen in dieser Zuordnung aufgrund der in diesem Bereich langjährigen bestehenden Er-

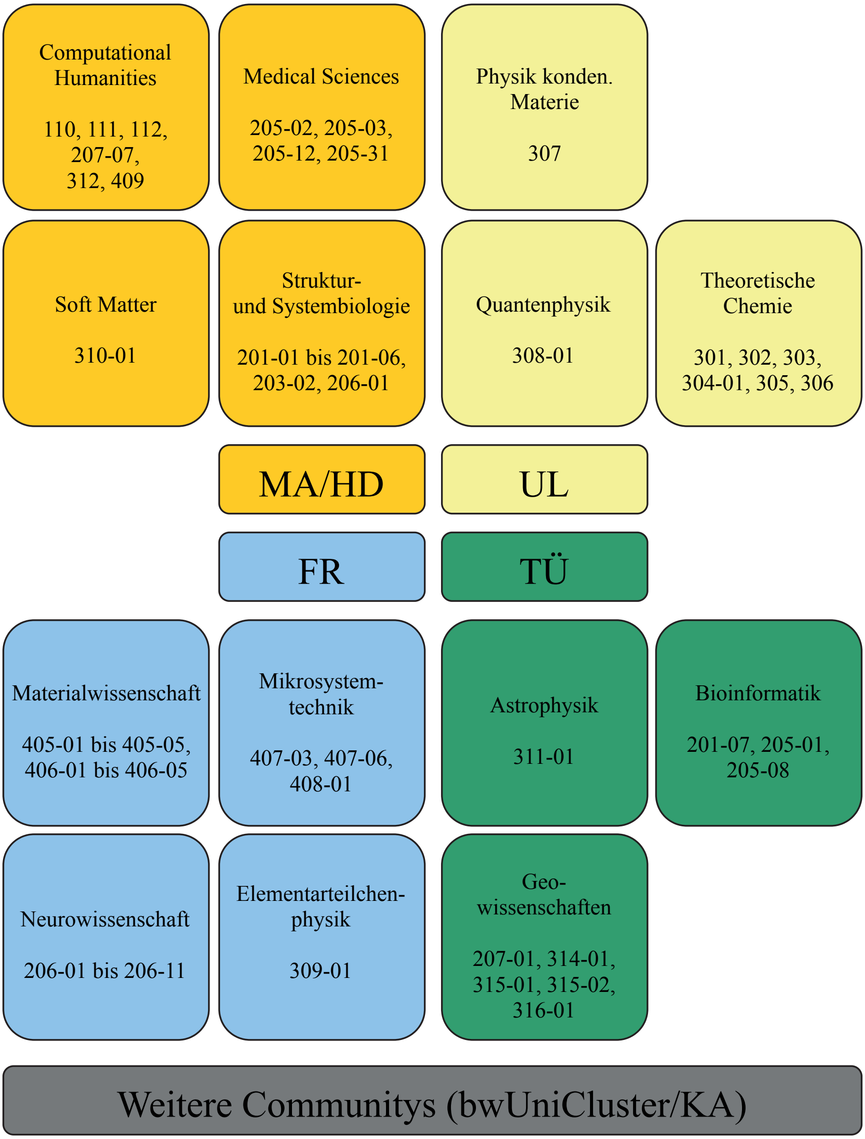


Abbildung 1: Künftige fachliche Ausdifferenzierung der HPC-Systeme der Ebene 3.

fahrung in der HPC-Nutzung eine Sonderrolle ein. Neben der Grundversorgung auf dem bwUniCluster wird darüber hinausgehender Bedarf direkt auf den Systemen auf Tier-2 und Tier-1 abgedeckt. Das Kompetenzzentrum Ingenieurwissenschaften wird deshalb auch über Ebenen hinweg ausgerichtet und profitiert dabei von den

Erfahrungen in Karlsruhe und Stuttgart. Die erfolgreiche Rolle des bwUniClusters als allgemeines Versorgungssystem für die Universitäten des Landes wird fortgeschrieben.

Die im Rahmen des Landesprojektes bwCloud aufgebauten Infrastrukturen und Dienste ergänzen mit der landesweiten Compute-Cloud-Infrastruktur die Angebote für Nutzer mit geringerem Rechenbedarf, Bedarf an anderen Betriebsmodellen (z. B. primär interaktive Nutzung, permanent laufende Dienste wie Workflow-Engines, Science-Portale) oder speziellen Betriebsumgebungen.

5 Data Intensive Computing und Large Scale Scientific Data Management

Gemäß der Wissenschaftsrats-Empfehlung wird der höchsten Leistungsklasse im HPC (Tier-1) vor allem der Bereich Capability Computing zugeordnet, während die Systeme der nächst niedrigeren Leistungsklasse (Tier-2) sowohl die Anforderungen des Capacity Computing bedienen als auch das Capability Computing adressieren. Die Systeme des Tier-3 dienen im Wesentlichen dem Capacity Computing in den Hochschulen mit Fokus auf individuelle wissenschaftliche Communitys. Baden-Württemberg hat mit seinem bwHPC-Konzept einen leistungsfähigen Rahmen für die Kooperation und Abstimmung zwischen den genannten Ebenen und damit ein sehr gutes Umfeld für die Wissenschaften geschaffen. Diese Grundlagen werden mit dem vorliegenden Konzept weiterentwickelt.

Der zunehmende Einsatz von Methoden zur Datenanalyse, also der Erkenntnisgewinn aus gemessenen oder anderweitig gewonnenen Daten, wurde als aktueller Trend bereits genannt. Häufig sind diese neuen Methoden nicht nur sehr rechenintensiv, sondern sie erfordern zusätzlich auch eine leistungsfähige Verwaltung sehr großer Datenmengen und die Verfügbarkeit großer Datenmengen aus unterschiedlichen Quellen auf HPC-Systemen. Man spricht in diesem Kontext von Data Intensive Computing (DIC).

Ausgehend vom Modell des Data-Life-Cycle lassen sich folgende, typische Workflows und Abläufe identifizieren. Beispielsweise Simulationen auf HPC-Systemen generieren große Datenmengen, die zunächst auf den angeschlossenen, sehr schnellen parallelen Dateisystemen lokal gespeichert werden. Experimente liefern ebenfalls große Datenmengen, die meist nach einer Vorselektion in sogenannte Large Scale

Data Facilities (LSDF) transferiert werden. Die so gewonnenen Daten werden zu einem späteren Zeitpunkt auf speziell dafür ausgelegten Systemen weiter analysiert oder visualisiert. Diese sind häufig aus Performancegründen mit eigenen Speichersystemen ausgestattet. Dies bedingt entweder eine Übertragung großer Datenvolumina von parallelen Dateisystemen oder LSDFs zu diesen Speichersystemen oder aber einen hoch performanten direkten Zugriff auf die jeweilige LSDF. Die erzielten Ergebnisse werden veröffentlicht, und die zugrundeliegenden Daten werden zur Nachnutzung in Repositorien abgelegt. Dies erfordert erneut eine Übertragung und Verarbeitung von Daten auf geeignete Zielsysteme. Am Ende eines Forschungsvorhabens werden die gesammelten Daten für eine spätere erneute Nutzung archiviert und dazu an ein für langfristige Archivierungszwecke ausgelegtes Datenspeichersystem übertragen. Ein Data-Life-Cycle am Beispiel eines Bioinformatik-Workflows ist in Abbildung 2 dargestellt.

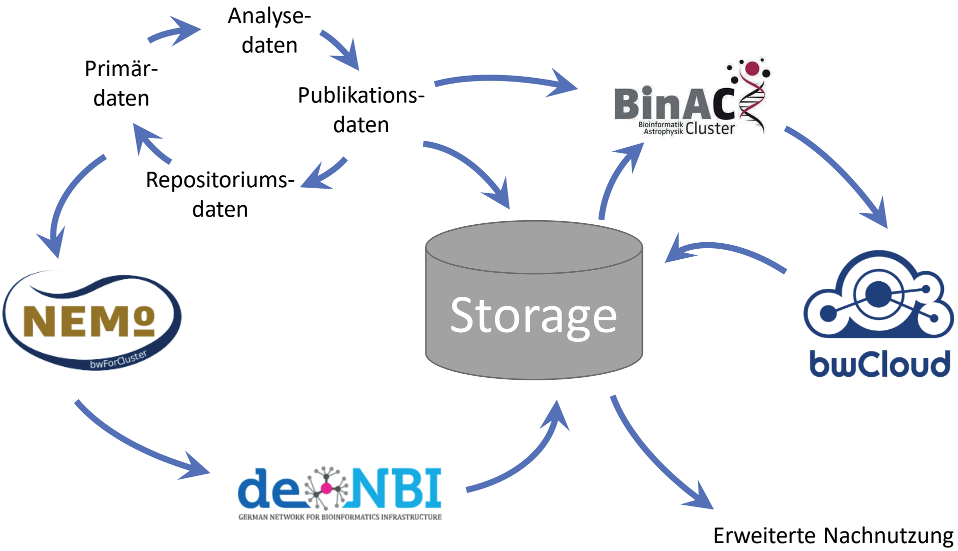


Abbildung 2: Data-Life-Cycle am Beispiel eines Bioinformatik-Workflows.

Bei all diesen Datenübertragungen müssen Informationen zur Provenienz der Daten, wissenschaftliche Metadaten, Autorisierungs- und ggf. auch Authentifizierungsinformationen mit übermittelt werden. Teilweise sind zusätzlich auch Formatkonversionen oder weitere Verarbeitungsschritte wie Aggregation oder Anonymisierung erforderlich. Die zugrundeliegenden technischen Prozesse können (und sollen) in vielen

Fällen nicht vollständig automatisiert ablaufen, da entsprechende Standards, sofern vorhanden, sehr spezifisch für individuelle wissenschaftliche Communitys sind. Zudem generiert die Interaktion der Wissenschaftler mit den wissenschaftlichen Daten durch Qualitätssicherung und Anreicherung der Metadaten echten Mehrwert.

Ausgehend von den bisher gemachten Erfahrungen beim Betrieb der Dienste im Bereich der Simulation und Daten, soll die bisherige vorwiegend technische Integration auf Basis des föderierten Identitätsmanagement weiter ausgebaut werden. Dabei wird neben einer engeren technischen Verzahnung von Rechen- und Datendiensten insbesondere das Ziel verfolgt, die verschiedenen Dienste und deren Supportstrukturen für die wissenschaftlichen Nutzer in abgestimmten Prozessen und unter einheitlicher Governance anzubieten.

6 Forschungsdatenmanagement und Datenföderation

Das Land Baden-Württemberg hat im Rahmen des gemeinsam entwickelten Fachkonzeptes »E-Science: Wissenschaft unter neuen Rahmenbedingungen – Fachkonzept zur Weiterentwicklung der wissenschaftlichen Infrastruktur in Baden-Württemberg« (MWK-BW, 2014) die Bedeutung des Forschungsdatenmanagements für eine nachhaltige Nutzung von Forschungsdaten betont und zwei Förderprogramme (Forschungsdatenmanagement und Virtualisierte Forschungsumgebungen) eingerichtet. FDM, die nachhaltige und zukunftsorientierte Organisation von Forschungsdaten, ihre Verfügbarmachung und Publikation sind heute ein Fokus moderner Forschungsprozesse. Die Werkzeuge des FDMs sind insbesondere notwendig, wenn die Vorgaben individueller Datenmanagementpläne (DMP) umgesetzt werden müssen und dazu ein Metadatenmanagement erfordern.

Die im Rahmen des vorgelegten Umsetzungskonzeptes zu entwickelnden Maßnahmen werden den Empfehlungen des Rats für Informationsinfrastrukturen (RfII) folgen, indem sie die Weiterentwicklung vorhandener Infrastrukturen mit dem Aufbau einer Umgebung für FDM verschränken. So sollen im Rahmen des Gesamtkonzepts die beantragten Großgeräte (Erweiterung der LSDF in Heidelberg, bwSFS in Freiburg, Tübingen und Ulm) den Forschern nicht nur große Speichersysteme zur Verfügung stellen, sondern ihnen mittelfristig auch die notwendigen Werkzeuge für das FDM integriert anbieten. Ebenfalls sollen diese Speichersysteme in einer Datenföderation koordiniert werden. Bei der Etablierung der Föderation und zur logischen

Verbindung einzelner Datenmanagementsysteme kann auf verschiedene Quellen zurückgegriffen werden: Existierende Initiativen in der Helmholtz Gemeinschaft (Programm Supercomputing & Big Data), Aktivitäten auf nationaler (BMBF, DFG: LIS-Programme), europäischer (EUDAT, Lecarpentier u. a., 2013; Ardestani u. a., 2015, künftig evtl. EOSC) und internationaler Ebene (Research Data Alliance).

Es hat sich gezeigt, dass für den Zugang zu den Speichersystemen eine Vielfalt an Protokollen angeboten werden muss, um den vielfältigen Anforderungen der Wissenschaftler gerecht zu werden. Dazu gehören neben NFS und CIFS/SMB auch eher aus dem Cloud Umfeld kommende Schnittstellen wie S3. Parallel dazu mussten insbesondere zur Kopplung von Daten und HPC Systemen auch sehr leistungsfähige Anbindungen realisiert werden. Die Umsetzung des Zugangs mit geringen Latenzen und hohen Bandbreiten erfordert entweder kurze Distanzen oder spezielle Dienste wie direkte optische Verbindungen im Weitverkehrsnetz.

In vielen Anwendungsbereichen bestehen für Daten komplexe und restriktive Nutzungsregeln, die besonders bei der zentralen Datenspeicherung berücksichtigt werden müssen. Diese Einschränkungen stehen oft in Zusammenhang mit personenbezogenen (z. B. Medizin, Sozialwissenschaften, Psychologie und Mobilitätsforschung) oder geheimen Daten (z. B. Wirtschaftswissenschaften und Ingenieurwissenschaften). In einer Datenföderation müssen daher auch die Möglichkeiten geschaffen werden, lokale Datenquellen performant anzubinden und beim Datentransfer Mechanismen wie Anonymisierung/Pseudonymisierung oder Aggregation zu unterstützen.

Die Basis für diese Datenföderation stellt vorhandene und zusätzlich aufzubauende Infrastruktur von Datenmanagement- und Datenspeichersystemen dar. Dazu gehören die parallelen Dateisysteme der HPC-Systeme, LSDFs, Datenanalyse-Spezialsysteme, Repositorien und Archivierungssysteme wie bwDataArchiv und bwDataDiss. Ebenso gehört dazu neben lokalen Speichersystemen und Repositorien auch die Anbindung von nationalen und internationalen Systemen, die durch die jeweiligen Fachcommunitys betrieben werden. Diese Systeme sollen und können dabei nicht zu einem oder wenigen zentralen Systemen zusammengeführt werden, sondern grundsätzlich eigenständig Teile einer Föderation bleiben. Dieser Ansatz ermöglicht wie bei bwHPC für viele Wissenschaftsbereiche die bewährte Fokussierung der einzelnen Betreiber auf ihre jeweiligen wissenschaftlichen Communitys und ermöglicht bei Bedarf zusätzlich eine erhöhte Datensicherheit durch Georedundanz. Für Daten, die besonderen datenschutzrechtlichen Anforderungen unterliegen wie z. B. in der Medizin oder den Sozialwissenschaften, müssen auch dezentrale Ansätze verfolgt und zusätzliche Mechanismen wie Pseudonymisierung, Anonymisierung oder Aggregation der Daten vor einen Transfer etabliert werden. Der Austausch und die Möglichkeit zur Korrelation der Daten müssen unter Beibehaltung der Datenhoheit weitgehend automatisiert und hoch performant ablaufen.

7 Weiterentwicklung der Support-Strukturen

Im Rahmen des Begleitprojektes bwHPC-S5 werden die notwendigen Absprachen zwischen allen Betreibern von Teilsystemen der Föderation analysiert, formalisiert und anschließend langfristig etabliert. Das Betriebsprojekt wird auch die angesprochenen Fragen notwendiger Formatkonversionen, Abbildungen von Metadaten und von Autorisierungs- und Authentifizierungsinformation adressieren. Die Benutzerunterstützung stellt neben den technischen Systemen für Datenmanagement und Datenspeicherung sowie der Datenmanagement-Software die dritte wesentliche Säule bei der Etablierung der Datenföderation dar.

Kernelement des neu ausgerichteten, erweiterten Begleitprojektes bwHPC-S5 (Scientific Simulation und Storage Support Service) ist die Weiterentwicklung von föderativen Unterstützungsstrukturen. Bei der Integration der Systeme und des Datenmanagements steht der sichere und performante Zugriff auf entfernte Daten sowie die Integration in den Data-Life-Cycle im Projektfokus, um das datenintensive Rechnen zu ermöglichen. Im Bereich Infrastruktur und föderatives Dienstemanagement wird das Augenmerk auf Monitoring, Reporting und betriebliche Optimierung gelegt. Unverzichtbar ist die Ausweitung des landesweiten HPC-Schulungsprogramms unter Berücksichtigung von DIC und LS^{2}DM. Bei allen Projektzielen wird verstärkt auf die Abstimmung mit den Tier-1- und Tier-2-Kompetenzen geachtet.

8 Governance

Die wissenschaftsgeleitete Steuerung der Nutzung und Weiterentwicklung dieser Infrastruktur wird gewährleistet durch eine übergreifende und in den letzten Jahren etablierte Governance- Struktur (Wesner, Suchodoletz u. a., 2017; Wesner, Walter u. a., 2016). Sie besteht aus den folgenden Elementen: HLRS-Lenkungsausschuss (für HPC Tier-1 und Tier-2), Landesnutzerausschuss (für HPC Tier-3), bwHPC-Lenkungskreis (für HPC Tier-3) sowie bwDATA-Steuerungskreis.

Der HLRS-Lenkungsausschuss stellt das seit langem etablierte Aufsichtsgremium für das Tier-1-System in Stuttgart und das Tier-2-System in Karlsruhe dar. Er ist besetzt mit Wissenschaftlerinnen und Wissenschaftlern, die je zur Hälfte auf Vorschlag der DFG und der Landesrektorenkonferenz vom Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg berufen werden. Er realisiert ein wissenschaftsgeleitetes Zugangsverfahren zu den beiden Systemen und stellt die optimale und offene Nutzung der Ressourcen für die jeweiligen Anwendungen und die Durchlässigkeit zwischen den Systemen sicher.

Für die wissenschaftliche Steuerung der bwHPC-Dienste auf Tier-3 und die nutzerseitige Steuerung im Bereich der wissenschaftlichen Datenhaltung wurde der Landesnutzerausschuss Baden-Württemberg (LNA-BW) eingerichtet. Im LNA-BW sind alle Universitäten des Landes sowie Repräsentanten der Hochschulen für Angewandte Wissenschaften vertreten. Die Prorektoren/Vizepräsidenten für Forschung schlagen jeweils ein Mitglied vor, das dann vom Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg in den LNA-BW berufen wird. Zudem sind die Hochschulen für Angewandte Wissenschaften mit einem gemeinsamen Vertreter Mitglied im LNA. Die Aufgaben des Landesnutzerausschusses umfassen aktuell die Diskussion der Regularien bezüglich des Ressourcenzugangs, die Beurteilung von Clusterauslastungsdaten sowie die Diskussion der Fachbereichszuordnung der Kompetenzzentren. Darüber hinaus behandelt der Ausschuss die Regulierung von Clustererweiterungen und die Vertretung der Nutzerwünsche bezüglich des operativen Betriebes gegenüber den Betreibern. Dazu gehören die Rückmeldung des nutzerseitigen Bedarfs an neuen Technologien an die Betreiber, die Ermittlung des Bedarfs an Software-Lizenzen, Bedarf an Quotas sowie Anpassungen bei Job-Queues bezüglich Job-Laufzeiten.

Der bwHPC-Lenkungskreis ist besetzt mit Mitgliedern aus dem Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg, dem LNA-BW, dem ALWR und den Hochschulen für angewandte Wissenschaften. Er übernimmt die Rolle des Lenkungsausschusses für die Projekte im Umfeld von bwHPC, insbesondere für das erweiterte Begleitprojekt bwHPC-S5. Der bwHPC-Lenkungskreis begleitet den LNA-BW und den ALWR-BW in allen Fragen betreffend Konzept und Evaluierung von bwHPC, Bedarfserhebungen, Ressourcen, Betrieb und Begleitprojekte.

Der bwDATA-Steuerungskreis überwacht analog zum bwHPC-Lenkungskreis die Umsetzung der im Rahmenkonzept bwDATA festlegten Projekte und Maßnahmen für große wissenschaftliche Daten. Ihm gehören die Mitglieder des ALWR-BW an sowie mindestens zwei Vertreter der Hochschulen für Angewandte Wissenschaften, ein Vertreter der Dualen Hochschule Baden-Württemberg (DHBW), sowie bei Bedarf je ein Vertreter der Pädagogischen Hochschulen, des Landesarchivs, der Landesbibliotheken, der Kunst- und Musikhochschulen und der vier Universitätskliniken. Zu den zentralen Aufgaben des Steuerkreises gehört auch die Wahrnehmung der Verantwortung für die kontinuierliche konzeptionelle Fortführung des Rahmenkonzeptes bwDATA. An den Sitzungen nimmt ein Vertreter des MWK als ständiger Gast teil.

Im Zuge der Einführung der Datenföderation für Baden-Württemberg soll die Governance ergänzt werden, um die weiteren Aufgaben dieses Konzeptes abzudecken. So ist eine Erweiterung der Aufgaben des Landesnutzerausschusses von den bisher zentral behandelten Fragen des High Performance Computings hin zu datenintensiven Diensten vorgesehen. Darüber hinaus ist eine Zusammenführung von bwHPC-Lenkungskreis und bwDATA-Steuerkreis geplant, um die in diesem Umsetzungskonzept adressierten Herausforderungen effizient zu bewältigen.

Gekürzte Fassung Dieser Text ist die gekürzte Fassung des von der DFG begutachteten »Umsetzungskonzepts der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM)« von Schneider, Heuveline, Horstmann, Neumair, Hätscher u. a. (2019).

Autoren

Gerhard Schneider, Universität Freiburg
Vincent Heuveline, Universität Heidelberg
Karl-Wilhelm Horstmann, Universität Hohenheim
Bernhard Neumair, Karlsruher Institut für Technologie
Petra Hätscher, Universität Konstanz
Josef Kolbitsch, Universität Mannheim
Simone Rehm, Universität Stuttgart
Michael Resch, Universität Stuttgart
Thomas Walter, Universität Tübingen

Stefan Wesner, Universität Ulm
Peter Castellaz, Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg

Korrespondenzautoren

Gerhard Schneider: gerhard.schneider@rz.uni-freiburg.de
Rechenzentrum, Albert-Ludwigs-Universität, Freiburg
Thomas Walter: thomas.walter@uni-tuebingen.de
ZDV, Eberhard-Karls-Universität, Tübingen

ORCID

Gerhard Schneider https://orcid.org/0000-0002-3214-002X
Josef Kolbitsch https://orcid.org/0000-0002-7601-1553
Thomas Walter https://orcid.org/0000-0002-8656-2340
Stefan Wesner https://orcid.org/0000-0002-7270-7959

Literatur

Ardestani, S. B. u. a. (2015). »B2share: An open escience data sharing platform«. In: *2015 IEEE 11th International Conference on e-Science (e-Science)*. IEEE, S. 448–453.

Barthel, R. und J. Salk (2019). »bwHPC-S5: Scientific Simulation and Storage Support Services. Unterstützung von Wissenschaft und Forschung beim leistungsstarken und datenintensiven Rechnen sowie großskaligem Forschungsdatenmanagement«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg*. Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 17–28. DOI: 10.15496/publikation-29039.

Lecarpentier, D. u. a. (2013). »EUDAT: a new cross-disciplinary data infrastructure for science«. In: *International Journal of Digital Curation* 8.1, S. 279–287. DOI: 10.2218/ijdc.v8i1.260.

MWK-BW (2014). *E-Science: Wissenschaft unter neuen Rahmenbedingungen. Fachkonzept zur Weiterentwicklung der wissenschaftlichen Infrastruktur in Baden-Württemberg.* Fachkonzept. Ministerium für Wissenschaft, Forschung und Kunst. URL: https://www.baden-wuerttemberg.de/fileadmin/redaktion/m-mwk/intern/dateien/Anlagen_PM/2014/066_PM_Anlage_E-Science_Web.pdf (besucht am 25. 02. 2019).

MWK-BW (2017). *HPC-Landesstrategie: Eine halbe Milliarde Euro für digitale Infrastruktur und Supercomputer.* Pressemitteilung Nr. 99/2017. Ministerium für Wissenschaft, Forschung und Kunst. URL: `https://mwk.baden-wuerttemberg.de/fileadmin/redaktion/m-mwk/intern/dateien/Anlagen_PM/2017/099_PM_Super-Computing_Eine_halbe_Milliarde_Euro_f%C3%BCr_digitale_Infrastruktur.pdf` (besucht am 07. 02. 2019).

Schneider, G., V. Heuveline, K.-W. Horstmann, B. Neumair, P. Hätscher u. a. (2019). *Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM).* Konzept. Hochschulen des Landes Baden-Württemberg. URN: `urn:nbn:de:bsz:21-dspace-864846`.

Schneider, G., V. Heuveline, K.-W. Horstmann, B. Neumair, M. Waldvogel u. a. (2018). *Rahmenkonzept der Hochschulen des Landes Baden-Württemberg für datenintensive Dienste – bwDATA (2015-2019).* de. Konzept. Hochschulen des Landes Baden-Württemberg. DOI: `10.15496/publikation-21187`.

Wesner, S., D. von Suchodoletz, B. Wiebelt, G. Schneider und T. Walter (2017). »Overview on governance structures in bwHPC«. In: *Proceedings of the 3rd bwHPC-Symposium: Heidelberg 2016.* (2016). Hrsg. von S. Richling, M. Baumann und V. Heuveline. Heidelberg: heiBOOKS. DOI: `10.11588/heibooks.308.418`.

Wesner, S., T. Walter, B. Wiebelt, D. von Suchodoletz und G. Schneider (2016). »Strukturen und Gremien einer bwHPC-Governance – Momentaufnahmen und Perspektiven«. In: *Kooperation von Rechenzentren Governance und Steuerung – Organisation, Rechtsgrundlagen, Politik.* Hrsg. von D. von Suchodoletz, J. C. Schulz, J. Leendertse, H. Hotzel und M. Wimmer. de Gruyter, S. 315–329. ISBN: 978-3-11-045888-6. DOI: `10.1515/9783110459753-027`.

Wissenschaftsrat (2014). *Bedeutung und Weiterentwicklung von Simulation in der Wissenschaft.* Positionspapier. WR. URL: `https://www.wissenschaftsrat.de/download/archiv/4032-14.pdf` (besucht am 07. 02. 2019).

— (2015). *Empfehlungen zur Finanzierung des Nationalen Hoch- und Höchstleistungsrechnens in Deutschland.* Empfehlungen. WR. URL: `https://www.wissenschaftsrat.de/download/archiv/4488-15.pdf` (besucht am 07. 02. 2019).

bwHPC-S5: Scientific Simulation and Storage Support Services

Unterstützung von Wissenschaft und Forschung beim leistungsstarken und datenintensiven Rechnen sowie großskaligem Forschungsdatenmanagement

Robert Barthel[*] Jürgen Salk[†]

[*]Karlsruher Institut für Technologie
[†]Universität Ulm

Das Projekt bwHPC-S5 ist das aktuelle Begleitprojekt zum Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das Hochleistungsrechnen (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM) und ist das Bindeglied zwischen Wissenschaft und den Infrastrukturen für HPC, DIC und LS^{2}DM. Es beinhaltet eine landesweit aufgestellte Benutzerbetreuung und unterstützt den Übergang auf höhere HPC-Leistungsebenen. Phase 1 des Projekts läuft von Juli 2018 bis Dezember 2020 und wird vom Ministerium für Forschung, Wissenschaft und Kunst Baden-Württemberg (MWK) finanziert.

1 Einleitung

Umsetzungs- bzw. Rahmenkonzepte der Universitäten für das Hochleistungsrechnen bzw. datenintensive Dienste haben in Baden-Württemberg eine lange Tradition, vgl. bwGRiD (von 2008 bis 2013, Schulz, Janne Christian [Hrsg.] u. a., 2014), bwHPC (von 2013 bis 2018, Schneider, Hebgen u. a., 2018a; Hartenstein u. a., 2013), sowie bwDATA Phase 1 (von 2013 bis 2014, Schneider, Hebgen u. a., 2018b) und bwDATA

DOI: 10.15496/publikation-29039

(von 2015 bis 2019, Schneider, Heuveline u. a., 2018), und ermöglichten und erbringen erfolgreiche Kooperationen von landesweit verfügbaren Diensten und deren Nutzerunterstützung für WissenschaftlerInnen und Studierende. Der Wissenschaftsrat (2015) hat dabei bwHPC als beispielhafte HPC-Landesstrategie hervorgehoben.

Beginnend mit bwGRiD wurde in Baden-Württemberg erstmals eine weitgehend homogene HPC-Infrastruktur auf der Ebene 3 (vgl. Wissenschaftsrat, 2015, Seite 9 ff.) standortübergreifend betrieben. Hier konnten erfolgreich Strukturen für die enge Zusammenarbeit der Betreiberstandorte sowie eine landesweit vereinheitlichte Benutzerumgebung und flexible Ressourcenadressierung (u. a. über ein Webportal) etabliert werden. Notwendige Weiterentwicklungen u. a. beim Ressourcenangebot zur Abdeckung der vielfältigen Anforderungen der verschiedenen Nutzergruppen, beim Ressourcenzugang zur Vereinfachung der Registrierungsprozesse, beim fachspezifischen Abdeckungsgrad der Nutzerunterstützung und bei der rechnernahen Dateninfrastruktur wurden im nachfolgenden bwHPC-Konzept adressiert.

Bei den Umsetzungen des bwHPC-Konzepts zwischen 2013 bis 2018 sind insbesondere die Ausprägung einer heterogenen HPC-Infrastruktur bestehend aus vier HPC-Forschungsclustern, genannt bwForCluster, an den Standorten Ulm, Mannheim/Heidelberg, Freiburg und Tübingen und einem HPC-Grundversorgungscluster, genannt bwUniCluster, sowie die durch das Projekt bwHPC-C5 (»Coordinated Compute Cluster Competence Centers«) etablierten bwHPC-Kompetenzzentren und dessen Nutzerunterstützungsdienste, als auch die Etablierung weiterer Steuerungsorgane hervorzuheben. So weisen diese neu etablierten bwForCluster eine auf definierte fachliche Schwerpunkte zugeschnittene Hardware- und Softwarearchitektur und an die besonderen Bedürfnisse zugeschnittenes Betriebsmodell auf.

Die etablierte föderierte Identitätsinfrastruktur (bwIDM) ermöglicht einen niederschwelligen Zugangsprozess bei HPC-Clustern und anderen Diensten (z. B. Supportportal). Durch die bwHPC-Kompetenzzentren und das Projekt bwHPC-C5 kann die HPC-Nutzerschaft auf ein umfangreiches HPC-Softwareportfolio auf den Clustern, ein Angebot auf Intensivunterstützung gezielte Expertenunterstützung durch sogenannte Tigerteam-Projekte bzw. Unterstützung beim Übergang auf Leistungsebenen 2 und 1, ein landesweit koordiniertes und angebotenes Schulungsprogramm, eine online verfügbare Best-Practices- und Nutzungsdokumentation sowie bedarfsgerechte Technologieevaluation blicken. Mit dem seit 2013 etablierten bwHPC-Symposium und Landesnutzerausschuss hat die Nutzerschaft zudem die Möglichkeit zum direkten Erfahrungsaustausch mit den HPC-Betreibern und -Dienstleistern bzw. deren Aussteuerung.

Parallel zur HPC-Infrastruktur und dem bwHPC-C5-Projekt wurde als Teil des bwDATA-Konzeptes eine umfangreiche Speicherinfrastruktur mit der Large Scale Data Facility (LSDF) in Karlsruhe und Heidelberg, dem bwDataArchiv in Karlsruhe und weiteren Landesdiensten aufgebaut. Dazu kommen noch weitere Infrastrukturen wie die bwCloud, eine Reihe von Spezialsystemen zur Datenanalyse (z. B. das Smart Data Innovation Lab) oder bwVisu zur Visualisierung.[1]

Aus Sicht der Nutzer ist das Management und die Verarbeitung von Forschungsdaten in getrennten Infrastrukturen wenig praktikabel. Entsprechenden Empfehlungen folgend (z. B. Rat für Informationsinfrastrukturen, 2016) werden mit dem neuen Umsetzungskonzept für HPC, DIC und LS^{2}DM des Landes Baden-Württemberg die bisher eigenständigen bwHPC- und bwDATA-Konzepte nun in eine gemeinsame Sicht überführt.

Zur Umsetzung der gemeinsamen Sicht auf HPC, DIC und LS^{2}DM werden im bwHPC-S5-Projekt alle erbrachten Dienste für die wissenschaftlichen Nutzer so organisiert, dass die begleitende Supportleistung, unabhängig ob es um das Thema HPC, Daten oder einer Kombination aus beiden, über die gleichen Schnittstellen angeboten wird. Das Konzept »one face to the customer« gilt dabei nicht nur für die Schulungen, Best-Practices-Dokumentationen, Dienstschnittstellen oder die allgemeine Nutzerunterstützung sondern insbesondere auch für die gemeinsam zwischen Betreibern und Nutzern durchgeführten Tigerteam-Unterstützungsprojekte.

Das Projekt bwHPC-S5 ist für den langen Zeitraum des neuen Umsetzungskonzepts in mehrere Phasen unterteilt, um auf geänderte Bedarfslagen und Anforderungen flexibel reagieren zu können. Die Unterteilung in Phasen und deren Evaluierung ermöglicht u. a. die gezielte Nachsteuerung der Umsetzungsmaßnahmen. Der Projektphase 1 vom Juli 2018 bis Dezember 2020 sind die folgenden sechs Ziele auferlegt: 1. Ausbau der föderativen Wissenschaftsunterstützung, 2. Fortschreibung der fachlichen Ausprägung im Bereich HPC, 3. Umsetzung einer landesweiten Datenföderation, 4. bedarfsgetriebene gemeinsame Technologieevaluierung, 5. weitere Professionalisierung der Öffentlichkeitsarbeit sowie 6. Fortschreibung und Weiterentwicklung der gemeinsamen Software-Versorgung der HPC-Systeme.

[1] `https://www.rda.kit.edu`, `https://www.bw-cloud.org`, `https://www.sdil.de`, `http://www.bwvisu.de`

Der vielleicht bedeutendste Aspekt des bwHPC-S5-Projekts ist die Einführung von Leistungsmetriken zur Bewertung des (Gesamt-)Projektfortschritts. Diese werden neben den definierten Ergebnissen der Projektaufgaben zur Beurteilung des Projekts herangezogen und umfassen quantitative Kennzahlen zu: Systemnutzung (z. B. Auslastungsmetriken der Cluster), Nutzeraktivität (z. B. registrierte Rechen- und Speichervorhaben), wissenschaftliche Resultate (z. B. Publikationen in Fachzeitschriften), Unterstützungsmaßnahmen (z. B. Tigerteam-Projekte), Softwareangebot (z. B. bereitgestellte Softwaremodule und deren Nutzungsgrad), Schulungsaktivitäten (z. B. Präsenzkurse deren Teilnehmerherkunft), technische Dokumentation und Trainingsplattform (z. B. online verfügbare Best-Practices und deren Interaktionscharakteristiken), Öffentlichkeitsarbeit (z. B. Outreach-Wirkung) und Innovationen (z. B. Technologiesprints und deren Erkenntnisse).

2 Projektpartner und Governance

Das Konsortium des Projekts bwHPC-S5 Phase 1 besteht aus den Universitäten Freiburg, Heidelberg, Hohenheim, Konstanz, Mannheim, Stuttgart, Tübingen und Ulm, dem Karlsruher Institut für Technologie sowie der Hochschule Esslingen und der Hochschule für Technik Stuttgart.

Zur Realisierung und Überwachung des Stands und der Ziele des Projekts wird eine hierarchische Struktur für die Koordination umgesetzt. Strategische Entscheidungen über die Fortentwicklung der Infrastrukturen werden vom Arbeitskreis der Leiterinnen und Leiter der Rechenzentren bzw. Informationszentren der Universitäten des Landes Baden-Württemberg (ALWR-BW), eingerichtet durch die Landesrektorenkonferenz (LRK), sowie dem bwHPC-Lenkungs- und bwDATA-Steuerkreis getroffen, in dem auch Repräsentanten der Nutzer der Hochschulen des Landes und des Ministeriums für Wissenschaft, Forschung und Kunst (MWK) Baden-Württemberg vertreten sind.

Der bwHPC-Lenkungskreis führt ALWR-BW, Wissenschaftler (Nutzervertreter) und das Landesministerium für Wissenschaft, Forschung und Kunst (Ministeriumsvertreter) zusammen, um einen verbindlichen Rahmen für die Interaktion der verschiedenen Stakeholder im Hinblick auf die Steuerung von Umsetzung und Weiterentwicklung des Landeskonzepts HPC, DIC, und LS^{2}DM zu schaffen. Für die wissenschaftliche Steuerung der HPC-Systeme der Ebene 3 und die nutzerseitige

Steuerung im Bereich der wissenschaftlichen Datenhaltung ist der Landesnutzerausschuss Baden-Württemberg (LNA-BW) zuständig. Die Prorektoren bzw. Vizepräsidenten für Forschung schlagen jeweils ein Mitglied vor, das dann vom MWK Baden-Württemberg in den LNA-BW berufen wird. Zudem sind die Hochschulen für Angewandte Wissenschaften mit einem gemeinsamen Vertreter Mitglied im LNA-BW.

Das Projekt bwHPC-S5 wird aufgrund seiner Komplexität und Bedeutung für das Umsetzungskonzept HPC, DIC und LS^{2}DM von zwei Vertretern des ALWR-BW begleitet und verantwortet. Für die operative Leitung des Projekts ist dagegen das Projektbüro (PMO) verantwortlich. Fortschritt und Risiken auf Projektebene bzw. Zusammenwirken der Arbeitspakete (AP) werden im Kernteam thematisiert. Das Kernteam setzt sich aus allen Leitern der Arbeitspakete und dem Projektbüro zusammen. Aufgaben der Arbeitspaketleiter umfassen die technische Leitung, Planung und Überwachung ihrer Arbeitspakete aber auch das Berichtswesen an die übergeordneten Gremien. Die bwHPC-Kompetenzzentren stellen zudem je einen stimmberechtigten Vertreter für das sogenannte Clusterauswahlteam (CAT), einem Team zur Zuweisung der Rechen- und Speichervorhaben an eine Cluster- oder Speicherressource.

Für die Abstimmung in betrieblichen Belangen der HPC- und Datenföderation, u. a. bezüglich Entwicklung und Fortschreibung der Betriebsmodelle sowie der betrieblichen Aspekte in der Produktion, wurde ein Technical Advisory Board (TAB) etabliert, in welches jede Universität ein Mitglied entsendet. Die Hochschulen entsenden ebenfalls ein Mitglied in das TAB.

Die oben genannten Gremien treffen sich regelmäßig, um innerhalb des Projekts die notwendige Abstimmung zu vollziehen. Neben Präsenztreffen stimmen sich Kernteam bzw. Technical Advisory Board in mindestens zweiwöchentlich stattfindenden Video- und Telefonkonferenzen ab.

3 Projektplan

Zur Optimierung der Umsetzung aller Aufgaben und Erreichung der definierten Ziele findet eine Untergliederung der Aufgabenbereiche anhand der Bezugsgruppen des Projekts, d. h. WissenschaftlerInnen und Studierenden sowie Dienst- und Servicebetreibern, statt. Daraus ergeben sich folgende drei Aktivitätsebenen: 1. Nutzerbezogene Aktivitäten und Öffentlichkeitsarbeit (d. h. AP 1.1, 1.2, 1.3), 2. Föderativer Betrieb und systembezogene Aktivitäten (d. h. AP 2.1, 2.2, 2.3, 2.4) und 3. Innovations- und Evaluationsaktivitäten (AP3).

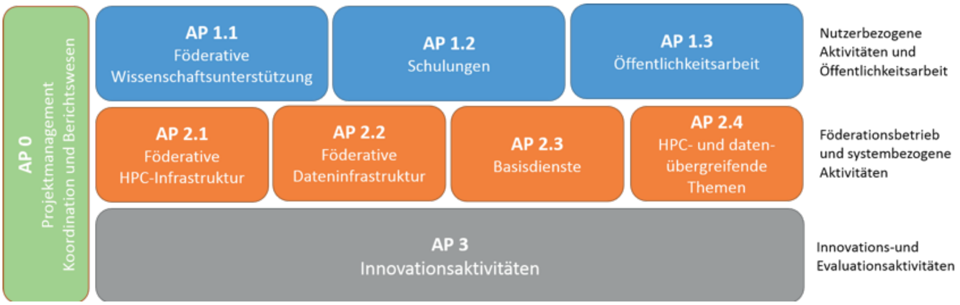

AP 0
Projektmanagement
Koordination und Berichtswesen
AP 1.1
Föderative Wissenschaftsunterstützung
AP 1.2
Schulungen
AP 1.3
Öffentlichkeitsarbeit
Nutzerbezogene Aktivitäten und Öffentlichkeitsarbeit
AP 2.1
Föderative HPC-Infrastruktur
AP 2.2
Föderative Dateninfrastruktur
AP 2.3
Basisdienste
AP 2.4
HPC- und datenübergreifende Themen
Föderationsbetrieb und systembezogene Aktivitäten
AP 3
Innovationsaktivitäten
Innovations-und Evaluationsaktivitäten

Abbildung 1: Arbeitspakete des Projekts bwHPC-S5 Phase 1. Die Größe der Boxen stellen nicht den Aufwandsumfang dar.

Aktivitäten umfassen neben in Projekten typischerweise definierten Aufgaben, Meilensteine und Liefergegenstände in bwHPC-S5 die kontinuierliche Erfassung der Leistungsmetriken.

Aktivitätsebene 1

Aktivitätsebene 1 mit den Arbeitspaketen AP 1.1, AP 1.2 und AP 1.3 stellt die direkte Schnittstelle zu Nutzern und anderen (auch externen) Interessensgruppen sowie der Öffentlichkeit dar. Jedes einzelne dieser drei Arbeitspakete beinhaltet sowohl HPC-bezogene Anteile als auch Themenbereiche mit Bezug zum datenintensiven Rechnen (Data Intensive Computing, DIC) und zum Umgang mit umfangreichen wissenschaftlichen Datenmengen (Large Scale Scientific Data Management, LS^{2}DM).

Im AP 1.1 »Föderative Wissenschaftsunterstützung« steht die zielgerichtete und fachlich fundierte Unterstützung von Nutzern der landesweiten bwHPC- und Datensysteme durch fachspezifisch ausgerichtete bwHPC-Kompetenzzentren im Vordergrund. Die Unterstützung schließt auch Pflege, Optimierung und Ausbau des wissenschaftlichen Software-Portfolios, Erstellung von technischen Anwender-Dokumentationen,[2] Zuweisung von Rechen- und Speicherressourcen sowie Beratung für den Übergang auf höhere HPC-Leistungsebenen ein. Bei der Integration einer

[2] `https://wiki.bwhpc.de`

Nutzerunterstützung zu den Datensystemen wird man u. a. auf die Erfahrungen und Erkenntnisse der Data Life Cycle Labs[3] zurückgreifen.

Unter einem bwHPC-Kompetenzzentrum ist eine Organisationsstruktur zu verstehen, in der Fachkompetenzen zur Anwenderunterstützung in verschiedenen Wissenschaftsbereichen gebündelt werden. Die personelle Zusammensetzung der Kompetenzzentren erfolgt standortübergreifend zur optimalen Ausnutzung des landesweit vorhandenen Expertenwissens. Mit dem landesweiten Ticketsystem[4] werden alle üblichen Anfragen zentral erfasst und unabhängig vom Standort des Nutzers oder des Dienstes vom fachlich am besten geeigneten Expertenteam bearbeitet. Dies wird bei betriebsnahen Themen oft vom jeweiligen Betreiberstandort erbracht, aber bei spezifischen und fachlich tiefergehenden Anfragen erfolgt dies standortübergreifend.

Komplexe und tiefergehende Fragestellungen, die nicht mit den üblichen Unterstützungsstrukturen schnell und effizient erbracht werden können, werden durch Bildung standortübergreifender Tigerteams adressiert. Ein Tigerteam ist Teil eines Kompetenzzentrums und wird zeitlich befristet, meist in enger Kooperation mit einer wissenschaftlichen Arbeitsgruppe, zur Umsetzung von konkreten Unterstützungs- und Optimierungsmaßnahmen aufgestellt. Die personelle Zusammensetzung eines Tigerteams kann standortübergreifend erfolgen, um das an unterschiedlichen Standorten vorhandene Expertenwissen optimal zu nutzen. Mit Projektphase 1 wird zudem ein Kompetenzzentrum für Global Systems Science[5] etabliert.

Ergänzend zu AP 1.1 werden in AP 1.2 »Schulungen« durch gezielte Schulungen die Fachkompetenzen an eine breite Nutzerschaft vermittelt. Dazu wird in Grundlagen- und Aufbaukursen sowohl Basiswissen zur effizienten Nutzung der bwHPC-Systeme für rechen- und datenintensive Aufgaben als auch spezielle Anwendungsbereiche zu HPC und Datenmanagement (z. B. zur parallelen Programmierung oder zum Einsatz spezieller Bibliotheken und Werkzeuge zur Performancesteigerung) sowie komplexer Softwaresysteme adressiert. Neben Präsenzkursen sind auch digitale Lehrformen Teil des Projekt-Portfolios, um möglichst viele Nutzer erreichen zu können. Pflege der Schulungsplattform,[6] Umfang und Aufbau eines E-Learning-Programms sowie die Erprobung eines Webinars gehört zu den erweiterten

[3] `https://www.scc.kit.edu/ueberuns/8057.php`

[4] `https://www.bwhpc.de/ticketsystem.php`

[5] `https://ec.europa.eu/digital-single-market/en/global-systems-science`

[6] `https://www.bwhpc.de/kursangebote.php`

Aufgaben dieses Arbeitspakets in der Projektphase 1. Die bwHPC-Kompetenzzentren unterstützen die Erstellung von Schulungsmaterialien und beteiligen sich ggf. als Tutoren oder Vortragende bei den Präsenzkursen.

Das AP 1.3 »Öffentlichkeitsarbeit« adressiert neben bestehenden Nutzern auch Zielgruppen. Dazu gehören neben potenziellen Nutzern, die bisher auf lokalen Rechensystemen arbeiten und über die besonderen Möglichkeiten zentraler Infrastruktur informiert werden sollen, auch Entscheidungsträger an den Universitäten, die HPC- und Daten-Community und die interessierte Öffentlichkeit. Ziele sind hier die Landesinitiative bwHPC bekannter zu machen, die Kernideen zu vermitteln und insgesamt die Anzahl der Nutzer in Baden-Württemberg weiter zu steigern. Essenzielle Aufgaben dieses Arbeitspakets sind daher die Ausgestaltung und Umsetzung der Nutzerumfragen, Zusammenstellung der wissenschaftlichen Resultate[7] auf der Projektwebseite, Verbreitung von Mitteilungen über bwHPC-Newletter, Presse und Social-Media sowie Erstellung von Informations- und Werbematerialien.

Aktivitätsebene 2

Aktivitätsebene 2 fasst alle Aktivitäten zusammen, welche den landesweiten, föderierten und koordinierten Betrieb der Rechen- und Dateninfrastruktur sicherstellen. Mit den Arbeitspaketen AP 2.1 bis AP 2.4 werden dazu interne Dienste und Dienstleistungen zur Verfügung gestellt, die nicht unmittelbar nach außen sichtbar sind, sondern auf technischer und administrativer Ebene auf die produktive Umsetzung des vorliegenden Landeskonzeptes ausgerichtet sind.

Weiterhin stellen diese Arbeitspakete ein internes Unterstützungsangebot für die darüber liegende nutzerbezogene Schicht, insbesondere für AP 1.1, dar. Im Gegensatz zu der in der Aktivitätsebene 1 gewählten themenübergreifenden Struktur wurde hier eine Aufteilung in föderative HPC-Infrastruktur, föderative Dateninfrastruktur, Basisdienste und Querschnittsthemen gewählt. Diese Aufteilung erlaubt somit die strukturelle Trennung von mehr technisch orientierten und mehr dienstorientierten Aktivitäten, die oft von verschiedenen Personen bearbeitet werden.

Die technisch fokussierten Aktivitäten in den Bereichen HPC (AP 2.1 »Föderative HPC-Infrastruktur«) und Daten (AP 2.2 »Föderative Dateninfrastruktur«) sind im Bereich der Föderation in unterschiedlichen Entwicklungsstadien. Im HPC-Bereich ist die Föderierung der bestehenden Systeme bereits sehr erfolgreich umgesetzt. Auf-

[7] `https://www.bwhpc.de/publikationen_der_nutzer.php`

gaben von AP 2.1 betreffen Optimierungsmaßnahmen des Betriebs der HPC-Cluster (z. B. verbessertes Scheduling, On-Demand-Dateisysteme), Infrastrukturerneuerung und Weiterentwicklung der Betriebsmodelle, Weiterentwicklung und Optimierung der Software-Versorgung bzgl. Funktionalitätsprüfung und Nutzungshandhabung sowie Aufbau und Weiterentwicklung von Infrastrukturen für wohldefinierte Forschungsumgebungen. Im Bereich Daten müssen die erfolgreichen Anbindungen einzelner Systeme noch in eine landesweite Föderation umgesetzt werden.

Primäre Arbeiten von AP2.2 liegen daher in der Entwicklung, Konfiguration und Erprobung. Dazu gehört im ersten Schritt die Anbindung der LSDF-Infrastruktur, bwDataArchiv und der Speichersysteme der bwForCluster und des bwUniCluster unter Nutzung der bestehenden Möglichkeiten von bwIDM und unter Nutzung IP-basierter Standardprotokolle. Zeitlich überlappend mit den oben beschriebenen Aktivitäten werden für ausgewählte Systeme u. a. Mechanismen zur intelligenten und automatisierten Replikation (Caching) oder Annotation mit Metadaten und Provenienz untersucht und in einzelnen Bereichen implementiert.

Die bisher erfolgreich im HPC-Umfeld etablierten Basisdienste (z. B. zentrale Antragsseite für Rechenvorhaben, Monitoring, Webseite zum Softwareportfolio, Ticketsystem) werden im eigenständigen AP 2.3 »Basisdienste« so erweitert und angepasst, dass alle Aspekte der Datenförderation berücksichtigt bzw. neue Anforderungen (z. B. Ressourcenstatus und Nutzungsstatistiken) erfüllt werden. Neben der Vereinheitlichung der Basisdienste gibt es eine Reihe von Technologiebereichen, die nicht klar HPC oder Daten zugeordnet werden können und daher in AP 2.4 »HPC- und datenübergreifende Themen« eigenständig und übergreifend umgesetzt und angeboten werden. Dazu zählen die Bereitstellung von Werkzeugen und Bibliotheken zur Softwareentwicklung und Performance-Analysen, Ausbau des RPM-basierten Frameworks für Software, systematisches und automatisiertes Profiling der Ressourcennutzung sowie Werkzeuge zur Überwachung des clusterübergreifenden Datentransfers.

Aktivitätsebene 3

Die Aufgaben der Aktivitätsebenen 1 und 2 stellen aufgrund der notwendigen Ausrichtung über Standortgrenzen hinweg bei allen Aufbau- und Integrationsaufgaben in den Bereichen HPC, DIC und LS^{2}DM eine hohe Anforderung an darauf abgestimmte neue und innovative Lösungen, müssen aber gleichzeitig die Erwartungshaltung der Anwender erfüllen, die einen verlässlichen und möglichst störungsarmen Betrieb fordern. Dies schränkt die Innovationsmöglichkeiten ein und führt zu einer Fokussierung auf bewährte Technologiebereiche und -lösungen. Die Dynamik im Bereich der verfügbaren Technologien und der zugehörigen Software- und Betriebsmodelle erfordert jedoch für zukünftige Systeme ggf. noch nicht für den Betrieb geeignete Technologien bereits jetzt zu untersuchen. Um nun diese konträren Forderungen abzubilden, werden die eher konservativen Innovationen in Aktivitätsebene 2 durch prototypische und damit experimentelle Technologieevaluationen ergänzt.

Als Teil von AP 3 »Innovationsaktivitäten« werden Technologietrends beobachtet und verfolgt und insbesondere auch auf Ergebnisse von Forschungsaktivitäten im nationalen und internationalen Umfeld geachtet, die ein Potenzial haben bereits angebotene Dienste zu verbessern oder neue aufzubauen. Evaluierung soll primär in Sprints, d. h. mit begrenztem Personaleinsatz für ca. 1-2 Monate, stattfinden. Vorgewählte Themen sind u. a. alternative Prozessorarchitekturen, flexibleres Scheduling, virtualisierte Forschungsumgebungen, Objektspeicher sowie die Einbindung weiterer Landesdienste wie bwVisu und bwCloud.

In Abstimmung mit den Projektpartnern ergeben sich grundsätzlich drei mögliche weitere Schritte: (i) die Evaluierung kommt zur prototypischen Umsetzung in produktionsnaher Umgebung, da die Technologie eine Verbesserung der Angebote in der Aktivitätsebene 1 und 2 verspricht, (ii) die Evaluationsergebnisse sind wenig überzeugend und werden bis auf weiteres nicht weiter verfolgt oder (iii) die Ergebnisse kommen zu keinem eindeutigen Ergebnis und erfordern unter Schärfung der Auswertungskriterien eine weitere Untersuchung ggf. auch erst zu einem späteren Zeitpunkt. Arbeitspaket 3 fungiert somit als Vorstufe für die Überführung von innovativen Konzepten in den produktiven Betrieb und stellt damit eine Schnittstelle zwischen der auf den Produktionsbetrieb ausgelegten Aktivitätsebene 2 und den eher prototypischen Neuentwicklungen weiterer Landesprojekte dar.

4 Fazit

Die Etablierung des Projekts bwHPC-S5 stellt einen der notwendigen Schritte zur Verknüpfung von HPC- und Forschungsdateninfrastrukturen im Land Baden-Württemberg dar. Neben Fortschreibungen seit bwHPC bestehender Aktivitäten liegen in der Projektphase 1 dabei die Entwicklung einer landesweiten Datenföderation und die Erweiterung der bestehenden HPC-Nutzerunterstützung mit Kompetenzen zu DIC und LS^{2}DM im Fokus.

Danksagungen

Die Autoren danken dem Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg für die Finanzierung des Projekts bwHPC-S5.

Korrespondenzautor

Robert Barthel: `robert.barthel@kit.edu`
Karlsruhe Institute of Technology, Steinbuch Centre for Computing
Zirkel 2, 76131 Karlsruhe, Germany

Literatur

Hartenstein, H., T. Walter und P. Castellaz (2013). »Aktuelle Umsetzungskonzepte der Universitäten des Landes Baden-Württemberg für Hochleistungsrechnen und datenintensive Dienste«. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 36.2. DOI: `10.1515/pik-2013-0007`.

Rat für Informationsinfrastrukturen (2016). *Leistung aus Vielfalt: Empfehlungen zu Strukturen, Prozessen und Finanzierung des Forschungsdatenmanagements in Deutschland.* Göttingen. `urn:nbn:de:101:1-201606229098`.

Schneider, G., M. Hebgen u. a. (2018a). *Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das Hochleistungsrechnen.* de. Konzept. Hochschulen des Landes Baden-Württemberg. DOI: `10.15496/publikation-21185`.

— (2018b). *Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für datenintensive Dienste – bwDATA Phase I (2013-2014).* de. Konzept. Hochschulen des Landes Baden-Württemberg. DOI: `10.15496/publikation-21188`.

Schneider, G., V. Heuveline u. a. (2018). *Rahmenkonzept der Hochschulen des Landes Baden-Württemberg für datenintensive Dienste – bwDATA (2015-2019).* de. Konzept. Hochschulen des Landes Baden-Württemberg. DOI: `10.15496/publikation-21187`.

Schulz, Janne Christian [Hrsg.] und Hermann, Sven [Hrsg.] (2014). *Hochleistungsrechnen in Baden-Württemberg – Ausgewählte Aktivitäten im bwGRiD 2012 : Beiträge zu Anwenderprojekten und Infrastruktur im bwGRiD im Jahr 2012.* KIT Scientific Publishing. DOI: `10.5445/ksp/1000039516`.

Wissenschaftsrat (2015). *Empfehlungen zur Finanzierung des Nationalen Hoch- und Höchstleistungsrechnens in Deutschland*. Stuttgart, Drs. 4488-15. https://www.wissenschaftsrat.de/download/archiv/4488-15.pdf.

bwForCluster NEMO

Forschungscluster für die Wissenschaft

Michael Janczyk Dirk von Suchodoletz Bernd Wiebelt

eScience Abteilung, Albert-Ludwigs-Universität, Freiburg, Deutschland

In den ersten zweieinhalb Jahren seiner Betriebszeit entwickelte sich der bwForCluster NEMO zu einem signifikanten Baustein in den landesweiten Forschungsinfrastrukturen für das »High Performance Computing«. Der in der Zwischenzeit erhebliche Ausbau und die Erweiterung des Systems durch Shareholder ist ein Beleg für die Tragfähigkeit seines Betriebsmodells und das Vertrauen in das landesweite HPC-Konzept. Hierzu steuert nicht nur die lokale und landesweite Governance bei, sondern ebenfalls der enge Austausch innerhalb der NEMO-Community. Mit dem System wird eine stabile Umgebung für die diversen Bedürfnisse der Wissenschafts-Communitys bereitgestellt. Parallel dazu werden neue Betriebs- und Monitoring-Konzepte entwickelt und getestet. Aktuelle und neuartige Herausforderungen liegen in der Unterstützung von »Virtualisierten Forschungsumgebungen« und zukünftigen digitalen Workflows ebenso wie in der Containerisierung und der Implementierung effektiver Betriebsmodelle gemeinsam mit den am Standort Freiburg betriebenen Cloud-Infrastrukturen.

1 Einleitung

Der bwForCluster NEMO[1] adressiert Forscher*innen auf Tier-Ebene 3, dem Einstiegssegment des »High Performance Computings« (HPC). Prinzipbedingt durch die Unterstützung verschiedener wissenschaftlicher Communitys muss mit einer Mischung aus unterschiedlichen Benutzerprofilen und entsprechend heterogenen Erwartungshaltungen geplant und gearbeitet werden. Zur Versorgung der Fach-Com-

[1] Zum Zeitpunkt des Verfassens ist der Cluster zweieinhalb Jahre (08/2016 – 01/2019) im Produktivbetrieb und damit bereits bei der Hälfte seiner auf 5 Jahre ausgelegten Betriebszeit.

DOI: 10.15496/publikation-29041

munitys kommt hinzu, dass es für einige Arbeitsgruppen die erste Berührung mit Rechnen jenseits des Desktops ist, während andere Forschende bereits auf (eigenen) Clustern Erfahrungen sammeln konnten. Zusätzlich wurden Arbeitsgruppen akquiriert, welche die Forschungsinfrastruktur durch eigene finanzielle Beteiligungen vergrößert und sich damit als Shareholder erweiterte Nutzungsrechte erworben haben. Aus Betreibersicht muss eine sich ausdehnende Landschaft von Compute- und Storagesystemen in komplexer werdenden wissenschaftlichen Workflows, die beispielsweise ein Pre-Processing in der Cloud und eine spätere Visualisierung großer Datenmengen vorsehen, effektiv gemanagt werden. Um ein Austarieren der vielfältigen Interessen und einen harmonischen Betrieb zu gewährleisten, wurden entsprechende Governance- und Betriebsstrukturen geschaffen, die sich in den zweieinhalb Jahren Laufzeit bewährt haben (Wesner u. a., 2016; Wiebelt u. a., 2016).

2 Die beteiligten Wissenschafts-Communitys

Der bwForCluster NEMO am Standort Freiburg bedient im Landesverbund von bwHPC die Bedürfnisse der Wissenschaft aus den Bereichen Elementarteilchenphysik, Neurowissenschaft, Mikrosystemtechnik und Materialwissenschaft (ENM). Im neuen »Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM)« von Schneider u. a. (2019) werden die Schwerpunkte noch detaillierter nach der DFG-Fachsystematik bis auf die Ebene der einzelnen Fächer aufgeteilt.[2] Das NEMO zugeordnete HPC-Kompetenzzentrum ENM unterstützt nach dieser die in Tabelle 1 dargestellten Schwerpunkte (vgl. auch die grafische Darstellung in Abbildung 1a).

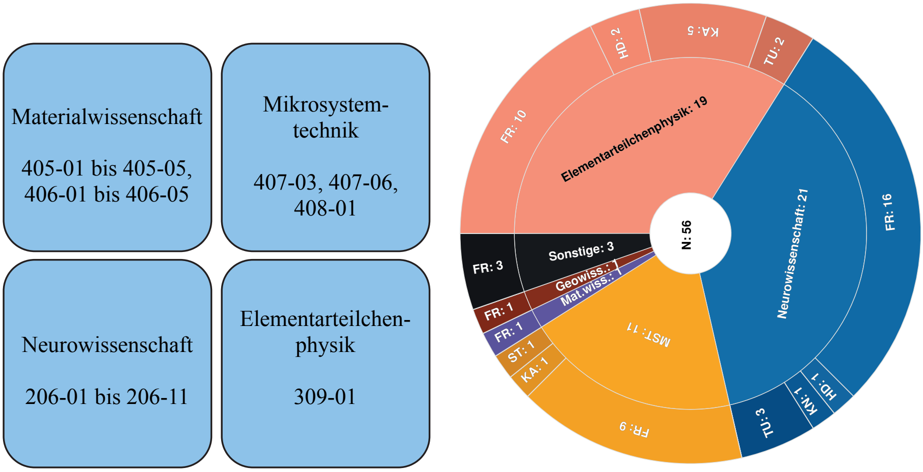
Die Schwerpunkte spiegeln sich ebenfalls in den Rechenvorhaben der Nutzer*innen wider. Ein Rechenvorhaben stellt dabei einen Projektantrag für Rechenressourcen dar. Es dient der Auswahl des Forschungsschwerpunkts für das Rechenvorhaben seitens des Antragstellers und der Einteilung zu einem Forschungscluster durch ein über alle Clusterstandorte agierendes Clusterauswahlteam (3 Governance). Abbildung 2a zeigt die angemeldeten Rechenvorhaben auf dem bwForCluster NEMO in den ersten zweieinhalb Betriebsjahren. Die Rechenvorhaben teilen sich auf die ursprünglichen drei Schwerpunkte Elementarteilchenphysik, Neurowissenschaft und

[2] Fächer nach DFG-Fachsystematik: http://www.dfg.de/dfg_profil/gremien/fachkollegien/liste/index.jsp (besucht am 18. 07. 2018).

Schwerpunkte	**DFG-Fachsystematik**	**Fächer**
Neurowissenschaft	206 Neurowissenschaft	206-01 – 206-11
Elementarteilchenphysik	309 Teilchen, Kerne und Felder	309-01
Mikrosystemtechnik	407 Systemtechnik 408 Elektrotechnik und Informationstechnik	407-03, 407-06 408-01
Materialwissenschaft	405 Werkstofftechnik 406 Materialwissenschaft	405-01 – 405-05 406-01 – 406-05

Tabelle 1: NEMO Schwerpunktbildung nach DFG-Fachsystematik.

Mikrosystemtechnik auf. Je ein Rechenvorhaben aus den Fachgebieten Materialwissenschaft und Geowissenschaften wurden von Shareholdern gestellt. Investitionen von Forschungsgruppen waren bereits beim Antrag des Clusters ein wichtiger Erfolgsfaktor, um eine Konsolidierung der ursprünglich dezentralen und in Eigenregie betriebenen Ressourcen der Fach-Communitys am Rechenzentrum der Universität Freiburg zu erreichen.[3]

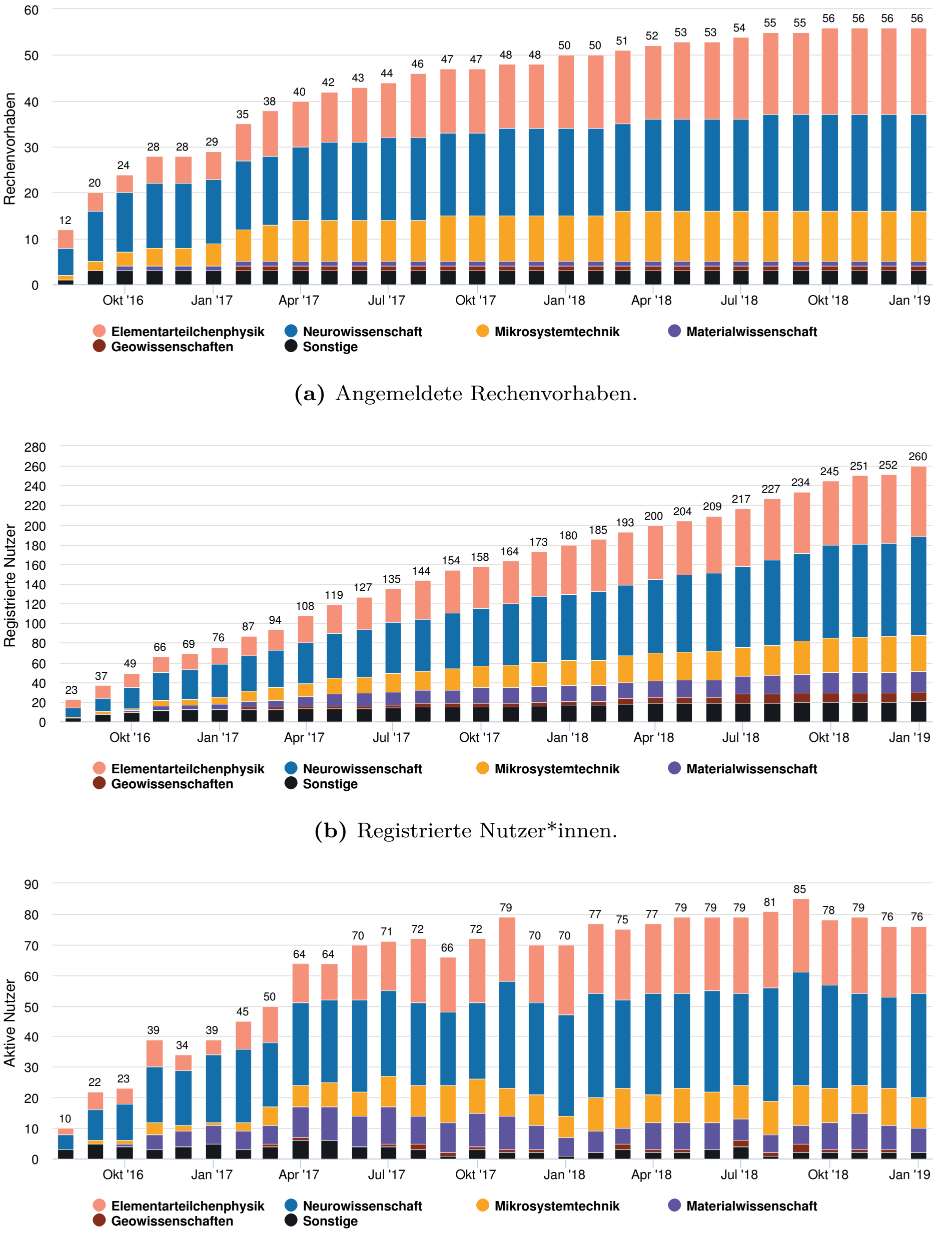


(a) Schwerpunktbildung nach DFG-Fachsystematik.

(b) Rechenvorhaben je Schwerpunkt und Standort.

Abbildung 1: Schwerpunktbildung des bwForClusters NEMO.

[3] Shareholder können alle Arbeitsgruppen aus dem Land werden und profitieren von einem für ihre Fachbereiche optimierten Forschungscluster.



(a) Angemeldete Rechenvorhaben.

(b) Registrierte Nutzer*innen.

(c) Aktive Nutzer*innen pro Monat.

Abbildung 2: Rechenvorhaben- und Nutzerentwicklung in den ersten zweieinhalb Jahren aufgeteilt nach Fachbereichen auf dem Forschungscluster NEMO.

Die Materialwissenschaft wie auch in kleinerem Maße die Geowissenschaften hatten, obwohl nur je ein Rechenvorhaben angemeldet wurde, einige registrierte (Abbildung 2b) und aktive (Abbildung 2c) Nutzer*innen. In der Materialwissenschaft lasteten die aktiven Nutzer*innen die Cluster-Ressourcen wahrnehmbar aus (Abbildung 8a). Dieses Profil führte dazu, dass das für den bwForCluster NEMO zuständige HPC-Kompetenzzentrum ENM im »Umsetzungskonzept II« um den Schwerpunkt Materialwissenschaft ergänzt wurde (Schneider u. a., 2019).

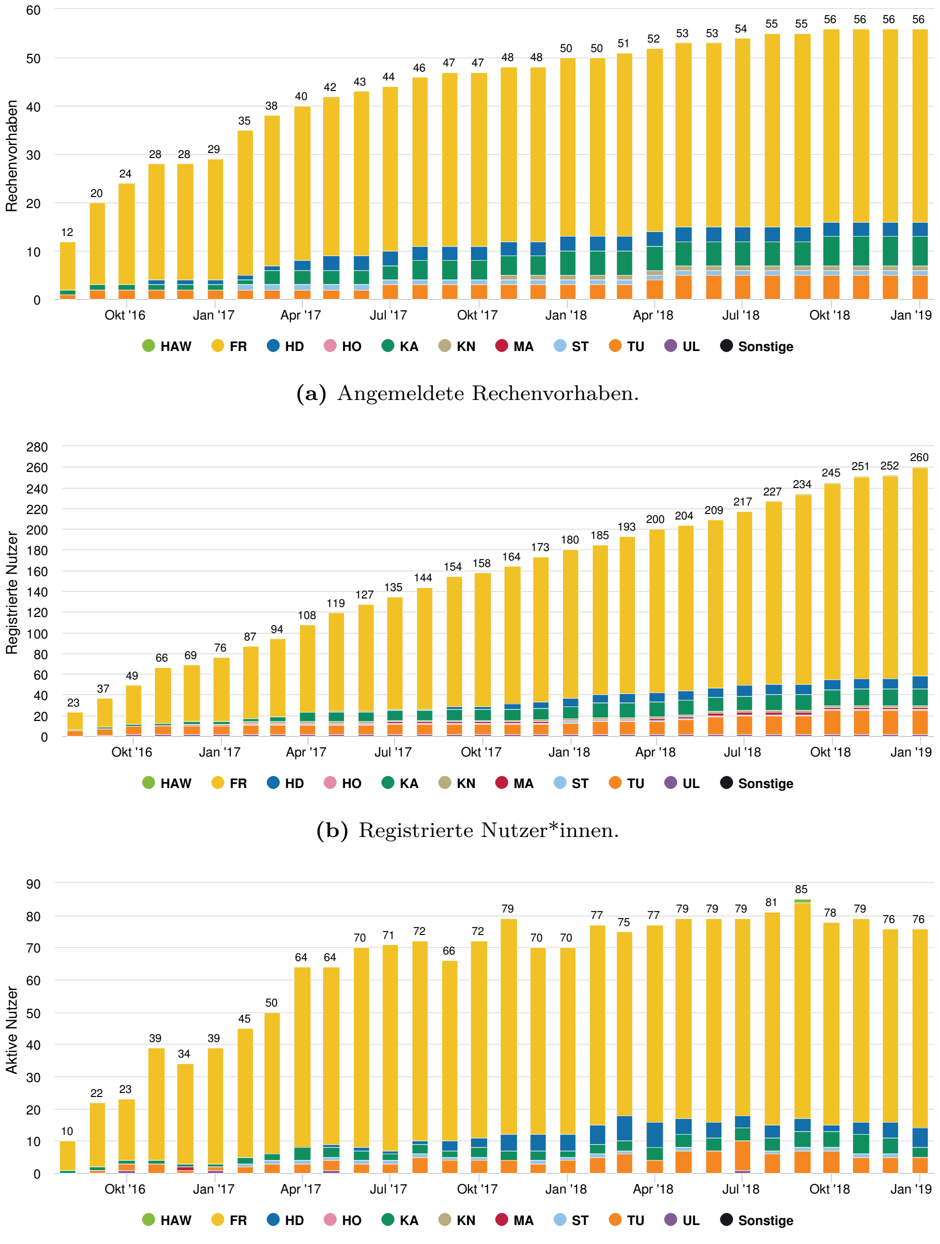
In Abbildung 2b wird deutlich, dass seit der offiziellen Inbetriebnahme des Clusters stetig neue Nutzer*innen dazu gekommen sind. Diese sind in einem hohen Maße aktiv. Abbildung 2c zeigt nur die Anzahl der Forschenden an, die mindestens einen Job im betreffendem Monat abgeschickt haben. In den ersten Monaten stieg diese Anzahl, bis sie sich in einem stabilen Fenster zwischen 70 und 80 aktiven Nutzer*innen pro Monat einpendelt. Diese Zahl dürfte wohl noch größer ausfallen, wenn Nutzer*innen innerhalb der Virtualisierten Forschungsumgebungen (VFU) einbezogen würden.[4]

Die Schwerpunktbildung in Baden-Württemberg erfolgte im Vorfeld des Clusterantrags nach den am Standort aktivsten wissenschaftlichen Communitys im HPC-Umfeld (Hartenstein u. a., 2013). Dies erklärt zudem die starke Konzentration der Rechenvorhaben aus Freiburg, wie in Abbildung 3a dargestellt. Die registrierten (Abbildung 3b) und aktiven (Abbildung 3c) Forschenden verteilen sich ähnlich. Die genaue Aufteilung der Rechenvorhaben auf die jeweiligen Felder und Standorte lässt sich Abbildung 1b entnehmen.

Die in Abbildung 8b gezeigte Auslastung verhält sich weniger eindeutig. Zwar stammt hier ebenfalls ein großer Teil der Auslastung aus Freiburg, aber insbesondere in den ersten Monaten überwiegen Wissenschaftler*innen aus Karlsruhe. Vergleicht man die Anzahl der aktiven Karlsruher Forschenden (Abbildung 3c) mit der Auslastung, fällt auf, dass teilweise ein einzelner Cluster-Nutzer bis zu einem Drittel des gesamten Clusters verwendet (Abbildung 4).

Diese Grafik stellt auch die Auslastung des Clusters der Nutzung durch Virtualisierte Forschungsumgebungen gegenüber. Diese decken sich insbesondere in den ersten Monaten mit der Auslastung durch Nutzer*innen aus Karlsruhe. Die VFU der »Compact Muon Solenoid Collaboration« (CMS) am CERN der Karlsruher Elementarteilchenphysik wurde bereits am vorherigen Testcluster aufgesetzt und lief

[4] Die VFU wird von einem Nutzer*innen der Arbeitsgruppe gestartet. Nutzer*innen innerhalb einer VFU sind für das System nicht sichtbar und werden von der Statistik nicht erfasst.



(a) Angemeldete Rechenvorhaben.

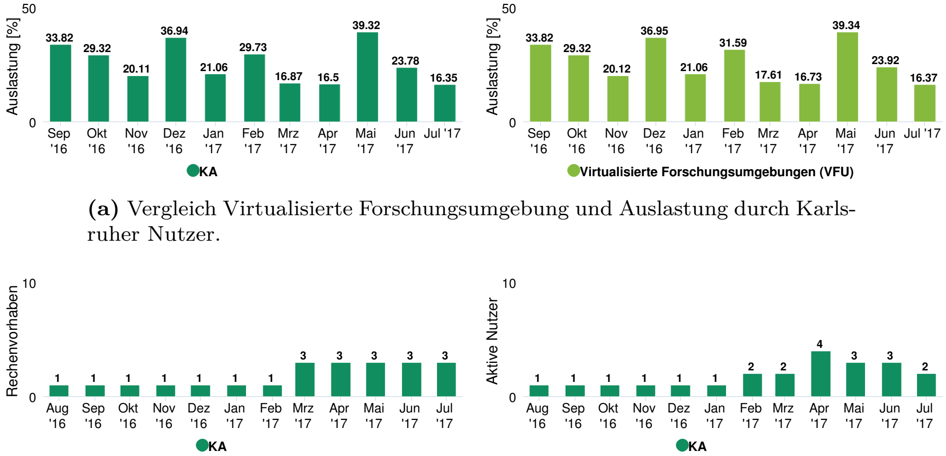
(b) Registrierte Nutzer*innen.

(c) Aktive Nutzer*innen.

Abbildung 3: Rechenvorhaben- und Nutzerentwicklung in den ersten zweieinhalb Jahren aufgeteilt nach Standorten auf dem Forschungscluster NEMO.

dort einige Monate stabil (Meier, 2017). Deshalb konnte die VFU CMS zu Beginn einen signifikanten Teil des Clusters auslasten (Abbildung 4a).

Die virtuelle Maschine, welche die VFU CMS zur Verfügung stellt, wird nur durch einen einzigen Nutzer gestartet. In der VFU können jedoch alle Karlsruher CMS-Gruppen rechnen. Deshalb ist der Vergleich aktive Nutzer*innen zu Auslastung des Clusters bei VFUs verzerrt. Die VFU wird mittelfristig auf einen VFU-Nutzer pro Wissenschafts- oder Arbeitsgruppe aufgeteilt werden. Eine direkte Abbildung der Cluster-Nutzer auf Forscher*innen in den jeweiligen VFUs ist zukünftig geplant. Spätestens bei der Nutzung von Containerisierungslösungen wie »Singularity« sind die Nutzer*innen in der Wirts- und Containerumgebung identisch.[5]



(a) Vergleich Virtualisierte Forschungsumgebung und Auslastung durch Karlsruher Nutzer.

(b) Rechenvorhaben und aktive Nutzer*innen aus Karlsruhe.

Abbildung 4: Vergleich der Statistiken für den Standort Karlsruhe mit der Auslastung durch VFUs im ersten Jahr von NEMO.

3 Governance

Für einen fairen Ausgleich der Interessen und einen reibungslosen Betrieb einer großen Forschungsinfrastruktur wie NEMO ist es wichtig, die Benutzer*innen frühzeitig, regelmäßig und in angemessener Art und Weise in Entscheidungsprozesse zu involvieren. Hierzu zählen sowohl anstehende Hardwareerweiterungen, Aufnahme neuer Shareholder, mögliche Erweiterungen der NEMO-Community oder Weiter-

[5] Singularity `https://www.sylabs.io/singularity` (besucht am 12. 02. 2019).

entwicklung des Betriebsmodells. Das Rechenzentrum als Betreiber des Forschungsclusters NEMO sieht sich in der Rolle des Dienstleisters der vier Fach-Communitys und greift hierfür auf deren Beratung und Vorschläge zurück. Aufgrund der hohen Anzahl an Beteiligten aus den ENM-Communitys wurde ein zweistufiges Modell aus großer Nutzerversammlung und kleinem Cluster-Beirat etabliert (Suchodoletz, Wiebelt und Janczyk, 2017). Die breit aufgestellte Nutzerversammlung, die einmal im Jahr tagen sollte, erlaubt es, ein Gesamtbild über Zufriedenheit und zukünftige Anforderungen aller involvierten Anwender*innen zu erhalten. In diesem Gremium erfolgen die Berichte durch das NEMO-Team, die Abstimmung der ENM-Communitys und die Vorstellung anstehender Entwicklungen.

Gleichzeitig werden aus den Reihen der wissenschaftlichen Communitys, der Betriebsgruppe und der Shareholder Vertreter in einen kleinen, handlungsfähigen Cluster-Beirat entsandt, der sich in halbjährlichen und bei Bedarf noch engeren Zyklen trifft und operative Belange des Clusters erörtert. Mitglieder des Cluster-Beirats sind Forschende der aktuell rechnenden Gruppen der ENM-Communitys, ein Vertreter des Landesnutzerausschusses (LNA-BW) sowie des NEMO Technical Advisory Boards (TAB), Vertreter der Shareholder sowie die operative Leitung des bwForCluster NEMO und bei Bedarf zusätzliche Expert*innen in beratender Funktion. Der Cluster-Beirat unterstützt die operative Leitung des bwForClusters NEMO in Belangen des Berichtswesens und der lokalen Governance sowie das HPC-Kompetenzzentrum ENM (Barthel u. a., 2019) inklusive dessen Entscheidungen im Cluster Auswahl Team (CAT). In den ersten zweieinhalb Jahren Laufzeit des Clusters tagte die Nutzerversammlung zwei Mal und der Cluster-Beirat fünf Mal.

Die NEMO-Governance funktioniert über eine enge Rückkopplungsschleife mit den Fach-Communitys, wie sie ähnlich bereits in der Antragsphase sowie zur Ausschreibung und Beschaffung erfolgte. Das dient gleichzeitig der Entlastung übergeordneter Ebenen wie Landesnutzerausschuss oder bwHPC-Lenkungskreis von ENM-spezifischen Belangen. Gleichwohl bleiben die übergeordneten Ebenen die letzte Instanz bei Problemen, die das bwHPC-Konzept als Ganzes betreffen, wie beispielsweise im Fall längerer Wartezeiten in der Queue (Wesner u. a., 2016).

Zu den Empfehlungen des Cluster-Beirats zählen die Einführung eines Technical Advisory Boards, die Einführung von sogenannten Memory-Knoten mit 256 bzw. 512 GiB RAM oder den Verzicht auf den Ausbau der XEON-Phi-Kapazitäten zugunsten klassischer Rechenknoten. Das NEMO-TAB umfasst die Administratoren beziehungsweise technikaffinen Mitglieder der einzelnen Forschungsgruppen und bespricht betriebliche Belange des Clusters, um diese dann kompakt in den Cluster-Beirat zu tragen.

4 Die Ausrichtung des Forschungsclusters

Bei der Beschaffung des Clusters wurden die Wünsche der im Förderantrag gesammelten Forschungs-Communitys als Grundlage genommen. Daraus wurde eine sehr einheitliche Hardwarekonfiguration destilliert, die bisher bei den Erweiterungen beibehalten wurde. Der Forschungscluster NEMO ist für sich gesehen die größte zusammenhängende Maschineninstallation am Rechenzentrum der Universität Freiburg. Hinzu kommen inzwischen weitere Compute-Systeme, die ebenfalls von der Abteilung eScience administriert werden. Bei den nutzenden Communitys der weiteren Systeme bestehen eine Reihe von Überschneidungen, so dass sowohl bei der Auswahl der Hardware als auch der Nutzung von Ressourcen wie Speichersystemen gemeinsame Interessen bestehen, die im Betriebsmodell abgebildet werden.

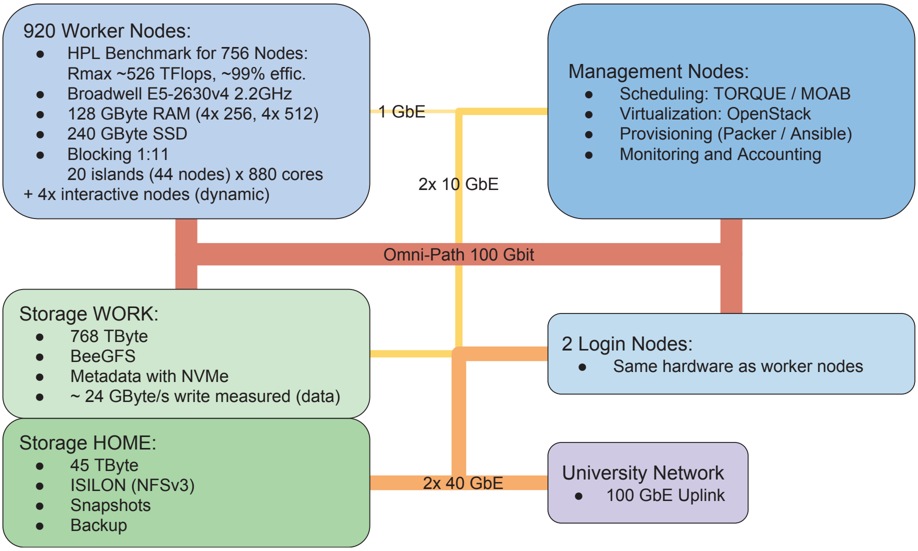

920 Worker Nodes:
● HPL Benchmark for 756 Nodes: Rmax ~526 TFlops, ~99% effic.
● Broadwell E5-2630v4 2.2GHz
● 128 GByte RAM (4x 256, 4x 512)
● 240 GByte SSD
● Blocking 1:11
20 islands (44 nodes) x 880 cores
+ 4x interactive nodes (dynamic)
Management Nodes:
● Scheduling: TORQUE / MOAB
● Virtualization: OpenStack
● Provisioning (Packer / Ansible)
● Monitoring and Accounting
1 GbE
2x 10 GbE
Omni-Path 100 Gbit
Storage WORK:
● 768 TByte
● BeeGFS
● Metadata with NVMe
● ~ 24 GByte/s write measured (data)
2 Login Nodes:
● Same hardware as worker nodes
Storage HOME:
● 45 TByte
● ISILON (NFSv3)
● Snapshots
● Backup
2x 40 GbE
University Network
● 100 GbE Uplink

Abbildung 5: Aufbau und Netzwerkanbindung des bwForClusters NEMO.

Abbildung 5 zeigt die schematische Darstellung des bwForClusters NEMO. In der derzeitigen Konfiguration besteht NEMO inzwischen aus 920 reinen Rechenknoten. Zu Beginn hatte der Cluster 748 Rechenknoten und wurde dann schrittweise auf 920 Knoten vergrößert. Vor diesen Erweiterungen wurde der MKL-optimierte »High Performance Computing Linpack Benchmark« (Intel Corporation, 2018) auf 756 identischen Rechenknoten durchgeführt und erreichte den Wert 525,714 TFlop/s bei einem theoretischen Maximalwert von 532,224 TFlop/s und einer Effizienz von etwa 98,78 %. Mit diesem Wert erreichte der Cluster im Juni des Jahres 2016 den Platz 214 der TOP500-Liste.[6]

Die Rechenknoten »Worker« verfügen jeweils über 128 GiB Hauptspeicher, einer SSD mit 240 GB Speicherplatz und einen 100 Gbit/s-Adapter für das Omni-Path-Hochleistungsnetzwerk. Jeweils 44 Maschinen sind zu einer Insel per Omni-Path und Gigabit-Ethernet verbunden. Jeder Omni-Path-Switch ist mit 4 × 100 Gbit/s an die Omni-Path-Spine-Ebene angebunden. Das ergibt einen Blocking-Faktor von 11:1. Pro Schrank sind zwei Inseln verbaut. Diese Konfiguration vermindert die Verkabelung, da nur wenige Kabel schrankübergreifend gezogen werden müssen.[7] Die meisten Kabel verbleiben innerhalb eines Schrankes. Der Cluster besteht aus 20 Inseln mit je 880 Kernen, die jeweils bei Bedarf non-blocking verwendet werden können. In der Praxis wird das von den Nutzer*innen aber nicht genutzt, da dann die Wartezeiten in der Jobqueue steigen. Aber auch ohne Non-Blocking-Konfiguration werden Jobs mit teilweise über 2000 Kernen auf NEMO gerechnet.

Der zentrale Parallelspeicher hatte zu Beginn 576 TB Speicherplatz, wurde aber bereits bei der ersten Erweiterung auf 768 TB vergrößert (nutzbare Kapazität). Die Metadaten liegen hierbei für den schnellen parallelen Zugriff auf NVMEs. Eingesetzt wird BeeGFS vom Fraunhofer ITWM.[8] Bei einem Test konnten vor der Erweiterung Nutzdaten mit über 24 GB/s übertragen und gespeichert werden.

4.1 Betriebsmodell

Das zugrundeliegende Betriebsmodell wurde auf eine effiziente Beschaffung, Inbetriebnahme und Erweiterbarkeit ausgelegt. Es nutzt hierzu das bereits länger etablierte Konzept des »Netzwerk-Bootens« (Schmelzer u. a., 2014), wodurch auf eine lokale Betriebssysteminstallation auf den einzelnen Knoten verzichtet werden kann.

[6] TOP500-Liste bwForCluster NEMO: `https://www.top500.org/system/178839` (besucht am 21. 08. 2018).

[7] Zwölf Glasfaserkabel werden pro Schrank herausgeführt, je vier Kabel pro Omni-Path-Switch und zwei pro Ethernet-Switch.

[8] Web-Präsenz des parallelen Cluster File Systems BeeGFS: `https://www.beegfs.io` (besucht am 12. 02. 2019).

Dem Laden und Starten des eigentlichen Betriebssystems ist ein Boot-Auswahl-Server vorgelagert, der es erlaubt in den Bootvorgang einzugreifen (Bauer, Messner u. a., 2019). Damit wird es möglich, Rechenknoten kurzfristig in einen anderen Betriebsmodus zu versetzen oder ohne großen Aufwand eine aktualisierte Softwareumgebung für NEMO auszuprobieren.

Im Rahmen des »ViCE-Projekts«[9] (Bauer, Suchodoletz u. a., 2019) wurde die Unterstützung von Virtualisierung auf Basis von OpenStack (Meier, 2017; Suchodoletz, Wiebelt, Meier u. a., 2017) etabliert. Das beinhaltet die Bereitstellung geeigneter Cloud-Infrastruktur-Komponenten und die Provisionierung der entsprechenden Softwarepakete auf den Rechenknoten. Die Virtualisierten Forschungsumgebungen der CMS- bzw. ATLAS-Gruppen (»A Toroidal LHC ApparatuS« Experiment am CERN) der Experimentellen Elementarteilchenphysik nutzen die auf NEMO verfügbare OpenStack-Infrastruktur und steuern diese mittels des Ressourcebrokers ROCED (Suchodoletz, Wiebelt, Meier u. a., 2017; Bührer u. a., 2018). Die Auslastung des Clusters NEMO durch VFUs in den ersten zweieinhalb Jahren ist in Abbildung 6 dargestellt.

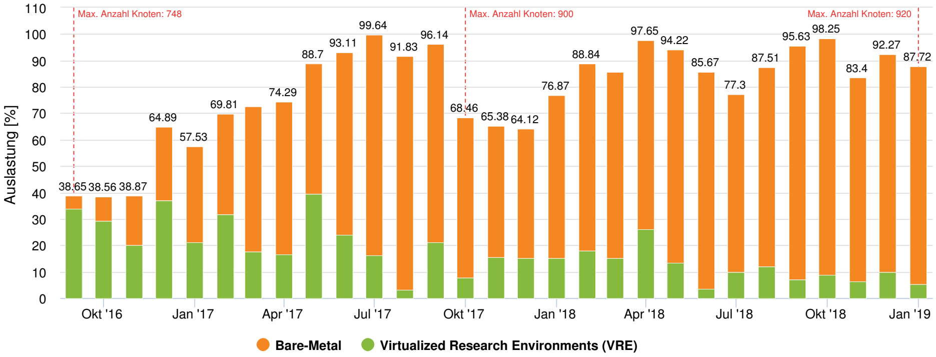


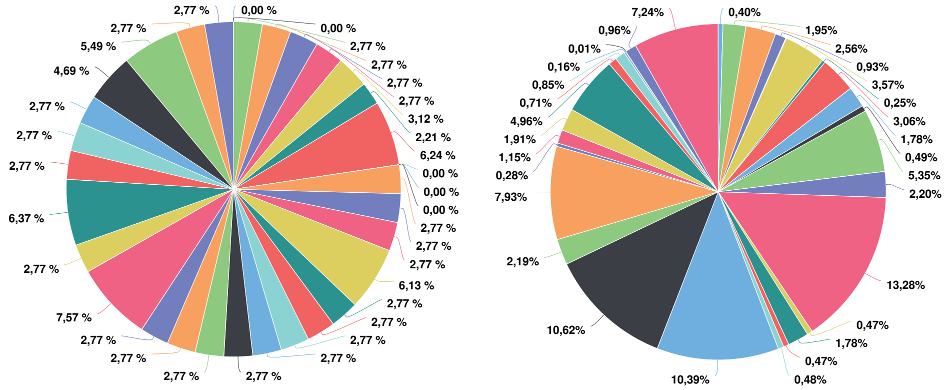
Abbildung 6: Auslastung des Clusters NEMO durch VFUs (Schätzung).

4.2 Stakeholder, Shareholder und Fairshare

Damit die Investitionen der Shareholder auf den Cluster abgebildet werden und gleichzeitig die Stakeholder aus den Forschungsschwerpunkten ihren Anteil nutzen können, muss die Clusternutzung kontingentiert werden. Hier spielt das »Fairshare-

[9] Gefördert im Rahmen der zweiten Linie der eScience-Initiative des Landes Baden-Württemberg.

Modell« eine Rolle (Wiebelt u. a., 2016).[10] Die Stake- und Shareholdergruppen bekommen einen Anteil am Cluster zugeteilt. Dabei wird der Stakeholderanteil auf die Arbeitsgruppen der ENM-Schwerpunkte aufgeteilt. Die Shareholder bekommen den Teil ihres Investments am Gesamtcluster zugeteilt. Diese Einstellung wird im Accounting des Schedulers konfiguriert. Derzeit beträgt der Shareholderanteil am Cluster etwa 28 %. Die Stakeholderanteile verteilen sich gleichmäßig auf die aktiven Rechenvorhaben. Im Monat Januar 2019 waren 26 Arbeitsgruppen auf Seiten der Stakeholder aktiv und hatten je 2,77 % Anteil am Cluster (»Fairshare Value«, Abbildung 7a).



(a) Anteile der Arbeitsgruppen im Januar 2019.

(b) Clusterauslastung der Arbeitsgruppen im Gesamtjahr 2018.

Abbildung 7: Die Anteile der jeweiligen Arbeitsgruppen an NEMO und deren Auslasung des Clusters im Jahr 2018.

Arbeitsgruppen können bis zur Höhe ihres Anteils den Cluster nutzen. Wenn sie unterhalb ihres Anteils liegen, bekommen die Jobs eine höhere Priorität in der Warteschleife und werden beim Scheduling bevorzugt. Liegen sie jedoch über ihrem Share, bekommen die Jobs negative Prioritäten, werden aber trotzdem in die Warteschlange eingefügt und können nach Abarbeitung der höher priorisierten Jobs anlaufen.

Die Anteilseigner geben der Gemeinschaft ihre Anteile. Diese kann damit diese Anteile mitbenutzen sogar über ihren eigenen Share hinaus. Die Anteilseigner können dann wiederum bei Bedarf über ihren Anteil hinaus den Cluster verwen-

[10] Vgl. Abschnitt 5.3, Seite 419 ff. der Scheduler-Dokumentation von Adaptive Computing Enterprises, Inc (2018).

den. Damit ein langes Ansparen von Rechenzeit nicht möglich ist, wird der aktuelle Fairshare-Wert jeweils über die letzten drei Monate berechnet. Dabei werden 32 Zeitschritte zu je drei Tagen mit einem Verfallsfaktor von 0,95 je Schritt verwendet.

Ein Beispiel zur Berechnung ist in Auflistung 1 dargestellt. Das aktuelle Intervall wird voll eingerechnet, während die drei Tage zuvor nur noch zu 95 % eingerechnet werden. Dabei entspricht »Target« dem Anteil der Arbeitsgruppe und (%) dem derzeitigen Fairshare-Wert der Gruppe.

```
FSInterval     % Target    0     1     2     3     4     5     6     7 ..
FSWeight     ---    --- 1.00  0.95  0.90  0.85  0.81  0.77  0.73  0.69 ..
AGx         9.92   2.77 9.50 11.21 10.73  9.86  9.75  9.68  9.88  9.71 ..
```

Auflistung 1: Ausgabe eines beispielhaften Fairshare-Werts inkl. Intervallen mit dem Kommando `mdiag -f`.

Die aktuelle Verteilung der Stake- und Shareholder ist in Abbildung 7a dargestellt. Vergleicht man diese Aufteilung mit der tatsächlichen Verteilung bei der Auslastung im Jahr 2018 (Abbildung 7b), kann man das Prinzip des Fairshare gut erkennen. Arbeitsgruppen, die einen permanent hohen Rechenbedarf haben, profitieren davon, dass nicht alle Gruppen ständig ihren vollen Anteil ausschöpfen. Im Ausgleich dazu können Arbeitsgruppen mit stark schwankendem Rechenbedarf schneller bedient werden oder punktuell deutlich mehr Ressourcen bekommen, als es ihrem Anteil zusteht. Die inaktiven Rechenvorhaben werden ohne Anteile dargestellt (0,00 %).

Sollten in Zukunft zusätzliche Steuerungsmöglichkeiten notwendig werden, stehen mit »Preemption« (Adaptive Computing Enterprises, Inc, 2018, Kapitel 21) oder »Rollback Reservations« (Adaptive Computing Enterprises, Inc, 2018, Abschnitt 6.6.2) zwei Möglichkeiten zur Verfügung, Quality-of-Service-Anforderungen im Scheduling durchzusetzen.

4.3 NEMO Erweiterungen

NEMO wurde bereits in der Antragsphase um Eigenanteile von Forschungsgruppen erweitert, die am Standort Freiburg ebenfalls in die vorgenannte Zuordnung fielen. Zudem gab und gibt es erneute Beteiligungen nach Inbetriebnahme, wie beispielsweise das FIT[11] oder die ATLAS-Arbeitsgruppen, welche die Hochleistungsrechenressource durch eigene finanzielle Beteiligungen vergrößert und sich damit erweiterte Nutzungsrechte erworben haben (Suchodoletz, Wesner u. a., 2016).

Eine weitere, geplante Entwicklungsrichtung besteht in der Einrichtung von Visualisierungsknoten, die eine entfernte grafische Ausgabe auf den Geräten der Forschenden erlauben, ohne hierzu signifikante Anforderungen an diese zu stellen. Um ein leichtes Deployment und die Koexistenz verschiedener Gruppen zu erlauben, werden Container-basierte Ansätze im Rahmen des ViCE-Projekts entwickelt und erprobt.

Neu seit 2019 ist eine Maschine mit zwei Höheneinheiten einer AMD-CPU mit 32 Kernen sowie acht NVIDIA Tesla V100 Grafikbeschleunigern. Diese wird nach einer kurzen Testphase der NEMO-Community für Machine-Learning-Applikationen zur Verfügung gestellt werden.

4.4 Betriebsstatistiken

Ein wesentlicher Qualitätsmaßstab für die Forscher*innen ist die Wartezeit bis zum Start der eigenen Jobs. Durch ein strategisches Herangehen bei der Beschaffung und die Nutzung der Option der Aufnahme neuer Shareholder durch Aufwuchsfinanzierung konnte eine langfristig gute Auslastung mit regelmäßigen Erweiterungen des Systems verbunden werden. So wurden schrittweise in den letzten zweieinhalb Jahren zusätzliche Kapazitäten geschaffen, welche die Auslastung in einem für die Forschenden vorteilhaften Rahmen hielt.[12]

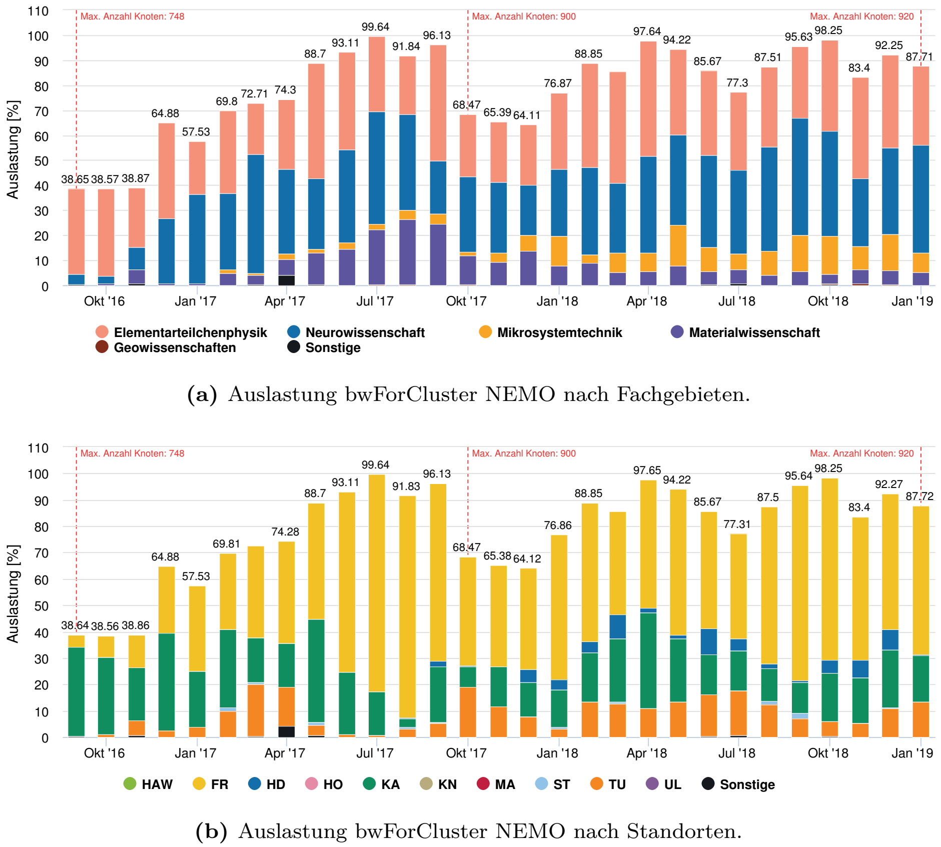
Abbildung 8 stellt die Auslastung des Clusters der ersten zweieinhalb Jahre aufgeschlüsselt nach Fachbereichen (8a) und nach Standorten (8b) dar. Zu Beginn bestand der Cluster aus 748 Rechenknoten. Im Oktober 2017 wurde der Cluster in einer ersten Erweiterung um 152 auf insgesamt 900 Rechenknoten vergrößert. Die Auslastung fiel dabei wieder auf ein niedrigeres Niveau, da mehr Knoten zur Verfügung standen, die die Forschenden zunächst wieder auslasten mussten.[13] Eine weitere kleinere Erweiterung erfolgte schließlich im Januar 2019 um 20 weitere Rechenknoten. Alle Erweiterungen erfolgten mit Rechenknoten des gleichen Typs, um

[11] Freiburger Zentrum für interaktive Werkstoffe und bioinspirierte Technologien, `http://www.fit.uni-freiburg.de` (besucht am 12. 02. 2019).

[12] Durch Erweiterungen konnte die Warteschlange der Jobs jeweils verkürzt werden.

[13] Bei der Erweiterung im Oktober 2017 kam es beim Update der BeeGFS-Server zu Problemen, so dass der Cluster mehrere Tage offline war.

die Aufgaben auf der administrativen Seite zu vereinfachen. Es gab lediglich eine Erweiterung des Hauptspeichers bei acht Rechenknoten und die Beschaffung des Machine-Learning-Knotens mit acht NVIDIA Tesla V100 Grafikbeschleunigern.[14]



(a) Auslastung bwForCluster NEMO nach Fachgebieten.

(b) Auslastung bwForCluster NEMO nach Standorten.

Abbildung 8: Auslastung des bwForClusters NEMO in den ersten zweieinhalb Jahren.

5 Aktuelle und zukünftige Entwicklungen

Dynamische Entwicklungen im »Scientific Computing« generieren fortlaufend neue Anforderungen an den Betrieb großer Forschungsinfrastrukturen wie NEMO. Der

[14] Eine Insel mit 44 Rechenknoten wurde bereits zu Beginn mit einem Mainboard gekauft, das doppelt so viele Speicherbausteine aufnehmen kann, um diese Erweiterungen zu ermöglichen. Dabei wurden alle RAM-Riegel wiederverwendet (4 × 512 GiB ergibt 4 × 256 GiB ohne Zusatzkosten).

sich reduzierende technologische Abstand und die Wahrnehmung seitens der Nutzer*innen lösen die klassische Dichotomie von Cloud und HPC zusehends auf. Die Softwareumgebungen gleichen sich vermehrt und für spezielle Aufgaben, wie Pre- oder Post-Processing oder »Remote Visualization« wird die Cloud als Compute-Plattform zunehmend relevant.

Das steigende Interesse am maschinellen Lernen in immer mehr Fachdisziplinen führt dazu, dass das »General-purpose computing on graphics processing units« (GPGPU) immer stärker nachgefragt wird, weshalb Grafikbeschleuniger einer breiteren Nutzerschicht – zu Beginn auch zu Evaluations- und Testzwecken – zugänglich gemacht werden sollten. Deshalb wurde 2019 ein erster Knoten für das Machine-Learning beschafft.

Mit dem steigenden Umfang von Forschungsdaten in einzelnen Projekten und im Wissenschaftsbetrieb insgesamt wird die Lokalität der Daten wieder relevant, da es sehr ineffizient sein kann, große Datenmengen für verhältnismäßig kleine Berechnungen über lange Strecken zu kopieren. Dieses unter dem Stichwort »Data Intensive Computing« beschriebene Phänomen erfordert ein verstärktes »Zusammendenken« der Forschungsinfrastrukturen Compute und Datenhaltung (Schneider u. a., 2019) und wird am NEMO-Standort durch die Beteiligung am Infrastrukturprojekt bwSFS (»Storage for Science«) gemeinsam mit dem BinAC-Standort Tübingen vorangetrieben (Suchodoletz, Hahn u. a., 2019).

Durch Anbinden zusätzlicher lokaler Wissenschaftsspeicher wie bwSFS lassen sich cluster-lokale Parallelspeicher wie BeeGFS als schnelle, kurzfristige Zwischenspeicher verwenden. Arbeitsordner können mehr als bisher rein zu Berechnungen genutzt werden und nach Jobende mit Metadaten versehen an weitere Speicher weitergeleitet werden. Dadurch muss der am Cluster direkt angeschlossene Parallelspeicher nicht mehr so groß bemessen werden und kann bereits bei kleinen und mittelgroßen Clustern aus Solid State Disks bestehen.

Da sich auf Tier-Ebene 3 die Architektur auf X86 beschränkt und konkrete Vorhersagen für langfristige Bedarfe einzelner Compute-Umgebungen schwer zu treffen sind, arbeitet das Betriebsteam am Standort Freiburg an zukünftigen Deployment-Modellen (Bauer, Messner u. a., 2019), die von einem sehr einfachen Basis-Setup eines Rechenknotens ausgehen. Hierzu wird dieser wie bisher über das lokale Netzwerk »gebootet« und mit einer sehr schlanken Softwareausstattung versehen. Gleichzeitig wird der Knoten in die jeweiligen Netzwerkumgebung mit Zugriff auf die relevanten

Ressourcen versetzt. Davon abhängig sollen die gebooteten Knoten in einem weiteren Schritt für eine HPC- oder Cloud-Nutzung konfiguriert werden. Zusätzliche Softwaremodule für konkretere Nutzungsszenarien, wie der Einsatz von GPUs oder die Verwendung als Login-Knoten werden bedarfsbezogen in einem weiteren Schritt dynamisch nachgezogen.

Für die HPC-Knoten ist angedacht, in weiteren Schritten die von den Forschenden erwarteten Softwareumgebungen sowohl über das traditionelle Modulsystem jedoch zunehmend auch durch Containerisierung verfügbar zu machen. Gerade durch Letzteres können individuelle Forschungsgruppen ihre Rechenumgebungen selbst zusammenstellen. Sie können diese einer eigenen Versionskontrolle unterstellen und damit die Reproduzierbarkeit ihrer Forschungsworkflows verbessern.

Weiterhin wird für HTC-Jobs auf einen verstärkten Einsatz von Cloud-Ressourcen gesetzt. Dieses erlaubt deutlich längere Walltimes[15] als sie in der aktuellen HPC-Umgebung von NEMO gewährt werden können. Ebenso bieten sich Cloud-Ressourcen für interaktive Jobs an. Aus der verstärkten administrativen Zusammenführung der Ressourcen entstehen neue Herausforderungen im Scheduling und in der Abrechnung. Hier besteht noch Forschungs- und Entwicklungsbedarf, der im Zuge des Projektes bwHPC-S5 angegangen wird (Barthel u. a., 2019).

In den aktuellen Überlegungen wird dabei noch nicht an eine automatische Rekonfiguration des Gesamtsystems gedacht. Jedoch soll eine deutlich höhere Dynamisierung der Ressourcenzuteilung je nach aktuellen Projekten und Anforderungen seitens der Forschenden erreicht werden. Nicht genutzte Cloud-Ressourcen können beispielsweise dafür verwendet werden, um eine sich aufgestaute Queue im HPC abzuarbeiten. Zusätzlich erlaubt die dynamische Ressourcenzuteilung eine verstärkte Berücksichtigung von Green-IT-Elementen. So lässt sich nicht nur eine schnelle Integration neuer Knoten ins Gesamtsystem erreichen, sondern auch eine verbesserte »Packung« der verschiedenen Compute-Jobs auf dem Gesamtsystem. Partielle Unterauslastungen können vermieden werden und die durch Konsolidierung frei gezogenen Knoten temporär bei Nicht-Nutzung außer Betrieb nehmen. So wurden bereits temporär zusätzliche Ressourcen NEMO zur Verfügung gestellt, um längere Queues unter der Woche abzuarbeiten.[16] Dabei müssen die Geldgeber und deren Anforderungen jeweils berücksichtigt werden, so dass in der Endabrechnung die Kontingente der einzelnen Parteien wieder stimmen.

[15] Gesamtlaufzeit eines Jobs.

[16] Vergleiche beispielsweise Monate Juli 2017 oder Oktober 2018 in Abbildung 8.

6 Fazit und Ausblick

Mit der Zunahme verteilter Landesinfrastrukturen (Schneider u. a., 2019) – Freiburg arbeitet gemeinsam mit den Kollegen aus Tübingen an der Bereitstellung eines größeren an die HPC-Cluster angedockten Speichersystems mit Forschungsdatenmanagementkomponente (bwSFS) – werden Fragen der Steuerung und Abstimmung relevanter. Hier kann die erfolgreiche Governance auf Landesebene, insbesondere auch die von NEMO eine Vorlage bieten.

Das Basissystem des Clusters wurde nicht nur als stabile Softwareumgebung für die rechnenden Communitys genutzt, sondern bildete ebenso die Grundlage für Experimente und das Sammeln von Erfahrungen im Umgang mit Virtualisierten Forschungsumgebungen im Rahmen von ViCE (Meier, 2017). Die VFUs erlauben Forschenden eine komplett eigene Softwareumgebung zu nutzen, wie sie beispielsweise für bestimmte Rechnungen der ATLAS- oder CMS-Gruppen notwendig ist. Als Alternative zur vollständigen Virtualisierung bietet sich die Containerisierung an, die ebenfalls eigene Softwareerweiterungen oder komplette Forschungsumgebungen ermöglicht. Diese könnten in zukünftigen Betriebsmodellen eine stärkere Konvergenz von HPC und Cloud für das High Throughput Computing erlauben. Diese Überlegungen lassen sich zudem auf zukünftige wissenschaftliche Workflows anwenden, in denen verschiedene Forschungsinfrastrukturen nacheinander genutzt werden und beispielsweise ein Pre- oder Post-Processing in der Cloud ebenso vorsehen wie die Remote-Visualisierung auf einem spezialisierten System.

Mit der Entwicklung des Boot-Auswahl-Servers werden die bereits mit ViCE angefangenen Optionen flexibler Betriebsmodelle weitergeführt (Bauer, Messner u. a., 2019). Die Basiskonfiguration nutzt dabei Entwicklungen aus anderen Landesprojekten wie bwLehrpool (Suchodoletz, Münchenberg u. a., 2014) nach. Das »Distributed Network Block Device Version 3« (DNBD3) bietet spezielle Funktionalität für die performante Versorgung einer großen Zahl von Cluster-Knoten mit einem Root-Filesystem als auch für das Failover für den Fall der Nichterreichbarkeit eines DNBD3-Servers im Verbund (Rettberg u. a., 2019).

Auch für die Restlaufzeit von NEMO ist weiterhin eine schrittweise Erneuerung durch abgestimmte Ersatzinvestitionen und Äquivalenztausch[17] angedacht. Die ur-

[17] In dem Sinne, dass Erweiterungen beispielsweise in Form von Arbeitsspeicher für bestehende Systeme erfolgen, wenn hier der größte Bedarf und gemeinsame Nutzen besteht. Hierfür erhält der Investierende Anteile aus dem bestehenden Gesamtpool statt beispielsweise neue Knoten hinzuzufügen.

sprünglich mehr als zwei Jahre stabil gehaltene Hardware-Landschaft wird dabei diverser; sowohl die Bereitstellung des GPGPUs als Compute-Alternative als auch die Aufnahme der AMD-Plattform und eine Anzahl von Memory-Knoten macht die Auswahl für die Forschenden größer. Anstehende Technologiewechsel werden bei zukünftigen Erweiterungen entsprechend berücksichtigt und mit den Share- und Stakeholdern abgestimmt.

Danksagungen

An dieser Stelle möchten die Autoren dem Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg und der DFG für die finanzielle Förderung des bwForClusters NEMO und die hervorragende Unterstützung der Projekte ViCE und bwHPC-S5 danken.

Korrespondenzautor

Michael Janczyk: michael.janczyk@rz.uni-freiburg.de
eScience Abteilung, Rechenzentrum Albert-Ludwigs-Universität Freiburg,
Hermann-Herder-Str. 10, 79104 Freiburg, Deutschland

ORCID

Michael Janczyk https://orcid.org/0000-0003-4886-736X
Dirk von Suchodoletz https://orcid.org/0000-0002-4382-5104
Bernd Wiebelt https://orcid.org/0000-0003-2771-4524

Literatur

Adaptive Computing Enterprises, Inc (2018). *Moab Workload Manager. Administrator Guide 9.1.3.* Administrator Guide. Version 9.1.3. Adaptive Computing Enterprises, Inc. URL: http://docs.adaptivecomputing.com/9-1-3/MWM/Moab-9.1.3.pdf (besucht am 12. 02. 2019).

Barthel, R. und J. Salk (2019). »bwHPC-S5: Scientific Simulation and Storage Support Services. Unterstützung von Wissenschaft und Forschung beim leistungsstarken und datenintensiven Rechnen sowie großskaligem Forschungsdatenmanagement«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 17–28. DOI: `10.15496/publikation-29039`.

Bauer, J., M. Messner u. a. (2019). »A Sorting Hat For Clusters. Dynamic Provisioning of Compute Nodes for Colocated Large Scale Computational Research Infrastructures«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 217–229. DOI: `10.15496/publikation-29055`.

Bauer, J., D. von Suchodoletz, J. Vollmer und H. Rasche (2019). »Game of Templates. Deploying and (re-)using Virtualized Research Environments in High-Performance and High-Throughput Computing«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 245–262. DOI: `10.15496/publikation-29057`.

Bührer, F. u. a. (2018). »Dynamic Virtualized Deployment of Particle Physics Environments on a High Performance Computing Cluster«. In: *Computing and Software for Big Science.* arXiv: `1812.11044 [physics.comp-ph]`.

Hartenstein, H., T. Walter und P. Castellaz (2013). »Aktuelle Umsetzungskonzepte der Universitäten des Landes Baden-Württemberg für Hochleistungsrechnen und datenintensive Dienste«. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 36.2. DOI: `10.1515/pik-2013-0007`.

Intel Corporation (2018). *Intel Math Kernel Library for Linux. Intel MKL 2019 – Linux.* Developer Guide. Version 2019, Revision 065. Intel Corporation. URL: `https://software.intel.com/sites/default/files/mkl-2019-developer-guide-linux.pdf` (besucht am 08. 02. 2019).

Meier, K. (2017). »Infrastrukturkonzepte für virtualisierte wissenschaftliche Forschungsumgebungen«. Diss. Albert-Ludwigs-Universität Freiburg im Breisgau.

Rettberg, S., D. von Suchodoletz und J. Bauer (2019). »Feeding the Masses: DNBD3. Simple, efficient, redundant block device for large scale HPC, Cloud and PC pool installations«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 231–243. DOI: `10.15496/publikation-29056`.

Schmelzer, S., D. von Suchodoletz, M. Janczyk und G. Schneider (2014). »Flexible Cluster Node Provisioning in a Distributed Environment«. In: *Hochleistungsrechnen in Baden-Württemberg. Ausgewählte Aktivitäten im bwGRiD 2012.* Beiträge zu Anwenderprojekten und Infrastruktur im bwGRiD im Jahr 2012. Hrsg. von J. C. Schulz und S. Hermann. KIT Scientific Publishing, Karlsruhe, S. 203–219. ISBN: 978-3-7315-0196-1. DOI: 10.5445/KSP/1000039516. URN: urn:nbn:de:0072-395167.

Schneider, G. u. a. (2019). »Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM)«. Gekürzte Fassung. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 3–16. DOI: 10.15496/publikation-29040.

Suchodoletz, D. von, U. Hahn, B. Wiebelt, K. Glogowski und M. Seifert (2019). »Storage infrastructures to support advanced scientific workflows. Towards research data management aware storage infrastructures«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Hrsg. von M. Janczyk, D. von Suchodoletz und B. Wiebelt. TLP, Tübingen, S. 263–279. DOI: 10.15496/publikation-29058.

Suchodoletz, D. von, J. Münchenberg u. a. (2014). »bwLehrpool – ein landesweiter Dienst für die Bereitstellung von PC-Pools in virtualisierter Umgebung für Lehre und Forschung«. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 37.1, S. 33–40. DOI: 10.1515/pik-2013-0046.

Suchodoletz, D. von, S. Wesner und G. Schneider (2016). »Überlegungen zu laufenden Cluster-Erweiterungen in bwHPC«. In: *Kooperation von Rechenzentren: Governance und Steuerung – Organisation, Rechtsgrundlagen, Politik.* Hrsg. von D. von Suchodoletz, J. C. Schulz, J. Leendertse, H. Hotzel und M. Wimmer. De Gruyter, S. 331–342. ISBN: 978-3-11-045888-6. DOI: 10.1515/9783110459753.

Suchodoletz, D. von, B. Wiebelt und M. Janczyk (2017). »bwHPC Governance of the ENM community«. In: *Proceedings of the 3rd bwHPC-Symposium.* (2016). Hrsg. von S. Richling, M. Baumann und V. Heuveline. Heidelberg: heiBOOKS. DOI: 10.11588/heibooks.308.418.

Suchodoletz, D. von, B. Wiebelt, K. Meier und M. Janczyk (2017). »Flexible HPC: bwForCluster NEMO«. In: *Proceedings of the 3rd bwHPC-Symposium.* (2016). Hrsg. von S. Richling, M. Baumann und V. Heuveline. Heidelberg: heiBOOKS. DOI: 10.11588/heibooks.308.418.

Wesner, S., T. Walter, B. Wiebelt, D. von Suchodoletz und G. Schneider (2016). »Strukturen und Gremien einer bwHPC-Governance – Momentaufnahmen und Perspektiven«. In: *Kooperation von Rechenzentren Governance und Steuerung – Organisation, Rechtsgrundlagen, Politik.* Hrsg. von D. von Suchodoletz, J. C. Schulz, J. Leendertse, H. Hotzel und M. Wimmer. de Gruyter, S. 315–329. ISBN: 978-3-11-045888-6. DOI: 10.1515/9783110459753-027.

Wiebelt, B. u. a. (2016). »Strukturvorschlag für eine bwHPC-Governance der ENM-Community«. In: *Kooperation von Rechenzentren Governance und Steuerung – Organisation, Rechtsgrundlagen, Politik.* Hrsg. von D. von Suchodoletz, J. C. Schulz, J. Leendertse, H. Hotzel und M. Wimmer. de Gruyter, S. 343–354. ISBN: 978-3-11-045888-6. DOI: 10.1515/9783110459753-029.

Data Analysis for Improving High-Performance Computing Operations and Research

An Eucor Seed Money Project

Florina M. Ciorba[*] Gerhard Schneider[†] Dirk von Suchodoletz[†]
Aurélien Cavelan[*] Thierry Sengstag[§] Sabine Gless[¶]
Nicolas Lachiche[‡] Ahmed Samet[‡]

[*]Department of Mathematics and Computer Science, University of Basel, Switzerland
[†]eScience, University of Freiburg, Freiburg, Germany
[§]Scientific Computing Center (sciCORE), University of Basel, Switzerland
[¶]Faculty of Law, University of Basel, Switzerland
[‡]ICube Laboratory, University of Strasbourg, France

This work addresses the challenges associated with analysis of data generated by high-performance computing (HPC) systems under data protection and privacy requirements. The HPC systems are the workhorse of simulation science, enabling unique insights across many disciplines (climate modeling, life sciences, weather forecast, etc.). System monitoring and analysis of monitoring data are highly significant for the efficient operation and research in performance optimization of HPC systems. Such systems generate various and large volumes of data as they operate, constituting a case of Big Data that challenges key data protection and privacy principles. This paper describes the *Data Analysis for Improving High Performance Computing Operations and Research* (DA-HPC-OR) project funded through the Eucor – The European Campus EVTZ via the Seed Money program[1]. The main goal in this project is the analysis of data collected since July 2016 on the HPC system (NEMO) at the University of Freiburg in order to improve their

[1] www.eucor-uni.org

DOI: 10.15496/publikation-29042

research and operations activities. Data collected on the sciCORE cluster in Basel will be used to validate the knowledge extracted from NEMO. This knowledge will be used to improve the monitoring, operational, and research activities of the three HPC systems (Freiburg, Basel, and Strasbourg). Data protection requires legal monitoring of the relevant Swiss, German, and EU legislation. Compliance with such laws will be ensured via data de-identification and anonymization prior to analysis. We leverage the HPC, legal, and data analysis expertise of the consortium to develop solutions that can be transferred to other Eucor members at no additional legislative inquiries or overheads.

1 Introduction

Each of the four pillars (experiments, theory, simulation, and data) of the scientific method produce and consume large amounts of data. Breakthrough science will occur at the interface between empirical, analytical, computational, and data-based observation. Parallel computing systems are the workhorse of the third pillar: simulation science. These systems are highly complex ecosystems, with multiple layers, ranging from the hardware to the application layer (cf. Figure 1). Monitoring solutions exist at every single layer.

Access to the monitoring data varies between the different communities, which also have different interests in the data. For instance, computational scientists in general have access to the monitoring data from the application layer and in certain cases also from the application environment layer. Their interests may include the running time of their applications but also understanding the application performance by profiling or tracing. Computer scientists may have access to monitoring data from application environment and cluster software layers. They are typically interested in performance data from both software and hardware components and subsystems. System administrators typically have access to data monitored at the hardware, system software, and cluster software layers. The monitoring interests cover both operational aspects of the system (e. g., availability, fair usage), as well as research-oriented aspects of the system (e. g., scheduling of batch jobs, resource utilization, fault-tolerance).

Regardless of the community interest, the access to and the sharing of monitoring data of HPC systems for analysis purposes requires (i) non-disclosure agreements between the data owners and/or (ii) de-identification and anonymization of the sensitive information within the data. The former requirement prevents any public release of data, thus hindering the reproducibility of the insights and of the research results derived from the data. The latter requirement imposes compliance with the data protection and privacy regulations in force at the location where the data is produced and collected.

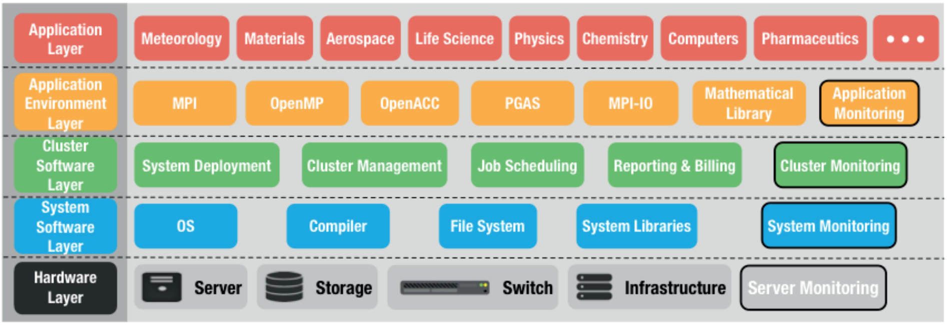


Figure 1: A typical HPC ecosystem with a layer-based monitoring approach

1.1 Goals and Expected Results

The focus of this project is on satisfying the second requirement (de-identification and anonymization that complies with data protection and privacy regulations) for the data produced and collected on the systems of the HPC centers at the member institutions: NEMO at University of Freiburg (NEMO-UniFR), sciCORE at University of Basel (sciCORE-UniBas), and HPC at University of Strasbourg (HPC-UniStra). This focus raises important legal and regulatory questions regarding the data protection and privacy laws in force in Germany, Switzerland, and France that this project needs to consider and comply with. Answering these questions requires legal expertise and a thorough analysis of the applicable individual, national, and European legislation.

1.1.1 Goal

The goal of this project is to analyze the (de-identified and anonymized) data collected at one of the three HPC centers of the consortium (NEMO-UniFR) to improve their research and operations activities, as well as offer monitoring, operational, and research insights for improving the activities at the other two HPC centers (sciCORE-UniBas and HPC-UniStra). The rationale for concentrating on NEMO-UniFR is due to the monitoring and data integration activities running at UniFR since August 2016.

1.1.2 Approach

The approach towards achieving this goal involves: Monitoring of software and hardware components; Use of de-identification and anonymization methods; and Analysis of the data processed in the previous step. Monitoring data is collected (in a first step), under various types, formats, and sizes. Therefore, meaningful integration of the various types and formats represents a significant challenge. This challenge can be addressed by ensuring that the HPC monitoring data follows the FAIR (findable, accessible, interoperable, and reusable) data principles (Wilkinson et al., 2016) already in the data collection stage.

To comply with the data protection and privacy laws of the three partner countries, the monitoring data needs to be de-identified and anonymized (in a second step). The de-identification and anonymization also need to preserve the data usefulness. This can be addressed by extending the FAIR principles with another U (usefulness) principle, resulting in FAIRU.

Via a meaningful analysis (using well established data mining methods) of the FAIRU HPC data (in a third step) actionable insights can be generated that will result in improvements both for HPC operation as well as for the research performed in the context of performance optimization of HPC applications and systems.

The novelty of this project is that no effort towards data analysis for HPC has explored the legal challenges crosscutting any transferrable solution or knowledge.

1.1.3 Methods

The methods employed for achieving the above goals include: monitoring, FAIRU data principles, de-identification and anonymization, legal data controlling, data aggregation and mining, and insight extraction. Of these, the consortium has expertise in the following methods: monitoring, de-identification and anonymization, legal data controlling, and data mining. The consortium will develop monitoring solutions for all member institutions, apply the FAIR and useful data principles, perform legal data controlling, data aggregation and mining, and extract insights.

Expected outcome The expected outcome of this project is in the form of solutions for improving the HPC operations and research at the member institutions, that comply with the diverse applicable data protection and privacy legislations.

1.1.4 Significance

The significance of this work is in improving the research and operations activities of three HPC centers within Eucor. The solutions proposed herein will be transferable to improve the HPC operations and research at other Eucor member institutions, at the benefit of no (or minimal) additional legislative inquiries and data management overhead.

2 Current State and Challenges

Monitoring of HPC systems and applications generates various and large volumes of data, constituting a case of Big Data (FDPIC, 2017b) that challenges key privacy and data protection principles, as highlighted in (Koops et al., 2014). To help reduce the risks associated with the use and analysis of Big Data, a resolution for Big Data has been proposed (ICDPPC, 2015). Any effort towards Big Data analytics in a data protection- and privacy-aware manner needs to carefully examine and abide by the laws applicable in the country where the data is produced. In the present, in Switzerland, this corresponds to the Swiss Data Protection (SDP) law (FDPIC, 2017a). In Germany, the Federal Data Protection Act[2] (FDPA) applies at the moment, while Baden-Württemberg has its own Data Protection (BWDP) law (LfDI Baden-Württemberg, 2017b). The law 78-17 of 6 January 1978 on information technology, data files and civil liberties (ITDFCIL) (CNIL, 2014) presently applies in France. Starting in May of 2018, France and Germany, as member states of the European Union, will enforce the EU guidelines for data protection (EUGDP) (LfDI Baden-Württemberg, 2017a).

This will render legal monitoring between France and Germany (and throughout the EU) easier than before. However, legal monitoring between Switzerland and any EU country will remain difficult. This difficulty is also captured by a very recent census of privacy and data protection authorities (ICDPPC, 2017). Nonetheless,

[2] Federal Data Protection Act: http://byds.juris.de/byds/012_1.4_BDSG_1990_Inhaltsuebersicht.html, (visited on 27. 09. 2017)

data protection and privacy cannot be hardcoded (Koops et al., 2014) in HPC systems, runtime systems, or in programming languages. Therefore, ensuring that data complies with all legal provisions within sectoral, state, national, and European legislation, that contain data protection requirements is a non-trivial, yet critical task for this project.

A recent overview of the state of the art monitoring solutions in HPC systems can be found in (Jha et al., 2017; Layton, 2017; Brown et al., 2017). Monitoring is also used proactively for system maintenance (Röder, 2016) as well as for application optimization. Faults have become a major threat for the execution of HPC applications at scale (Geist, 2016; Snir et al., 2014; Cappello et al., 2009). Resilience has been identified as one of the top ten Exascale challenges (Dongarra et al., 2011). In this context, the analysis of system logs (Martino et al., 2014; Tiwari et al., 2015; Sridharan et al., 2015) paves the way for understanding the distribution of faults and their impact on the system and its performance. Research in this direction is urgently needed to improve current fault-detection methods, fault models, and other resilience techniques (Benoit et al., 2016). Existing work on (manual) data analysis or (automatic) data mining for HPC includes fault prediction (Gainaru et al., 2012) and coping with recall and precision of soft error detectors (Bautista-Gomez et al., 2016), respectively. However, there is room for improvement. Specifically, the use of data mining to HPC fits into Industry 4.0 where there is a strong interest on improving the performance of processes via log mining. This is described in a recent manifesto on process mining (Aalst et al., 2011).

3 Conclusion and Outlook

This seed money project will significantly raise the visibility of Eucor via the use of its label on each of the members' website, publications and via their network channels. The DA-HPC-OR project already tightens the collaboration between scientists or the three neighboring countries beyond this project.

Corresponding Author

Florina M. Ciorba: `florina.ciorba@unibas.ch`
Department of Mathematics and Computer Science,
University of Basel, Switzerland

ORCID

Florina M. Ciorba https://orcid.org/0000-0002-2773-4499
Gerhard Schneider https://orcid.org/0000-0002-3214-002X
Dirk von Suchodoletz https://orcid.org/0000-0002-4382-5104
Aurélien Cavelan https://orcid.org/0000-0002-1784-0730

References

Aalst, W. van der et al. (2011). *Process Mining Manifesto.* URL: http://www.win.tue.nl/ieeetfpm/lib/exe/fetch.php?media=shared:process_mining_manifesto-small.pdf (visited on 27. 09. 2017).

Bautista-Gomez, L. et al. (2016). »Coping with recall and precision of soft error detectors«. In: *Journal of Parallel and Distributed Computing , 2016, 98, 8 - 24* 98, pp. 8–24.

Benoit, A., A. Cavelan, Y. Robert and H. Sun (2016). »Optimal Resilience Patterns to Cope with Fail-Stop and Silent Errors«. In: *Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS).* Chicago, USA.

Brown, R. and O. Graß (2017). *Application Monitoring with openITCOCKPIT.* URL: http://www.admin-magazine.com/Archive/2017/37/Application-Monitoring-with-openITCOCKPIT (visited on 27. 09. 2017).

Cappello, F. et al. (2009). »Toward Exascale Resilience«. In: *Int. Journal of High Performance Computing Applications* 23, pp. 374–388.

CNIL (2014). *Act n°78-17 of 6 January 1978 on information technology, data files and civil liberties.* URL: https://www.cnil.fr/sites/default/files/typo/document/Act78-17VA.pdf (visited on 27. 09. 2017).

Dongarra, J. and et al. (2011). »The International Exascale Software Project Roadmap«. In: *Int. J. High Perform. Comput. Appl.* 25, pp. 3–60.

FDPIC (2017a). *Data Protection.* URL: https://www.edoeb.admin.ch/datenschutz/index.html?lang=en (visited on 27. 09. 2017).

— (2017b). *Explanatory notes on Big Data.* URL: https://www.edoeb.admin.ch/datenschutz/00683/01169/index.html?lang=en (visited on 27. 09. 2017).

Gainaru, A., F. Cappello, M. Snir and W. Kramer (2012). »Fault Prediction Under the Microscope: A Closer Look into HPC Systems«. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis.* Salt Lake City.

Geist, A. (2016). *How To Kill A Supercomputer: Dirty Power, Cosmic Rays, and Bad Solder.* URL: `https://spectrum.ieee.org/computing/hardware/how-to-kill-a-supercomputer-dirty-power-cosmic-rays-and-bad-solder`.

ICDPPC (2015). *Resolution Big Data.* URL: `https://icdppc.org/wp-content/uploads/2015/02/Resolution-Big-Data.pdf` (visited on 27. 09. 2017).

— (2017). *Counting on Commissioners: High level results of the ICDPPC Census 2017.* URL: `https://icdppc.org/wp-content/uploads/2017/09/ICDPPC-Census-Report-1.pdf` (visited on 27. 09. 2017).

Jha, S. et al. (2017). »Holistic Measurement Driven System Assessment«. In: *Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications (HPCMASPA).* Honolulu HI, USA.

Koops, B.-J. and R. Leenes (2014). »Privacy regulation cannot be hardcoded. A critical comment on the 'privacy by design' provision in data-protection law«. In: *International Review of Law, Computers & Technology* 28 (2), pp. 159–171.

Layton, J. (2017). *Resource Monitoring For Remote Applications.* URL: `http://www.admin-magazine.com/HPC/Articles/REMORA` (visited on 27. 09. 2017).

LfDI Baden-Württemberg (2017a). *EU guidelines on data protection.* URL: `https://www.baden-wuerttemberg.datenschutz.de/eu-richtlinien-zum-datenschutz/` (visited on 27. 09. 2017).

— (2017b). *Laws / Regulations.* URL: `https://www.baden-wuerttemberg.datenschutz.de/gesetzeverordnungen/` (visited on 27. 09. 2017).

Martino, C. D. et al. (2014). »Lessons Learned from the Analysis of System Failures at Petascale: The Case of Blue Waters«. In: *Proceedings of the 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks.* DSN '14, pp. 610–621.

Röder, D. (2016). *Proactive Monitoring.* URL: `http://www.admin-magazine.com/Articles/Proactive-Monitoring` (visited on 27. 09. 2017).

Snir, M. and et al. (2014). »Addressing Failures in Exascale Computing«. In: *Int. J. High Perform. Comput. Appl.* 28, pp. 129–173.

Sridharan, V. et al. (2015). »Memory errors in modern systems: The good, the bad, and the ugly«. In: *ACM SIGPLAN Notices.* Vol. 50, pp. 297–310.

Tiwari, D., S. Gupta, G. Gallarno, J. Rogers and D. Maxwell (2015). »Reliability Lessons Learned From GPU Experience With The Titan Supercomputer at Oak Ridge Leadership Computing Facility«. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.* Austin, Texas.

Wilkinson, M. D. and et al (2016). »The FAIR Guiding Principles for scientific data management and stewardship«. In: *Scientific Data* 3. DOI: `10.1038/sdata.2016.18`.

II Scientific Contributions

Performance of the bwHPC cluster in the production of $\mu \to \tau$ embedded events used for the prediction of background for $H \to \tau\tau$ analyses

Janek Bechtel Sebastian Brommer Artur Gottmann
Günter Quast Roger Wolf

Institut für Experimentelle Teilchenphysik, Karlsruher Institut für Technologie, Karlsruhe, Germany

In high energy physics, a main challenge is the accurate prediction of background events at a particle detector. These events are usually estimated by simulation. As an alternative, data-driven methods use observed events to derive a background prediction and are often less computationally expensive than simulation. The τ lepton embedding method presents a data-driven method to estimate the background from $\mathrm{Z} \to \tau\tau$ events for Higgs boson analyses in the same final state. $\mathrm{Z} \to \mu\mu$ events recorded by the CMS experiment are selected, the muons are removed from the event and replaced with simulated τ leptons with the same kinematic properties as the removed muons. The resulting hybrid event provides an improved description of pile-up and the underlying event compared to the simulation of the full proton-proton collision. In this paper the production of these hybrid events used by the CMS collaboration is described. The production relies on the resources made available by the bwHPC project. The data used for this purpose correspond to 65 million di-muon events collected in 2017 by CMS.

DOI: 10.15496/publikation-29043

1 The Higgs boson in the Standard Model

In the Standard Model of particle physics, the requirement of a mechanism to achieve electroweak symmetry breaking leads to the prediction of the existence of at least one neutral scalar particle, known as the Higgs boson (Guralnik et al., 1964; Higgs, 1964; Englert et al., 1964). A particle compatible with the required properties of this boson was observed in 2012 by the ATLAS and the CMS experiment at the Large Hadron Collider (LHC) at CERN via its decay into photons, Z bosons and W bosons. (ATLAS collaboration, 2012; CMS collaboration, 2012a; CMS collaboration, 2013). Additionally, a Standard Model Higgs boson is expected to generate mass for particles of half-integer spin, known as fermions. To establish this mechanism known as Yukawa coupling, a measurement of the direct Higgs boson coupling to fermions is necessary. As the coupling of the Higgs bosons to fermions is proportional to its mass, the most promising decay channel is the decay into two oppositely charged τ leptons ($H \to \tau\tau$) due to the large event rate compared to lighter fermions, and smaller contribution from background events compared to the decay of the Higgs boson into heavy quarks. After being observed already in the combined measurements of the ATLAS and CMS experiments during Run-1, the first observations of this decay into τ leptons by single experiments were presented in 2018 by the CMS and ATLAS collaborations (CMS collaboration, 2018a; ATLAS collaboration, 2018), paving the way for precision measurements of the couplings of the Higgs boson to fermions on the growing dataset collected by the CMS experiment. In addition to these precision measurements, many promising supersymmetric extensions to the Standard Model predict additional heavy neutral Higgs bosons with possibly enhanced couplings to down-type fermions such as the τ lepton.

The challenges of these analyses include an accurate description of background events with the same event signature as decays of the Higgs boson, the most prominent one being the decay of the neutral force carrier of the weak force, the Z boson, into two τ leptons ($Z \to \tau\tau$). A common way to estimate this background is to simulate proton-proton collisions to obtain samples of simulated $Z \to \tau\tau$ events. This from-scratch simulation of particle collisions is computationally expensive, and has difficulties in describing many complicated processes at the LHC such as additional jets due to multiple proton-proton collisions or the underlying event of the hard collision. In this paper a data-driven method which provides a computing-efficient way to solve these difficulties is presented.

2 The τ lepton embedding method

The decay of Z bosons into τ leptons ($Z \to \tau\tau$) constitutes one of the most prominent irreducible backgrounds for the search and analysis of Higgs boson events in the di-τ lepton final state. Instead of purely relying on simulation, this background can mostly be estimated from data. This estimation is done by making use of lepton universality in the Standard Model of particle physics, which in this specific case refers to the equal coupling strength of the Z boson to all leptons such as muons or τ leptons. Z boson events with two muons ($Z \to \mu\mu$) are selected. The tracker hits and all energy deposits of the initially reconstructed muons are removed from the event and replaced by simulated τ lepton decays with the same kinematic properties as for the removed muons, creating a partially simulated hybrid event. In such hybrid events only the well understood τ lepton decay relies on simulation and parts of the event that are difficult to describe, like the underlying event or the production of additional jets, are estimated from data. Since the simulated τ lepton decays are embedded into the remaining environment of a $Z \to \mu\mu$ event after removal of the muons, this approach is referred to as τ lepton embedding method. A visualization of this method is given in Figure 1.

The technique has been successfully used in the past by the ATLAS and CMS Collaboration for the search and analysis of Higgs boson events in the context of the Standard Model of particle physics and its minimal supersymmetric extension on the LHC Run-1 dataset (ATLAS collaboration, 2015; CMS collaboration, 2011; CMS collaboration, 2012b; CMS collaboration, 2014a; CMS collaboration, 2014b). Embedded events produced on bwHPC resources using the Run-2 dataset have served as a cross-check of the estimation of the background from $Z \to \tau\tau$ events from simulation, for the first CMS search for additional heavy Higgs bosons in the $\tau\tau$ final state at 13 TeV in the context of the MSSM (CMS collaboration, 2018b).

With a runtime of $\mathcal{O}(10\,\mathrm{s})$ per event, the τ lepton embedding method requires considerably less computational effort compared to a from-scratch simulation of proton-proton collisions at the CMS detector, which takes $\mathcal{O}(\text{minutes})$ per event. Still, computational challenges arise due to the large number of events that are needed. The availability of a large amount of computing resources is necessary for coping with these challenges. During the development and production of the embedded events in both 2017 and 2018, the vast majority of events was processed

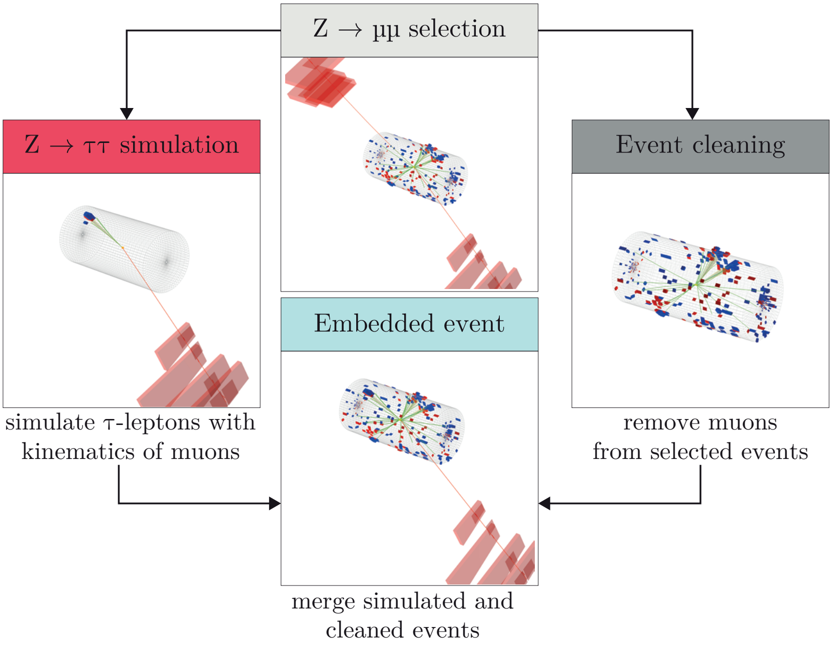

Z → μμ selection
Z → ττ simulation
Event cleaning
Embedded event
simulate τ-leptons with kinematics of muons
remove muons from selected events
merge simulated and cleaned events

Figure 1: Visualization of the τ lepton embedding method. $Z \rightarrow \mu\mu$ events are selected from data recorded by the CMS experiment. The muon tracks as well as their deposits in the calorimeters are removed from the recorded event. The decay of two τ leptons is now simulated in an empty detector environment and merged into the cleaned event. The resulting hybrid event is used to describe decays of the Z boson into τ leptons. From Ref. (Bechtel, 2017).

on the Cluster NEMO[1] located in Freiburg. In the following, the production of these embedded events is described. In Section 3, an overview of the computational setup used for the production of the embedded events is given. Section 3.1 provides a statistical insight into the performance of the computing resources.

[1] `https://www.hpc.uni-freiburg.de/nemo`

3 Production of embedded events at the bwHPC

The production of embedded events starts with the full dataset collected by the CMS experiment in 2017 tagged to contain two muon candidates. This dataset amounts to 219 million events. A full and technical description of the CMS detector can be found in Ref. (CMS collaboration, 2008). For the purpose of the τ lepton embedding method, the uncompressed detector information is used. As this dataset contains all events in which two muon candidates could be identified, a first **pre-selection** step reduces the sample to all events where a Z boson candidate with sufficient invariant mass can be constructed. 3 out of 10 events survive this selection, ultimately stored on disk to be used for the subsequent steps. The storage sites used for this work-flow are both the dedicated high energy physics (HEP) storage at GridKa, located at KIT, Karlsruhe, and the HEP storage at DESY, Hamburg.

On average, the required disk space per uncompressed event is 3 MB, resulting in a total disk usage of 206 TB for 65 million events. These events are then transfered to the computing center in batches of 1000 events for subsequent hybrid event production. The production is split up into four steps executed in sequence, each producing an input file for the following step.

1. **Selection of the di-muon events.** The selection step is repeated to ensure compatibility with changes in the reconstruction software and to adapt the selection criteria. Here, requirements on the energy of the two muon candidates and a requirement of at least one Z boson candidate in the event are set.
   `Average runtime per event: 5.0 s`
2. **Cleaning of the two muons from the event.** All hits in the silicon tracker and muon detection systems of the CMS detector that are used to reconstruct the tracks of the two muons, and all entries in the calorimetry that are related to the muons are deleted from the event.
   `Average runtime per event: 4.4 s`
3. **Simulation step.** The reconstructed kinematics of the two muons are used to create two τ leptons with the same kinematics. The decay of these τ leptons is then simulated in an empty detector environment. In this step, a filter is applied which only selects events in which the τ leptons decay into a predefined final state, the latter being defined by containing an electron, a muon or a harmonically decaying τ lepton. This filter allows to use the selected events

multiple times for each di-τ lepton final state.

`Average runtime per event: 4.9 s`

4. **Merging of cleaned and simulated event.** The simulated decay of the τ leptons and the underlying event from which the two muon candidates have been selected are merged to form a hybrid *embedded* event. The merging is done on the level of reconstructed objects at the earliest possible stage, which is at the level of reconstructed tracks in the tracking system and hits in the calorimeters, and before the reconstruction of particle candidates by the particle-flow algorithm (CMS collaboration, 2017). The hybrid event now provides a data-driven estimate of $Z \to \tau\tau$ events in the CMS detector and can be used as background prediction for Higgs analyses.

   `Average runtime per event: 1.1 s`

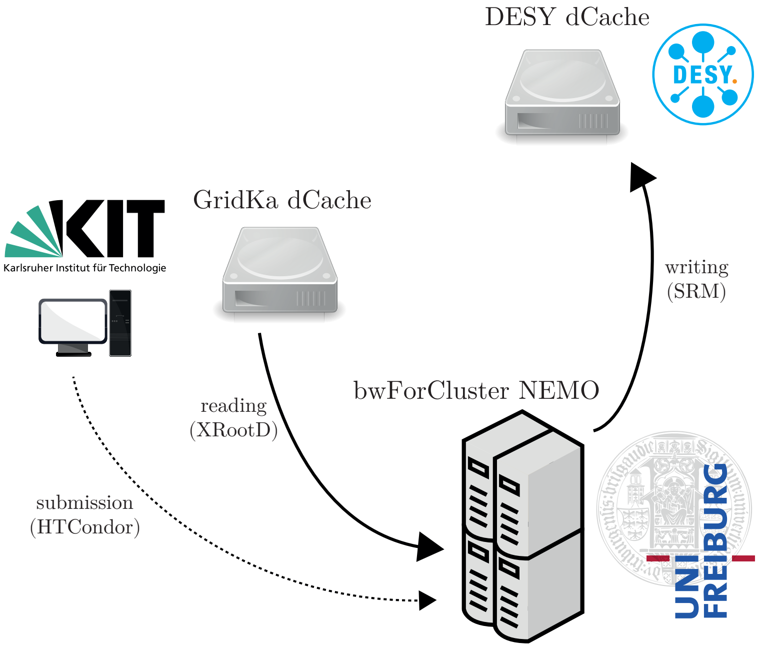


Figure 2: Schematic view of the computational setup used for the production of embedded events at the NEMO cluster. The input files amounting to 206 TB are stored on the GridKa storage located in Karlsruhe. They are then transferred in batches of 1000 events (∼ 3 GB) to the computing center NEMO in Freiburg. Finally, a single output file which has been compressed to ∼ 20 MB is stored at the HEP storage at DESY. From here, the files are published to the Data Aggregation System of the CMS collaboration to be used by the analysts.

The average CPU-time of an event in the event loop is 15.4 s. The CPU efficiency of the setup, measured with a local file, is 92%. Most of the idle time is spent on reading and writing the output files. When running this workflow at a computing center, this efficiency is further decreased by transfer times of the input files as a result of the limited bandwidth between the computing center and the HEP storage. A schematic view of the computational setup is shown in Figure 2.

The four steps are coded as python scripts and submitted via the job submission tool `grid-control` (Stober et al., 2017), using the distributed computing software `HTCondor`[2] as a backend. Details on the dynamic integration of resources as well as the specialized software environments that allow the processing of these jobs on opportunistic resources such as NEMO are given in Ref. (Heidecker et al., 2019).

The preselected files are stored on the HEP storage of located at KIT and are read from the computing center via the data access framework `XRootD` (Dorigo et al., 2005). The files are then processed on NEMO machines and written to the HEP storage at the DESY Tier-2 center via `gridFTP` (Allcock et al., 2005). The output size of all 65 million input events combined lies in the order of 1 TB. This decrease in data size by a factor of 200 is a result of both the compression from the full event description as delivered by the detector to a format suitable and sufficient for physics analyses, as well as a loss of events due to kinematic filters which are implemented in both the selection and the simulation step, removing events that are not relevant for the analysis due to their kinematic properties.

Run label	$\mathcal{L}$ in fb^{-1}	# files	# events	Size in TB	# jobs
Run2017 B	4.8	2930	5,632,077	15.82	5,633
Run2017 C	9.6	9034	16,627,325	47.25	16,628
Run2017 D	4.2	11684	7,178,226	20.17	7,179
Run2017 E	9.3	25866	15,323,608	51.87	15,324
Run2017 F	13.5	11684	20,408,930	70.94	20,409
Total per final state	41.3	89505	65,170,166	206.10	65,173
Total (6 final states)					391,038

Table 1: Selected $Z \to \mu\mu$ candidate events used for production of embedded event samples.

The total number of events available for processing are given in Table 1. For the production of $\mu \to \tau$ embedded events on the 2017 dataset collected by the CMS

[2] `https://research.cs.wisc.edu/htcondor/`

experiment, a total of 65 million events are available, which are processed six times. First, they are processed for each of the di-τ lepton final states in which the analysis is performed. As a τ lepton decays into an electron (e), muon (μ) or into light hadrons (τ_h), the four final states are $\tau\tau \rightarrow (e\mu,\ e\tau_h,\ \mu\tau_h,\ \tau_h\tau_h)$. The two final states $\tau\tau \rightarrow (ee,\ \mu\mu)$ are neglected due to their high contamination by Z boson decays into these leptons. Furthermore, the events are processed for $\mu \rightarrow \mu$ and $\mu \rightarrow e$ validation samples in which the initial muons are replaced by muons and electrons respectively. The four final states and two validation samples results in processing all 65 million input events at a total of six times.

For submission, the events are split depending on their run period of the LHC before being further split into input batches of 1000 events. This split by events results in a total number of 65,173 jobs for each validation and final state sample, and therefore 391,038 jobs in total, with a single job being expected to run for around six hours.

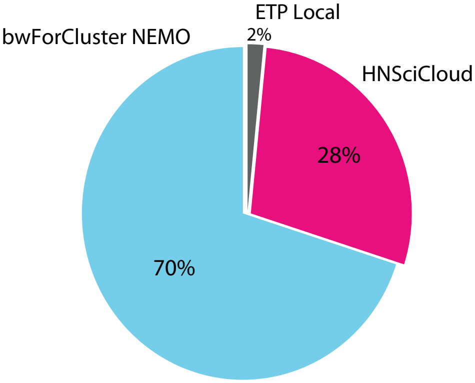


Figure 3: Relative fractions of total number of finished jobs at the available providers. The majority of cores was provided by NEMO.

3.1 Evaluation of site performance

In April 2017, three computing resources were available for the processing of these jobs, and the submission via HTCondor allows to submit them to any cluster, with the job being sent to the next available core. The three resources consisted of

- **bwForCluster NEMO**[3] for Elementary Particle Physics, Neuroscience and Microsystems Engineering, supported by the bwHPC project.
- **Helix Nebula Science Cloud**[4], a partnership between commercial cloud providers and research centers.
- **Local ETP resources** of the computing infrastructure of the Institute for Experimental Particle Physics (ETP) at the Karlsruhe Institute of Technology.

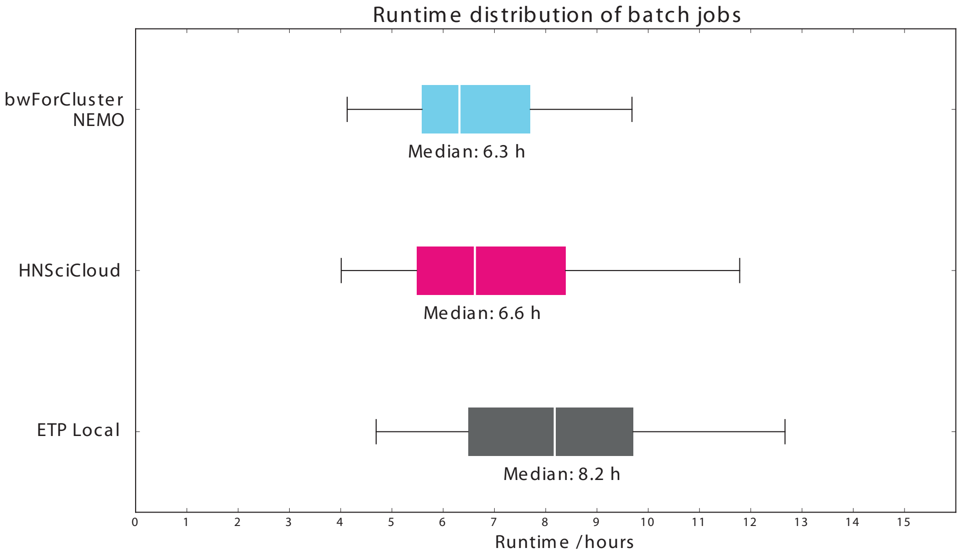


Figure 4: Distribution of job runtime shown in a box-and-whisker representation. The edges of the central box represent the first and third quartile respectively, with the median represented by a white line within the box. This results in 50% of all job runtimes lying within the box. The two outer whiskers represent the 5th and 95th percentiles, meaning only 5% of all values fall below the lower whisker and 5% fall above the upper whisker, and 90% of values fall between them. The job runtimes are compared for the three sites that were utilized for the production of embedded events. The shortest average job runtime was achieved at the machines of NEMO, which also showed the least upward fluctuations, resulting in a reliable processing of events.

The fraction of provided resources of these three is shown in Figure 3. The majority of jobs were completed at NEMO. Without these resources, the production of embedded events would have been prolonged by a factor of four, resulting in a delay

[3] `https://www.hpc.uni-freiburg.de/nemo`
[4] `http://www.helix-nebula.eu/about-hnscicloud`

between data-taking and the reliable background estimation by embedded events of over four months, which would negate a large benefit of this method. Figure 4 compares the runtime of jobs at the three sites. The cores at NEMO provide the best performance, as indicated by both the shortest average runtime of jobs as well as the small fluctuations in runtime.

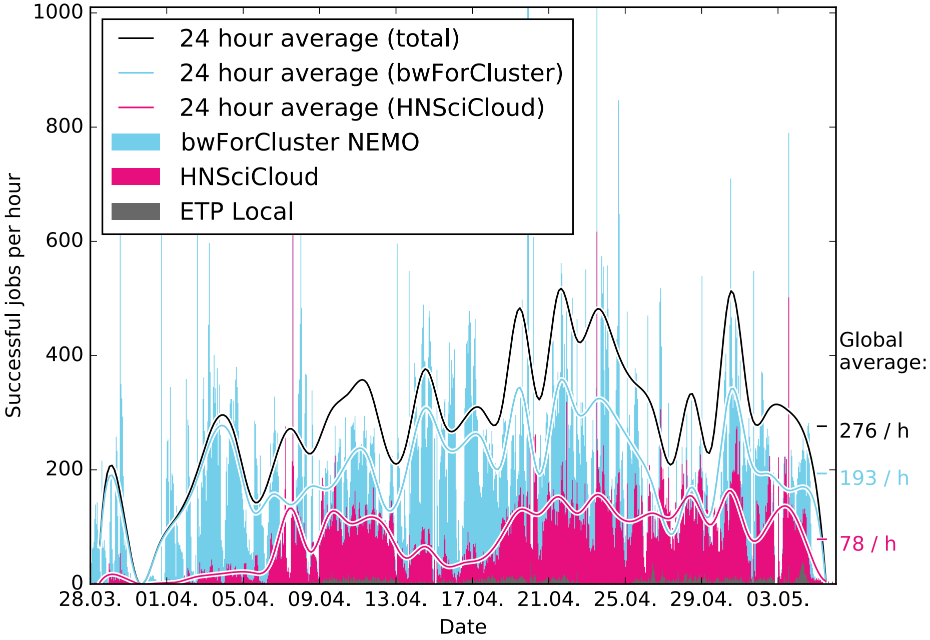


Figure 5: Histogram of successful jobs per hour for the production of embedded events over a timespan of five weeks between 28. 03. 2018 and 05. 05. 2018. In addition, the floating 24 hour averages are shown. All sites used have shown a reliable performance and continuous availability during the five weeks in which the $\mu \to \tau$ embedded samples were produced.

Figure 5 shows a histogram of successful jobs over a timespan of one month at the three sites and in bins of one hour by the time-stamp of job completion. Here, the dependence of the production on NEMO becomes apparent – it supplied the majority of cores, resulting in an average of 193 jobs being successfully completed at each given hour in the five-week timespan.

4 Conclusion

The τ lepton embedding method provides a valuable way to describe the expected background from $Z \to \tau\tau$ decays. The advantages of the method over the use of simulation lie in the description of complicated event characteristics of proton-proton collisions at the LHC, such as of the underlying event and the production of additional jets. Many corrections that are needed for simulated events become obsolete with the use of embedded events. In this regard, embedded events supplied an important cross-check already during the search for heavy neutral Higgs bosons using 2016 data, an improvement which has only been possible as a result of the opportunistic computing resources supplied by the bwHPC project.

The samples using data collected by the CMS detector in 2017, produced in April 2018 at the bwForCluster NEMO in Freiburg, will ultimately be used for the publication of the upcoming analysis of decays of the Standard Model Higgs boson into two τ leptons.

Acknowledgements

The work presented in this paper would not have been possible without the support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG.

Corresponding Author

Janek Bechtel: janek.bechtel@kit.edu
Institut für Experimentelle Teilchenphysik,
Karlsruher Institut für Technologie, Wolfgang-Gaede-Str. 1, Karlsruhe, Germany

ORCID

Janek Bechtel https://orcid.org/0000-0001-5245-7318
Sebastian Brommer https://orcid.org/0000-0001-8988-2035
Artur Gottmann https://orcid.org/0000-0001-6696-349X
Roger Wolf https://orcid.org/0000-0001-9456-383X

References

Allcock, W., J. Bresnahan, R. Kettimuthu and M. Link (2005). »The Globus Striped GridFTP Framework and Server«. In: *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pp. 54–54. DOI: 10.1109/SC.2005.72.

ATLAS collaboration (2012). »Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC«. In: *Physics Letters B* 716.1, pp. 1–29. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.020.

— (2015). »Modelling $Z \to \tau\tau$ processes in ATLAS with τ-embedded $Z \to \mu\mu$ data«. In: *JINST* 10.09, P09018. DOI: 10.1088/1748-0221/10/09/P09018. arXiv: 1506.05623 [hep-ex].

— (2018). »Cross-section measurements of the Higgs boson decaying to a pair of tau leptons in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector«. In: *Technical Report ATLAS-CONF-2018-021*.

Bechtel, J. (2017). »Cross-check of the CMS search for additional MSSM Higgs bosons in the di-τ final state using $\mu \to \tau$ embedded events«. MA thesis. Karlsruhe Institute of Technology. URL: https://ekp-invenio.physik.uni-karlsruhe.de/record/48943.

CMS collaboration (2008). »The CMS experiment at the CERN LHC«. In: *Journal of Instrumentation* 3.08, S08004. URL: http://stacks.iop.org/1748-0221/3/i=08/a=S08004.

— (2011). »Search for Neutral Minimal Supersymmetric Standard Model Higgs Bosons Decaying to Tau Pairs in pp Collisions at $\sqrt{s} = 7\,\text{TeV}$«. In: *Physical Review Letters* 106, p. 231801. DOI: 10.1103/PhysRevLett.106.231801. arXiv: 1104.1619 [hep-ex].

— (2012a). »Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC«. In: *Physics Letters B* 716, p. 30. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].

— (2012b). »Search for neutral Higgs bosons decaying to tau pairs in pp collisions at $\sqrt{s} = 7\,\text{TeV}$«. In: *Physics Letters B* 713, p. 68. DOI: 10.1016/j.physletb.2012.05.028. arXiv: 1202.4083 [hep-ex].

— (2013). »Observation of a new boson with mass near 125 GeV in pp collisions at $\sqrt{s} =$ 7 and 8 TeV«. In: *JHEP* 06, p. 081. DOI: 10.1007/JHEP06(2013)081. arXiv: 1303.4571 [hep-ex].

— (2014a). »Evidence for the 125 GeV Higgs boson decaying to a pair of τ leptons«. In: *JHEP* 05, p. 104. DOI: 10.1007/JHEP05(2014)104. arXiv: 1401.5041 [hep-ex].

— (2014b). »Search for neutral MSSM Higgs bosons decaying to a pair of tau leptons in pp collisions«. In: *JHEP* 10, p. 160. DOI: 10.1007/JHEP10(2014)160. arXiv: 1408.3316 [hep-ex].

— (2017). »Particle-flow reconstruction and global event description with the CMS detector«. In: *JINST* 12, P10003. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965 [physics.ins-det].

— (2018a). »Observation of the Higgs boson decay to a pair of τ leptons with the CMS detector«. In: *Physics Letters* B779, pp. 283–316. DOI: 10.1016/j.physletb.2018.02.004. arXiv: 1708.00373 [hep-ex].

— (2018b). »Search for additional neutral MSSM Higgs bosons in the $\tau\ \tau$ final state in proton-proton collisions at $\sqrt{s}$=13 TeV«. In: *Journal of High Energy Physics* 2018.9, p. 7. ISSN: 1029-8479. DOI: 10.1007/JHEP09(2018)007.

Dorigo, A., P. Elmer, F. Furano and A. Hanushevsky (2005). »XROOTD - A highly scalable architecture for data access«. In: *WSEAS Transactions on Computers* 4, pp. 348–353.

Englert, F. and R. Brout (1964). »Broken Symmetry and the Mass of Gauge Vector Mesons«. In: *Physical Review Letters* 13, pp. 321–323. DOI: 10.1103/PhysRevLett.13.321.

Guralnik, G. S., C. R. Hagen and T. W. Kibble (1964). »Global Conservation Laws and Massless Particles«. In: *Physical Review Letters* 13, pp. 585–587. DOI: 10.1103/PhysRevLett.13.585.

Heidecker, C., M. J. Schnepf, F. von Cube, M. Giffels and G. Quast (2019). »Dynamic Resource Extension for Data Intensive Computing with Specialized Software Environments on HPC systems«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. TLP, Tübingen, pp. 161–172. DOI: 10.15496/publikation-29051.

Higgs, P. W. (1964). »Broken Symmetries and the Masses of Gauge Bosons«. In: *Physical Review Letters* 13, pp. 508–509. DOI: 10.1103/PhysRevLett.13.508.

Stober, F. et al. (2017). »The swiss army knife of job submission tools: grid-control«. In: *CoRR* abs/1707.03198. arXiv: 1707.03198 [cs.DC].

doi: 10.15496/publikation-29062

Simulating tactoids of chiral rod-like particles

Anja Kuhnhold

Institute of Physics, University of Freiburg, Freiburg, Germany

We present simulations of rod-like particles that assemble into spindle-shaped droplets, so-called tactoids, in presence of depleting but non-self interacting (Asakura-Oosawa) spheres. The shape and structure of these objects depend on the density of the depletants and on the molecular parameters of the model rod-like particles. A special property of the simulated rods is their chirality that is induced by having a helical arrangement of point charges at the surface of the rods. An equilibrium bulk phase of such particles is a cholesteric liquid crystal phase. However, in the small assemblies studied in the present work, the formation of the cholesteric phase is suppressed.

1 Introduction

Systems of rod-like particles show a variety of equilibrium bulk phases (Gennes et al., 1995). Starting from an isotropic phase at low concentration (uniformly distributed positions and orientations) phase transitions to anisotropic (liquid crystalline) phases are found for increasing concentration. In the nematic phase the positions are still uniformly distributed, but the rods are oriented along a common direction (defined by the director). This phase actually is a special case of the cholesteric (or chiral nematic) phase, where the director rotates through space with some periodicity called the cholesteric pitch. For the nematic phase the cholesteric pitch is infinite. Increasing the concentration further leads to the transition to smectic phases, which are characterized by a one-dimensional layering. Within the layers

doi: 10.15496/publikation-29044

the positions are still uniformly distributed and the director may either coincide with the layer normal (Smectic-A) or be tilted to that (Smectic-C). For even larger concentrations the system shows solid phases. The cholesteric phase (which we are most interested in) occurs for example in systems of particles with chiral interactions (distinction between left- and right-handed). Many natural and synthetic molecules fall into this category; to name a few these, are cellulose nanocrystals (Lagerwall et al., 2014), *fd* virus (Grelet et al., 2003; Dogic et al., 2006), DNA (Livolant, 1991), chiral biphenyls (Solladié et al., 1996), or copolymers consisting of γ-benzyl glutamate and γ-alkyl glutamate (Watanabe et al., 1987).

Between (either in terms of the transition or indeed in space) bulk isotropic and bulk liquid crystalline phases the tactoids, (chiral) nematic droplets, are found (Wang et al., 2018; Park et al., 2014; Kim et al., 2013). Due to fluctuations in density, some parts of the system transit from isotropic to nematic phase, resulting in an interface between the phases. To have stable tactoids, the energy cost for this interface needs to be balanced by the free energy gain due to the formation of the (favoured) nematic phase. I. e. the tactoid only survives when its volume is large enough compared to its surface area. For spherical particles and the vapour-liquid transition, the tactoids (droplets) would be spherical to minimize the surface area for a given volume. For rod-like (elongated) particles there are two more effects: The particles prefer to have a uniform orientation and (usually) prefer to be in parallel alignment with the interface (planar anchoring). The competition of all effects determines the resulting shape (spindle-like or spheroidal) and structure (uniform, bipolar or twisted director field) of the tactoids.

The big goal would be to control the properties of liquid crystals starting from controlling the properties of the tactoids and their directed coalescence (without topological defects). Especially cholesteric liquid crystals are useful for optical applications, like displays or sensors (Dreher et al., 1973; Yeh et al., 2010; Saha et al., 2012; Picot et al., 2013). With our simulations we want to contribute to this goal by relating the microscopic (model) parameters and system conditions to the properties of the formed tactoids.

2 Theory

The above described competition of different effects is theoretically described by a free energy $F = F_E + F_S$, which is the sum of the Frank elastic energy F_E and a surface or interfacial free energy F_S. Shape and structure of the tactoids are adjusted to minimize this free energy. The Frank elastic energy is described by the volume integral (Virga, 1995; Prinsen et al., 2003; Prinsen et al., 2004a; Prinsen et al., 2004b):

$$F_E = \frac{1}{2} \int_V \mathrm{d}^3\mathbf{r} \left[K_{11} \left(\nabla \cdot \mathbf{n} \right)^2 + K_{22} \left(\mathbf{n} \cdot \nabla \times \mathbf{n} \right)^2 + K_{33} \left(\mathbf{n} \times \nabla \times \mathbf{n} \right)^2 \right] , \tag{1}$$

where the three terms correspond to splay, twist and bend deformation of the director field $\mathbf{n}(\mathbf{r})$, respectively. This part acts as a penalty for any deformation of the director field from being uniform ($\mathbf{n}(\mathbf{r}) = \mathbf{n}_0$), and the Frank elastic constants K_{11}, K_{22}, K_{33} determine the strength of this penalty for each kind of deformation. The interfacial free energy is described by the surface integral:

$$F_S = \tau \int_S \mathrm{d}^2\mathbf{r} \left(1 + \omega \left(\mathbf{q} \cdot \mathbf{n} \right)^2 \right) , \tag{2}$$

where τ is the interfacial tension, ω is the anchoring strength and $\mathbf{q}(\mathbf{r})$ defines the normal of the surface. The first term acts to minimize the surface of the tactoid and the second term acts as a penalty for non-planar alignment of the director to the surface. The strength of this penalty increases with interfacial tension and anchoring strength.

3 Simulations

To simulate droplets of an anisotropic phase surrounded by an isotropic phase, we use two species. One is a chiral rod-like particle and the other is a spherical particle acting as depletant to the rods. The rods are so-called helical Yukawa rods (Wensink et al., 2011; Honorato-Rios et al., 2016): The core is a hard spherocylinder of length L and diameter D and it is decorated with n^{pc} point charges in a helical arrangement with internal pitch p^{int}, cf. Fig. 1. Those charges interact via a Yukawa potential:

$$U_{\mathrm{Y}}(r) = k_{\mathrm{B}}T \left(\frac{Z}{n^{\mathrm{pc}}}\right)^2 \lambda_{\mathrm{B}} \, \frac{\exp\left[-\kappa r\right]}{r} \ , \tag{3}$$

with r the distance between the point charges, Z the sum of the strength of the n^{pc} charges, λ_{B} the Bjerrum length, κ the Debye screening constant, k_{B} Boltzmann's constant, and T the temperature. The solvent is implicit and determines λ_{B} and κ. Using the elementary charge e, the permittivity of the solvent $\epsilon_0\epsilon_{\mathrm{r}}$, the number density of rods $\rho = N/V$ and the added salt concentration c_{s}, λ_{B} and κ are defined as:

$$\lambda_{\mathrm{B}} = e^2/(4\pi\epsilon_0\epsilon_{\mathrm{r}}k_{\mathrm{B}}T) \tag{4}$$

$$\kappa = \sqrt{4\pi\lambda_{\mathrm{B}}(Z\rho + 2c_{\mathrm{s}})} \ . \tag{5}$$

Note, that counter ions and salt ions are not simulated explicitly, but act implicitly by determining the screening length κ^{-1}.

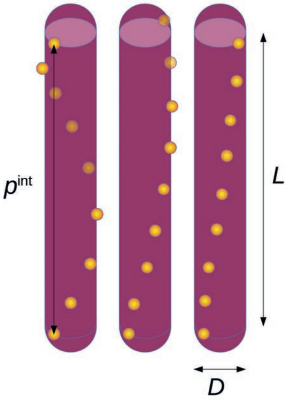


Figure 1: Sketch of helical Yukawa rods with length L, diameter D and different internal pitches (left $p^{int} = L$, middle $p^{int} = 2L$, right $p^{int} = 4L$), with the number of point charges $n^{\mathrm{pc}} = 9$. Reprinted from Front. Mater. 3, 00021, 2016 (published under the terms of the Creative Commons Attribution License (CC BY)).

The total energy of a system is the sum of all Yukawa interactions between charges on different rods with distance $r < r_{\mathrm{c}}$, where the cutoff distance r_{c} is between $2D$ and $4D$.

The spheres are so-called Asakura-Oosawa spheres (Asakura et al., 1954): They do not interact with each other, but they have a hard core of diameter $D_{\text{ao}} = 2D$ w. r. t. the rods. So they act as depletants to the rods leading to an effective attraction of the rods and therefore allowing for the formation of assemblies. There is no additional soft interaction between the spheres and the rods.

To find the equilibrium configuration of the systems, we apply Metropolis Monte Carlo simulations. I. e. we propose single particle moves (translation, rotation of the long axis or rotation around the long axis) of randomly chosen rods and accept the new configuration according to the Metropolis acceptance criterion: If the Boltzmann factor $\exp\left(-(E^{\text{new}} - E^{\text{old}})/k_{\text{B}}T\right)$ of the energy difference between old and new configuration is larger than a random number (uniformly drawn from [0,1)) the new configuration is accepted, otherwise it is rejected. Because of the hard core of the rods, overlaps between rods or rods and spheres are not allowed; thus, moves to overlapping configurations are rejected. The step sizes for the particle moves are adjusted to have an acceptance rate of 0.5.

At this point we are not interested in the nucleation process. Therefore, initial configurations already contain an assembly of rods. The number density of Asakura-Oosawa spheres needs to be large enough to keep the demixed state stable. The spheres are not stored as individual objects, but are inserted around the rods one at a time following the scheme described in (Glaser et al., 2015). This is called implicit depletant simulation and saves memory compared to the explicit simulation, in which the number of spheres would exceed several millions. The simulation box is chosen to be large enough to avoid interactions of rods across the periodic boundaries.

We use compute clusters to identify interesting regions of the parameter space by scanning through sets of parameters in parallel. The code itself is not highly parallelized: Neighbor lists are used for the energy calculation and two parallel threads share this task.

We measure density profiles, director fields and nematic order parameter profiles, and volume and aspect ratio of the tactoids. The nematic order parameters and directors are determined locally. They are defined as the largest eigenvalue and the corresponding eigenvector of the order parameter tensor:

$$Q_{ij} = \frac{1}{N_{\mathrm{L}}} \sum_{\alpha=1}^{N_{\mathrm{L}}} \frac{3}{2} \left(u_\alpha^i u_\alpha^j - \frac{1}{3} \delta_{ij} \right) , \tag{6}$$

where u_α is a unit vector pointing along the long axis of rod α, $\{i,j\} = \{x,y,z\}$ its components and N_{L} the number of rods that are considered locally.

We vary the number of rods, the number of depletants and the chirality (no charge, different internal pitch).

4 Results

We study systems of $N = 500$ to $N \approx 4000$ rods of aspect ratio $L/D = 10$. The number density of Asakura-Oosawa spheres ranges from $\rho_s\, D^3 = 0.46$ to 1.8. A snapshot of a tactoid with $N = 2916$ and $\rho_s\, D^3 = 1.0$ is shown in Fig. 2.

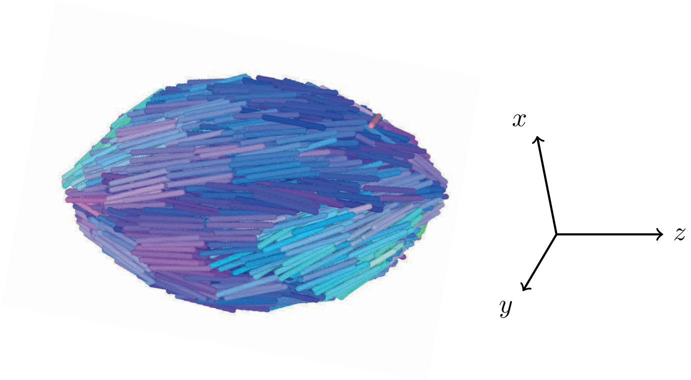


Figure 2: Snapshot of a tactoid with $N = 2916$ helical Yukawa rods with internal pitch $p^{\mathrm{int}}/D = 20$ and depletant density $\rho_s D^3 = 1.0$. The Asakura-Oosawa spheres surrounding the tactoid are not shown. The z-axis of the coordinate system is chosen to coincide with the long axis of the tactoid.

4.1 Effect of tactoid volume

In an earlier simulation study using uncharged hard rods (Trukhina et al., 2009), it was found that small tactoids have a uniform director field, whereas larger ones have a bipolar director field; which is in accord with theoretical predictions (Prinsen et al., 2003; Prinsen et al., 2004a; Prinsen et al., 2004b; Kaznacheev, Bogdanov and Taraskin, 2002; Kaznacheev, Bogdanov and Sonin, 2003). The reason is that surface free energy and elastic free energy scale differently with the size of the tactoid. For small tactoids the penalty to have non-planar alignment of the rods with the interface is smaller than that of deforming the uniform director field. For large tactoids it is much easier to pay for the deformation to a bipolar director field, this way reducing the interfacial energy by planar alignment. Thus, in the limit of infinite volume, the tactoids would be bipolar spheres.

Using the helical Yukawa rods instead, the qualitative behavior is similar, but there are two effects due to the repulsive chiral interaction between the rods: It is easier to deform the director field (the equilibrium phase is cholesteric, which is by definition non-uniform). And the anchoring strength is enhanced, because there is no electrostatic interaction between the rods and the Asakura-Oosawa spheres. So the assembly tries to expose as many charges as possible to the surface. Hence, also quite small tactoids show a less uniform director field compared to the hard rod equivalent.

4.2 Effect of chiral interaction

As discussed in the previous subsection, the repulsive chiral interaction increases the anchoring strength and eases the deformation of the director field. There are many parameters that influence the interaction, e. g. the ionic strength due to counter ions and added salt or the internal pitch of the charge helix. It is still difficult to predict the macroscopic equilibrium properties of a bulk cholesteric liquid crystal from the model parameters (or microscopic parameters of mesogens in general) and system conditions. We found that the internal pitch has a huge effect, especially on the alignment of rods (Kuhnhold and Schilling, 2016; Kuhnhold, Giesen et al., 2018) (large pitch enhances alignment, small pitch suppresses it). So we tested the effect of different internal pitches $p^{\mathrm{int}}/D = 20$ or 40.

We show the director fields in cylindrical coordinates in Fig. 3 (z-axis as indicated in Fig. 2, and $\rho^2 = x^2 + y^2$). We assume cylindrical symmetry and average directors and nematic order parameters over the azimuthal angle φ. The color encodes the orientation of the director w. r. t. the ρ-z plane: red: parallel to the z-axis, blue/black: tilted within the plane, and green/magenta/turquoise: increasingly tilted out of the plane. A perfectly uniform tactoid would be red everywhere; a perfectly bipolar tactoid would be red in the center and blue towards the surface; and a twisted tactoid would show different colors in a periodic fashion. The internal pitch in the left panel is $p^{\mathrm{int}}/D = 20$ and in the right panel it is 40. We observe a clear difference in the director field. The largest out-of-plane tilt (turquoise color) is found in Fig. 3b for the larger pitch. But for the smaller pitch (a and c) larger parts of the director fields are tilted out-of-plane (best seen by the magenta parts). However, for none of the cases studied so far we find a truly twisted director field; the reason for this being the rather small size of the tactoids.

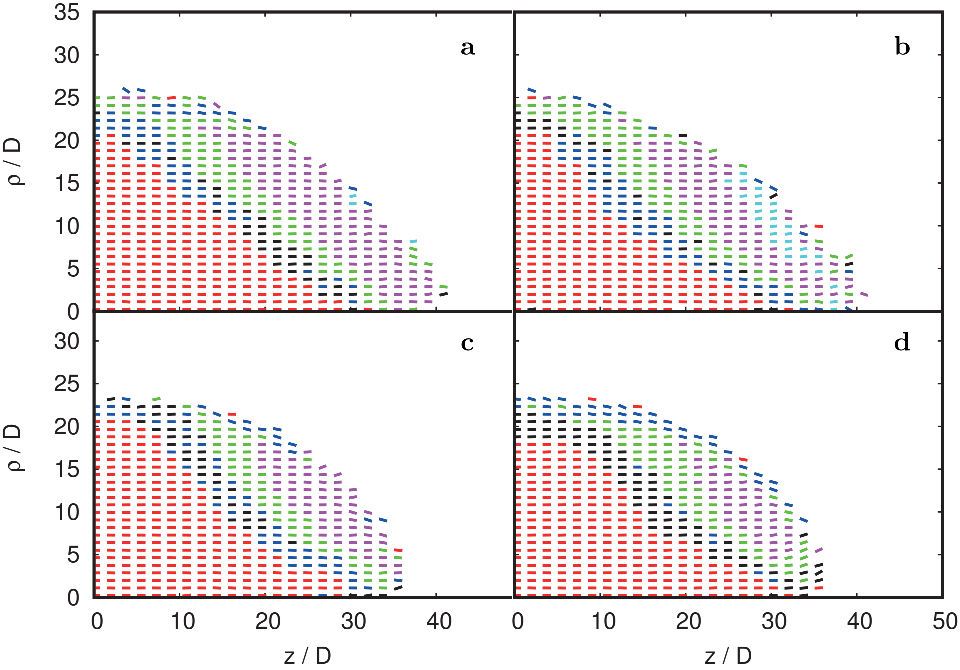


Figure 3: Director field of nematic tactoids for different internal pitch p^{int} and depletant density ρ_s shown in a cylindrical coordinate system ρ, z (symmetric in φ and for $z \to -z$). a: $p^{\mathrm{int}}/D = 20$, $\rho_s\, D^3 = 0.68$; b: $p^{\mathrm{int}}/D = 40$, $\rho_s\, D^3 = 0.68$; c: $p^{\mathrm{int}}/D = 20$, $\rho_s\, D^3 = 1.00$; d: $p^{\mathrm{int}}/D = 40$, $\rho_s\, D^3 = 1.00$. The bars represent local directors and the color supports the identification of the director orientation: red: parallel to long axis, blue/black: tilted within the ρ-z plane, green/magenta/turquoise: tilted out of the plane by increasing amount.

4.3 Effect of depletant density

The density of depletant Asakura-Oosawa spheres ρ_s has two effects. First, a higher density increases the effective attraction of the rods and therefore leads to smaller tactoid volumes, and second, the interfacial tension τ also increases with increasing ρ_s. In the top panel of Fig. 3 the depletant density is $\rho_s D^3 = 0.68$ and in the bottom panel it is 1.0. The larger density leads to more compact tactoids (smaller volume, higher concentration of rods within tactoid). Those tactoids show a much less fraction of out-of-plane tilt due to the denser packing. Due to the increased interfacial tension, those tactoids are closer to the perfect bipolar structure, which is indicated by the blue color at the tactoid surface.

5 Conclusion

We showed results of Monte Carlo simulations of chiral rod-like particles assembled to nematic droplets, so-called tactoids. The shape and structure of the tactoids is determined by the interplay between elastic and interfacial free energy, and also depends on the tactoid volume. Because of the chiral interaction, which leads to a cholesteric bulk phase, the formation of non-uniform tactoids is easier compared to assemblies of non-interacting rods. We observe a clear dependence of the director field on the internal pitch of the charge helix of our model mesogen. To properly interpret our results, more simulations with different model parameters are needed.

Acknowledgement

We thank Tanja Schilling and Paul van der Schoot for helpful discussions about the tactoids.

Simulations were carried out using the HPC facilities of the University of Luxembourg[1] (Varrette et al., 2014) and the NEMO cluster of the bwHPC facilities funded by the Ministry of Science Research and the Arts Baden-Württemberg (MWK) and the Germany Research Foundation[2] (DFG).

[1] see `http://hpc.uni.lu`

[2] see `http://www.bwhpc.de`

Corresponding Author

Anja Kuhnhold: anja.kuhnhold@physik.uni-freiburg.de
Institute of Physics, University of Freiburg, Freiburg, Germany

ORCID

Anja Kuhnhold https://orcid.org/0000-0003-2538-5392

References

Asakura, S. and F. Oosawa (1954). »On Interaction between Two Bodies Immersed in a Solution of Macromolecules«. In: *The Journal of Chemical Physics* 22.7, pp. 1255–1256. ISSN: 0021-9606. DOI: 10.1063/1.1740347.

Dogic, Z. and S. Fraden (2006). »Ordered phases of filamentous viruses«. In: *Current Opinion in Colloid & Interface Science* 11.1, pp. 47–55. ISSN: 1359-0294. DOI: 10.1016/j.cocis.2005.10.004.

Dreher, R. and G. Meier (1973). »Optical Properties of Cholesteric Liquid Crystals«. In: *Physical Review A* 8.3, pp. 1616–1623. DOI: 10.1103/PhysRevA.8.1616.

Gennes, P. G. de and J. Prost (1995). *The Physics of Liquid Crystals*. Clarendon Press. 620 pp. ISBN: 978-0-19-851785-6.

Glaser, J., A. S. Karas and S. C. Glotzer (2015). »A parallel algorithm for implicit depletant simulations«. In: *The Journal of Chemical Physics* 143.18, p. 184110. ISSN: 0021-9606. DOI: 10.1063/1.4935175.

Grelet, E. and S. Fraden (2003). »What Is the Origin of Chirality in the Cholesteric Phase of Virus Suspensions?« In: *Physical Review Letters* 90.19, p. 198302. DOI: 10.1103/PhysRevLett.90.198302.

Honorato-Rios, C. et al. (2016). »Equilibrium Liquid Crystal Phase Diagrams and Detection of Kinetic Arrest in Cellulose Nanocrystal Suspensions«. In: *Frontiers in Materials* 3. ISSN: 2296-8016. DOI: 10.3389/fmats.2016.00021.

Kaznacheev, A. V., M. M. Bogdanov and A. S. Sonin (2003). »The influence of anchoring energy on the prolate shape of tactoids in lyotropic inorganic liquid crystals«. In: *Journal of Experimental and Theoretical Physics* 97.6, pp. 1159–1167. ISSN: 1063-7761, 1090-6509. DOI: 10.1134/1.1641899.

Kaznacheev, A. V., M. M. Bogdanov and S. A. Taraskin (2002). »The nature of prolate shape of tactoids in lyotropic inorganic liquid crystals«. In: *Journal of Experimental and Theoretical Physics* 95.1, pp. 57–63. ISSN: 1063-7761, 1090-6509. DOI: 10.1134/1.1499901.

Kim, Y.-K., S. V. Shiyanovskii and O. D. Lavrentovich (2013). »Morphogenesis of defects and tactoids during isotropic–nematic phase transition in self-assembled lyotropic chromonic liquid crystals«. In: *Journal of Physics: Condensed Matter* 25.40, p. 404202. ISSN: 0953-8984. DOI: 10.1088/0953-8984/25/40/404202.

Kuhnhold, A., S. M. Giesen and T. Schilling (2018). »Compression of a suspension of helical Yukawa rods«. In: *Molecular Physics* 116.21, pp. 2806–2811. ISSN: 0026-8976. DOI: 10.1080/00268976.2018.1471232.

Kuhnhold, A. and T. Schilling (2016). »Isotropic-nematic transition and cholesteric phases of helical Yukawa rods«. In: *The Journal of Chemical Physics* 145.19, p. 194904. ISSN: 0021-9606. DOI: 10.1063/1.4967718.

Lagerwall, J. P. F. et al. (2014). »Cellulose nanocrystal-based materials: from liquid crystal self-assembly and glass formation to multifunctional thin films«. In: *NPG Asia Materials* 6.1, e80. ISSN: 1884-4057. DOI: 10.1038/am.2013.69.

Livolant, F. (1991). »Ordered phases of DNA in vivo and in vitro«. In: *Physica A: Statistical Mechanics and its Applications* 176.1, pp. 117–137.

Park, J. H. et al. (2014). »Macroscopic Control of Helix Orientation in Films Dried from Cholesteric Liquid-Crystalline Cellulose Nanocrystal Suspensions«. In: *ChemPhysChem* 15.7, pp. 1477–1484. ISSN: 1439-7641. DOI: 10.1002/cphc.201400062.

Picot, O. T. et al. (2013). »A real time optical strain sensor based on a cholesteric liquid crystal network«. In: *RSC Advances* 3.41, pp. 18794–18798. ISSN: 2046-2069. DOI: 10.1039/C3RA42986E.

Prinsen, P. and P. v. d. Schoot (2004a). »Continuous director-field transformation of nematic tactoids«. In: *The European Physical Journal E* 13.1, pp. 35–41. ISSN: 1292-8941, 1292-895X. DOI: 10.1140/epje/e2004-00038-y.

— (2004b). »Parity breaking in nematic tactoids«. In: *Journal of Physics: Condensed Matter* 16.49, p. 8835. ISSN: 0953-8984. DOI: 10.1088/0953-8984/16/49/003.

Prinsen, P. and P. v. d. Schoot (2003). »Shape and director-field transformation of tactoids«. In: *Physical Review E* 68.2, p. 021701. DOI: 10.1103/PhysRevE.68.021701.

Saha, A. et al. (2012). »Irreversible visual sensing of humidity using a cholesteric liquid crystal«. In: *Chemical Communications* 48.38, pp. 4579–4581. ISSN: 1364-548X. DOI: 10.1039/C2CC16934G.

Solladié, G., P. Hugelé, R. Bartsch and A. Skoulios (1996). »Bildung von enantiomerenreinen Flüssigkristallen aus axial-chiralen Biphenylen«. In: *Angewandte Chemie* 108.13, pp. 1640–1642. ISSN: 1521-3757. DOI: 10.1002/ange.19961081329.

Trukhina, Y., S. Jungblut, P. van der Schoot and T. Schilling (2009). »Osmotic compression of droplets of hard rods: A computer simulation study«. In: *The Journal of Chemical Physics* 130.16, p. 164513. ISSN: 0021-9606. DOI: 10.1063/1.3117924.

Varrette, S., P. Bouvry, H. Cartiaux and F. Georgatos (2014). »Management of an academic HPC cluster: The UL experience«. In: *2014 International Conference on High Performance Computing Simulation (HPCS)*, pp. 959–967. DOI: 10.1109/HPCSim.2014.6903792.

Virga, E. G. (1995). *Variational Theories for Liquid Crystals*. Google-Books-ID: LgbQebzpxCAC. CRC Press. 404 pp. ISBN: 978-0-412-39880-3.

Wang, P.-X. and M. J. MacLachlan (2018). »Liquid crystalline tactoids: ordered structure, defective coalescence and evolution in confined geometries«. In: *Philosophical Transactions of the Royal Society A* 376.2112, p. 20170042. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2017.0042.

Watanabe, J., M. Goto and T. Nagase (1987). »Thermotropic polypeptides. 3. Investigation of cholesteric mesophase properties of poly(.gamma.-benzyl L-glutamate-co-.gamma.-dodecyl L-glutamates) by circular dichroic measurements«. In: *Macromolecules* 20.2, pp. 298–304. ISSN: 0024-9297. DOI: 10.1021/ma00168a011.

Wensink, H. H. and G. Jackson (2011). »Cholesteric order in systems of helical Yukawa rods«. In: *Journal of Physics: Condensed Matter* 23.19, p. 194107. ISSN: 0953-8984. DOI: 10.1088/0953-8984/23/19/194107.

Yeh, P. and C. Gu (2010). *Optics of Liquid Crystal Displays*. Google-Books-ID: 0XhtwBpMtA8C. John Wiley & Sons. 787 pp. ISBN: 978-0-470-18176-8.

doi: 10.15496/publikation-29062

Neutron star oscillations

Linking gravitational waves to microphysics

Andreas Boden Daniela D. Doneva Kostas D. Kokkotas

Theoretical Astrophysics, Eberhard-Karls University of Tübingen, 72076 Tübingen, Germany

Fast rotating isolated neutron stars are strong sources of gravitational waves when deformed. One possible source of such a deformation is the occurrence of unstable oscillation modes in the star. Their properties strongly depend on the equation of state, describing the behavior of matter at extremely high densities. A future detection of gravitational waves from rotating neutron stars can thus provide insights not only for relativistic astrophysics, but for nuclear physics as well.

1 Scientific background

When a star reaches the end of its lifespan, the nuclear fusion powering it comes to an end, resulting in the collapse of the stellar core. Depending on the mass of the initial star, the result of the collapse can be one of three objects: a white dwarf, a neutron star or a black hole.

Lighter stars (such as our Sun) leave a white dwarf, which is supported against further collapse by the degeneracy pressure of the electrons. A white dwarf as massive as the Sun is only about as large as Earth.

The remnant of higher mass stars can not be supported in the same manner. Instead, it collapses further, squeezing electrons and protons tight enough together to form neutrons, whose degeneracy pressure now stabilizes the newly formed neutron star (NS). These objects have densities comparable to an atomic nucleus, meaning that a typical 1.4 solar mass NS is only about 10 km in radius.

doi: 10.15496/publikation-29046

Even heavier stars form a black hole as the remnant of their collapse. Not even letting light escape their influence, black holes generate the most extreme gravitational effects. They do not, however, possess any material qualities and their behavior is completely determined by a very small number of parameters.

1.1 Neutron stars

NSs are the densest material astrophysical objects and as such provide unique opportunities to study matter under most extreme conditions. Their existence was already proposed in the 1930s, while the first direct detection happened in the 1960s. Since then, several thousand NS have been observed, some of them with additional interesting properties such as extremely intense magnetic fields or very high rotation rates of nearly 1 kHz. While those observations all relied on electromagnetic observations, in 2017 the first detection of gravitational waves from the merger of two NSs was accomplished (B. P. Abbott, R. Abbott, T. D. Abbott, Acernese et al., 2017).

1.2 Gravitational wave astronomy

Being the subject of the Nobel prize in physics 2017, gravitational wave astronomy has recently gained a lot of attention. While classical astronomy relies on electromagnetic waves as carriers of information (be it radio, optical or gamma-rays), the first direct detection of gravitational waves (GWs) in 2015 (B. P. Abbott, R. Abbott, T. D. Abbott, Abernathy et al., 2016) has opened up a completely new window for the observation of astrophysical objects.

Unlike electromagnetic waves, gravitational waves are essentially unaffected by matter they pass through, since they do not rely on electromagnetic fields to propagate but only on the fabric of spacetime itself, as described by Einstein's general theory of relativity. Locally, they manifest as tiny changes in distance between otherwise fixed points. In order to detect GWs, those tiny changes have to be measured to very high precision which is realized by extremely sensitive interferometer setups.

The strongest sources of gravitational waves are the mergers of very massive objects such as black holes or NSs. When two such objects orbit each other, they emit GWs and thus loose orbital energy, bringing them closer together. The closer they are, the stronger the emitted GWs are, and the faster their orbits shrink. While it can take millions of years until the two objects come close enough together to touch and merge, it will happen inevitably. Just when they touch, the gravitational wave signal is strongest and gets weaker, while the objects merge and form a single new black hole or NS. Therefore, the typical signal for such a merger is characterized by a signal with a distinctive peak in amplitude and a monotonously increasing frequency. Several such signals have been observed over the past years and provided valuable information about the masses and several additional parameters of the objects involved.

1.3 Gravitational waves from neutron stars

While these mergers yield the strongest signals (and therefore the easiest to be detected), they are not the only sources of GWs. In this project we are mainly interested in the continuous signal that can be emitted by fast rotating deformed NSs. A perfectly axial-symmetric rotating NS does not emit GWs, but as soon as there is a deviation from symmetry, GWs are being generated.

Just as a water drop in zero gravity (or any other physical system for that matter), a NS has distinct frequencies it can naturally vibrate at. These vibrations can involve a plethora of hydrodynamic and general relativistic quantities and forces, as well as different geometrical patterns. A large number of these vibration patterns (or oscillation modes) lead to at least short-lived deviations from axial symmetry. Since such oscillations are typically damped by one mechanism or another, the GWs created by these asymmetries are usually very weak.

However, if certain conditions on the star's rotation rate and/or the oscillation frequency are fulfilled, it is possible for one or more oscillation modes to become *unstable* via the so-called Chandrasekhar-Friedman-Schutz (CFS) mechanism (Chandrasekhar, 1970; Schutz et al., 1975). An unstable mode can siphon energy off the star's rotation and increase its oscillation amplitude further and further. With the amplitude increasing, so does the deviation from axial symmetry and therefore also the gravitational wave amplitude.

Since we can observe NSs rotating fast enough to drive one or more modes unstable, this process cannot go on indefinitely (which would mean ripping the NS apart), but saturates at a certain (unknown) amplitude. With several models for the saturation mechanism being proposed, knowledge of the saturation amplitude is an important ingredient in properly understanding the mechanism and also gaining information on hydrodynamic properties of the NS (e. g. viscosity).

The CFS mechanism can therefore lead to rather strong gravitational wave emission from fast rotating NSs, the detection of which would allow to draw conclusions on the structure of the NS itself.

In the immediate future, however, the detection of GWs from mergers of binary NS systems is more likely, with one such event already recorded. While the broad structure of those signals primarily depends on the NS masses, its finer details still carry imprints of the matter oscillations. Given a clean enough signal, similar conclusions can be drawn.

However, all of this requires precise knowledge of different features of the oscillation modes, the most important one being the oscillation frequency. These properties strongly depend on the behavior of matter under the extreme conditions inside a neutron star, which is encoded in the equation of state.

1.4 The equation of state

The matter in the interior of NSs has very unique properties that can neither be found in any other astrophysical environment nor be recreated in a laboratory. In the core of a NS, particles are squeezed together more tightly than in an atomic nucleus. The behavior of matter under such conditions is one of the greatest unsolved problems of modern nuclear physics, the solution of which would allow us to better understand the very early history of the universe.

For our purposes, all the microphysical interactions can be encoded in the so-called equation of state (EoS), connecting macroscopic, thermodynamical quantities such as pressure, (energy) density or temperature. Many different models, computations and theories have led to a huge number of proposed EoSs. Determining which is the »correct« one is a very important step in solving this problem.

Since the EoS describes the behavior of the matter the NS consists of, it has a huge impact on its properties. Different EoSs will lead to a different mass-radius relationship for example, but also to different properties of the oscillation modes. By investigating the behavior of possibly unstable oscillation modes and the resulting gravitational wave emission under the assumption of different EoSs, we can therefore constrain the EoS with future GW observations from fast rotating NS. This will boost our understanding of the behavior of supranuclear matter and thus the early universe.

1.5 State of the art

Although oscillations of relativistic stars have been studied for more than half a century and many of the oscillation properties have been revealed, the focus was on non-rotating or slowly rotating objects. The main reasons were the numerical challenges presented in the non-linear treatment and the size of the perturbation equations in linear theory.

During the last decade, our group considerably advanced the field by studying the oscillations and instabilities of relativistic stars in the Cowling approximation, i. e. freezing the spacetime perturbations (Gaertig and Kokkotas, 2008; Gaertig and Kokkotas, 2009; Krüger et al., 2010; Kastaun et al., 2010). The Cowling approximation is known to provide very good qualitative results, which, however, are known to deviate from the exact solutions by typically 10-30 %, at least in the case of non-rotating stars. Thus, in this linear formalism our group developed asteroseismological relations which can be used to extract the parameters (mass, radius, rotation rate and EoS) of the stable and unstable fast rotating neutron stars (Gaertig and Kokkotas, 2011; Doneva, Gaertig et al., 2013; Doneva and Kokkotas, 2015). Furthermore, the criteria for the onset of CFS instability and its efficiency were also discussed in (Gaertig, Glampedakis et al., 2011; Passamonti et al., 2013), while the saturation amplitude of the instabilities was studied in (Pnigouras et al., 2015; Pnigouras et al., 2016).

This expertise led to proposing a novel scenario, according to which the post-merger remnant of colliding neutron stars, if it is not so massive as to collapse directly to a black-hole, will rotate quite fast, near its Kepler limit and will be rotationally unstable (Doneva and Kokkotas, 2015). The efficiency of the process depends on three factors, the accurate knowledge of the oscillation frequency, the growth time of the instability and the saturation amplitude. All three cases are known in Cowling approximation, but in order for the process to be detectable and astrophysicaly relevant, i. e. to be able to extract the parameters of the star, more accurate calculations are necessary.

In this direction we are concentrating our effort, hoping to be ready for the next binary neutron star mergers and to contribute to the global effort.

2 Numerical setup

The main goal of this project is to accurately investigate the properties of various oscillation modes of NSs and thus create a better understanding of the microphysics of matter under extreme conditions via future gravitational wave observations. To this end, we want to perform a number of fully non-linear relativistic three-dimensional numerical simulations of the evolution of isolated NSs. Each simulation assumes a certain NS model by choosing a mass, rotation rate and EoS. For each model, several different initial perturbations of the equilibrium model have to be considered in order to excite various oscillation modes. These simulations largely rely on the Einstein Toolkit[1] (Löffler et al., 2012), an open source software platform for relativistic astrophysics, and are mostly carried out on the bwForCluster BinAC[2].

The results of each simulation need to be analyzed further in order to extract the relevant properties of the oscillation modes. This task is carried out on the same machine by a custom Python code.

2.1 System of equations

For a working numerical model of the evolution of a NS, we need two key ingredients. The first is a formulation of the field equations of general relativity, since the Newtonian theory of gravity is not applicable to the extremely strong gravitational fields governing the evolution of a NS. The second are the equations of relativistic hydrodynamics, as we can use a perfect fluid description for the matter inside the star, with all the microphysics encoded in the EoS. The actual time integration is carried out using the method of lines. This way, we do not have to take care of the coupling between the hydrodynamic quantities and the spacetime quantities explicitly.

[1] Einstein Toolkit: Open software for relativistic astrophysics: `http://einsteintoolkit.org/`, (visited on 17. 06. 2018).

[2] bwForCluster BinAC: `https://www.binac.uni-tuebingen.de/`, (visited on 17. 06. 2018).

2.1.1 The field equations

Einstein's field equations describe the relation between the mass-energy content of the universe and its spacetime geometry:

$$G_{\mu\nu} = 8\pi T_{\mu\nu}. \tag{1}$$

Here, $G_{\mu\nu}$ denotes the Einstein tensor describing the curvature of spacetime. The stress-energy tensor $T_{\mu\nu}$, on the other hand, describes the energy content of matter and fields. Due to the strong interconnection shown by these equations, we cannot treat the evolution of spacetime and matter separately, but have to include the state of the one in the evolution equations of the other.

However, being four-dimensional tensor equations describing the entire spacetime geometry at once, the field equations are not very well suited for numerical treatment. Instead, it is possible to pose the problem as a time succession of three-dimensional space geometries, i. e. an initial value problem.

Here we are using the very successful BSSN (Baumgarte, Shapiro, Shibata, Nakamura) formulation (Shibata et al., 1995; Baumgarte et al., 1998). It achieves such a reformulation by decomposing the field equations into three parts: a set of time-evolution equations for a number of abstract spacetime quantities, a set of constraint equations that need to be fulfilled at all times and a set of gauge conditions.

The first set is used to evolve the spacetime quantities from one timestep to the next, the second set is important in the construction of initial data and can serve as an error estimate during the evolution, while the proper choice of gauge conditions is important to obtain a stable evolution.

2.1.2 Hydrodynamics

While for the evolution of the rather smooth spacetime variables a finite-differences method works very well, applying the same to the hydrodynamic quantities would introduce large errors. The reason is that fluid variables can vary rather rapidly in space and develop discontinuities (shock waves) even from smooth initial data. Treating a shock wave with a finite-difference method would quickly smear out the (physical) discontinuity and thus fail to describe the correct evolution.

Therefore, we use a finite volume method for the treatment of the hydrodynamic evolution instead. To this end, we recast the equation of relativistic hydrodynamics (the continuity equation, the energy equation and the Euler equation) into so-called flux-conservative form:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}^i}{\partial x_i} = \mathbf{S}. \tag{2}$$

It is possible to find »conserved« variables $\mathbf{q}$ instead of the »primitive« hydrodynamic variables, such as pressure, density or velocity, for which the equations of relativistic hydrodynamics takes this particularly simple form. This way, we can split up the computation of the right-hand-side for the method of lines into the treatment of a source term $\mathbf{S}$ and a flux-term $\mathbf{F}^i$, which are both functions of the $\mathbf{q}$.

The main advantage of this procedure becomes apparent when we take the cell average of these equations by integrating over a single grid cell $\mathcal{V}$ of volume V

$$\frac{1}{V}\int_{\mathcal{V}} \mathrm{d}V\, \partial_t \mathbf{q} + \frac{1}{V}\int_{\mathcal{V}} \mathrm{d}V\, \partial_i \mathbf{F}^i = \frac{1}{V}\int_{\mathcal{V}} \mathrm{d}V\, \mathbf{S}. \tag{3}$$

If we now choose the discretization scheme for the hydrodynamic variables $\mathbf{q}_{(i,j,k)}$ to represent the cell average (instead of the value at the cell center) over the cell with spatial indices (i, j, k), we arrive at

$$\partial_t \mathbf{q}_{(i,j,k)} + \frac{1}{V}\int_{\partial\mathcal{V}} \mathrm{d}\mathbf{A}_i\, \mathbf{F}^i_{(i,j,k)} = \mathbf{S}_{(i,j,k)}, \tag{4}$$

using Gauss' law for the second term and thus transforming the volume integral to a surface integral. To compute this, we need to evaluate the flux terms on the six cell interfaces, even though we only know the cell-averaged $\mathbf{q}$. This is where the last crucial ingredient for a shock-capturing scheme comes into play. Instead of trying to interpolate the $\mathbf{q}$ and computing the flux terms from this (which would essentially result in the same disadvantages as using a finite difference scheme), a so-called Riemann problem is set up at each cell interface and the fluxes are being computed from the solution to this. A Riemann problem is set up as initial data with a single discontinuity between two constant states on its left and right hand side.

The solution to such a problem can be obtained (semi-) analytically and captures shock waves properly (after all, this initial data results in the formation of a single shock, a contact discontinuity and a rarefaction wave). For the expert reader, let us note that we are typically using the Marquina solver with PPM reconstruction.

A more extensive description of the implementation can be found in the description of the GRHydro component of the Einstein Toolkit (Baiotti et al., 2005). The EoS can in principle be provided either as an analytic expression, e. g. for a polytrope, or as tabulated values for more realistic descriptions.

2.2 Grid setups

We are using two different three-dimensional grid setups for our simulations. The first one is a traditional Cartesian grid with several layers of mesh refinement, as provided by the Carpet module of the Einstein Toolkit (Schnetter et al., 2004).

The second one is a spherical grid setup we developed specifically for this project. One of the limiting factors for such simulations on Cartesian grids is the occurrence of numerical viscosity effects, which tend to damp out the very oscillations under investigation after a relatively short period of time. The main contribution to this effect stems from the mismatch of the more or less spherical NS surface and the Cartesian grid.

The spherical grid has performed much better in this regard in initial tests using the Cowling approximation (i. e. keeping the spacetime background static and only evolving the hydrodynamic quantities). We hope to see similar advantages for the full implementation we are currently working on.

The main drawback of the spherical grid is the clustering of grid cells at the center and the axis, which severely limits the possible time step via the Courant–Friedrichs–Lewy (CFL) condition (Courant et al., 1928). Additionally, it scales worse to higher numbers of cores used, since it typically uses fewer grid cells than the Cartesian setup. However, the code can still produce results in a reasonable amount of time on the current hardware. Utilizing accelerators (GPUs) for this setup could possibly increase the performance in the future.

2.3 Post-processing

Using the ingredients sketched above (and many more we could not mention here), we can simulate the evolution of our chosen NS model in full general relativity. This is, however, only the first step. The results of the simulation are time series for different physical quantities at a large number of grid points. In order to extract information on the oscillation modes, this data needs to be analyzed further.

To this end, we are in the process of developing a flexible post-processing code to extract the frequencies and eigenfunctions of the individual oscillation modes present in the star, which can in turn be used to compute further properties such as the damping times.

Providing this information for a large number of NS models and different EoSs will be crucial for posing constraints on the EoS from future observations.

2.4 Computational cost

The simulations are computationally very expensive and could not be realized without a cluster as provided by the bwForCluster BinAC or the bwUniCluster of the bwHPC initiative.

While the individual simulations are tailored to the specific problem (size of the NS, required resolution etc.), we here want to provide some ballpark numbers on the computational cost involved for the Cartesian setup. Since the spherical grid setup is not yet ready for production use, we do not have comparable numbers for it.

A typical simulation run in the Cowling approximation (keeping the spacetime fixed) requires keeping track of three time levels of ten hydrodynamic quantities and several helper variables and of one time level of 24 constant spacetime variables as well as another large number of helper variables such as coordinates, quality estimates and so on. At a grid size of 100^3 cells, memory consumption is at least about 5 GB, but can be much higher in certain steps of the computation. After over 60,000 time steps, such a simulation creates 200 GB of relevant data and takes about 35 hours on 6 nodes (168 processors) of BinAC.

For the simulations in full general relativity, also the spacetime variables need three time levels. Additionally, we need a much larger computational domain to also keep track of gravitational waves. This is accomplished with so-called box-in-box mesh-refinement techniques, which allow for a much higher resolution at the center than at the outer boundaries by placing one refinement level at half the size and twice the resolution in the center of the next coarser level. We typically use 6 refinement levels with 100^3 cells each. This leads to over 35 GB of minimum memory consumption and creates more than 350 GB of data. Such a run takes about 48 hours on 10 nodes (280 processors).

The post-processing code works in two steps, the first one of which takes about 5 hours on 6 processors to prepare the second step, which takes about 45 minutes per investigated mode (typically less than 10) on a single processor. However, this post-processing code is still under development.

We have already performed production simulations of 10 models and want to do 20 more in the next 6 months, once the post-processing code is completed.

3 State of the project and current efforts

In a number of experimental simulations, different Cartesian grid resolutions, placements and treatments of the boundary as well as several gauge choices have been tested for their impact on performance and accuracy.

The thoroughly investigated BU series (the second, uniformly rotating series of Dimmelmeier et al. (2006)) of increasingly rapidly rotating polytropes at a fixed central energy density has been chosen as a first benchmark set of models for simulation. The evolution of all ten members of the BU series has successfully been simulated with the chosen set of parameters. This data is also being used to assess the performance of the post-processing code.

Additionally, first steps were taken to handle not only polytropic, but also more realistic EoSs that cannot be described by an analytic function, but are typically provided as tabulated data. This requires a different numerical treatment, both in the construction of the initial data and in the actual evolution.

The analysis code provides convenient access to the various data produced by both simulation codes. Extracting frequencies and wave forms is already possible for non-rotating stars. Our current efforts in this regard are to extend the extraction procedure to rotating stars, and to employ this data to obtain damping times for the various modes using the quadrupole formula.

The code operating on the spherical grid setup can handle fast rotating neutron stars in the Cowling approximation. Those simulations yield promising results at greatly reduced numerical viscosity. We are looking forward to the results the full general relativistic version of the code (currently under development) will produce.

Acknowledgements

The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG.

DD would like to thank the European Social Fund, the Ministry of Science, Research and the Arts Baden-Württemberg for their support.

AB and DD are indebted to the Baden-Württemberg Stiftung for the financial support of this research project by the Eliteprogramm für Postdocs.

Corresponding Author

Andreas Boden: andreas.boden@uni-tuebingen.de
Theoretical Astrophysics, Eberhard-Karls University of Tübingen,
72076 Tübingen, Germany

ORCID

Andreas Boden https://orcid.org/0000-0001-9669-7938
Daniela D. Doneva https://orcid.org/0000-0001-6519-000X
Kostas D. Kokkotas https://orcid.org/0000-0001-6048-2919

References

Abbott, B. P., R. Abbott, T. D. Abbott, M. R. Abernathy et al. (2016). »Observation of Gravitational Waves from a Binary Black Hole Merger«. In: *Phys. Rev. Lett.* 116 (6), p. 061102. DOI: 10.1103/PhysRevLett.116.061102.

Abbott, B. P., R. Abbott, T. D. Abbott, F. Acernese et al. (2017). »GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral«. In: *Phys. Rev. Lett.* 119 (16), p. 161101. DOI: 10.1103/PhysRevLett.119.161101.

Baiotti, L. et al. (2005). »Three-dimensional relativistic simulations of rotating neutron star collapse to a Kerr black hole«. In: *Phys. Rev. D* 71, p. 024035. DOI: 10.1103/PhysRevD.71.024035. arXiv: gr-qc/0403029.

Baumgarte, T. W. and S. L. Shapiro (1998). »Numerical integration of Einstein's field equations«. In: *Phys. Rev. D* 59 (2), p. 024007. DOI: 10.1103/PhysRevD.59.024007.

Chandrasekhar, S. (1970). »Solutions of Two Problems in the Theory of Gravitational Radiation«. In: *Phys. Rev. Lett.* 24 (11), pp. 611–615. DOI: 10.1103/PhysRevLett.24.611.

Courant, R., K. Friedrichs and H. Lewy (1928). »Über die partiellen Differenzengleichungen der mathematischen Physik«. In: *Mathematische annalen* 100.1, pp. 32–74.

Dimmelmeier, H., N. Stergioulas and J. A. Font (2006). »Non-linear axisymmetric pulsations of rotating relativistic stars in the conformal flatness approximation«. In: *Monthly Notices of the Royal Astronomical Society* 368.4, pp. 1609–1630. DOI: 10.1111/j.1365-2966.2006.10274.x. eprint: http://mnras.oxfordjournals.org/content/368/4/1609.full.pdf+html.

Doneva, D. D., E. Gaertig, K. D. Kokkotas and C. Krüger (2013). »Gravitational wave asteroseismology of fast rotating neutron stars with realistic equations of state«. In: *Phys. Rev. D* 88.4, 044052, p. 044052. DOI: 10.1103/PhysRevD.88.044052. arXiv: 1305.7197 [astro-ph.SR].

Doneva, D. D. and K. D. Kokkotas (2015). »Asteroseismology of rapidly rotating neutron stars: An alternative approach«. In: *Phys. Rev. D* 92.12, 124004, p. 124004. DOI: 10.1103/PhysRevD.92.124004. arXiv: 1507.06606 [astro-ph.SR].

Gaertig, E., K. Glampedakis, K. D. Kokkotas and B. Zink (2011). »f-Mode Instability in Relativistic Neutron Stars«. In: *Physical Review Letters* 107.10, 101102, p. 101102. DOI: 10.1103/PhysRevLett.107.101102. arXiv: 1106.5512 [astro-ph.SR].

Gaertig, E. and K. D. Kokkotas (2008). »Oscillations of rapidly rotating relativistic stars«. In: *Phys. Rev. D* 78.6, 064063, p. 064063. DOI: 10.1103/PhysRevD.78.064063. arXiv: 0809.0629 [gr-qc].

— (2009). »Relativistic g-modes in rapidly rotating neutron stars«. In: *Phys. Rev. D* 80.6, 064026, p. 064026. DOI: 10.1103/PhysRevD.80.064026. arXiv: 0905.0821 [astro-ph.SR].

— (2011). »Gravitational wave asteroseismology with fast rotating neutron stars«. In: *Phys. Rev. D* 83.6, 064031, p. 064031. DOI: 10.1103/PhysRevD.83.064031. arXiv: 1005.5228 [astro-ph.SR].

Kastaun, W., B. Willburger and K. D. Kokkotas (2010). »Saturation amplitude of the f-mode instability«. In: *Phys. Rev. D* 82.10, 104036, p. 104036. DOI: 10.1103/PhysRevD.82.104036. arXiv: 1006.3885 [gr-qc].

Krüger, C., E. Gaertig and K. D. Kokkotas (2010). »Oscillations and instabilities of fast and differentially rotating relativistic stars«. In: *Phys. Rev. D* 81.8, 084019, p. 084019. DOI: 10.1103/PhysRevD.81.084019. arXiv: 0911.2764 [astro-ph.SR].

Löffler, F. et al. (2012). »The Einstein Toolkit: A Community Computational Infrastructure for Relativistic Astrophysics«. In: *Class. Quantum Grav.* 29.11, p. 115001. DOI: 10.1088/0264-9381/29/11/115001. arXiv: 1111.3344 [gr-qc].

Passamonti, A., E. Gaertig, K. D. Kokkotas and D. Doneva (2013). »Evolution of the f-mode instability in neutron stars and gravitational wave detectability«. In: *Phys. Rev. D* 87.8, 084010, p. 084010. DOI: 10.1103/PhysRevD.87.084010. arXiv: 1209.5308 [astro-ph.SR].

Pnigouras, P. and K. D. Kokkotas (2015). »Saturation of the f-mode instability in neutron stars: Theoretical framework«. In: *Phys. Rev. D* 92.8, 084018, p. 084018. DOI: 10.1103/PhysRevD.92.084018. arXiv: 1509.01453 [astro-ph.HE].

— (2016). »Saturation of the f -mode instability in neutron stars. II. Applications and results«. In: *Phys. Rev. D* 94.2, 024053, p. 024053. DOI: 10.1103/PhysRevD.94.024053. arXiv: 1607.03059 [astro-ph.HE].

Schnetter, E., S. H. Hawley and I. Hawke (2004). »Evolutions in 3-D numerical relativity using fixed mesh refinement«. In: *Class. Quantum Grav.* 21, pp. 1465–1488. DOI: 10.1088/0264-9381/21/6/014. arXiv: gr-qc/0310042.

Schutz, B. F. and J. L. Friedman (1975). »Gravitational radiation instability in rotating stars«. In: *The Astrophysical Journal* 199, pp. L157–L159.

Shibata, M. and T. Nakamura (1995). »Evolution of three-dimensional gravitational waves: Harmonic slicing case«. In: *Phys. Rev. D* 52 (10), pp. 5428–5444. DOI: 10.1103/PhysRevD.52.5428.

Testing Einstein's theory of gravity with simulations of tidal disruption events

Gela Hämmerling[*] Kostas D. Kokkotas[*] Pablo Laguna[†]

[*]Theoretical Astrophysics, IAAT, Eberhard-Karls University of Tübingen, 72076 Tübingen, Germany

[†]Center for Relativistic Astrophysics and School of Physics, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

Although Einstein's theory of gravity has already passed many tests in the last century, it is still one of the most discussed theories in physics. Its strong-field regime can only be accessed through the study of ultra-compact objects like, e. g., black holes. In this report, we propose using black holes with metrics that deviate from the standard Kerr solution in order to probe the strong-field regime of gravity. For measuring the effects induced by the deviation, we study tidal disruption events from which large amounts of energy in electromagnetic and gravitational radiation are released. We developed an approach to implement arbitrary metrics that are not solutions of Einstein's equations into the framework of a fully general relativistic code. The results from our numerical simulations can then be compared with data from actual observed events and place constraints on the metric deviation parameters, hence assessing the validity of general relativity in gravity's very extremes.

1 Introduction

For over a century, Einstein's theory of relativity has been the best available theory of gravity and describes compact astrophysical objects like black holes (BHs) and

doi: 10.15496/publikation-29048

neutron stars with high accuracy. Still, it does not fit perfectly with other major physical theories, e. g., it is not compatible with quantum field theory. Up to now, general relativity (GR) has easily passed all observational and experimental tests in the weak-field regime, i. e., within the solar system. In order to probe the strong-field regime, which is more difficult to access, one needs to study ultra-compact objects, the physical properties of which are not fully understood.

A popular test in the strong-field regime involves the so-called *no-hair theorem*, which states that BHs in GR are uniquely characterized by their masses, spins, and charges and are described by the Kerr-Newman metric. In the case of uncharged BHs, the Kerr metric is used for rotating BHs and the Schwarzschild metric for BHs without spin. One way of testing the theorem is via observations of phenomena in the vicinity of the Kerr metric and comparing them with those taking place near parametrically deviated metrics. These types of observations and measurements performed in the strong-field regime can put constraints on these deviations from the Kerr metric and thereby test GR itself.

Tidal disruptions of stars by BHs are fascinating and violent cosmic events that release large amounts of energy in electromagnetic radiation and are accompanied by gravitational wave (GW) emission in a broad range of frequencies. The radiation patterns provide information about both the BH and the internal structure of the disrupted star. As there is a huge variety of observational signals from these events, with the electromagnetic spectrum already being detectable for many years and the GW spectrum hopefully soon accessible, and as they involve compact objects tightly interacting with each other, tidal disruption events (TDEs) provide a perfect opportunity for testing the strong-field regime of GR.

In order to study complex astrophysical processes like TDEs, observations are not always sufficient to describe these processes with reasonable accuracy. Instead, we make use of numerical simulations which are necessary and useful tools for this type of research. TDEs involve compact objects and thus require the application of full GR, since Newtonian gravity is not enough. Therefore, the evolution of these objects is described by Einstein's equations. These evolution equations are complex tensor equations which are second order in time and describe the whole spacetime geometry at once. For the numerical implementation, we have to make some adjustments to these field equations in order to retrieve a set of equations that is more suitable for that purpose.

For our simulations we use a method developed in the Master thesis »Simulations of tidal disruption events with parametrically deformed black holes« (Hämmerling, 2017). With the help of a newly written code module, it is possible to perform simulations of tidal disruption using arbitrary metrics which are not solutions of Einstein's equations. The module is embedded in the framework of the fully general relativistic Maya code, which is developed and maintained by the Atlanta group. Thereby, the Maya code was extended as the deviated metrics are not a solution of Einstein's equations and thus do not obey the evolution equations.

The performance of the code has already been successfully tested in the Master thesis (Hämmerling, 2017). In this report, we will give an overview of the current status of the ongoing project.

2 Scientific background

The ongoing research is organized in two major topics: we want to probe GR in the strong-field regime by testing the no-hair theorem, which includes a) a thorough study of alternative theories of gravity and of the theorem itself, and b) the simulation of TDEs.

2.1 Testing the no-hair theorem

Although having successfully passed a large set of observational and experimental tests since its first formulation by Einstein in 1916 (Einstein, 1916), GR it is still one of the most discussed theories with a large variety of alternatives, for many of which there are good reasons to be seriously taken into account. Most of the tests that have been performed probed only the weak-field regime of gravity, as this can be very thoroughly tested through experiments in the solar system. Binary pulsar data and the recently discovered GWs from BHs (B. P. Abbott, R. Abbott, T. D. Abbott, Abernathy et al., 2016a; B. P. Abbott, R. Abbott, T. D. Abbott, Abernathy et al., 2016b; B. P. Abbott, R. Abbott, T. D. Abbott, Acernese et al., 2017b; B. P. Abbott, R. Abbott, T. D. Abbott, Acernese et al., 2017a; B. P. Abbott, R. Abbott, T. D. Abbott, Acernese et al., 2017c) and Neutron Stars (B. P. Abbott, R. Abbott, T. D. Abbott, Acernese et al., 2017d) allow for the first serious tests of the strong-field regime. In general, the physics and astrophysics of the objects involved here (BHs and neutron stars) are not completely understood and alternative candidates for ultra-compact objects have been seriously studied in the last decade. The most fundamental issue is that the astrophysical phenomena and objects associated with the strong-field regime have no counterparts in Newtonian gravity.

The so-called no-hair theorem states that astrophysical BHs are uniquely characterized by their masses, spins, and charges (Israel, 1967; Carter, 1971; Hartle et al., 1972). Their geometry is described by the Kerr-Newman metric (Newman, Couch et al., 1965), which reduces to the Kerr metric (Kerr, 1963) in the case of a rotating but uncharged BH. The latter should be the case for astrophysical black holes, because any residual electric charge would quickly neutralize. In other words, the Kerr metric is the only stationary, axisymmetric, asymptotically flat vacuum spacetime in GR that has an event horizon but no closed timelike curves outside the horizon.

In principle, if a deviation from the Kerr metric can be detected, there are two possible interpretations. Assuming the validity of GR, the object under investigation cannot be a BH, but rather a stable stellar configuration or a new exotic compact object, like a *gravastar* (Mazur et al., 2001; Visser et al., 2004) or a BH surrounded by a scalar field (C. A. R. Herdeiro et al., 2014; C. Herdeiro et al., 2016). There are already suggestions that some GW »echoes« may have already been seen in the post-merger data of black holes (Abedi et al., 2017; Westerweck et al., 2018; Maselli et al., 2017). On the other hand, if the no-hair theorem is violated, but the candidate shows features of an event horizon, then GR can only be *approximately* valid in the strong-field regime. The latter outcome naturally leads to the quest of finding a more complete and satisfactory theory of gravity.

Because of the no-hair theorem, all parametric deviations of the Kerr metric within GR have to violate at least one of the basic hypotheses of the theorem, leading to mathematical and physical issues. These spacetimes may contain either naked singularities (i. e., singularities without an event horizon) or regions with closed timelike curves outside the event horizon. Requiring that the new metric is free of these pathologies makes theses studies a very demanding task.

A variety of non-Kerr metrics have been developed in the past 7 years and many of them have already been tested in astrophysics, e. g., the approach by Johannsen and Psaltis (JP) (Johannsen and Psaltis, 2011a; Johannsen and Psaltis, 2011b; Johannsen, 2013a; Johannsen, 2013b). Here, the authors have parametrized strong-field deviations introducing polynomial corrections into the Schwarzschild metric of a nonrotating BH, showing that through the Newman-Janis algorithm (Newman and Janis, 1965) this procedure leads to a Kerr-like spacetime. The JP solution is regular and free of unphysical properties outside the event horizon, and it can be used to describe BHs spinning up to the maximum value of the angular momentum allowed by the deformation. This metric is therefore suitable to study astrophysical phenomena close to the event horizon and then provide genuine strong-gravity tests of the no-hair theorem.

A new approach deriving Kerr-like metrics has been proposed recently in (Papadopoulos et al., 2018), allowing for an easy way in constructing Kerr-like spacetimes that admit a Carter-like third constant of motion (Carter, 1968).

The JP metric used in this project so far has been improved even further by Johannsen (Johannsen, 2013a) in order to also include a Carter constant. For that purpose, Johannsen introduced several more deviating functions, resulting in a metric which can now also be mapped to known BH solutions in alternative theories of gravity.

The nonrotating case of the JP metric has been used in the work of Bambi et al. (Pei et al., 2015), studying the scattering of particles by these deformed BHs. They studied the excitation of axial quasinormal modes of deformed nonrotating BHs and compared the associated GW signal with that expected in GR from a standard Schwarzschild BH. As a result, they state that the measurement of the GW spectrum could in principle distinguish among different spacetimes and thus constrain the deviation parameter. Bambi is now investigating, among others, the Johannsen metric by studying the reflection spectrum of accretion disks around deformed BHs (Nampalliwar et al., 2018; Bambi et al., 2018).

Beside the JP approach, Konoplya et al. (Konoplya et al., 2016) have proposed another promising parametric framework. In this approach, deviations from GR are described by mixing continuous fraction and Taylor expansions of the radial and angular variables, which guarantee an excellent convergence of the metric to known solutions, as those given by the Einstein-Dilaton-Gauss-Bonnet (EDGB) BHs (Kanti et al., 1996; Kleihaus et al., 2016; Kokkotas et al., 2017).

In our research project we want to take the next step and investigate the strong-gravity tidal effect of the JP and EDGB-type metrics in a dynamical setup by considering the case of the tidal interaction between an intermediate-mass BH and different kinds of stars.

2.2 Tidal disruption events

TDEs are a powerful tool for the analysis of both the BH and the internal structure of the disrupted star. It is also an effective method for the detection of central BHs in quiescent massive galaxies, which are hard to detect due to a surrounding gas environment leading to no significant emissions. With the growing importance of GW detection, a deeper understanding and a thorough examination of tidal disruption events will facilitate the interpretation of measured GW signals and thus provide a new method for studying the nature of compact objects.

The modeling of TDEs was pioneered by Rees (Rees, 1988), Phinney (Phinney, 1989) and Evans & Kochanek (C. R. Evans et al., 1989) in the late 1980s. They proposed that the tidal disruption of a star closely passing by a massive BH provides a mechanism which may fuel a low-luminosity active galactic nucleus by leading to an intense accretion flare, whose signature might hint to the presence of a BH. In numerical calculations, they examined the distribution of debris orbits concentrating on post-disruption evolution. This requires detailed hydrodynamical calculations due to the complex balance between orbital circulation, cooling, viscous accretion, and debris infall. In these early years, numerical simulations including the calculation of the tidal disruption rate were already performed by various groups, including Shapiro and Marchant in 1978 (Shapiro et al., 1978), whereas important aspects of the physics of stellar disruption were first understood by Lacy, Townes and Hollenbach in 1982 (Lacy et al., 1982).

Observations of TDEs have the potential to unveil supermassive black holes (SMBHs) at the center of galaxies. In galaxies with quiescent BHs, accretion-powered nuclear activity is absent and tidal disruption signatures are principally easy to identify. In the last decade, observational evidence for TDEs has been presented in increasing numbers. Thanks to the huge variety of signals from these events and due to the fact that they involve compact objects hugely influencing the spacetime in the vicinity, TDEs are the ideal processes to probe the strong-field regime of GR by testing the no-hair theorem.

Summary The aim of our research project is to test the no-hair theorem by simulating TDEs, using BHs that are described by parametrically deviated metrics. The resulting observational data will be compared to measurements of actual events.

Thus, it will be possible to put constraints on these parameters and thereby test the validity of GR in the strong-field regime.

3 Numerical methods

As the process of a tidal disruption happens in the strong gravitational field of a SMBH, a general relativistic description and calculation of gravity is required. Thankfully, access and usage of the numerical relativity code Maya is provided for the proposed project. The Maya code has demonstrated excellent performance in handling fluid flows in the vicinity of BHs in the study of binary BH mergers (Bode, Shoemaker et al., 2008; Bode, Laguna et al., 2009; Bode, Bogdanović et al., 2011), tidal disruptions of white dwarfs by intermediate-mass BHs (Haas et al., 2012; Shcherbakov et al., 2013; Shcherbakov et al., 2012), and tidal disruptions of solar-type stars by SMBHs (C. Evans et al., 2015).

The Maya code is based on the open source numerical relativity code `Einstein Toolkit` (Zilhão et al., 2013; Löffler et al., 2012), which is itself based on the `Cactus` framework (Goodale et al., 2002). Within this framework, the code is composed of several different modules called »thorns«, whereas each one has only one main functionality in order to sustain the modularity of the code. The `Einstein Toolkit` is developed by researchers from different institutions throughout the world and is in active continuous development. It uses the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) formulation (Baumgarte et al., 1998; Shibata et al., 1995) for the spacetime evolution of Einstein's equations with a finite-volume general-relativistic hydrodynamics solver.

However, the ultimate goal of this research is to perform simulations of TDEs with parametrically deformed BHs that are *not* a solution of Einstein's equations. Therefore, an evolution of the BH spacetime with the BSSN formulation is impossible and an adapted approach for the numerical implementation of these modified metrics by programming a new thorn called `AddBH` has been developed in the Master thesis (Hämmerling, 2017).

Computational infrastructure The simulations proposed in this work will be performed on the computational resources bwUniCluster and BinAC, supported by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) with grant no. INST 39/963-1 FUGG. Additionally, the TAT group owns a 24 core machine for small runs and postprocessing routines.

For the purpose of our project, we have an allocation on the BinAC cluster for a computational project with a total of a million CPU hours per year. As it is only used by scientists in the field of bioinformatics and astrophysics, the queueing times are relatively short, which enables fast computation and quick results.

4 Status Report

In this report, we want to present the current status of our research project. The newly written code module `AddBH`, developed in (Hämmerling, 2017), has so far been tested by simulating TDEs with intermediate-mass BHs described by the standard Schwarzschild metric and the Kerr metric used for rotating black holes, which are both a solution of Einstein's equations.

4.1 First tests and troubleshooting

Throughout the testing phase, several problems have been encountered and solved. For example, the curvature singularity of the BH initially led to diverging data at its center. We solved this problem by introducing a smoothing function which overwrites the data within the event horizon with some predetermined values. As the overwriting happens well within the event horizon, no information about it can reach a possible observer, i. e., the observational signal of the tidal disruption event will not be influenced by that. The coordinate singularity at the event horizon is treated by transforming the respective metric to a more suitable coordinate system.

Additionally, putting the solar-type star into a parabolic orbit around the black hole turned out to be a huge problem. The first few simulations revealed that the perfect fluid of the star does not move very smoothly but starts to dissipate over a large area. This is caused by a faulty assignment of the star's velocity by the thorns which set the initial data for the hydrodynamical variables. For now, we decided to simply overwrite the data as a temporary solution, which leads to an acceptable amount of spreading of the stellar material. Nevertheless, this is not a permanent solution and we are currently addressing this problem in optimization runs.

4.2 Simulation parameters

Based on the study of ultra-close encounters of stars with intermediate mass black holes by (C. Evans et al., 2015), we also consider a black hole of mass $M_{\mathrm{BH}} = 10^5 M_\odot$ and a solar-type main sequence star. For the simulation of our compact objects, we used the parameters listed in table 1, by column: total system mass M_{sys}, black hole mass M_{BH}, mass of the star M_*, radius of the star R_*, central density of the star ρ_{c} and polytropic exponent Γ.

M_{sys}	M_{BH}	M_*	R_*	ρ_{c}	Γ
$1\,M_{\mathrm{sys}}$	$0.99999\,M_{\mathrm{sys}}$	$5.76 \times 10^{-6}\,M_{\mathrm{sys}}$	$3.85\,M_{\mathrm{sys}}$	$1.29 \times 10^{-6}\,M_{\mathrm{sys}}^{-2}$	$4/3$
$10^5\,M_\odot$	$\sim 10^5\,M_\odot$	$0.576\,M_\odot$	$0.82\,R_\odot$	$79.9\,\mathrm{g/cm^3}$	$4/3$

Table 1: Parameters used for the simulations in system units, which means geometrized units in addition to a scaling with the total system mass M_{sys}, and in CGS units beneath.

The star's mass, radius and central density is indeed comparable to the sun's properties.[1] In the following, we will only use the system units, i. e., we use geometrical units that are additionally scaled with the total system mass M_{sys}. The star is modeled as an ideal fluid described by the Tolman-Oppenheimer-Volkoff equations (Tolman, 1939; Oppenheimer et al., 1939), its equation of state is given by a polytrope with polytropic exponent $\Gamma = 4/3$. With initial velocities of $\texttt{velx} = -0.1133$, $\texttt{vely} = 0.0504$ and zero velocity in z-direction, the star takes a parabolic orbit around the black hole with an initial separation of $d = 130\,M_{\mathrm{sys}}$. The black hole in the rotating case has a spin parameter of $a = 0.5$.

Regarding the grid, we employ six levels of mesh refinements around the star and none around the black hole, as we are not interested in the evolution of the black hole. All but the coarsest mesh have 70^3 grid points, with the coarsest having 164^3. The resolution on the finest mesh is $R_*/30$.

4.3 Disruptions by standard black holes

Up until now, we tested the code by simulating TDEs with intermediate-mass BHs described by the standard Schwarzschild and Kerr metrics. The correct implementation of parametrically deformed metrics will be addressed in our future work.

[1] The central density of the sun is $\rho_{\mathrm{c}} \approx 160\,\mathrm{g/cm^3}$.

Schwarzschild black hole The simulations of the tidal disruption of a solar-type main sequence star by a standard Schwarzschild black hole were performed with the parameters for the compact objects displayed in table 1. In general, we are able to see the beginning of the tidal disruption process. Nevertheless, the simulations stop quite early and we cannot observe the whole disruption. As soon as the disrupted material of the star performs one circular orbit around the black hole and falls back onto its own tail, the hydrodynamical routines produce various errors. The main errors result from the star's fluid velocity exceeding the speed of sound. Additionally, certain routines have problems in the numerical transformation of hydrodynamical variables under these extreme conditions. All in all, the accumulation of errors finally results in an early termination of the simulation. In the future, we will optimize these routines so that the fluid variables are treated correctly near the event horizon and we are able to fully simulate the disruption process.

Figure 1 shows the rest mass density plots of the simulation. Each figure includes three snapshots of the tidal disruption and each panel depicts the xy-plane of the computational domain with a range of $[-160\,M_{\mathrm{sys}}, 160\,M_{\mathrm{sys}}]$ for both axes. The left panel shows the moment when matter first reaches the black hole, which is positioned in the center of the picture. The snapshot in the middle panel is taken shortly before the disruption sets in and the first material of the star is ejected outward. The right panel is the last iteration of the simulation and shows the instant when the disrupted debris falls back onto its own tail.

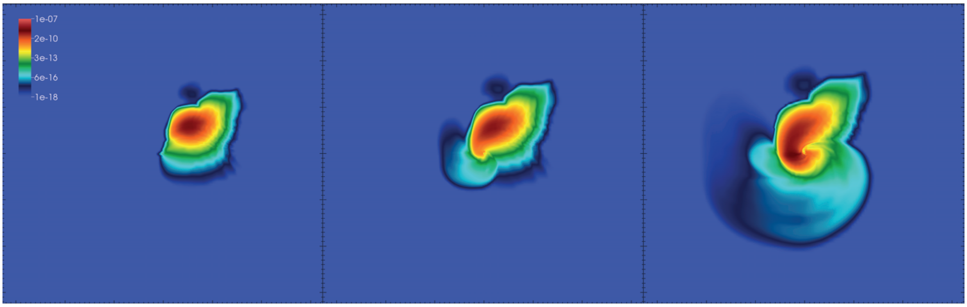


Figure 1: Rest mass density plots of a star moving on a parabolic orbit around a Schwarzschild black hole. The snapshots are taken when the stellar material first reaches the black hole (left panel), a few moments before the tidal disruption starts (middle panel) and at the end of the simulation (right panel).

Kerr black hole The simulations of the tidal disruption of a solar-type main sequence star by a standard Kerr black hole was performed with the parameters for the compact objects displayed in table 1 and an angular momentum parameter of $a = 0.5$. In fact, the simulations with the Kerr black hole are quite successful. In contrast to those with a Schwarzschild black hole, they don't stop when the disrupted debris falls back onto itself after one orbit around the black hole. Instead, the whole star gets disrupted and we can observe the formation of an accretion torus. Additionally, the tidal disruption produces a shockwave of debris that is ejected outward, in the end even reaching beyond the computational domain. Therefore, the statement by (Lacy et al., 1982), that nearly half of the debris is bound by the black hole in an accretion torus and the other half is ejected, seems to hold true for our simulation. If we compare the simulations between the disruptions by the Schwarzschild and by the Kerr black hole qualitatively, the only statement we can make is that the star seems to be spread over a larger area when reaching the black hole in the Kerr case. The reason for this is the increased gravitational pull of the Kerr black hole due to its rotation. The comparison of any other quantities is hardly possible as the simulations with the Schwarzschild black hole terminated early.

The rest mass density plots in the figures 2 and 3 show a succession of snapshots depicting this process.[2] Figure 2 shows the xy-plane of the computational domain in a range of $[-160\,M_{\mathrm{sys}}, 160\,M_{\mathrm{sys}}]$ for both axes, whereas the full domain of $[-460\,M_{\mathrm{sys}}, 460\,M_{\mathrm{sys}}]$ is shown in figure 3. In order to show the different dimensions of the figures, the right panel of figure 2 shows the same snapshot as the left panel of figure 3. Additionally, it is taken at a time comparable to the termination point of the simulations with the Schwarzschild black hole discussed before, i. e., it shows the moment when the debris performed one orbit around the black hole and falls back onto its own tail. The left panel of figure 2 shows the moment when the stellar matter first reaches the black hole, whereas the middle snapshot is taken just shortly after the estimated onset of tidal disruption.

Summary We are able to perform simulations of TDEs with intermediate-mass BHs described by the standard Schwarzschild and Kerr metrics using our alternative approach with promising results. It is possible to observe the disruption of the simulated main sequence, solar-type star and, in the Kerr case, the formation of

[2] For a video showing the full tidal disruption process see `https://drive.google.com/open?id=0B8Hmz8zgvMApZmFEbFV0UzAxX2c`.

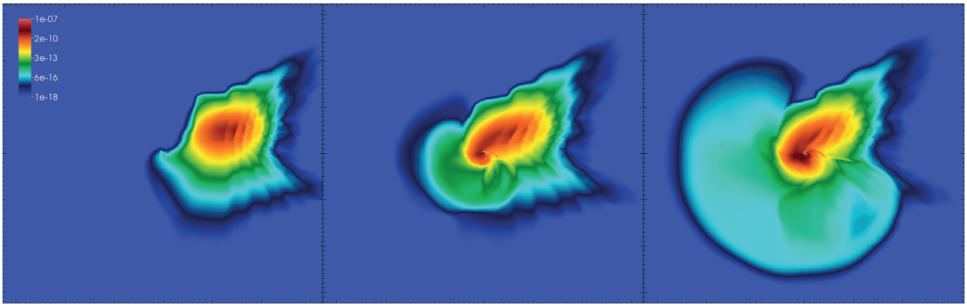


Figure 2: Rest mass density plots of a star moving on a parabolic orbit around a Kerr black hole with an angular momentum parameter of $a = 0.5$. The snapshots are taken when the stellar material first reaches the black hole (left panel), a few moments after the tidal disruption starts (middle panel) and when the debris falls back onto its own tail after one orbit around the black hole (right panel). The xy-plane of the computational domain is shown in a range of $[-160\,M_{\mathrm{sys}}, 160\,M_{\mathrm{sys}}]$.

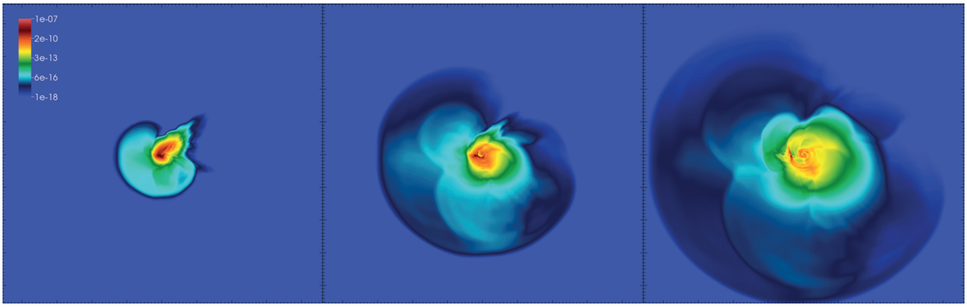


Figure 3: Continued rest mass density plots of 2 but shown in the full computational domain of $[-460\,M_{\mathrm{sys}}, 460\,M_{\mathrm{sys}}]$. The left panel is identical to the right panel of figure 2. The other two snapshots show the progress of the disruption. The star gets totally disrupted and an accretion torus forms around the Kerr black hole.

an accretion torus. Additionally, a shockwave of debris that is strongly ejected outwards becomes evident in the density plots. This supports the statement by ref. (Lacy et al., 1982), that nearly half of the debris is bound by the BH in an accretion torus and the other half is ejected.

There are many possible ways to improve the simulations. The optimization of the code and the actual implementation of the parametrically deviated BH metrics together with the calculation of observational data are all part of our planned work.

5 Outlook

With the help of the thorn `AddBH`, it is possible to perform simulations of tidal disruption with arbitrary metrics that are not a solution of Einstein's equations. This is the basis of our project titled »Testing Einstein's theory of gravity with simulations of tidal disruption events«.

The simulations performed with this module still require improvement. By testing various calculation and simulation parameters, it should be possible to further optimize the simulated TDEs. Possible approaches here are, for example, to check different stellar compositions, i. e., to not only simulate main sequence, solar-type stars but also white dwarfs.

The ultimate goal of this work is to probe the strong-field regime of GR by testing the no-hair theorem with simulations of TDEs containing parametrically deformed BHs. For that purpose, it is necessary to find a proper coordinate transformation for each meaningful alternative metric we will use in order to avoid diverging data at the event horizon. Then we can implement the metric in the code and be able to perform stable simulations.

We will not only study the JP and the Johannson metric, but also other metrics will be used. Actually, thanks to the framework developed in (Papadopoulos et al., 2018), meaningful deformations can be trivially constructed.

In order to evaluate the effect of the parametrically deviated metrics for the TDEs, the calculation of important observational data should complement the simulations in a postprocessing step. Only then will it be possible to tell the difference between a tidal disruption by a standard BH described by GR or by a deviated BH. For example, it is possible to calculate the rate of mass accretion onto the BH via measuring the flux through a spherical surface near the event horizon. Further post-disruption behavior can be studied by applying the Atlanta smoothed particle hydrodynamics code (Bogdanović et al., 2004). Then, the calculated mass accretion rate and light curves can be compared with measurements from actual observed TDEs.

Additionally, the GW emission can be calculated, although the signal is expected to be negligible with respect to amplitude and therefore detectability if compared to the already available electromagnetic spectrum.

To sum up, with our successfully developed code we possess the necessary tools to simulate TDEs, which will eventually enable us to test the no-hair theorem and thus probe the still largely unexplored strong-field regime of gravity.

Acknowledgments

The authors acknowledge the support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG. GH is indebted to the Baden-Württemberg Stiftung for the financial support of this research project by the Postdoctoral Fellowship for Leading Early Career Researchers and acknowledges financial support from the German National Academic Foundation (Studienstiftung des deutschen Volkes).

Corresponding Author

Gela Hämmerling: gela.haemmerling@uni-tuebingen.de
Theoretical Astrophysics, IAAT, Eberhard-Karls University of Tübingen,
72076 Tübingen, Germany

ORCID

Kostas Kokkotas iD https://orcid.org/0000-0001-6048-2919
Pablo Laguna iD https://orcid.org/0000-0002-2539-3897

References

Abbott, B. P., R. Abbott, T. D. Abbott, M. R. Abernathy et al. (2016a). »GW151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence«. In: *Physical Review Letters* 116.24, p. 241103.

Abbott, B. P., R. Abbott, T. D. Abbott, F. Acernese et al. (2017a). »GW170608: Observation of a 19 solar-mass binary black hole coalescence«. In: *The Astrophysical Journal Letters* 851.2, p. L35.

Abbott, B. P., R. Abbott, T. D. Abbott, F. Acernese et al. (2017b). »GW170104: Observation of a 50-Solar-Mass Binary Black Hole Coalescence at Redshift 0.2«. In: *Phys. Rev. Lett.* 118 (22), p. 221101. DOI: 10.1103/PhysRevLett.118.221101.

— (2017c). »GW170814: A three-detector observation of gravitational waves from a binary black hole coalescence«. In: *Physical Review Letters* 119.14, p. 141101.

Abbott, B. P., R. Abbott, T. D. Abbott, F. Acernese et al. (2017d). »GW170817: observation of gravitational waves from a binary neutron star inspiral«. In: *Physical Review Letters* 119.16, p. 161101.

Abbott, B. P., R. Abbott, T. D. Abbott, M. R. Abernathy et al. (2016b). »Observation of gravitational waves from a binary black hole merger«. In: *Physical Review Letters* 116.6, p. 061102.

Abedi, J., H. Dykaar and N. Afshordi (2017). »Echoes from the abyss: Tentative evidence for Planck-scale structure at black hole horizons«. In: *Phys. Rev. D* 96 (8), p. 082004. DOI: 10.1103/PhysRevD.96.082004.

Bambi, C. et al. (2018). »RELXILL_NK: A Relativistic Reflection Model for Testing Einstein's Gravity«. In: *Universe* 4, p. 79. DOI: 10.3390/universe4070079. arXiv: 1806.02141 [gr-qc].

Baumgarte, T. W. and S. L. Shapiro (1998). »Numerical integration of Einstein's field equations«. In: *Physical Review D* 59.2, p. 024007.

Bode, T., T. Bogdanović et al. (2011). »Mergers of supermassive black holes in astrophysical environments«. In: *The Astrophysical Journal* 744.1, p. 45.

Bode, T., P. Laguna et al. (2009). »Binary black hole evolutions of approximate puncture initial data«. In: *Physical Review D* 80.2, p. 024008.

Bode, T., D. Shoemaker, F. Herrmann and I. Hinder (2008). »Robustness of binary black hole mergers in the presence of spurious radiation«. In: *Physical Review D* 77.4, p. 044027.

Bogdanović, T., M. Eracleous, S. Mahadevan, S. Sigurdsson and P. Laguna (2004). »Tidal disruption of a star by a black hole: observational signature«. In: *The Astrophysical Journal* 610.2, p. 707.

Carter, B. (1968). »Global Structure of the Kerr Family of Gravitational Fields«. In: *Phys. Rev.* 174 (5), pp. 1559–1571. DOI: 10.1103/PhysRev.174.1559.

— (1971). »Axisymmetric Black Hole Has Only Two Degrees of Freedom«. In: *Phys. Rev. Lett.* 26 (6), pp. 331–333. DOI: 10.1103/PhysRevLett.26.331.

Einstein, A. (1916). »Die Grundlage der allgemeinen Relativitätstheorie«. In: *Annalen der Physik* 354.7, pp. 769–822.

Evans, C. R. and C. S. Kochanek (1989). »The tidal disruption of a star by a massive black hole«. In: *The Astrophysical Journal* 346, pp. L13–L16.

Evans, C., P. Laguna and M. Eracleous (2015). »Ultra-close Encounters of Stars with Massive Black Holes: Tidal Disruption Events with Prompt Hyperaccretion«. In: *The Astrophysical Journal Letters* 805.2, p. L19.

Goodale, T. et al. (2002). »The cactus framework and toolkit: Design and applications«. In: *International Conference on High Performance Computing for Computational Science*. Springer, pp. 197–227.

Haas, R., R. V. Shcherbakov, T. Bode and P. Laguna (2012). »Tidal disruptions of white dwarfs from ultra-close encounters with intermediate-mass spinning black holes«. In: *The Astrophysical Journal* 749.2, p. 117.

Hämmerling, G. (2017). »Simulations of tidal disruption events with parametrically deformed metrics«. In: *Master thesis* University of Tübingen.

Hartle, J. B. and S. W. Hawking (1972). »Solutions of the Einstein-Maxwell equations with many black holes«. In: *Communications in Mathematical Physics* 26, pp. 87–101. DOI: 10.1007/BF01645696.

Herdeiro, C. A. R. and E. Radu (2014). »Kerr Black Holes with Scalar Hair«. In: *Phys. Rev. Lett.* 112 (22), p. 221101. DOI: 10.1103/PhysRevLett.112.221101.

Herdeiro, C., E. Radu and H. Rúnarsson (2016). »Kerr black holes with Proca hair«. In: *Classical and Quantum Gravity* 33.15, p. 154001. URL: http://stacks.iop.org/0264-9381/33/i=15/a=154001.

Israel, W. (1967). »Event Horizons in Static Vacuum Space-Times«. In: *Phys. Rev.* 164 (5), pp. 1776–1779. DOI: 10.1103/PhysRev.164.1776.

Johannsen, T. (2013a). »Regular black hole metric with three constants of motion«. In: *Physical Review D* 88.4, p. 044002.

— (2013b). »Systematic study of event horizons and pathologies of parametrically deformed Kerr spacetimes«. In: *Physical Review D* 87.12, p. 124017.

Johannsen, T. and D. Psaltis (2011a). »Metric for rapidly spinning black holes suitable for strong-field tests of the no-hair theorem«. In: *Physical Review D* 83.12, p. 124015.

— (2011b). »Testing the no-hair theorem with observations of black holes in the electromagnetic spectrum«. In: *Advances in Space Research* 47.3, pp. 528–532. ISSN: 0273-1177. DOI: 10.1016/j.asr.2010.10.019.

Kanti, P., N. E. Mavromatos, J. Rizos, K. Tamvakis and E. Winstanley (1996). »Dilatonic black holes in higher curvature string gravity«. In: *Physical Review D* 54.8, p. 5049.

Kerr, R. P. (1963). »Gravitational Field of a Spinning Mass as an Example of Algebraically Special Metrics«. In: *Phys. Rev. Lett.* 11 (5), pp. 237–238. DOI: 10.1103/PhysRevLett.11.237.

Kleihaus, B., J. Kunz, S. Mojica and E. Radu (2016). »Spinning black holes in Einstein–Gauss-Bonnet–dilaton theory: Nonperturbative solutions«. In: *Physical Review D* 93.4, p. 044047.

Kokkotas, K. D., R. A. Konoplya and A. Zhidenko (2017). »An analytical approximation for the Einstein-dilaton-Gauss-Bonnet black hole metric«. In: *Physical Review D* 96.6, 064004, p. 064004. DOI: 10.1103/PhysRevD.96.064004. arXiv: 1706.07460 [gr-qc].

Konoplya, R., L. Rezzolla and A. Zhidenko (2016). »General parametrization of axisymmetric black holes in metric theories of gravity«. In: *Phys. Rev. D* 93 (6), p. 064015. DOI: 10.1103/PhysRevD.93.064015.

Lacy, J. H., C. H. Townes and D. J. Hollenbach (1982). »The nature of the central parsec of the Galaxy«. In: *The Astrophysical Journal* 262, pp. 120–134.

Löffler, F. et al. (2012). »The Einstein Toolkit: a community computational infrastructure for relativistic astrophysics«. In: *Classical and Quantum Gravity* 29.11, p. 115001.

Maselli, A., S. H. Völkel and K. D. Kokkotas (2017). »Parameter estimation of gravitational wave echoes from exotic compact objects«. In: *Physical Review D* 96.6, 064045, p. 064045. DOI: 10.1103/PhysRevD.96.064045. arXiv: 1708.02217 [gr-qc].

Mazur, P. O. and E. Mottola (2001). »Gravitational Condensate Stars: An Alternative to Black Holes«. In: *ArXiv General Relativity and Quantum Cosmology e-prints.* arXiv: gr-qc/0109035.

Nampalliwar, S., C. Bambi, K. D. Kokkotas and R. A. Konoplya (2018). »Iron line spectroscopy with Einstein-dilaton-Gauss-Bonnet black holes«. In: *Physics Letters B* 781, pp. 626–632. DOI: 10.1016/j.physletb.2018.04.053. arXiv: 1803.10819 [gr-qc].

Newman, E. T., E. Couch et al. (1965). »Metric of a Rotating, Charged Mass«. In: *Journal of Mathematical Physics* 6.6, pp. 918–919. DOI: 10.1063/1.1704351.

Newman, E. T. and A. Janis (1965). »Note on the Kerr Spinning-Particle Metric«. In: *Journal of Mathematical Physics* 6.6, pp. 915–917.

Oppenheimer, J. R. and G. M. Volkoff (1939). »On Massive Neutron Cores«. In: *Phys. Rev.* 55 (4), pp. 374–381. DOI: 10.1103/PhysRev.55.374.

Papadopoulos, G. O. and K. D. Kokkotas (2018). »Preserving Kerr symmetries in deformed spacetimes«. In: *Classical and Quantum Gravity* 35.18, 185014, p. 185014. DOI: 10.1088/1361-6382/aad7f4. arXiv: 1807.08594 [gr-qc].

Pei, G. and C. Bambi (2015). »Scattering of particles by deformed non-rotating black holes«. In: *The European Physical Journal C* 75.11, p. 560.

Phinney, E. (1989). *In IAU Symposium 136, The Center of Our Galaxy, ed. M. Morris.*

Rees, M. J. (1988). »Tidal disruption of stars by black holes of 10 to the 6th-10 to the 8th solar masses in nearby galaxies«. In: *Nature* 333, pp. 523–528.

Shapiro, S. and A. Marchant (1978). »Star clusters containing massive, central black holes-Monte Carlo simulations in two-dimensional phase space«. In: *The Astrophysical Journal* 225, pp. 603–624.

Shcherbakov, R. V. et al. (2012). »Prompt emission from tidal disruptions of white dwarfs by intermediate mass black holes«. In: *EPJ Web of Conferences*. Vol. 39. EDP Sciences, p. 02007.

— (2013). »GRB060218 as a Tidal Disruption of a White Dwarf by an Intermediate-mass Black Hole«. In: *The Astrophysical Journal* 769.2, p. 85.

Shibata, M. and T. Nakamura (1995). »Evolution of three-dimensional gravitational waves: Harmonic slicing case«. In: *Physical Review D* 52.10, p. 5428.

Tolman, R. C. (1939). »Static Solutions of Einstein's Field Equations for Spheres of Fluid«. In: *Phys. Rev.* 55 (4), pp. 364–373. DOI: 10.1103/PhysRev.55.364.

Visser, M. and D. L. Wiltshire (2004). »Stable gravastars—an alternative to black holes?« In: *Classical and Quantum Gravity* 21.4, p. 1135.

Westerweck, J. et al. (2018). »Low significance of evidence for black hole echoes in gravitational wave data«. In: *Phys. Rev. D* 97.12, 124037, p. 124037. DOI: 10.1103/PhysRevD.97.124037. arXiv: 1712.09966 [gr-qc].

Zilhão, M. and F. Löffler (2013). »An introduction to the Einstein Toolkit«. In: *International Journal of Modern Physics A* 28.22n23, p. 1340014.

HPC with Python: An MPI-parallel implementation of the Lattice Boltzmann Method

Lars Pastewka Andreas Greiner

Department of Microsystems Engineering, University of Freiburg, Germany

The Lattice Boltzmann Method is well suited for high-performance computational fluid dynamics. We show by means of a common two-dimensional test case, the lid-driven cavity problem, that excellent parallel scaling can be achieved in an implementation based on pure Python, using the `numpy` library and the Message Passing Interface. We highlight opportunities and pitfalls for the implementation of parallel high-performance codes in the high-level language Python.

1 Introduction

The Boltzmann transport equation (BTE) was introduced by Ludwig Boltzmann in the context of kinetic gas theory (Boltzmann, 1896) and is a statistical model for the transport of molecular constituents during flow (Cercignani, 1988). The *Lattice* Boltzmann method (LBM) is a numerical scheme based on a discretized version of the BTE, introduced by McNamara and Zanetti in 1988 (McNamara et al., 1988). LBM models have been used for the last three decades to study the dynamics of fluids in many different applications, from multiphase flow (Gunstensen, Rothman et al., 1991; Grunau et al., 1993), porous media (Aharonov et al., 1993; Gunstensen and Rothman, 1993) to microfluidics (Zhang, 2011). A thorough review of its applications in fluid dynamics and beyond can be found in Succi (2001).

DOI: 10.15496/publikation-29049

2 The Boltzmann transport equation

The BTE describes the time of evolution of the probability density $f(\mathbf{v}, \mathbf{r}, t)$ for finding a molecule with mass m and velocity $\mathbf{v}$ at position $\mathbf{r}$ as a function of time t. The moments of this probability density define the mass density $\rho(\mathbf{r}, t)$, the momentum density $\mathbf{j}(\mathbf{r}, t)$ and the temperature $T(\mathbf{r}, t)$ in D-dimensional space,

$$\rho(\mathbf{r}, t) = m \int d^D v \, f(\mathbf{v}, \mathbf{r}, t) \tag{1}$$

$$\mathbf{j}(\mathbf{r}, t) = \rho(\mathbf{r}, t)\mathbf{u}(\mathbf{r}, t) = m \int d^D v \, \mathbf{v} f(\mathbf{v}, \mathbf{r}, t) \tag{2}$$

$$k_B T(\mathbf{r}, t) = \frac{m^2}{D\rho(\mathbf{r}, t)} \int d^D v \, \left[\mathbf{v} - \mathbf{u}(\mathbf{r}, t)\right]^2 f(\mathbf{v}, \mathbf{r}, t), \tag{3}$$

where we have defined the average velocity $\mathbf{u}(\mathbf{r}, t)$ at position $\mathbf{r}$. We have here assumed a monoatomic system with molecules of mass m and k_B is the Boltzmann constant.

The BTE describes the total time-rate of change of this probability distribution, df/dt. We know that for large times t it must relax towards statistical equilibrium, given by the Maxwell velocity distribution function (Huang, 1987),

$$f^{\text{eq}}(\mathbf{v}; \rho, \mathbf{u}, T) = \frac{\rho}{m} \left(\frac{m}{2\pi k_B T}\right)^{D/2} \exp\left\{-\frac{m(\mathbf{v} - \mathbf{u})^2}{2k_B T}\right\}. \tag{4}$$

A common approximation is to assume relaxation of f towards f^{eq} with a single characteristic time τ, as suggested by Bhatnagar, Gross, and Krook (BGK) in 1954 (Bhatnagar et al., 1954),

$$\frac{df(\mathbf{v}, \mathbf{r}, t)}{dt} = -\frac{f(\mathbf{v}, \mathbf{r}, t) - f^{\text{eq}}(\mathbf{v}; \rho(\mathbf{r}, t), \mathbf{u}(\mathbf{r}, t), T(\mathbf{r}, t))}{\tau}. \tag{5}$$

Eq. (5) is the BGK-Boltzmann equation. Note that the total differential of f is

$$\frac{df}{dt} = \frac{\mathbf{F}(\mathbf{r})}{m} \frac{\partial f(\mathbf{v}, \mathbf{r}, t)}{\partial \mathbf{v}} + \mathbf{v} \frac{\partial f(\mathbf{v}, \mathbf{r}, t)}{\partial \mathbf{r}} + \frac{\partial f(\mathbf{v}, \mathbf{r}, t)}{\partial t}, \tag{6}$$

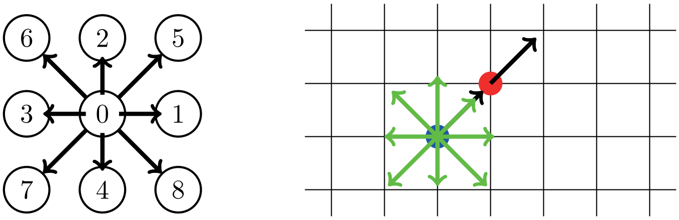
where $\mathbf{F}(\mathbf{r})$ is an external force acting on the molecules.

We will remain within the context of this single (BGK) relaxation time approximation, but modern developments of the method aim at introducing multiple relaxation times, for example by relaxing the cumulants of f individually (Geier, Greiner et al., 2006; Geier, Schönherr et al., 2015).

2.1 The Lattice Boltzmann Method in two dimensions

The BTE is discretized in space, velocity and time. Discretizing Eq. (5) on a regular (square) lattice in space is straightforward. However, we also require a suitable discretization of velocity space and the time step. Both are chosen such that the distance between interpolation points in velocity space multiplied by the time step equals the distance between points on the spatial lattice; in other words, a molecule traveling on the spatial lattice travels between integer number of lattice points during one time step.

The particular realization of the velocities used by us is shown in Fig. 1a. The velocity set contains 9 directions. Each direction is assigned the index shown in Fig. 1a. Direction 0 zero hence describes the population of molecules at rest. This specific discretization in two-dimensional space ($D = 2$) with nine directions is commonly denoted by D2Q9.



(a)

(b)

Figure 1: Discretization of the BTE. (a) Discretization of velocity space into nine directions. The numbers uniquely identify the direction. (b) Regular two-dimensional lattice used for the spatial discretization.

The discrete velocities are therefore given by the directions to the eight neighbors divided by the time step. We have nine velocity vectors $\mathbf{c}_i$, with $i = 0, \ldots, 8$ for each direction i. We now also require the occupation numbers for these nine directions. The probability distribution $f(\mathbf{r}, \mathbf{v})$ is hence represented by nine discrete $f_i(\mathbf{x}_j, t)$

where $\mathbf{x}_j$ is the discrete lattice point and i denotes the direction. Note that the moments of the probability density Eqs. (1) and (2) become

$$\rho(\mathbf{x}_j, t) = \sum_i f_i(\mathbf{x}_j, t) \tag{7}$$

$$\mathbf{u}(\mathbf{x}_j, t) = \frac{1}{\rho(\mathbf{x}_j, t)} \sum_i \mathbf{c}_i f_i(\mathbf{x}_j, t), \tag{8}$$

where $\rho(\mathbf{x}_j, t)$ is now the number density, i. e. we assume unit molecular mass.

The discretized BGK-Boltzmann equation (5) then reads

$$f_i(\mathbf{x}_j + \mathbf{c}_i \cdot \Delta t, t + \Delta t) = f_i(\mathbf{x}_j, t) - \omega[f_i(\mathbf{x}_j, t) - f_i^{\mathrm{eq}}(\mathbf{x}_j, t)] \tag{9}$$

where Δt is a time step used for the discrete dynamical propagation of the occupation numbers. We have introduced the normalized relaxation parameter $\omega = \Delta t/\tau$ and the external forces $\mathbf{F}(\mathbf{r})$ are set to zero. Note that the left hand side and the first term on the right hand side of Eq. (9) is a discretization of the total differential of f. The expression for the equilibrium distribution function in the nine directions is (Mohamad, 2011; Wolf-Gladrow, 2000)

$$f_i^{\mathrm{eq}}(\mathbf{x}_j, t) = w_i \rho(\mathbf{x}_j, t) \left\{ 1 + 3\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}_j, t) + \frac{9}{2} \left[\mathbf{c}_i \cdot \mathbf{u}(\mathbf{x}_j, t)\right]^2 - \frac{3}{2} u^2(\mathbf{x}_j, t) \right\}, \tag{10}$$

with $w_i = 4/9$, $1/9$ and $1/36$ for directions 0, 1-3 and 4-8, respectively. Note that like Eq. (4) for the continuous distribution function, Eq. (10) is obtained by applying the maximum entropy principle under the constraint of mass and momentum conservation to the discrete distribution function (Mohamad, 2011; Wolf-Gladrow, 2000).

To give a rough idea of how Eq. (9) propagates the occupation numbers for the individual directions, we describe as an example what happens to f_5. Assume we have a finite value for direction f_i including f_5, their magnitude given by the length of the green arrows, respectively, and based at the position shown by the blue spot in Fig. 1b. At time t the r.h.s. of Eq. (9) relaxes f_5 towards the local equilibrium given by the black arrow based at the blue spot, which is commonly called the collision operation (Boltzmann, 1896). Time propagation is described by the l.h.s. of Eq. (9). After one time step Δt, the occupation f_5 will be handed to the red spot and occupy direction 5 there. This is commonly called the streaming step. An algorithmic implementation of Eq. (9) is conveniently split into these streaming and collision steps.

2.2 Implementation in Python and `numpy`

The quantity that specifies the state of our simulation are the occupation numbers $f_i(\mathbf{x}_j)$. We represent these as a single contiguous `numpy`[1] array called `f_ikl`. Note that the suffixes in our specific naming scheme indicate the number of array dimensions and their function. In the present case, `ikl` stands for three array dimensions: `i` is a direction (array dimension of size 9), `k` is the lattice position in x-direction, and `l` (size n_x) the lattice position in y-direction (size n_y). Below we also encounter `c`, which denotes a Cartesian array dimension (size 2). This naming convention eases readability of the code and translation of formulas containing linear algebra into `numpy` operations and is borrowed from the GPAW code (Mortensen et al., 2005; Enkovaara et al., 2010). Note that because `numpy`'s default storage order is row-major, the array `f_ikl` is stored in what is commonly called the structure of arrays (SoA) storage order (Obrecht et al., 2011; Qi, 2017).

The streaming part can be cast into the form listed in Code Listing 1, which uses the `numpy.roll` function and automatically takes care of periodic boundary conditions. `numpy.roll` rolls the data on the lattice into the direction specified by `axis` in the code listing by a distance given by `c_ic[i]`. Note that `c_ic[i]` is a one-dimensional array of length 2 that specifies the Cartesian coordinates of direction i. Taking the example of the `f_ikl` array, a roll operation on occupation number 1 and `axis` 0 with unit distance will assign $f_{1,k,l} \to f_{1,k+1,l}$.

```python
import numpy as np
c_ic = np.array([[0,  1,  0, -1,  0,  1, -1, -1,  1],
                 [0,  0,  1,  0, -1,  1,  1, -1, -1]]).T

def stream(f_ikl):
    for i in range(1, 9):
        f_ikl[i] = np.roll(f_ikl[i], c_ic[i], axis=(0, 1))
```

Code Listing 1: Implementation of the streaming step

[1] `http://www.numpy.org/`, Version 1.14.3

The collision part of Eq. (9) is also straightforward to implement. To translate an expression like Eq. (9) into vectorized `numpy` code, it can be useful to write it down in Einstein notation. Einstein notation can be cast directly in Python code using the `numpy.einsum` function. Code Listing 2 shows a naive but straightforward implementation of the collision step, split into the computation of the equilibrium distribution function in `equilibrium` and the final collision step in `collide`, where `omega` is the relaxation parameter ω of Eq. (5).

```python
import numpy as np
w_i = np.array([4/9, 1/9, 1/9, 1/9, 1/9, 1/36, 1/36, 1/36, 1/36])

def equilibrium(rho_kl, u_ckl):
    cu_ikl = np.dot(u_ckl.T, c_ic.T).T
    uu_kl = np.sum(u_ckl**2, axis=0)
    return (w_i*(rho_kl*(1 + 3*cu_ikl + 9/2*cu_ikl**2 - 3/2*uu_kl)).T).T

def collide(f_ikl, omega):
    rho_kl = np.sum(f_ikl, axis=0)
    u_ckl = np.dot(f_ikl.T, c_ic).T/rho_kl
    f_ikl += omega*(equilibrium(rho_kl, u_ckl) - f_ikl)
    return rho_kl, u_ckl
```

Code Listing 2: Naive implementation of the collision step

It is worth pointing out that the collision implementation of Code Listing 2 does not lead to optimal performance. The Python code can be further optimized by unrolling all multiplications with `c_ic` and eliminating the terms that vanish due to zeros in `c_ic`. This implementation is our optimized reference Python implementation. We additionally benchmark this Python implementation against a C++ collision kernel integrated into our Python code with `pybind11`[2] and `eigen`[3].

2.3 Parallelization strategy

We parallelize the LBM using spatial domain decomposition and the message passing interface (MPI)[4]. The collision part of the LBM is then embarrassingly parallel. It is a spatially local operation and no communication is required. Note that we

[2] `https://github.com/pybind/pybind11`, Version 2.2.2
[3] `https://eigen.tuxfamily.org/`, Version 3.3.4
[4] `https://www.mpi-forum.org/`

could also have decided to parallelize in the direction space by distributing the occupation numbers for the individual directions onto separate MPI processes. This would then require communication during the collision step. This is, however, disadvantageous because there are just nine directions and parallelism would be limited to just nine parallel processes.

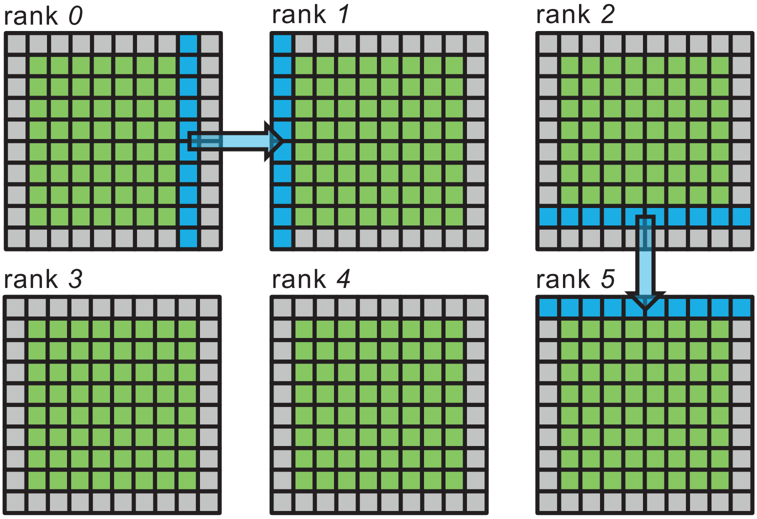


Figure 2: Domain decomposition and communication strategy. We decompose the full two-dimensional lattice into spatial domains of roughly equal size (green lattice points). We then add an additional ghost region of unit thickness surrounding these domains (gray lattice points). During each communication step, we communicate the outermost green active lattice into the adjacent outermost ghost lattice (as exemplified by the blue arrows). This requires four communication steps, two of which are indicated by the arrows.

Parallelization in the spatial domain requires communication during the streaming step. All occupation numbers that are moved past a domain boundary need to be communicated to the neighboring domain. We implement this by adding an additional ghost region to the simulation domain. Figure 2 illustrates the concept for a decomposition in 3 × 2 domains. The green lattice points are our active computational domain and the grey points are the ghost lattice points. We now communicate the region adjacent to the ghost points into the respective ghost points of the neighboring domain as shown in Fig. 2 *before* the streaming step. Within the streaming step, we then stream the relevant occupation numbers f_i from the ghost points into the active lattice points. This requires a total of four communication steps: Communication to the right and bottom as shown in Fig. 2 (for each domain) plus their reverse, communication to left and top.

We implement communication in Python using the `mpi4py`[5] bindings to the MPI library. The Python code responsible for communication is shown in Code Listing 3. We need onpe `Sendrecv` call to communicate in each of the four directions. `Sendrecv` takes care of sending data in one direction and simultaneously receiving it from the opposite direction. `ndx` and `ndy` in the code are the number of domains in x and y-direction.

```python
from mpi4py import MPI
comm = MPI.COMM_WORLD.Create_cart((ndx, ndy), periods=(False, False))
left_src, left_dst = comm.Shift(0, -1)
right_src, right_dst = comm.Shift(0, 1)
bottom_src, bottom_dst = comm.Shift(1, -1)
top_src, top_dst = comm.Shift(1, 1)

def communicate(f_ikl):
    comm.Sendrecv(f_ikl[:, 1, :], left_dst,
                  recvbuf=f_ikl[:, -1, :], source=left_src)
    comm.Sendrecv(f_ikl[:, -2, :], right_dst,
                  recvbuf=f_ikl[:, 0, :], source=right_src)
    comm.Sendrecv(f_ikl[:, :, 1], bottom_dst,
                  recvbuf=f_ikl[:, :, -1], source=bottom_src)
    comm.Sendrecv(f_ikl[:, :, -2], top_dst,
                  recvbuf=f_ikl[:, :, 0], source=top_src)
```

Code Listing 3: Implementation of the communication step

At the time of this writing, all `mpi4py` communication methods require contiguous `numpy` arrays. The input arrays in Code Listing 3 need therefore be cast into contiguous arrays using `numpy.ascontiguousarray` and a temporary contiguous buffer is required for receiving data. These intermediate steps have been omitted from the code excerpt for brevity.

3 Results for the lid-driven cavity

The lid-driven cavity is frequently used to test fluid dynamics simulation programs. Given a quadratic box with a sliding lid, as shown in Fig. 3a, we use equilibrium

[5] `https://mpi4py.scipy.org/`, Version 3.0.0

initial conditions, Eq. (10), for the discrete BGK-Boltzmann transport equations. The initial values were chosen as $\rho = 1.0$ and $\mathbf{u} = 0$ at time $t = 0$.

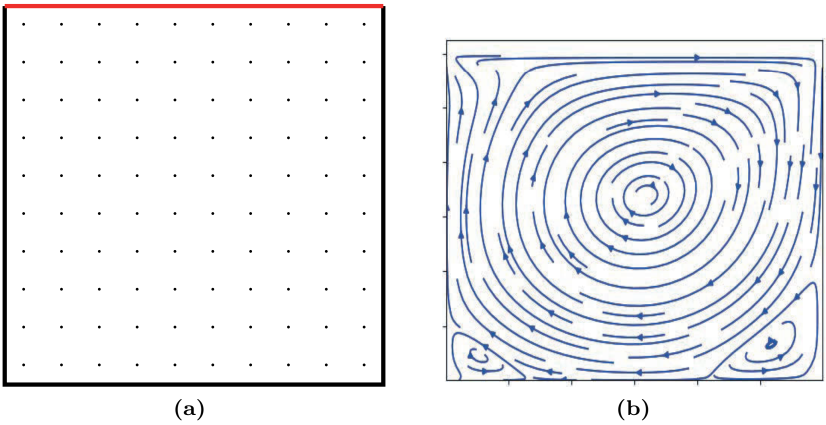


Figure 3: Lid-driven cavity. (a) The system consists of a quadratic box with hard walls drawn in black and a lid that slides to the right with a prescribed velocity, drawn in red. The lattice points all lie inside the box and the walls lie half way between boundary lattice points and (virtual) solid lattice points outside the box. (b) Streamlines in the steady state for a domain of 300 × 300 lattice points and $\omega = 1.7$, corresponding to Reynolds number 1000, given the typical velocity of the sliding lid as $u_{\text{lid}} = 0.1$. The figures is visualized using the `streamplot` function of the `matplotlib`[6]library. The lattice is resampled into a 30 × 30 lattice before determining the streamlines.

The lid moves with a given velocity u_{lid} in direction 1, i. e. to the east. We apply bounce-back boundary conditions on the black boundaries and the prescribed wall velocity boundary conditions on the red wall. The full boundary conditions can be written as

$$f_i = f_{i^*} - 6 w_i \rho_{\text{wall}} \mathbf{c}_i \cdot \mathbf{u}_{\text{lid}} \tag{11}$$

where i^* indicates the direction before bounce-back and ρ_{wall} is the fluid density at the wall. More information can be found in Ref. (Mohamad, 2011; Wolf-Gladrow, 2000). Note that we apply bounce back boundary conditions to all parallel domains, but we introduce ghost buffers only for domain boundaries in interior domains, not for the outer boundaries of edge domains. Using this approach, we do not need special treatment of boundary or interior domains. Bounce back of particles in an interior is simply overridden in the next communication step. Outer boundaries of edge domains are not communicated when specifying `periods=(False, False)` in

[6] `https://matplotlib.org/`

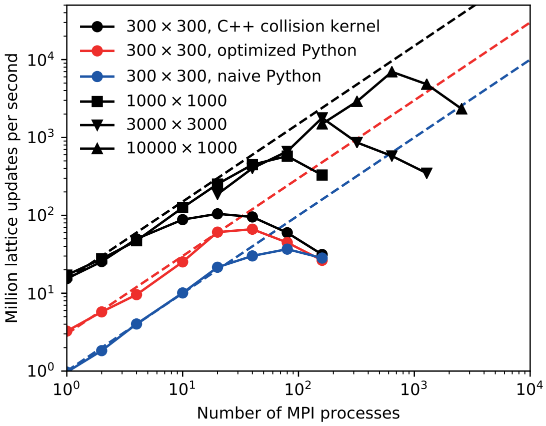
the call to `Create_cart` in Code Listing 3. This keeps our Python implementation as simple as possible and free of specific treatment of corner cases.

We calculated the velocity field in the steady state. Figure 3b shows the streamlines of the flow field after 1 million time steps for a lattice of 300 × 300 lattice points with $\omega = 1.7$ and the velocity of the lid chosen as $u_{\text{lid}} = 0.1$. The sliding lid induces a large vortex rotating clock wise. The two lower corners show small vortices rotating in the opposite direction.

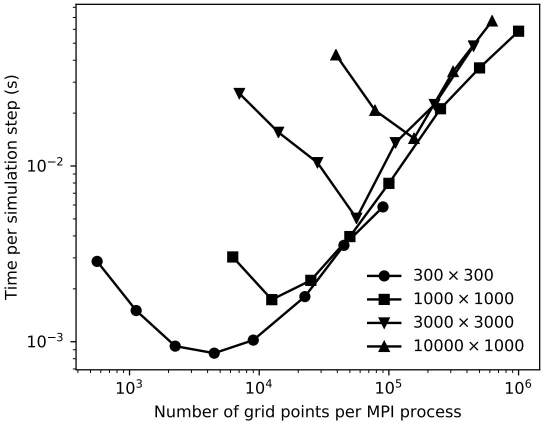
In order to demonstrate the scaling behavior for the parallel version of our code, we calculated the steady state solution of the problem for different sizes of the simulation domain 300×300, $1,000 \times 1,000$, $3,000 \times 3,000$ and $10,000 \times 10,000$ lattice points on bwForCluster NEMO at the University of Freiburg (2x Broadwell E5-2630v4 at 2.2 GHz per compute node with a 100 Gbit/s Omni-Path interconnect). The square computational lattice was divided in rectangular subregions, with each subregion assigned to one process.

A common measure for the speed of an implementation of the Lattice Boltzmann Method is the number of million lattice updates per second (MLUPS), which we compute for the lid-driven cavity. Fig. 4a reports this measure for the naive and optimized Python implementation as well as for the Python implementation using a C++ implementation of the collision kernel, all using double precision floating-point arithmetics. It is immediately clear that the Python&C++ implementation is fastest, reaching about 15 MLUPS on a single core, while the naive Python implementation reaches only 1 MLUPS. Scaling for all implementations is excellent and the dashed lines in Fig. 4a shows ideal scaling for the three implementations. For a $10,000 \times 10,000$ lattice, the Python&C++ implementation peaks at 7 billion lattice updates per second (BLUPS = 1000 MLUPS) when using 640 MPI process (32 NEMO compute nodes).

Figure 4b shows the execution time (per simulation step) as a function of number of lattice points per MPI process. For each lattice there is a minimum in execution time that moves to a smaller number of lattice points per MPI process with decreasing lattice size. This means the larger the overall lattice, the larger the portions that reside on each MPI process must be for scaling to be optimal.



(a)



(b)

Figure 4: Parallel scaling of the lid-driven cavity on bwForCluster NEMO. (a) Strong scaling test for different system sizes. The dashed lines show ideal scaling with 1 MLUPS, 3 MLUPS and 15 MLUPS per MPI process for the naive, optimized and C++ implementations. (b) Weak scaling test showing the execution time required for a single simulation step for the implementation using C++ collision kernels.

4 Discussion

Figure 4a shows that the three implementations achieve different execution speeds. The optimized Python implementation is already 3× faster than the naive implementation, while the C++ collision kernels gain another factor of 5× in speed. Scaling is good for all implementations and breaks down at a higher number of MPI processes for the slower implementations. For the 300×300 lattice, all three implementations achieve almost identical execution speed at 160 MPI processes. At this point, the execution time is entirely dominated by the cost of communication between the processes.

Parallel scaling breaks down at different points for different lattice sizes. More insights are obtained from the weak scaling test of Fig. 4b, which shows that there is a minimal number of lattice points per process, such that computation and not communication or idling dominates the aggregate cost. This number depends on the overall size of the lattice and is around $5,000$ for the small 300×300 lattice and $100,000$ for the largest $10,000 \times 10,000$ lattice. Assuming a square region per process, we end up with a surface to volume relation of about 0.06 to 0.01. Because the point of breakdown occurs at different surface to volume ratios, we believe that cost of communication is not its source. There are slight differences in run-time between the individual processes because the global lattice is not evenly divisible by the lattice used for domain decomposition, leading to slight variations in the size of the local domains. These differences in run-time are likely exacerbated at a larger number of total MPI processes, leading to some idling processes. More investigation is necessary to clarify this point. We also note that this result is specific for the D2Q9 model and is likely different for other two-dimensional and three-dimensional models.

The maximum performance of 7 BLUPS reached by our implementation is comparable to recent reports of 5 BLUPS reached on an implementation on multiple graphics processing units (GPUs) (Xu et al., 2018), albeit for a D3Q19 lattice and a multi-relaxation time collision operation, but (presumably) single precision floating-point arithmetics. Xu et al. (Xu et al., 2018) report 5 BLUPS on 12 NVIDIA K20M GPUs. Assuming linear scaling, we reach 5 BLUPS at 240 MPI processes or 24 NEMO compute nodes. An extensive test of the CPU-based LBM implementation *Musubi* (Hasert et al., 2014) using double precision floating-point arithmetics was recently presented by Qi (Qi, 2017). On Hazel Hen (Haswell E5-2680v3 at 2.50 GHz with a Cray Aries interconnect), *Musubi* achieves around 10 MLUPS per core for a D3Q19 lattice as compared to our 15 MLUPS per core for a D2Q9 lattice.

5 Conclusions

We have demonstrated a parallel implementation of the Lattice Boltzmann Method (LBM) in Python using the `numpy` library and the `mpi4py` bindings to the Message Passing Interface (MPI) that yields excellent scaling on bwForCluster NEMO. We have further shown that implementing the collision operation in the lower level programming language C++ yields a significant performance gain. It eliminates the creation of temporary buffers and avoids multiple loops over data structures. Our C++ optimization appears competitive with recently reported performances of implementations of the Lattice Boltzmann Method (Xu et al., 2018; Hasert et al., 2014), but one has to keep in mind that most published benchmarks are obtained for three-dimensional lattices while we here discuss only the two-dimensional case. Yet, our implementation can still be optimized further, e. g. by fusing streaming and collision steps, looping over data structures in a manner that maintains cache-coherency (Pohl et al., 2003) or by hiding communication behind computation. Future work will focus on extending the present code to three-dimensions and implementing further optimizations.

Implementing high-performance codes in Python has the advantage of fast development times and compact codes that can be used to test implementation and parallelization strategies before optimizing portions of the code in a lower-level language. Our naive parallel LBM implementation has 160 lines of Python code, 34 of which are a parallel implementation of `numpy.save` using MPI I/O. The simplicity of this code makes Python and MPI also suitable for teaching parallel computing. We note that more complex parallel simulation codes implemented largely in Python, for example the GPAW code (Mortensen et al., 2005; Enkovaara et al., 2010) and associated libraries (Larsen et al., 2017) for electronic structure calculations, have emerged over the past years. Python is maturing towards a language that allows rapid development of parallel simulation codes for high-performance computing systems. Libraries such as `pybind11` and `eigen` make extending Python with native numerical code straightforward.

Our full parallel implementation of the Lattice Boltzmann Method can be found online[7].

[7] `https://github.com/IMTEK-Simulation/LBWithPython`

Acknowledgements

The authors acknowledge the support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG as well as through the Research Unit FOR 2383 ProMiSe under Grant No. GR 2622/6-1.

Corresponding Author

Lars Pastewka: lars.pastewka@imtek.uni-freiburg.de
Department of Microsystems Engineering, University of Freiburg,
Georges-Köhler-Allee 103, 79110 Freiburg, Germany

ORCID

Lars Pastewka iD https://orcid.org/0000-0001-8351-7336

References

Aharonov, E. and D. H. Rothman (1993). »Non-Newtonian flow (through porous media): A lattice-Boltzmann method«. In: *Geophys. Res. Lett.* 20.8, pp. 679–682.

Bhatnagar, P. L., E. P. Gross and M. Krook (1954). »A model for collisionless processes in gases I: small amplitude processes in charged and neutral one-component systmes«. In: *Phys. Rev.* 94, pp. 511–525.

Boltzmann, L. (1896). *Vorlesungen über Gastheorie.* Barth.

Cercignani, C. (1988). *The Boltzmann Equation and its Applications.* Springer.

Enkovaara, J. et al. (2010). »Electronic structure calculations with GPAW: A real-space implementation of the projector augmented-wave method«. In: *J. Phys.: Condens. Matter* 22.25, p. 253202.

Geier, M., A. Greiner and J. G. Korvink (2006). »Cascaded digital Lattice Boltzmann automata for high Reynolds number flow«. In: *Phys. Rev. E* 73, p. 066705.

Geier, M., M. Schönherr, A. Pasquali and M. Krafczyk (2015). »The cumulant lattice Boltzmann equation in three dimensions: Theory and validation«. In: *Comput. Math. Appl.* 70.4, pp. 507–547.

Grunau, D., S. Chen and K. Eggert (1993). »A lattice Boltzmann model for multiphase fluid flows«. In: *Physics of Fluids A: Fluid Dynamics* 5.10, pp. 2557–2562.

Gunstensen, A. K. and D. H. Rothman (1993). »Lattice-Boltzmann studies of immiscible two-phase flow through porous media«. In: *J. Geophys. Res.* 98.B4, pp. 6431–6441.

Gunstensen, A. K., D. H. Rothman, S. Zaleski and G. Zanetti (1991). »Lattice Boltzmann model of immiscible fluids«. In: *Phys. Rev. A* 43.8, pp. 4320–4327.

Hasert, M. et al. (2014). »Complex fluid simulations with the parallel tree-based Lattice Boltzmann solver Musubi«. In: *J. Comput. Sci.* 5, pp. 784–794.

Huang, K. (1987). *Statistical Mechanics.* Wiley, New York.

Larsen, A. H. et al. (2017). »The Atomic Simulation Environment-A Python library for working with atoms«. In: *J. Phys.: Condens. Matter* 29, p. 273002.

McNamara, G. R. and G. Zanetti (1988). »Use of the Boltzmann equation to simulate lattice gas automata«. In: *Phys. Rev. Lett.* 56, pp. 2332–2335.

Mohamad, A. A. (2011). *Lattice Boltzmann Method.* Springer.

Mortensen, J. J., L. B. Hansen and K. W. Jacobsen (2005). »Real-space grid implementation of the projector augmented wave method«. In: *Phys. Rev. B* 71.3, p. 035109.

Obrecht, C., F. Kuznik, B. Tourancheau and J.-J. Roux (2011). »A new approach to the lattice Boltzmann method for graphics processing units«. In: *Comput. Math. Appl.* 61.12, pp. 3628–3638.

Pohl, T., M. Kowarschik, J. Wilke, K. Iglberger and U. Rüde (2003). »Optimization and profiling of the cache performance of parallel Lattice Boltzmann codes«. In: *Parallel Process. Lett.* 13.04, pp. 549–560.

Qi, J. (2017). »Efficient Lattice Boltzmann simulations on large scale high performance computing systems«. PhD thesis. Rheinisch-Westfälische Technische Hochschule Aachen.

Succi, S. (2001). *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Clarendon Press, Oxford.

Wolf-Gladrow, D. A. (2000). *Lattice Gas Cellular Automata and Lattice Boltzmann Models Mechanics.* Springer, Berlin.

Xu, L., A. Song and W. Zhang (2018). »Scalable parallel algorithm of multiple-relaxation-time Lattice Boltzmann Method with large eddy simulation on multi-GPUs«. In: *Sci. Program.* 2018, p. 1298313.

Zhang, J. (2011). »Lattice Boltzmann method for microfluidics: models and applications«. In: *Microfluid. Nanofluidics* 10.1, pp. 1–28.

DOI: 10.15496/publikation-29062

Numerical Investigation of Strongly Interacting Bosons at Zero Temperature

Laurent de Forges de Parny Frank Schäfer Jonas Mielke
Miguel A. Bastarrachea-Magnani Andreas Buchleitner

Physikalisches Institut, Albert-Ludwigs-Universität Freiburg, Germany

We review some numerical works carried out within the department for Quantum Optics and Statistics at the University of Freiburg's Institute of Physics, between September 2016 and June 2018. Our activities focus on quantum properties of matter at zero temperature, i. e., a regime where the thermal energy $k_B T$ is negligible with respect to the other energy scales of the considered system. This area of research, related to ultracold gases, has attracted a great deal of interest, both experimentally and theoretically, since the first realization of a Bose-Einstein condensate in 1995. In a context where the theoretical understanding of these systems still remains challenging, the growing power of computers offers a unique and efficient way to tackle such challenges. In our theory group, we particularly use powerful numerical methods that give *exact* results, in contrast to other theoretical approaches based on an *a priori* assumption, e. g., mean-field theory. To illustrate it, we focus on few typical results that would not be available other than by using high-performance computing. These results have been obtained by using three numerical methods: quantum Monte Carlo (QMC), Gutzwiller Monte Carlo (GMC), and the Multiconfigurational Time-dependent Hartree method for bosons (MCTDHX).

DOI: 10.15496/publikation-29050

1 Introduction

In the last century, high-performance computing has been of crucial importance in theoretical and applied sciences, e. g., for meteorological predictions and for nuclear studies. More precisely, the Manhattan project during World War II, is considered the starting point in the field of numerical computation. Then, the civil Fermi-Pasta-Ulam numerical simulation in 1953 definitely opened the perspective of solving physical problems not solvable otherwise (Dauxois et al., 2005). Nowadays, numerical simulations are used extensively in many fields of Science: Monte Carlo methods in statistical physics, particle physics, quantum chemistry, econometrics, *etc.*; Finite elements and Runge-Kutta for solving differential equations in mathematics, physics and for star and galaxy dynamics in astrophysics; Particle-in-Cell simulations and molecular dynamics in Biology, *etc.* (Ferrario et al., 2006).

In low-energy physics, numerical simulations are very practical tools for solving Hamiltonian models under active investigation in condensed matter and ultracold gases. The main constraint remains the diagonalization of the Hamiltonian matrix, a mathematical operation circumvented by many technics, e. g. exact diagonalization if possible, auxiliary-field, variational and quantum Monte Carlo, dynamical mean-field theory, the density matrix renormalization group (DMRG), and, more recently, tensor networks, MCTDHX and the time-DMRG (Dagotto, 1994; Scalapino, 2006; Román, 2014).

In this paper, we discuss a panel of numerical methods we used within the department for Quantum Optics and Statistics at the University of Freiburg's Institute of Physics, between September 2016 and June 2018. These methods allowed for extensive simulations of interacting bosons in a regime where the thermal energy $k_B T$ is negligible with respect to the energy scale of the Hamiltonian terms, i. e., in the zero temperature limit. The paper is organized as follows: In Sec. 2, we discuss the QMC method and the stochastic Green functions in the context of quantum magnetism. In Sec. 3, we present the GMC method: a flexible – but non-exact – numerical method we have developed and used for studying a recent system of interacting bosons placed in an optical lattice and inside a high-finesse optical cavity. Sec. 4 reports our dynamical investigation of ultracold bosons in a time-dependent double-well potential with the MCTDHX method. In Sec. 5, we summarize these results and give some final remarks.

2 QMC method: application to Bose-Einstein condensate with magnetic interactions

High-temperature superconductivity remains a central problem in solid-state physics. Although actively under investigation since the 90s, this phenomenon, which involves electricity transport without losing energy, is not described by a unique theory. The mechanism of superconductivity is inherent to quantum mechanics: free electrons – which are fermionic particles – are bound in pairs, called Cooper pairs, due to their interactions with the ionic lattice. The resulting Cooper pairs behave like bosons and condense into the ground state to form a Bose-Einstein condensate.[1] The consideration of the particles interactions are essential for the understanding of this mechanism. In this context, the Hubbard model, originally developed in 1963 to describe electrons in solids, has been proposed as a promising candidate for the understanding of high-temperature superconductivity (Nature Physics, 2013). Nevertheless, an exact solution exists only in 1D and higher-dimensional materials, i. e. most of the existing materials, are difficult to investigate. Once paired, the Cooper pairs lead to interacting bosons in an ionic lattice well described by the Bose-Hubbard model. This particularly simple model has not ceased to intrigue theoretical condensed-matter physicists, as the physics described by this model is rich and surprising. Indeed, it captures the essence of the paradigmatic superfluid-insulator transition, a transition of high interest in solid-state physics (Fisher et al., 1989). Since a general analytical solution of the (Bose)-Hubbard model does not exist in 2D and 3D, many computational techniques have been developed, e. g. quantum Monte Carlo, dynamical mean-field theory, and tensor networks, to cite a few (Dagotto, 1994; Scalapino, 2006; Román, 2014).

A remarkable idea in the context of atomic physics was the proposal in 1998 to implement the Bose-Hubbard model by loading a Bose-Einstein condensate into an optical lattice (Jaksch et al., 1998). This seminal idea has opened new perspectives in the field of quantum phase transition and led to the first observation of the Mott-superfluid transition in 2002 (Greiner et al., 2002). Subsequently, these experiments made possible the achievement of low-dimensional systems where new phases can emerge (Bloch et al., 2008). In particular, these systems allow for the investigation of more complex models, called *extended* Bose-Hubbard models, which

[1] cf. BCS theory, Nobel Prize in Physics in 1972.

consider additional ingredients like spin-spin interactions. This opens the possibility to investigate the interplay between magnetism and superfluidity (Vengalattore, Leslie et al., 2008; Vengalattore, Guzman et al., 2010) and, more recently, quantum phase transitions with spin degrees of freedom (Jiang et al., 2016).

We focus here on a spin-1 bosonic model for which a quantum magnetic state, the nematic state, arises from the magnetic interactions. This non-trivial nematic state is particularly surprising since the spin-rotation symmetry is spontaneously broken without magnetic ordering: the spins are randomly anti-aligned and the total magnetization vanishes (Zibold et al., 2016; Jacob et al., 2012; Forges de Parny, Yang et al., 2014). The extended Bose-Hubbard Hamiltonian describing this model reads (Mahmud et al., 2013; Imambekov et al., 2003; Imambekov et al., 2004; Ho, 1998)

$$\begin{aligned}\hat{\mathcal{H}} \;&=\; \underbrace{-t\sum_{\sigma,\langle\mathbf{r},\mathbf{r}'\rangle}\left(a^{\dagger}_{\sigma\mathbf{r}}a_{\sigma\mathbf{r}'}+\mathrm{h.}c.\right)}_{\text{kinetic term}}+\underbrace{\frac{U_0}{2}\sum_{\mathbf{r}}\hat{n}_{\mathbf{r}}\left(\hat{n}_{\mathbf{r}}-1\right)}_{\text{on-site repulsion}}+\underbrace{\frac{U_2}{2}\sum_{\mathbf{r}}\left(\hat{\mathbf{S}}^2_{\mathbf{r}}-2\hat{n}_{\mathbf{r}}\right)}_{\text{on-site spin-spin interaction}}\\ &\quad-\underbrace{q\sum_{\mathbf{r}}\hat{n}_{0\mathbf{r}}}_{\text{quadratic Zeeman}}-\underbrace{\mu\sum_{\sigma,\mathbf{r}}\hat{n}_{\sigma\mathbf{r}}}_{\text{chemical potential}}\quad,\end{aligned}\tag{1}$$

where operator $a_{\sigma\mathbf{r}}$ ($a^{\dagger}_{\sigma\mathbf{r}}$) annihilates (creates) a boson in the Zeeman state $\sigma=\{\pm1,0\}$ on site $\mathbf{r}$ of a periodic square optical lattice of size $L\times L$. The kinetic term allows particles to hop between neighboring sites $\langle\mathbf{r},r'\rangle$ with strength t. $N_\sigma\equiv\sum_{\mathbf{r}}\langle a^{\dagger}_{\sigma\mathbf{r}}a_{\sigma\mathbf{r}}\rangle$ denotes the total number of σ bosons, $\rho_\sigma\equiv N_\sigma/L^2$ the corresponding density, and $\rho\equiv\sum_\sigma\rho_\sigma$ the total density. The operator $\hat{\mathbf{S}}_{\mathbf{r}}=(\hat{S}_{x,\mathbf{r}},\hat{S}_{y,\mathbf{r}},\hat{S}_{z,\mathbf{r}})$ is the spin operator where $\hat{S}_{\alpha,\mathbf{r}}=\sum_{\sigma,\sigma'}a^{\dagger}_{\sigma\mathbf{r}}J_{\alpha,\sigma\sigma'}a_{\sigma'\mathbf{r}}$, $\alpha=\{x,y,z\}$ and the $J_{\alpha,\sigma\sigma'}$ are standard spin-1 matrices. The parameters $U_0>0$ and U_2 are the on-site spin-independent and spin-dependent interaction terms. In the following, we will consider antiferromagnetic $^{23}\mathrm{N}a$ atoms for which $U_2/U_0\simeq0.036$. For these atoms, the local magnetic moment $S^2(0)\equiv\frac{1}{L^2}\sum_{\mathbf{r}}\langle\hat{\mathbf{S}}^2_{\mathbf{r}}\rangle$ is minimized, which means that the spins of atoms anti-align to form a singlet if possible. Finally, μ is the chemical potential. We have previously derived the exact phase diagram without considering the quadratic Zeeman term, i. e., for $q=0$, and investigate the properties of the nematic state (Forges de Parny, Yang et al., 2014; Forges de Parny, Hébert et al., 2013). The purpose of our recent study was to show that the quadratic Zeeman effect – a control parameter in experiments – is a control parameter for tuning the nematic state and the Mott-superfluid transition (Forges de Parny and Rousseau, 2018). In the following, we discuss our numerical approach to tackling this question.

According to standard quantum mechanic rules, all the physical information is obtained by diagonalizing the Hamiltonian matrix associated with operator Eq. (1). Typically, in a bi-dimensional lattice experiment, $N \sim 10^5$ atoms are trapped in $\sim 10^8$ sites. The square matrix to diagonalize has a size of $\sim 10^{300000} \times 10^{300000}$ which is far from the total estimated number of atoms in our known Universe, i. e., 10^{80} atoms. Even if the Hamiltonian matrix is sparse, its diagonalization is impossible. To overcome this problem, we have used a powerful numerical method, the quantum Monte Carlo method, which gives exact results for the most probable state at equilibrium, i. e., the ground state $|\psi_g\rangle$ (Forges de Parny, Hébert et al., 2013; Forges de Parny and Rousseau, 2018). Like all methods, this one has its own limitations: the sign problem (Loh et al., 1990), long time convergence and CPU time cost, implementation difficulties, etc. Nevertheless, using the NEMO cluster and parallelized simulations, we were able to investigate $N = 288$ interacting bosons in $L \times L = 144$ sites, i. e., a system described by a Hamiltonian matrix of size $\sim 10^{118} \times 10^{118}$ which is far beyond the limits of the exact diagonalization methods (even with the Lanczos algorithm). More technically, for $N = 288$ bosons in $L \times L = 144$ sites, we obtained converged results with a thermalization time of 10 hours and a measurement time of 1 day and 10 hours, using MPI parallelized simulations with 20 processors per node and 2 GB of persistent memory on bwForCluster NEMO. Typically, the total ground state energy $E_g = \langle\psi_g|\hat{\mathcal{H}}|\psi_g\rangle$ – which converges faster than other quantities – is converged with an accuracy of 0.001%.

We have used QMC simulations with the stochastic Green function (SGF) algorithm, developed by Valy Rousseau (Rousseau, 2008). Whereas most of the QMC algorithms sample the partition function $\mathcal{Z}(\beta) = \sum_\psi \langle\psi|e^{-\beta\hat{\mathcal{H}}}|\psi\rangle$ (Fehske et al., 2007; Sandvik, 2010), with $\beta = 1/k_B T$ the inverse of temperature, the SGF algorithm samples the *extended* partition function

$$\mathcal{Z}(\beta,\tau) = \sum_\psi \langle\psi|e^{-(\beta-\tau)\hat{\mathcal{H}}}\hat{\mathcal{G}}e^{-\tau\hat{\mathcal{H}}}|\psi\rangle \ , \tag{2}$$

where $\hat{\mathcal{G}}$ is the the Green operator defined by

$$\hat{\mathcal{G}} = \sum_{p=0}^{+\infty}\sum_{q=0}^{+\infty} g_{pq} \sum_{\left\{i_p|j_q\right\}} \prod_{k=1}^{p} \hat{\mathcal{A}}_{i_k}^{\dagger} \prod_{l=1}^{q} \hat{\mathcal{A}}_{j_l}. \tag{3}$$

In Eq. (3), the notation $\left\{i_p|j_q\right\}$ denotes two subsets of site indices $i_1, i_2, \cdots, i_p$ and $j_1, j_2, \cdots, j_q$ with the constraint that all indices in subset i are different from the indices in subset j (but several indices in one subset may be equal), and g_{pq} is a matrix that depends on the application of the algorithm. In the occupation number representation, the normalized creation and annihilation operators,

$$\hat{\mathcal{A}}^{\dagger} = a^{\dagger}\frac{1}{\sqrt{\hat{n}+1}} \qquad \hat{\mathcal{A}} = \frac{1}{\sqrt{\hat{n}+1}}a, \tag{4}$$

satisfy the following relations for any state $\big|n\big\rangle$,

$$\hat{\mathcal{A}}^{\dagger}\big|n\big\rangle = \big|n+1\big\rangle \qquad \hat{\mathcal{A}}\big|n\big\rangle = \big|n-1\big\rangle, \tag{5}$$

with the particular case $\hat{\mathcal{A}}\big|0\big\rangle = 0$. The SGF algorithm can be applied to any lattice Hamiltonian of the form $\hat{\mathcal{H}} = \hat{\mathcal{V}} - \hat{\mathcal{T}}$, where $\hat{\mathcal{V}}$ is diagonal in the chosen occupation number basis and $\hat{\mathcal{T}}$ has only positive matrix elements (to avoid the sign problem). The extended partition function, expressed in terms of Feynman path integrals in continuous imaginary time, takes the form

$$\begin{aligned} \mathcal{Z}(\beta,\tau) \;=\; & \sum_{n\geq 0} \int_{0<\tau_1<\cdots<\tau_n<\beta} \big\langle\psi_0\big|e^{-\beta\mathcal{V}}\hat{\mathcal{T}}(\tau_n)\big|\psi_{n-1}\big\rangle\big\langle\psi_{n-1}\big|\hat{\mathcal{T}}(\tau_{n-1})\big|\psi_{n-2}\big\rangle \\ & \times\cdots\big\langle\psi_{L+1}\big|\hat{\mathcal{T}}(\tau_L)\big|\psi_L\big\rangle\big\langle\psi_L\big|\hat{\mathcal{G}}(\tau)\big|\psi_R\big\rangle\big\langle\psi_R\big|\hat{\mathcal{T}}(\tau_R)\big|\psi_{R-1}\big\rangle \\ & \times\cdots\big\langle\psi_2\big|\hat{\mathcal{T}}(\tau_2)\big|\psi_1\big\rangle\big\langle\psi_1\big|\hat{\mathcal{T}}(\tau_1)\big|\psi_0\big\rangle d\tau_1\cdots d\tau_n, \end{aligned} \tag{6}$$

where the sum $\sum_{n\geq 0}$ implicitly runs over complete sets of states $\left\{\big|\psi_k\big\rangle\right\}$, and $\hat{\mathcal{T}}(\tau)$ and $\hat{\mathcal{G}}(\tau)$ the time-dependent operators defined by

$$\hat{\mathcal{T}}(\tau) = e^{\tau\hat{\mathcal{V}}}\hat{\mathcal{T}}e^{-\tau\hat{\mathcal{V}}}, \quad \hat{\mathcal{G}}(\tau) = e^{\tau\hat{\mathcal{V}}}\hat{\mathcal{G}}e^{-\tau\hat{\mathcal{V}}}. \tag{7}$$

The main advantage of sampling this extended partition function is the possibility to measure high-order correlation functions, such as n-points Green functions. In addition, the strength of the SGF algorithm is the possibility to simulate Hamiltonians with high-order terms that cannot be treated by other methods, e. g., many species systems with transmutation. Also, this algorithm works in any dimension in both the canonical and grand-canonical ensembles with an acceptance rate of the global updates of 100% for any Hamiltonian. All details about the SFG algorithm are discussed in Refs. (Rousseau, 2008).

We now discuss the application to the SGF algorithm for the study of Hamiltonian Eq. (1). The aim is to emphasize the capacity of this method to bring reliable results far from approximate methods, typically the mean-field approximation largely used in the literature. The mean-field phase diagram of Eq. (1) for many values of the quadratic Zeeman parameter q is plotted in Fig. 1 (a). We observe incompressible Mott insulator (MI) lobes with density $\rho = 1, 2$ (in black for $q/U_0 = 100$), and a compressible superfluid phase otherwise. The false colors show the total condensate fraction $C^{MF} = \sum_\sigma |\langle a_\sigma \rangle|^2$, i. e., the signature of a quantum liquid, for $q/U_0 = 100$. The interesting fact is that the boundary of the $\rho = 2$ MI lobe is tunable with q. Also, the nature of the quantum phase transition changes with q: dashed (plain) lines indicate first (second) order transitions. The mean-field phase diagram of Eq. (1) takes few minutes to be calculated, but is not reliable for low dimensional systems (the higher is the dimensionality, the higher is the mean-field accuracy).

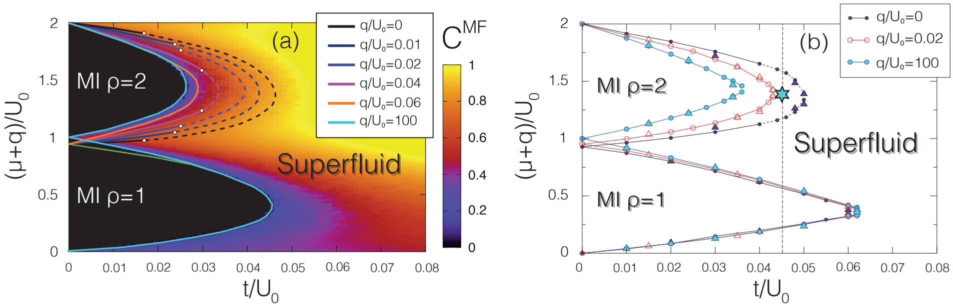


Figure 1: (Color online) (a) Mean field and (b) exact QMC phase diagram of model Eq. (1) for $L = 8$ (circles) and $L = 12$ (triangles) with respect to the quadratic Zeeman parameter q. Two phases are present: the Mott insulator (MI) with ρ particle per site and the superfluid phase. Contrary to the $\rho = 1$ MI region, q strongly affects the tip of the $\rho = 2$ Mott lobe. The dashed (plain) lines indicate a first- (second-) order transition. (a) False colors show the total condensate fraction $C^{MF} = \sum_\sigma |\langle a_\sigma \rangle|^2$ for $q/U_0 = 100$ and white dots indicate tricritical points. (b) The cyan star indicates the parameters chosen in Fig. 2.

The exact QMC phase diagram, using the SGF algorithm, is plotted in Fig. 1 (b). Such a plot takes less than 4 days to be calculated for many sizes L with parallelized simulations on the NEMO cluster. As expected, we observe MI and superfluid phases, and the shape of the diagrams is roughly the same as the mean-field ones (nevertheless, the QMC $\rho = 1$ MI lobe is sharper). The main difference is quantitative: the mean field is known for minimizing the quantum fluctuations. As a consequence, the tip of the MI lobe, where the MI-superfluid transition takes place, qualitatively differs: $t_c/U_0{}^{MF} < t_c/U_0{}^{QMC}$. The nature of the quantum phase transition also differs: the mean field suggests a first-order MI-superfluid transition at the tip of the Mott lobe for $\rho = 2$ and $q/U_0 = 0.02$. This statement is invalidated by our QMC results (Forges de Parny and Rousseau, 2018).

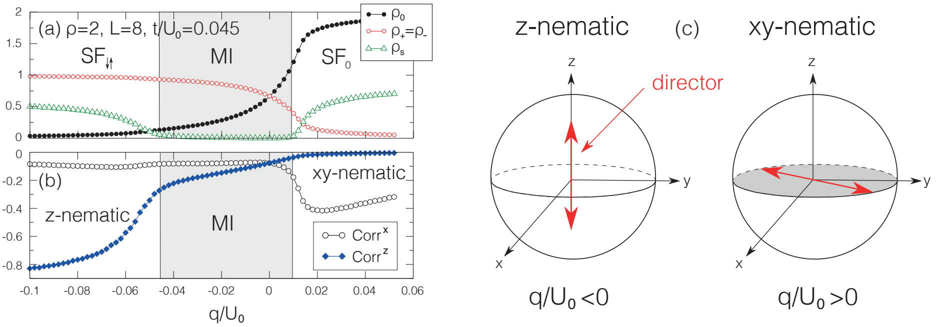


Figure 2: (Color online) (a) QMC data for $\rho = 2$ and $L = 8$ at fixed hopping $t/U_0 = 0.045$: q acts as a control parameter for the MI-SF transition. For $q \to -\infty$ the system is in the $\mathrm{SF}_{\downarrow\uparrow}$ with a nematic director along z-axis and enters in the MI phase at $q/U_0 \simeq -0.045$ when increasing q. Then, the system continuously adopts a SF_0 phase with a nematic director belonging to the xy-plane at $q/U_0 \simeq 0.01$. Both transitions are second order. (b) Sketch of the two observed nematic structures.

The QMC method also allows us to prove that the quadratic Zeeman parameter is a control parameter for both the MI-superfluid transition and for the nematic structure. To emphasize this effect, we fix $t/U_0 = 0.045$ such that the system is in the MI phase for $q = 0$ (cyan star in Fig. 1 (b)). Fig. 2 (a) shows how the spin populations evolve when varying q/U_0: for $q/U_0 \to -\infty$ $(+\infty)$ the system is populated by spins $\sigma = \pm 1$ $(\sigma = 0)$ only. More interestingly, q acts as a control parameter for the MI-superfluid transition: for $q \to -\infty$ the system is in the $\mathrm{SF}_{\downarrow\uparrow}$ phase with a nematic director along z-axis and enters in the MI phase at $q/U_0 \simeq -0.045$ when increasing q. Then, the system continuously adopts a SF_0 phase with a nematic director belonging to the xy-plane at $q/U_0 \simeq 0.01$. Both transitions are second order. The superfluid phase is indicated by a non-zero superfluid density ρ_s. The structure of the nematic order is captured by the spin-spin correlation functions $\mathrm{C}orr^{\alpha} = \frac{1}{L^4}\sum_{\mathbf{r},R\neq 0}\langle \hat{S}_{\alpha,\mathbf{r}}\hat{S}_{\alpha,\mathbf{r}+R}\rangle \neq 0$, with $\alpha = x, y, z$, and $\mathrm{C}orr^{x} = \mathrm{C}orr^{y}$, plotted in Fig. 2 (b). The spins anti-align along a director oriented in the z axis for $q/U_0 < 0$ ($\mathrm{C}orr^{z} \neq 0$), whereas the nematic director belongs to the xy plane for $q/U_0 > 0$ ($\mathrm{C}orr^{x,y} \neq 0$). The two nematic structures observed are drawn in Fig. 2 (c). These spin-spin correlation functions are not accessible with the mean-field theory.

3 GMC method: application to ultracold gases with infinite-range cavity-mediated interactions

As in Sec. 2, we discuss an extended Bose-Hubbard model implemented in ultracold atoms experiments. We especially focus on a system recently investigated in the Esslinger's group at ETH-Zurich (Landig et al., 2016), where a cloud of cold atoms is placed in an optical lattice and inside a high-finesse optical cavity. The field of the cavity mediates an effective infinite-range interaction between the atoms, which favors a density difference between neighboring sites of the optical lattice. This experiment attracted a strong interest as it provides one of the first observations of the elusive *supersolid phase*, a phase initially discussed by Penrose and Onsager in 1956 (Penrose et al., 1956; Gross, 1957). The supersolid phase is characterized by both long-range phase coherence and spatial ordering, i. e., simultaneous diagonal and off-diagonal long-range orders. Our motivation was to bring additional and crucial information regarding the thermodynamic stability of the supersolid phase and elucidate the nature of the quantum phase transitions.

The experimental system is well described by the extended Bose-Hubbard model with an additional long-range interaction (Landig et al., 2016)

$$\begin{aligned}\hat{\mathcal{H}} = & \underbrace{-t\sum_{\langle i,j\rangle}\left(a_i^{\dagger}a_j + \mathrm{H.}c.\right)}_{\text{kinetic term}} + \underbrace{\frac{U_s}{2}\sum_{i\in e,o}\hat{n}_i\left(\hat{n}_i - 1\right)}_{\text{on-site repulsion}} \\ & \underbrace{-\frac{U_l}{K}\left(\sum_{i\in e}\hat{n}_i - \sum_{i\in o}\hat{n}_i\right)^2}_{\text{long-range interaction}} - \underbrace{\mu\sum_{i\in e,o}\hat{n}_i}_{\text{chemical potential}}.\end{aligned} \tag{8}$$

This model describes spinless bosons on a square optical lattice inside a high-finesse optical cavity. The first term corresponds to the kinetic energy with tunneling amplitude t between nearest neighboring sites $\langle i,j\rangle$ defined on a square lattice with periodic boundary conditions and $L \times L$ sites. The kinetic term favors delocalization of bosons on the lattice and supports the superfluidity. The bosonic operator $a_i^\dagger$ (a_i) creates (annihilates) an atom at site i, and $\hat{n}_i = a_i^\dagger a_i$ is the corresponding number operator. The second term represents the on-site repulsive interactions between the atoms with strength $U_s > 0$. The index e and o denote respectively even and odd lattice sites. The third term describes the long-range interaction with amplitude $U_l > 0$ and favours imbalanced populations between even and odd sites, i. e. density oscillation. Finally, μ denotes the chemical potential.

The experimental phase diagram, plotted in Fig. 3, comprises four phases. For large negative detuning, the system adopts the superfluid phase for small interaction U_s/t, and the Mott insulator otherwise.

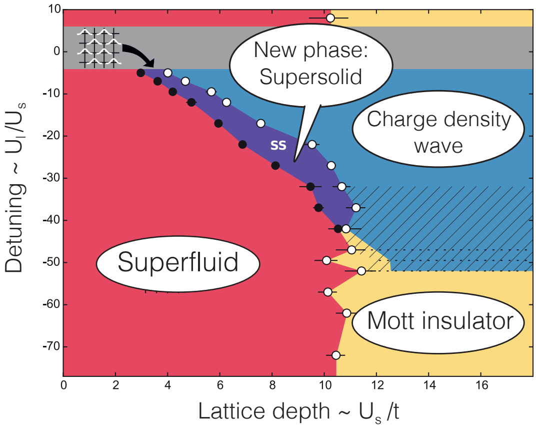


Figure 3: (Color online) Experimental phase diagram of Eq. (8) for one particle per site. Four phases are observed: a superfluid and a Mott insulator for small U_l, and a charge density wave and a supersolid phase for larger U_l. Taken from Ref. (Landig et al., 2016).

When the long-range interactions are turned on, the Mott insulator evolves in the charge density wave, i. e., an insulating phase with density oscillation, and more interestingly, a supersolid phase appears. The supersolid supports both the phase coherence of the superfluid and the density oscillation of the charge density wave. In the experimental study, the stability of the supersolid remained an open question. Also, the nature of the quantum phase transitions has not been fully elucidated.

In this context, we have performed numerical simulations to tackle these questions. More generally, the understanding of the spectral properties of Hamiltonian Eq. (8) has been extensively studied in our group by Jonas Mielke (MSc Thesis, 2018). At the ground state level, we have used QMC simulations, with the SGF algorithm discussed in Sec. 2, and developed an improved mean-field method, the Gutzwiller Monte Carlo method (Flottat et al., 2017). The GMC and QMC phase diagrams are respectively plotted in Fig. 4 (a) and (b).

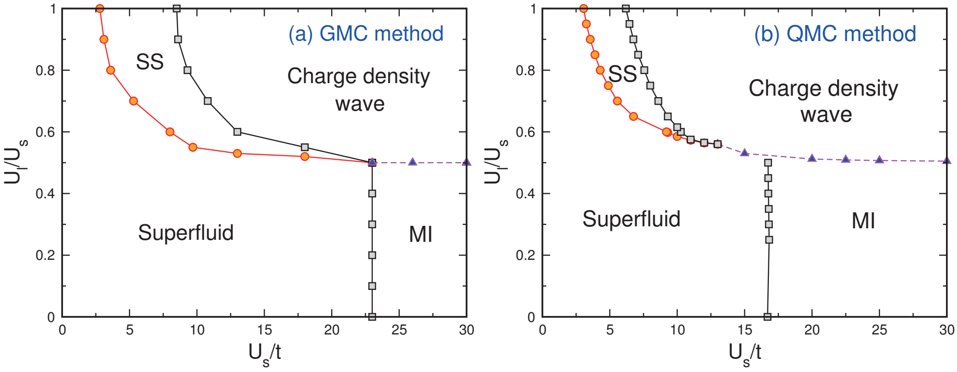


Figure 4: (Color online) (a) Zero temperature mean-field GMC and (b) exact QMC phase diagrams from Ref. (Flottat et al., 2017). The four experimental phases of Fig. 3 are found in both GMC and QMC phase diagrams. The QMC method reports a smaller SS region than the GMC one. Also, the superfluid to charge density wave transition is not observed in the GMC phase diagram.

As expected, we observe the four experimental phases of Fig. 3, namely the superfluid, the supersolid, the Mott insulator and the charge density wave. Both methods allow us to prove the stability of the supersolid phase with one particle per site and we unveiled the nature of the quantum phase transitions with QMC simulations. The differences observed in Fig. 4 (a) and (b) are not surprising since the GMC does not take into account the quantum fluctuations and therefore minimizes the critical hopping at the Mott-superfluid transition, i. e., $t_c^{GMC} < t_c^{QMC}$, then $U_s/t_c^{GMC} \sim 23 > U_s/t_c^{QMC} \sim 16$. This reason explains the size difference of the SS phase in Fig. 4 (a) and (b), i. e., the SS region is smaller in Fig. 4 (b). Also, the GMC methods does not allow for the direct observation of the superfluid to charge density wave transition. Nevertheless, the GMC method present real advantages:

- 3D systems are investigable (otherwise difficult with QMC);
- give access to the correlation functions (not standard in other mean-field formulations);
- gauge fields are investigable;
- take into account the thermal fluctuations in exact fashion;
- describes the Ising, XY and Berezinskii-Kosterlitz-Thouless transitions at finite temperature;
- easy to implement and requires small CPU and RAM resources (Fig. 4 (a) took less than 24h CPU).

We now describe the GMC method, based on Ref. (Hickey et al., 2014), developed by L. de Forges de Parny and T. Roscilde at ENS Lyon. This numerical method is built on the combination of both the Gutzwiller ansatz and the classical Monte Carlo method with Metropolis algorithm (Metropolis et al., 1953). This results in a *semi-classical lattice field theory* which preserves the U(1) symmetry, i. e. the gauge invariance, which is a great advantage compared to most of the mean-field approaches used in the literature. This method also allows for the *artificial reconstruction* of correlation functions on a $L \times L$ lattice cluster. More explicitly, the Gutzwiller mean-field state, written in the Fock basis $\{|\boldsymbol{n}_i\rangle\}$ with upper truncation n_{max}, takes the form

$$|\Psi(\mathbf{f})\rangle = \bigotimes_{i=1}^{L^2} |\psi_i\rangle = \bigotimes_{i=1}^{L^2} \left(\sum_{\boldsymbol{n}_i=0}^{n_{\max}} f_{\boldsymbol{n}_i}^{(i)} |\boldsymbol{n}_i\rangle \right) , \tag{9}$$

where L^2 is the total number of sites and $\mathbf{f} = \{f_{\boldsymbol{n}_i}^{(i)}\}$ is a vector of $(n_{\max} + 1) \times L^2$ complex coefficients, satisfying the normalization constraints $\sum_{\boldsymbol{n}} |f_{\boldsymbol{n}}^{(i)}|^2 = 1$ on each site. Because of this constraints, we have that

$$f_{\boldsymbol{n}_i}^{(i)} = A_{\boldsymbol{n}_i}^{(i)} e^{i\phi_{\boldsymbol{n}_i}^{(i)}} \tag{10}$$

with $0 \leq A_{\boldsymbol{n}_i}^{(i)} \leq 1$. In the program, the amplitudes $A_{\boldsymbol{n}_i}^{(i)}$ and phases $\phi_{\boldsymbol{n}_i}^{(i)}$are long float numbers stored in two arrays of dimension $(n_{max} + 1) \times L^2$, which is not a constraint for the sizes we have investigated ($L < 100$).

The GMC method consists of updating both the amplitudes $A^{(i)}_{\boldsymbol{n}_i}$ and the phases $\phi^{(i)}_{\boldsymbol{n}_i}$ using the metropolis algorithm with energy $E(\mathbf{f}) = \langle\Psi(\mathbf{f})|\hat{\mathcal{H}}|\Psi(\mathbf{f})\rangle$ which takes the explicit form

$$\begin{aligned} E(\mathbf{f}) \;\; = \;\; & -2t\sum_{\langle i,j\rangle}\sum_{\boldsymbol{n}_i,\boldsymbol{n}_j}\gamma_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j)\cos\left(\Delta\phi_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j)\right) \\ & +\sum_{i}\sum_{\boldsymbol{n}_i}(A^{(i)}_{\boldsymbol{n}_i})^2\Big[\,\frac{U_s}{2}n_i(n_i-1)-\mu n_i\,\Big] \qquad (11) \\ & -\frac{U_l}{K}\left(\sum_{i\in e}\sum_{\boldsymbol{n}_i}(A^{(i)}_{\boldsymbol{n}_i})^2 n_i-\sum_{i\in o}\sum_{\boldsymbol{n}_i}(A^{(i)}_{\boldsymbol{n}_i})^2 n_i\right)^2 \;, \end{aligned}$$

with

$$\begin{aligned} \gamma_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j) &= \sqrt{n_i n_j}\; A^{(i)}_{\boldsymbol{n}_i}A^{(i)}_{\boldsymbol{n}_i-1}A^{(j)}_{\boldsymbol{n}_j}A^{(j)}_{\boldsymbol{n}_j-1}, \\ \Delta\phi_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j) &= \left(\phi^{(i)}_{\boldsymbol{n}_i}-\phi^{(i)}_{\boldsymbol{n}_i-1}\right)-\left(\phi^{(j)}_{\boldsymbol{n}_j}-\phi^{(j)}_{\boldsymbol{n}_j-1}\right)\;. \qquad (12) \end{aligned}$$

The above model, Eq. (12), represents the Hamiltonian of a generalized XY model with fluctuating couplings. The XY spins live on a $(2D+1)$-dimensional lattice with sites $(i,\boldsymbol{n}_i)$, where the extra dimension is provided by the occupation number. At the heart of Gutzwiller Monte Carlo is the simplification of the Boltzmann weight:

$$\langle\Psi(\mathbf{f})|\; e^{-\beta\hat{\mathcal{H}}}\;|\Psi(\mathbf{f})\rangle \to e^{-\beta E(\mathbf{f})}. \qquad (13)$$

The classical Monte Carlo simulation contains two kinds of single-site Metropolis moves: amplitude moves, preserving the sum constraint, and phase moves. A subtle aspect concerns the transition probabilities for amplitude moves as the amplitude appears in the metric of the integrals defining the partition function with a linear term of the kind $A^{(i)}_{\boldsymbol{n}_i} = \exp[\log(A^{(i)}_{\boldsymbol{n}_i})]$. This means that an amplitude move on site i, changing $\{A^{(i)}_{\boldsymbol{n}_i}\}$ into $\{A'^{(i)}_{\boldsymbol{n}_i}\}$, has to be accepted with probability

$$P = \min\left(e^{-\beta[E(\{A'^{(i)}_{\boldsymbol{n}_i}\})-E(\{A^{(i)}_{\boldsymbol{n}_i}\})]}\times e^{\sum_{\boldsymbol{n}_i}\ln(A'^{(i)}_{\boldsymbol{n}_i}/A^{(i)}_{\boldsymbol{n}_i})},1\right). \qquad (14)$$

Our simulations were performed on a square lattice of size $L\times L$ with periodic boundary conditions. The average total energy is given by $E = \langle E(\mathbf{f})\rangle$ and the total density is defined by $\rho = \frac{1}{L^2}\langle\sum_i\sum_{\boldsymbol{n}_i}n_i(A^{(i)}_{\boldsymbol{n}_i})^2\rangle$, where $\langle . \rangle \equiv \frac{1}{N_{\mathrm{MC}}}\sum_n^{N_{\mathrm{MC}}}(.)$

is the Monte Carlo average. This part is implemented by a loop in the program. The signature of a long-range order phase coherence is given by a finite condensate fraction $C = \frac{1}{L^4}\sum_{i,j}\langle a_i^\dagger a_j\rangle$ given by

$$\begin{aligned} C = & \quad \frac{1}{L^4}\left\langle\left[\sum_{i,\boldsymbol{n}_i}\gamma_i(\boldsymbol{n}_i)\cos(\phi^{(i)}_{\boldsymbol{n}_i}-\phi^{(i)}_{\boldsymbol{n}_i-1})\right]^2\right\rangle \\ & +\frac{1}{L^4}\left\langle\left[\sum_{i,\boldsymbol{n}_i}\gamma_i(\boldsymbol{n}_i)\sin(\phi^{(i)}_{\boldsymbol{n}_i}-\phi^{(i)}_{\boldsymbol{n}_i-1})\right]^2\right\rangle \quad , \end{aligned} \tag{15}$$

with $\gamma_i(\boldsymbol{n}_i) = \sqrt{n_i}\ A^{(i)}_{\boldsymbol{n}_i}A^{(i)}_{\boldsymbol{n}_{i-1}}$. The phase coherence signal is also captured by the superfluid density ρ_s, defined as the second derivative of the free-energy density $f = -\frac{1}{L^2\beta}\log\mathcal{Z}$ with respect to a phase twist φ of the creation and annihilation operators along a given (x) direction such that $a_x^\dagger \to a_x^\dagger e^{i\varphi x/L}$ and $a_x \to a_x e^{-i\varphi x/L}$, and with $\mathcal{Z}$ the partition function and $\beta = 1/k_BT$. The superfluid density reads

$$\rho_s = \left.\frac{\partial^2 f}{\partial\varphi^2}\right|_{\varphi=0} = \frac{1}{L^2}\left.\left\langle\frac{\partial^2\mathcal{H}(\varphi)}{\partial\varphi^2}\right\rangle\right|_{\varphi=0} - \frac{\beta}{L^2}\left.\left\langle\left(\frac{\partial\mathcal{H}(\varphi)}{\partial\varphi}\right)^2\right\rangle\right|_{\varphi=0} . \tag{16}$$

In the framework of the GMC formulation and using the energy formulation Eq. (12), the superfluid density becomes

$$\begin{aligned} \rho_s &= \frac{2t}{L^2}\left\langle\sum_{\langle i,j\rangle_x}\sum_{\boldsymbol{n}_i,\boldsymbol{n}_j}\gamma_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j)\cos(\Delta\phi_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j))\right\rangle \\ &- \frac{4t^2}{TL^2}\left\langle\left(\sum_{\langle i,j\rangle_x}\sum_{\boldsymbol{n}_i,\boldsymbol{n}_j}\gamma_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j)\sin(\Delta\phi_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j))\right)^2\right\rangle \quad , \end{aligned} \tag{17}$$

where $\gamma_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j)$ and $\Delta\phi_{ij}(\boldsymbol{n}_i,\boldsymbol{n}_j)$ are defined by Eqs. (12). The information of the spatial order is given by the structure factor $S(\mathbf{k})$, i. e. the Fourier transform of the density-density correlation function, $S(\mathbf{k}) = \frac{1}{L^2}\sum_{\mathbf{r},r'} e^{i\mathbf{k}.(r-r'))}\langle n_{\mathbf{r}}n_{\mathbf{r}'}\rangle$. Particularly, the density oscillations observed in the charge density wave and in the supersolid phases are signaled by $S(\pi,\pi)\neq 0$.

At finite temperature, we have obtained promising results. For instance, Fig. 5 shows the expected scaling behavior of the condensate fraction Eq. (15) at a Berezinskii-Kosterlitz-Thouless transition, i. e., $C(T_{BKT}) \propto L^{-1/4}$. Also, we observe the associated well-known universal jump of the superfluid density $\rho_s(T_{BKT}) = 2T_{BKT}/\pi$.

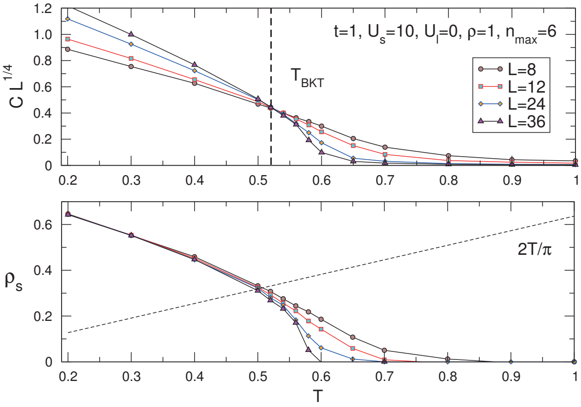


Figure 5: (Color online) (Up) Finite size scaling of the condensate fraction C, Eq. (15), and (down) of the superfluid density ρ_s, Eq. (17), as a function of the temperature T for the 2D standard Bose-Hubbard model ($U_l = 0$).

4 MCTDHX method: many-body tunneling dynamics of interacting bosons in a double well

This section reports on our numerical activities concerning quantum many-body dynamics of interacting bosons trapped in a double-well potential. This model has been extensively studied both experimentally and theoretically in the context of the Josephson oscillations in quantum gases experiments (Albiez et al., 2005). We have particularly focused on dynamical scenarios where the trap evolves in time from a single well to a double well by raising a central gaussian barrier with a ramping time T_{ramp}.

This system is modeled by N interacting spinless bosons of mass m trapped in a one-dimensional double well for which the many-body Hamiltonian reads

$$\hat{\mathcal{H}} \quad = \quad \sum_{i}^{N} \left(-\frac{1}{2} \frac{d^2}{dx_i^2} + V(x_i, t) \right) + \frac{\lambda}{2} \sum_{i \neq j} \delta(x_i - x_j) \; , \qquad (18)$$

with units $m = \hbar = 1$, and where the double-well potential

$$V(x_i, t) = \frac{x_i^2}{2} + A(t)e^{-x_i^2/2} \tag{19}$$

results from the combination of both a harmonic potential superimposed by a central gaussian barrier with time dependent amplitude $A(t)$ of the form

$$A(t) = A_{\max} \times \begin{cases} t/T_{\mathrm{r}amp}, & t < T_{\mathrm{r}amp}, \\ 1, & t \geq T_{\mathrm{r}amp}. \end{cases} \tag{20}$$

In Eq. (19), x_i denotes the coordinate of the ith particle. The repulsive inter-particle interaction strength, $\lambda > 0$, is determined by the s-wave scattering length a_s and the transverse confinement $\omega_\perp$ (Olshanii, 1998). The spectral structures and many-body tunneling dynamics of Hamiltonian Eq. (18) have been extensively studied in our group by Frank Schäfer in collaboration with Miguel Bastarrachea-Magnani (MSc Thesis, 2018). To this end, we have used many numerical methods. In this paper, we restrict our attention to the Multiconfigurational Time-dependent Hartree method for bosons (MCTDHX).

MCTDHX allows for the investigation of interacting particles in many scenarios, e. g. interacting bosons or fermions in optical lattices (Lode and Bruder, 2016) or bosons in double-well potentials (Zöllner et al., 2008; Zöllner et al., 2006; Streltsov et al., 2007). In our context, this method is useful for the investigation of N interacting bosons in a time-dependent double-well potential. Nevertheless, this method is not efficient for the calculation of the entire energy spectrum. In the following, we outline the basic steps towards the MCTDHX equations of motion, see Ref. (Alon et al., 2008) for extensive details regarding the method. The aim is to solve the time-dependent Schrödinger equation

$$i\frac{\partial}{\partial t}|\Psi\rangle = \hat{\mathcal{H}}|\Psi\rangle \ , \tag{21}$$

with many-body Hamiltonian $\hat{\mathcal{H}}$ defined by Eq. (18). To do so, we first formulate a general multiconfigurational ansatz for the wave function based on truncating the field operator

$$\Psi(x,t) = \sum_{k=1}^{\infty} a_k(t)\phi_k(x,t) \tag{22}$$

from an infinite to a finite sum of M operators, i. e.,

$$\Psi(x,t) = \sum_{k=1}^{M} a_k(t)\phi_k(x,t)\ . \tag{23}$$

Under this assumption, the bosonic ansatz for the many-body wave function reads

$$|\Psi\rangle = \sum_{\{\vec{n}\}} C_{\vec{n}}(t) \prod_{k=1}^{M} \frac{(a_k^\dagger(t))^{n_k}}{\sqrt{n_k!}} |\mathrm{v}ac.\rangle, \tag{24}$$

where the summation runs over all (symmetrized) basis states of the Hilbert space. The vector $\vec{n} = (n_1, n_2, \ldots, n_M)$ represents the occupations of the orbitals that preserve the total number of particles $n_1 + n_2 + n_3 + \cdots + n_M = N$, M is the number of orbitals $\phi_k(x,t)$, and $|\mathrm{v}ac.\rangle$ is the vacuum. The key idea of MCTDHX is to control the later assumption a posteriori to get a considerably reduction of the computational effort.

Using this ansatz, the time-dependent Schrödinger equation is solved by using the time-dependent variational principle for minimizing the action functional (Kramer et al., 1981)

$$\mathcal{S} = \int \mathrm{d}t \left[\langle\Psi(t)| \left(\hat{\mathcal{H}} - i\frac{\partial}{\partial t}\right) |\Psi(t)\rangle - \sum_{k,j=1}^{M} \mu_{kj}(t) \Big(\langle\phi_k(t)|\phi_j(t)\rangle - \delta_{kj} \Big) \right], \tag{25}$$

where the time-dependent Lagrange multipliers $\mu_{kj}(t)$ enforce the orthonormality of the orbitals. The minimization of the action $\mathcal{S}$ finally leads to the MCTDHX equations of motion, i. e., a coupled set of first-order non-linear differential equations (Alon et al., 2008)

$$\begin{aligned}
i\frac{\partial}{\partial t}C_{\vec{n}}(t) &= \sum_{\vec{m}} \langle \vec{n},t| \left(\hat{\mathcal{H}} - i\frac{\partial}{\partial t}\right) |\vec{m},t\rangle C_{\vec{m}}(t) \\
i\frac{\partial}{\partial t}|\phi_k\rangle &= \mathbf{P}\Bigg[\left(-\frac{1}{2}\frac{d^2}{dx^2} + V(x,t)\right)|\phi_k\rangle \\
&\quad +\lambda \sum_{\alpha\beta\gamma\delta}^{M} \{\rho^{(1)}\}^{-1}_{k\alpha}\rho^{(2)}_{\alpha\beta\gamma\delta}\phi^*_\beta(x,t)\phi_\delta(x,t)|\phi_\gamma\rangle \Bigg],
\end{aligned} \tag{26}$$

where $\mathbf{P} = 1 - \sum_{j=1}^{M}|\phi_j\rangle\langle\phi_j|$ denotes the projection operator, and where $\rho^{(1)}_{k\alpha} = \langle\Psi|a^\dagger_k a_\alpha|\Psi\rangle$ and $\rho^{(2)}_{\alpha\beta\gamma\delta} = \langle\Psi|a^\dagger_\alpha a^\dagger_\beta a_\gamma a_\delta|\Psi\rangle$ are respectively the matrix elements of the reduced single- and two-particle density matrices. The projector $\mathbf{P}$ vanishes *exactly* only in the limit $M \to \infty$, thus Eq. (26) becomes equivalent to the time-dependent Schrödinger equation. On the other side, the MCTDHX method with one orbital, i. e., $M = 1$, is equivalent to the Gross-Pitaevskii mean field where only one coefficient $C_{0,0,..,N,..,0}(t)$ contributes. Therefore, the accuracy of MCTDHX strongly depends on the choice of the number of orbitals M used in the simulations and the convergence of the MCTDHX results can be improved by increasing the number of orbitals M (Lode and Bruder, 2016).

We have used the freely available software implementation (Lode, Tsatsos et al., 2017), where the spatial discretization relies on a discrete variable representation (DVR) combined with a fast Fourier transformation (Beck et al., 2000). In practice, we have used $M = 20$ orbitals, $x_{max} = -x_{\min} = 12$ and $NDVR_x = 512$ grid points. With this choice of parameters, the convergence of the MCTDHX results for two interacting bosons in a harmonic trap, with respect to the exact ones, is found of the order of 10^{-4}–10^{-2}. See Refs. (Lode and Bruder, 2016; Lode, Sakmann et al., 2012; Fasshauer et al., 2016) for more details on the convergence of the method.

As an example, MCTDHX allows for the observation of different dynamical scenarios as a function of the ramping time T_{ramp}. We observe saturation or oscillations of the von Neumann entropy

$$S(t) = -\mathrm{Tr}\left[\rho^{(1)}(t)\ln\rho^{(1)}(t)\right] \ , \tag{27}$$

with $\rho^{(1)}(t)$ the reduced single-particle density matrix. As shown in Fig. 6, this statement is observed from $N = 2$ to $N = 100$ particles.

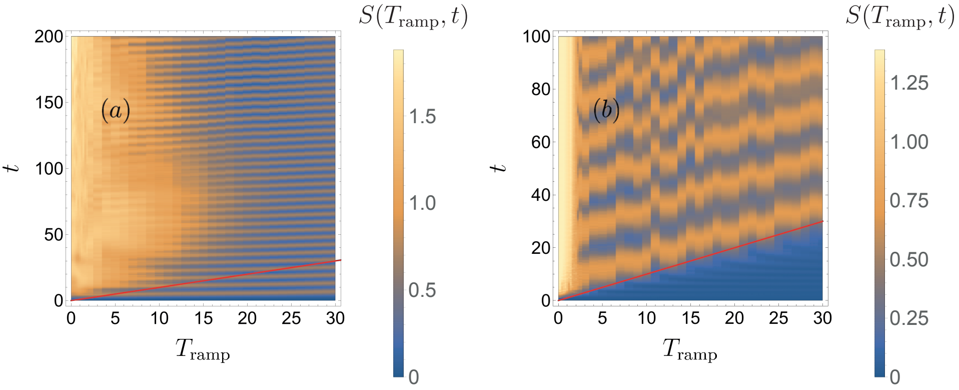


Figure 6: (Color online) Time evolution of the von Neumann entropy in false colors as a function of the ramping time T_{ramp} in the case of (a) 2 particles and (b) 100 particles with interaction $\lambda = 1$. Above the red line, $t = T_{ramp}$, the double-well potential is fixed.

In the saturating regime, the larger the interaction strength, the faster the entropy converges to the equilibrium value. In the oscillating regime, the oscillation has a well-defined frequency determined by the energy gap of the ground state with respect to the first excited state as $\nu(\lambda) = (E_1 - E_0)/\pi$.

5 Conclusion

We have presented a panel of the numerical methods employed in the Quantum Optics and Statistics department at the University of Freiburg's Institute of Physics between September 2016 and June 2018. Particularly, we have discussed three fundamentally different methods in three physical contexts:

(1) We have presented the quantum Monte Carlo method with the stochastic Green function algorithm. The advantages of using this method compared to the mean-field one in the context of quantum magnetism have been discussed. This exact method allowed us to derive the phase diagram of a spin-1 Bose-Hubbard model (Forges de Parny and Rousseau, 2018).

(2) We have described a new method we have developed: the Gutzwiller Monte Carlo (Flottat et al., 2017). This flexible variational method consists of a lattice field theory associated with a Metropolis Monte Carlo method. This method reduces to an extended mean field at zero temperature, but takes into account the thermal fluctuations at finite temperature. We highlighted the use of this method in the context of ultracold gases with infinite-range cavity-mediated interactions, a system under investigation in our group, see J. Mielke's MSc thesis, 2018.

(3) Lastly, we have presented the Multiconfigurational Time-dependent Hartree method for bosons in the context of many-body dynamics of interacting bosons in a double well, see F. Schäfer's MSc thesis, 2018. This method is particularly suited for the investigation of the dynamics of interacting bosons and fermions in many physical contexts.

Each method has its own strengths and limitations which have to be considered depending on the system under investigation. In quantum mechanics, the main constraint remains the diagonalization of the (huge) Hamiltonian matrix, a mathematical operation circumvented by the technics described in this paper. Nowadays, many perspectives are considered: algorithms based on machine learning, tensor networks and the time-DMRG – already used but still restricted at short time – and a direct quantum treatment with quantum computers, e. g., D-wave.

Acknowledgement

We thank V. G. Rousseau and A. U. J. Lode for their help regarding the implementation of the SGF algorithm and MCTDHX method, respectively. We also thank T. Roscilde for his contribution in the development of the GMC method. We acknowledge the support by the state of Baden-Württemberg through bwHPC NEMO cluster and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG. We are also grateful for the financial support from the Alexander von Humboldt-Foundation.

Corresponding Author

Andreas Buchleitner: abu@uni-freiburg.de
Physikalisches Institut, Albert-Ludwigs-Universität Freiburg,
Hermann-Herder-Straße 3, 79104 Freiburg, Germany

References

Albiez, M. et al. (2005). »Direct Observation of Tunneling and Nonlinear Self-Trapping in a Single Bosonic Josephson Junction«. In: *Phys. Rev. Lett.* 95 (1), p. 010402. DOI: 10.1103/PhysRevLett.95.010402.

Alon, O. E., A. I. Streltsov and L. S. Cederbaum (2008). »Multiconfigurational time-dependent Hartree method for bosons: Many-body dynamics of bosonic systems«. In: *Phys. Rev. A* 77 (3), p. 033613. DOI: 10.1103/PhysRevA.77.033613.

Beck, M. H., A. Jäckle, G. A. Worth and H.-D. Meyer (2000). »The multiconfiguration time-dependent Hartree (MCTDH) method: a highly efficient algorithm for propagating wavepackets«. In: *Physics Reports* 324.1, pp. 1–105. ISSN: 0370-1573. DOI: 10.1016/S0370-1573(99)00047-2.

Bloch, I., J. Dalibard and W. Zwerger (2008). »Many-body physics with ultracold gases«. In: *Rev. Mod. Phys.* 80 (3), pp. 885–964. DOI: 10.1103/RevModPhys.80.885.

Dagotto, E. (1994). »Correlated electrons in high-temperature superconductors«. In: *Rev. Mod. Phys.* 66 (3), pp. 763–840. DOI: 10.1103/RevModPhys.66.763.

Dauxois, T., M. Peyrard and S. Ruffo (2005). »The Fermi–Pasta–Ulam 'numerical experiment': history and pedagogical perspectives«. In: *European Journal of Physics* 26.5, S3. URL: http://stacks.iop.org/0143-0807/26/i=5/a=S01.

Fasshauer, E. and A. U. J. Lode (2016). »Multiconfigurational time-dependent Hartree method for fermions: Implementation, exactness, and few-fermion tunneling to open space«. In: *Phys. Rev. A* 93 (3), p. 033635. DOI: 10.1103/PhysRevA.93.033635.

Fehske, H., R. Schneider and A. Weiße (2007). *Computational Many-Particle Physics, Lecture Notes in Physics.* Ed. by Springer. Springer, Berlin, Heidelberg.

Ferrario, M., G. Ciccotti and K. Binder (2006). *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1.* Ed. by Springer. Springer, Berlin, Heidelberg.

Fisher, M. P. A., P. B. Weichman, G. Grinstein and D. S. Fisher (1989). »Boson localization and the superfluid-insulator transition«. In: *Phys. Rev. B* 40 (1), pp. 546–570. DOI: 10.1103/PhysRevB.40.546.

Flottat, T., L. de Forges de Parny, F. Hébert, V. G. Rousseau and G. G. Batrouni (2017). »Phase diagram of bosons in a two-dimensional optical lattice with infinite-range cavity-mediated interactions«. In: *Phys. Rev. B* 95 (14), p. 144501. DOI: 10.1103/PhysRevB.95.144501.

Forges de Parny, L. de, F. Hébert, V. G. Rousseau and G. G. Batrouni (2013). »Interacting spin-1 bosons in a two-dimensional optical lattice«. In: *Phys. Rev. B* 88 (10), p. 104509. DOI: 10.1103/PhysRevB.88.104509.

Forges de Parny, L. de and V. G. Rousseau (2018). »Phase diagrams of antiferromagnetic spin-1 bosons on a square optical lattice with the quadratic Zeeman effect«. In: *Phys. Rev. A* 97 (2), p. 023628. DOI: 10.1103/PhysRevA.97.023628.

Forges de Parny, L. de, H.-Y. Yang and F. Mila (2014). »Anderson Tower of States and Nematic Order of Spin-1 Bosonic Atoms on a 2D Lattice«. In: *Phys. Rev. Lett.* 113 (20), p. 200402. DOI: 10.1103/PhysRevLett.113.200402.

Greiner, M., O. Mandel, T. Esslinger, T. W. Hänsch and I. Bloch (2002). »Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms«. In: *Nature* 415 (15), pp. 39–44. DOI: 10.1038/415039a.

Gross, E. P. (1957). »Unified Theory of Interacting Bosons«. In: *Phys. Rev.* 106 (1), pp. 161–162. DOI: 10.1103/PhysRev.106.161.

Hickey, C. and A. Paramekanti (2014). »Thermal Phase Transitions of Strongly Correlated Bosons with Spin-Orbit Coupling«. In: *Phys. Rev. Lett.* 113 (26), p. 265302. DOI: 10.1103/PhysRevLett.113.265302.

Ho, T.-L. (1998). »Spinor Bose Condensates in Optical Traps«. In: *Phys. Rev. Lett.* 81 (4), pp. 742–745. DOI: 10.1103/PhysRevLett.81.742.

Imambekov, A., M. Lukin and E. Demler (2003). »Spin-exchange interactions of spin-one bosons in optical lattices: Singlet, nematic, and dimerized phases«. In: *Phys. Rev. A* 68 (6), p. 063602. DOI: 10.1103/PhysRevA.68.063602.

— (2004). »Magnetization Plateaus for Spin-One Bosons in Optical Lattices: Stern-Gerlach Experiments with Strongly Correlated Atoms«. In: *Phys. Rev. Lett.* 93 (12), p. 120405. DOI: 10.1103/PhysRevLett.93.120405.

Jacob, D. et al. (2012). »Phase diagram of spin-1 antiferromagnetic Bose-Einstein condensates«. In: *Phys. Rev. A* 86 (6), p. 061601. DOI: 10.1103/PhysRevA.86.061601.

Jaksch, D., C. Bruder, J. I. Cirac, C. W. Gardiner and P. Zoller (1998). »Cold Bosonic Atoms in Optical Lattices«. In: *Phys. Rev. Lett.* 81 (15), pp. 3108–3111. DOI: 10.1103/PhysRevLett.81.3108.

Jiang, J. et al. (2016). »First-order superfluid-to-Mott-insulator phase transitions in spinor condensates«. In: *Phys. Rev. A* 93 (6), p. 063607. DOI: 10.1103/PhysRevA.93.063607.

Kramer, P. and M. Saraceno (1981). »Geometry of the Time-Dependent Variational Principle in Quantum Mechanics«. In: *Lecture Notes in Physics* 140.

Landig, E. et al. (2016). »Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms«. In: *Nature* 532, pp. 476–479. DOI: 10.1038/nature17409.

Lode, A. U. J. and C. Bruder (2016). »Dynamics of Hubbard Hamiltonians with the multiconfigurational time-dependent Hartree method for indistinguishable particles«. In: *Phys. Rev. A* 94 (1), p. 013616. DOI: 10.1103/PhysRevA.94.013616.

Lode, A. U. J., K. Sakmann, O. E. Alon, L. S. Cederbaum and A. I. Streltsov (2012). »Numerically exact quantum dynamics of bosons with time-dependent interactions of harmonic type«. In: *Phys. Rev. A* 86 (6), p. 063606. DOI: 10.1103/PhysRevA.86.063606.

Lode, A. U. J., M. C. Tsatsos and E. Fasshauer (2017). *MCTDH-X: The time-dependent multiconfigurational Hartree for indistinguishable particles software.* URL: http://ultracold.org.

Loh, E. Y. et al. (1990). »Sign problem in the numerical simulation of many-electron systems«. In: *Phys. Rev. B* 41 (13), pp. 9301–9307. DOI: 10.1103/PhysRevB.41.9301.

Mahmud, K. W. and E. Tiesinga (2013). »Dynamics of spin-1 bosons in an optical lattice: Spin mixing, quantum-phase-revival spectroscopy, and effective three-body interactions«. In: *Phys. Rev. A* 88 (2), p. 023602. DOI: 10.1103/PhysRevA.88.023602.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller (1953). »Equation of State Calculations by Fast Computing Machines«. In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092. DOI: 10.1063/1.1699114.

Nature Physics (2013). »The Hubbard model at half a century«. In: *Nature Physics* 9, p. 523. DOI: 10.1038/nphys2759.

Olshanii, M. (1998). »Atomic Scattering in the Presence of an External Confinement and a Gas of Impenetrable Bosons«. In: *Phys. Rev. Lett.* 81 (5), pp. 938–941. DOI: 10.1103/PhysRevLett.81.938.

Penrose, O. and L. Onsager (1956). »Bose-Einstein Condensation and Liquid Helium«. In: *Phys. Rev.* 104 (3), pp. 576–584. DOI: 10.1103/PhysRev.104.576.

Román, O. (2014). »A practical introduction to tensor networks: Matrix product states and projected entangled pair states«. In: *Annals of Physics* 349, pp. 117–158. ISSN: 0003-4916. DOI: 10.1016/j.aop.2014.06.013.

Rousseau, V. G. (2008). »Directed update for the stochastic Green function algorithm«. In: *Phys. Rev. E* 78 (5), p. 056707. DOI: 10.1103/PhysRevE.78.056707.

Sandvik, A. W. (2010). »Computational Studies of Quantum Spin Systems«. In: *AIP Conf. Proc.* 1297, p. 135.

Scalapino, D. J. (2006). »Numerical Studies of the 2D Hubbard Model«. In: *Handbook of High Temperature Superconductivity*. arXiv: `cond-mat/0610710 [cond-mat.str-el]`.

Streltsov, A. I., O. E. Alon and L. S. Cederbaum (2007). »Role of Excited States in the Splitting of a Trapped Interacting Bose-Einstein Condensate by a Time-Dependent Barrier«. In: *Phys. Rev. Lett.* 99 (3), p. 030402. DOI: `10.1103/PhysRevLett.99.030402`.

Vengalattore, M., J. Guzman, S. R. Leslie, F. Serwane and D. M. Stamper-Kurn (2010). »Periodic spin textures in a degenerate $F = 1$ ^{87}Rb spinor Bose gas«. In: *Phys. Rev. A* 81 (5), p. 053612. DOI: `10.1103/PhysRevA.81.053612`.

Vengalattore, M., S. R. Leslie, J. Guzman and D. M. Stamper-Kurn (2008). »Spontaneously Modulated Spin Textures in a Dipolar Spinor Bose-Einstein Condensate«. In: *Phys. Rev. Lett.* 100 (17), p. 170403. DOI: `10.1103/PhysRevLett.100.170403`.

Zibold, T. et al. (2016). »Spin-nematic order in antiferromagnetic spinor condensates«. In: *Phys. Rev. A* 93 (2), p. 023614. DOI: `10.1103/PhysRevA.93.023614`.

Zöllner, S., H.-D. Meyer and P. Schmelcher (2006). »Ultracold few-boson systems in a double-well trap«. In: *Phys. Rev. A* 74 (5), p. 053612. DOI: `10.1103/PhysRevA.74.053612`.

— (2008). »Few-Boson Dynamics in Double Wells: From Single-Atom to Correlated Pair Tunneling«. In: *Phys. Rev. Lett.* 100 (4), p. 040401. DOI: `10.1103/PhysRevLett.100.040401`.

III Administrative and Technical Contributions

Dynamic Resource Extension for Data Intensive Computing with Specialized Software Environments on HPC Systems

Christoph Heidecker[*] Matthias J. Schnepf[*] R. Florian von Cube[†]
Manuel Giffels[*] Martin Sauter[*] Günter Quast[*]

[*]Institute for Experimental Particle Physics, Karlsruhe Institute of Technology, Germany
[†]Physics Institute 3A, RWTH Aachen University, Germany

Modern High Energy Physics (HEP) requires large-scale processing of extensive amounts of scientific data. The needed computing resources are currently provided statically by HEP specific computing centers. To increase the number of available resources, for example to cover peak loads, the HEP computing development team at KIT concentrates on the dynamic integration of additional computing resources into the HEP infrastructure. Therefore, we developed ROCED, a tool to dynamically request and integrate computing resources including resources at HPC centers and commercial cloud providers. Since these resources usually do not support HEP software natively, we rely on virtualization and container technologies, which allows us to run HEP workflows on these so called opportunistic resources. Additionally, we study the efficient processing of huge amounts of data on a distributed infrastructure, where the data is usually stored at HEP specific data centers and is accessed remotely over WAN. To optimize the overall data throughput and to increase the CPU efficiency, we are currently developing an automated caching system for frequently used data that is transparently integrated into the distributed HEP computing infrastructure.

DOI: 10.15496/publikation-29051

1 Introduction

Modern High Energy Physics (HEP) searches for the fundamental building blocks of matter by looking for new particles and rare processes at energies mankind has never reached before. This research requires both giant experimental setups and large scaled processing of huge amounts of data. While simulations are required to predict the behavior of characteristic quantities, data analyses scan for noticeable differences between measured data and theory prediction. Both require a huge amount of computing resources, which are currently deployed at HEP specific computing centers.

Since the demand for computing resources is heavily increasing in HEP, improvements of workflows and new computing concepts and resources are required. Whereas static computing resources that are dedicated to HEP serve the base load, the HEP computing development team at KIT concentrates on the dynamic integration of additional computing resources. This allows for example the covering peak loads caused by journal or conference deadlines. Furthermore, this allows the sharing of computing resources with other scientific communities to be guaranteed.

This makes the bwForCluster NEMO (Wiebelt et al., 2017), whose resources are shared among different scientific communities, ideally suited for our developments. At the bwForCluster NEMO, requested resources are allocated following a fair share policy and released again once there is no demand for additional HEP resources, so that they can be used by other communities. In order to provide HEP users easy access to these additional computing resources, they are integrated into one common batch system, offering one single point of entry to all available resources for HEP. For this, the Institute of Experimental Particle Physics (ETP) at the Karlsruhe Institute of Technology (KIT) developed a tool taking care of the automatic and dynamic integration of these so-called opportunistic computing resources.

In contrast to dedicated HEP computing resources, opportunistic resources such as the NEMO HPC cluster neither provide the HEP specific software environment nor the HEP infrastructure that is specialized for data-intensive processing. Hence, we need to adapt these resources to fully support HEP workflows. The deployment of a specialized software environment is enabled by the possibility to virtualize or containerize workflows at the NEMO HPC center. In contrast to traditional virtualization, the container technology as a more lightweight approach encapsulates processes within an own environment. Deploying a fully configured software environment using those technologies enables us to fulfill the HEP requirements. The KIT group is currently exploring both technologies in order to increase the amount of available non-HEP dedicated computing resources that can be utilized for HEP workflows.

HPC clusters are designed for CPU-intensive workflows with multiprocess interaction, data transfers, as well as CPU utilization suffer, when thousands of independent processes which run in parallel access data stored at remote HEP Grid storages. Therefore, we investigate how data-intensive processing can be enabled on opportunistic resources. In the caching concept developed at ETP, an intermediate cache layer between the worker nodes and the remote storage systems deploys fast access to repeatedly accessed data. Since most HEP workflows repeatedly process the same data, this concept reduces the pressure on the network and increases the data throughput and thus the CPU efficiency.

In the following, we present the concepts of dynamic integration of HPC computing resources, provisioning of software environments and caching infrastructure developed at KIT.

2 Resource Allocation and Integration

There are different kinds of providers of computing resources, such as HPC centers and commercial cloud providers. Depending on the provider, the computing resources have to be requested via common batch systems or via specialized APIs. At the bwForCluster NEMO, we request resources in the form of virtual machines via their MOAB[1] batch system in combination with an OpenStack installation (Meier et al., 2016).

After resource allocation, those resources are integrated into our overlay batch system, in our case HTCondor[2], which manages all HEP user jobs. The usage of an overlay batch system provides a single point of entry for the users and thus simplifies the usage of multiple resources. Due to the pull mechanism of HTCondor, the integration into the common HEP resource pool is done by simply starting an HTCondor client process on the virtualized worker nodes.

[1] Moab Cluster Suite, Adaptive Computing, Inc. URL: https://www.adaptivecomputing.com/

[2] HTCondor, University of Wisconsin-Madison, https://research.cs.wisc.edu/htcondor/

For the dynamic allocation and integration of resources, the HEP community at KIT developed the resource manager Responsive On-demand Cloud Enabled Deployment (ROCED)[3]. ROCED periodically monitors the job queue and the available resources in our overlay batch system. ROCED recognizes a demand for additional resources and automatically requests those at a resource provider. The design of ROCED using adapters allows for the utilization of a wide range of resources provided by different resource providers. When the allocated resources are not used anymore, the overlay batch system client stops after a certain period and releases those resources. This allows for dynamically adjusting the amount of allocated resources to the current demand for resources.

At the time of submission of this paper, ROCED has been used in production for integrating resources like the bwForCluster NEMO into our HTCondor batch system for over two years already. Figure 1 displays the automatic allocation of resources adjusted to the current demand for a period of six days. The number of dynamically allocated CPUs is compared to the number of jobs that are queued or running within the overlay batch system. It gives an example of the stable behavior of ROCED and the high utilization of NEMO resources by the KIT HEP community.

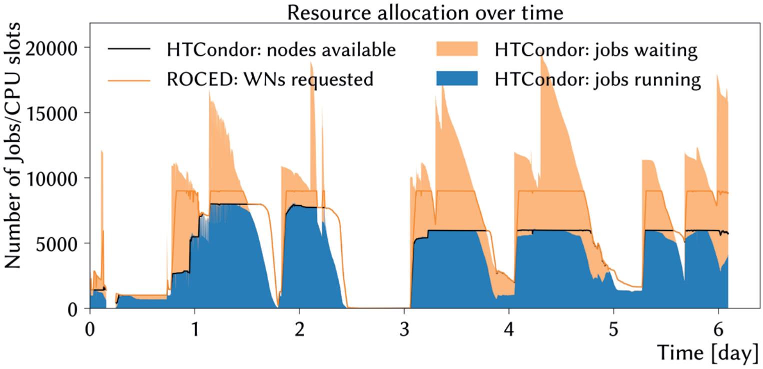


Figure 1: The demand for additional computing resources and their utilization at the NEMO HPC center is shown in units of CPU cores for an exemplary period of 6 days. The orange line shows the number of requested CPU cores, the black line shows the number of cores integrated into the overlay batch system. The blue area shows the number of the used CPU cores, whereas the stacked orange area shows the demand for more cores based on the overlay batch system's queue.

[3] ROCED, ROCED project, URL: https://github.com/roced-scheduler/ROCED

Currently, we are able to allocate up to 6000 CPU cores limited by fair share, which can be used for processing of HEP analyses. During testing phase, we were able to allocate up to 8000 CPU cores as shown in Figure 1. One of the analyses that profits heavily from the integration of NEMO resources is the Higgs analysis relying on the production of $\mu \rightarrow \tau$ embedded events (Bechtel et al., 2019), presented within these proceedings.

A more detailed view of the dynamic integration and provisioning of software environments is given in (Heidecker et al., 2018).

3 Provisioning of Software Environments

Scientific communities such as HEP have specific requirements regarding the environment to provide a verified software stack. This verified software environment ensures that the scientific results of the software can be relied on. Most analyses in the HEP community currently require Scientific Linux 6. However, over time, more and more analyses in HEP are switching to CERN CentOS 7. Both Scientific Linux 6 and CERN CentOS 7 are derivates of RedHat Enterprise Linux 6/7 with a specific set of installed tools. However, those restrictions to specific software environments usually limit the amount of resources that can be utilized. Since most opportunistic resources do not provide the required software environment natively, virtualization and container technologies can be utilized for the provisioning of the HEP software environment. In this chapter, we give an overview of the utilized technologies as well as the advantages and disadvantages of some implementations we tested.

Virtualization is probably the most common technology of provisioning dedicated operating systems and environments. A complete operating system runs on a virtualized hardware, which is called a virtual machine. The host system itself decides what resources are shared with the virtual machine. Since the virtual machine is completely isolated from the host system, the user inside the virtual machine can be given all permissions. This enables the provision of a fully configured software environment inside the virtual machine, isolated from the host system and other users. In this case the HTCondor client process runs inside the virtual machine, which provides the HEP software environment as well. However, the virtualization of resources leads to administrative overhead and a few percent performance degradation.

A more lightweight solution to providing a software environment is to use container technologies. In contrast to virtualization, a container is a process that runs on the host system in an isolated namespace. This isolates the processes inside the container from other process on the host. Also in this case, the host controls the resources the container can access. Additionally to a certain memory and the number of CPU cores, this can also include direct access to a file system of the host.

Container technologies lead to less overhead than virtual machines, resulting in a smaller performance degradation. One implementation of the container technology is DOCKER[4]. DOCKER provides many additional features such as network monitoring of the isolated processes. Additionally, HTCONDOR offers built-in support for DOCKER, so that the batch system client is able to directly start jobs inside of DOCKER containers. On most of the HEP specific computing resources at KIT, we use this method to provide the required software environment. This allows us to use a modern operating system on bare metal and at the same time provide the HEP software environment for batch jobs. Furthermore, it is possible to provide different software environments for different jobs, which makes updates or changes more flexible.

Due to security concerns, DOCKER is usually not installed on HPC clusters. However, as the processes inside the virtual machine are completely isolated from the host system, it is possible to use DOCKER inside a virtual machine. On the bwForCluster NEMO, we use this setup in production, which allows us to use all additional features of DOCKER. This enables for example the monitoring of network usage as shown in Figure 2 for a typical data-intensive HEP job.

In the future, we would like to use this additional information for advanced job scheduling, taking into account available and utilized network bandwidth.

An alternative container technology is SINGULARITY (Kurtzer et al., 2017), which is explicitly developed for the usage on shared computing resources like HPC centers. It is designed to run completely in user space, which resolves security concerns, but results in a limited feature set compared to DOCKER. For example, the above-mentioned network monitoring is not possible using SINGULARITY containers. However, it enables direct access to the hosts file system like the common parallel filesystem provided at HPC centers, where jobs can profit from high bandwidth to

[4] Docker, Docker, Inc. version 17.05.0-ce. URL: https://www.docker.com

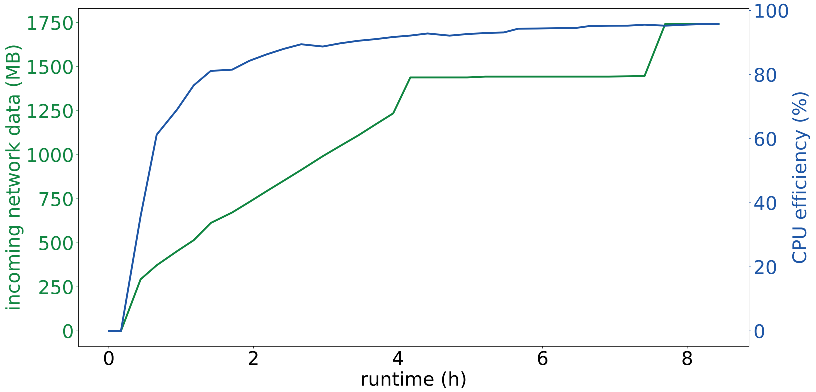


Figure 2: CPU efficiency (blue) and accumulated incoming network traffic (green) for a job of a typical HEP analysis workflow. The values were updated every 15 min. After the job's initialization, the job reads data from remote storage and starts processing. This specific job read 1742.2 MB, took 8.5 h and had an 95.7 % CPU efficiency on average.

data that is locally stored at the HPC center. This enables future optimizations of the data throughput for data-intensive HEP workflows.

Since Docker provides both the full feature set and the lowest overhead, we prefer setups using a pure Docker installation. Shared resources usually do not allow this, so we can either use virtual machines with integrated Docker support or singularity containers, depending on the resource provider. Here, one has to weigh the higher overhead caused by virtualization compared to containerization against the additional features provided by the combined setup. We prefer the combined setup using Docker containers within virtual machines, since we want to use the additional monitoring information provided by Docker to enhance our job to resource scheduling.

4 Optimization of data throughput

Workflows of the HEP community usually can be divided into two different types. Whereas simulations mainly require a lot of CPU hours, data analysis workflows require large datasets to be transferred to the worker nodes. The former benefit directly from the additional resources, but the latter require optimized infrastructures and adjusted hardware configurations.

Since opportunistic resources and especially HPC centers are not explicitly designed to support such data-intensive workflows, the HEP community at KIT is studying how to enable efficient processing of data. HPC centers as well as other opportunistic resources, usually do not provide sufficient long-term storage capacity to hold the huge amount of data produced by current HEP experiments. Hence, data analysis workflows transfer data from dedicated Grid storage systems to the HPC worker nodes for processing. The shared and limited network infrastructure between Grid storage and an HPC center limits the data transfers and can cause inefficient CPU utilization.

Here, we concentrate on a caching concept that stores parts of the accessed data on HPC storage volumes. This directly speeds up HEP analyses that repeatedly access the same data by exploiting the available high-performance parallel files systems available at the HPC center. Furthermore, it reduces the load on the shared external network connection and thus improves data throughput.

For the transparent integration of the caching approach into current HEP infrastructure, we concentrate on HEP specific data transfer protocols and the workflow management infrastructure HTCondor. We decided to concentrate on the XRootD protocol (Dorigo et al., 2005), which was developed by the HEP community for world-wide access to a huge amount of data. It supports a hierarchical infrastructure to manage distributed storage systems and already provides basic caching functionalities. For the intermediate caching infrastructure, caching proxies are installed, which allow transparent access to remote storage capacities while caching frequently used data on storage volumes at the data center. Repeated access to already cached data is redirected to the copy at the data center instead of accessing the Grid storage element. Therefore, we deployed XRootD cache proxies at the NEMO HPC center that utilize the distributed high-performance storage capacities to optimize data throughput for the processing of HEP workflows. The maximum bandwidth that the proxy can provide with this setup depends on both the performance of the storage system and the available bandwidth of the host running XRootD services. We can avoid this limitation, since the jobs can also directly read cached files from the parallel file system. Unfortunately, it is not allowed to mount the storage volumes inside a virtual machine, which we currently use to provide our required software environment. However, using the container technology Singularity allows for the use of the storage volumes without any security concerns.

In this case, the access request still goes through the XRootD hierarchy to allow access to remote data as well as on-the-fly caching of data while streaming it to the worker. However, access to already cached files is redirected to the local mount of the parallel file system profiting from the fast Omni-Path interconnects. When using Singularity containers, it is no longer as easy to determine the network traffic of the individual jobs as in the current setup. Hence, we currently prefer the first setup, which allows us to use Docker container within virtual machines providing monitoring data of the network utilization of jobs. However, on other HPC centers such as the ForHLR2 (Barthel et al., 2017) at Karlsruhe, which has no virtualization, the usage of Singularity as well as the local redirection for cached data is a useful alternative to the NEMO setup.

As part of a large-scale computing infrastructure for future HEP experiments, we want to install caches at each computing resource connected to the HEP infrastructure. In this case, the challenge will be improving the data throughput of the processed workflows and efficiently utilize the limited cache volumes. Therefore, the HEP community at the KIT studies the advantages of coordinated caching. Here, multiple caches are coordinated to reach efficient utilization of limited cache volumes. Furthermore, data-intensive jobs are matched to the most suitable computing resource that already has cached requested files. This enables an optimized utilization of the computing resources and thus shorter processing times of the processed workflows.

In order to introduce the concept of data locality into the HTCondor batch system, an extension is currently being developed. As shown in Figure 3, it fetches location information from the XRootD service and injects it into the job-to-resource matching. In addition, this new HTCondor extension also coordinates the cache location of newly arrived data for further processing.

First studies on a prototype infrastructure showed a big improvement of the data throughput and thus a better utilized CPU. The highest CPU utilization was achieved by a combined utilization of cache access and network transfers. The advantages of such an infrastructure optimization for opportunistic resources will be studied in detail at NEMO. A successful utilization of the caching concept at the NEMO HPC center will enable future HEP experiments to efficiently utilize different kinds of opportunistic resources for data processing.

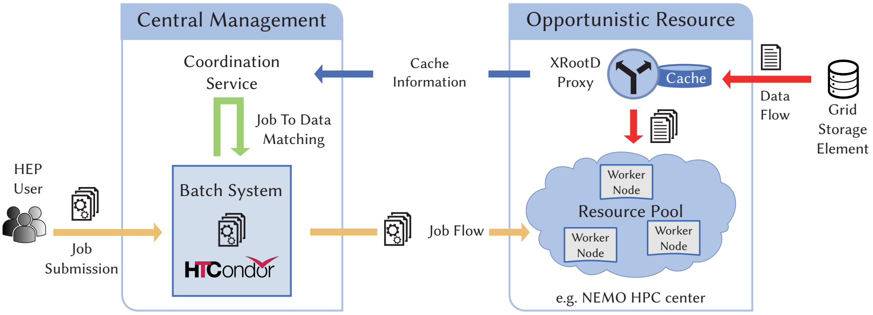


Figure 3: An XRootD proxy caches data transfered from Grid storage element on the fly. An extension of the HTCondor batch system injects the location of cached data into the job matching. Hence, jobs are matched to the most suitable computing resource that provides the best cache hit rate.

5 Summary

Opportunistic resources such as the shared HPC center NEMO provide a huge amount of computing resources.

Using virtualization and container technology, we are able to provide our HEP software environment on different resources such as HPC centers or commercial cloud providers. The preferred setup is to start containers inside virtual machines, which enables network monitoring for each of our batch jobs.

The HEP community at KIT developed ROCED, a tool to dynamically request resources and integrate them into common HEP computing infrastructure. At the bwForCluster NEMO, this setup has been in production for over two years, which allows us to efficiently utilize shared resources and thus cover peak loads.

Over time, we recognized CPU inefficiencies of data intensive jobs due to shared and limited network capacities. In order to avoid these inefficiencies, a concept of coordinated data caches utilizing the locally available parallel file system has been developed, which allows us to place frequently used data temporarily local to the workers. The concept developed in this manner is also applicable to other opportunistic resources as well as computing clusters with locally available storage attached.

With the presented techniques we are able to process all kind of batch jobs of HEP workflows on may different opportunistic resources in a dynamic and efficient way.

Acknowledgment

The authors acknowledge the support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG.

Corresponding Author

Christoph Heidecker: christoph.heidecker@kit.edu
Matthias Schnepf: matthias.schnepf@kit.edu
Institute of Experimental Particle Physics (ETP),
Karlsruhe Institute of Technology, Germany

ORCID

Christoph Heidecker iD https://orcid.org/0000-0002-8361-4520
Matthias Schnepf iD https://orcid.org/0000-0003-0623-0184
Manuel Giffels iD https://orcid.org/0000-0003-0193-3032

References

Barthel, R. and S. Raffeiner (2017). »ForHLR: a New Tier-2 High-Performance Computing System for Research. October 12th 2016, Heidelberg«. eng. In: *Proceedings of the 3rd bwHPC-Symposium*. Heidelberg: Universitätsbibliothek Heidelberg, pp. 73–75. ISBN: 978-3-946531-70-8. DOI: 10.11588/heibooks.308.418.

Bechtel, J., S. Brommer, A. Gottmann, G. Quast and R. Wolf (2019). »Performance of the bwHPC cluster in the production of $\mu \to \tau$ embedded events used for the prediction of background for $H \to \tau\tau$ analyses«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg*. Freiburg, September 2018. 5th bwHPC Symposium. TLP, Tübingen, pp. 61–73. DOI: 10.15496/publikation-29043.

Dorigo, A., P. Elmer, F. Furano and A. Hanushevsky (2005). »XROOTD/TXNetFile: A Highly Scalable Architecture for Data Access in the ROOT Environment«. In: *Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics*. TELE-INFO'05. Prague, Czech Republic: World Scientific, Engineering Academy and Society (WSEAS), 46:1–46:6. ISBN: 960-8457-11-4. URL: http://dl.acm.org/citation.cfm?id=1391157.1391203.

Heidecker, C., M. Giffels, G. Quast, K. Rabbertz and M. Schnepf (2018). »High precision predictions for particle collisions at the Large Hadron Collider«. eng. In: *Proceedings of the 4th bwHPC Symposium.* Tübingen: Universität Tübingen, pp. 28–31. DOI: `10.15496/publikation-25195`.

Kurtzer, G. M., V. Sochat and M. W. Bauer (2017). »Singularity: Scientific containers for mobility of compute«. In: *PLOS ONE* 12.5, pp. 1–20. DOI: `10.1371/journal.pone.0177459`.

Meier, K. et al. (2016). »Dynamic provisioning of a HEP computing infrastructure on a shared hybrid HPC system«. In: *Journal of Physics: Conference Series* 762.1, p. 012012. DOI: `10.1088/1742-6596/762/1/012012`.

Wiebelt, B., K. Meier, M. Janczyk and D. von Suchodoletz (2017). »Flexible HPC: bwForCluster NEMO. October 12th 2016, Heidelberg«. eng. In: *Proceedings of the 3rd bwHPC-Symposium.* Heidelberg: Universitätsbibliothek Heidelberg, pp. 128–130. ISBN: 978-3-946531-70-8. DOI: `10.11588/heibooks.308.418`.

Unified Container Environments for Scientific Cluster Scenarios

Benjamin Schanzel Mark Leznik Simon Volpert Jörg Domaschka
Stefan Wesner

Institute of Information Resource Management, Ulm University, Germany

Providing runtime dependencies for computational workflows in shared environments, like HPC clusters, requires appropriate management efforts from users and administrators. Users of a cluster define the software stack required for a workflow to execute successfully, while administrators maintain the mechanisms to offer libraries and applications in different versions and combinations for the users to have maximum flexibility. The Environment Modules system is the tool of choice on bwForCluster BinAC for this purpose. In this paper, we present a solution to execute a workflow which relies on a software stack not available via Environment Modules on BinAC. The paper describes the usage of a containerized, user-defined software stack for this particular problem using the Singularity and Docker container platforms. Additionally, we present a solution for the reproducible provisioning of identical software stacks across HPC and non-HPC environments. The approach uses a Docker image as the basis for a Singularity container. This allows users to define arbitrary software stacks giving them the ability to execute their workflows across different environments, from local workstations to HPC clusters. This approach provides identical versions of software and libraries across all environments.

1 Introduction

The reproducibility of experimental findings and results is an essential requirement of many scientific processes. Papers and reports presenting results and conclusions

DOI: 10.15496/publikation-29052

obtained from experiments therefore not only describe the final results, but also the experiment methodology and environment. This holds true for computational sciences as well (Sandve et al., 2013). Environmental influences on computational workflows – i. e. different operating systems (OS's) and various versions of software libraries and applications – interfere with the reproducibility of scientific results computed within such an environment. This implies that for a computational scientific workflow to be reproducible, the runtime environment, in which results are computed in, must be controlled, described, and provided alongside the results (Boettiger, 2015; Bartusch et al., 2018).

A number of solutions are available for the definition and reproducible provisioning of computational environments. In the industry of enterprise applications and particularly in cloud computing environments, hardware level virtualization is a widely adopted method for providing complete environments, called Virtual Machines (VM's) (Gartner Inc., 2016). The primary goals of this virtualization mechanism are the isolation of virtualized environments from the host system, as well as increasing the utilization of costly resources by hosting multiple VM's on one physical machine. Despite these objectives, VM's are also useful for providing environments in an automated and reproducible manner. VM images contain a predefined stack of software – from the OS, software libraries, applications, to user provided code. These images can be deployed and run on any compatible system. Hardware level virtualization thus is an effective approach for isolating a scientific computational workflow from external environmental influences. It might, however, restrict direct access to accelerated hardware (e. g. GPUs), and could therefore introduce significant performance regressions. Consequently, this contradicts the use in High-Performance Computing (HPC) environments, where users want their code to be executed as close as possible to real hardware for a maximum of performance.

Another, more recent approach to describing and providing complete software stacks for computational workflows is containerization. Here again, images of complete environments (i. e. the OS, libraries, etc.) are predefined and can be deployed and executed in a reproducible and platform-independent manner. Processes within a container are executed directly on the host system using lightweight virtualization features of the host OS. This does not involve any virtualization of hardware and is therefore more suitable for HPC use cases than hardware level virtualization.

One of the most widely adopted solutions for OS-level virtualization is Docker (Portworx Inc., 2017). Its main use cases are micro-service architectures and web-based cloud applications. In such environments, Docker containers are generally executed inside of VM's.

For HPC use cases, executing containerized software stacks would require running the containers directly on the host OS. Docker, however, is not perfectly suitable for this use case. The Docker host daemon is a root-owned process on the host OS, which means that containers are executed with the highest privileges available on the host system. It would therefore be theoretically possible for a user to run arbitrary code on the host OS with root privileges, effectively rendering the system compromised. This implies security concerns, hindering the adoption of Docker in HPC environments (Kurtzer et al., 2017). Other container technologies, such as Singularity, not only address these security downsides of Docker but also explicitly focus on the use of containers for scientific workflows in HPC environments (Kurtzer et al., 2017).

In this paper we present an exemplary use case of a machine learning (ML) workflow on the bwForCluster BinAC (Krüger et al., 2017) which is executed within a Singularity container. We argue why providing a user-defined software stack in the form of a container is a well-suited method for this use case in particular and for computational workflows in general. Furthermore, we present a two-layered solution with a combination of Docker and Singularity, which enables reproducible and platform independent execution of workflows. Section 2 describes the technical requirements of a specific use case leading to the solution based on a containerized software stack. Section 3 explains how the required software stack has been implemented and how it can be executed, either as a Docker or Singularity container. Section 4 concludes how the use of containers not only solves the specific use case described in this paper, but also helps to improve the reproducibility of computational workflows across different HPC and non-HPC environments in general.

2 Status Quo and Problem Statement

This section describes the use case of an ML workflow, intended to run on bwForCluster BinAC. We explain how the runtime environments for workflows are usually provided on the cluster and why this approach is not sufficient for this exemplary use case.

Providing software for computational jobs submitted by users on bwHPC clusters is commonly accomplished via the Environment Modules system. By this means, it is possible to dynamically modify the applications and software library environment available to the user. Figure 1 visualizes the process of providing software libraries on BinAC. It additionally shows a corresponding process in a non-HPC environment, like a local workstation.

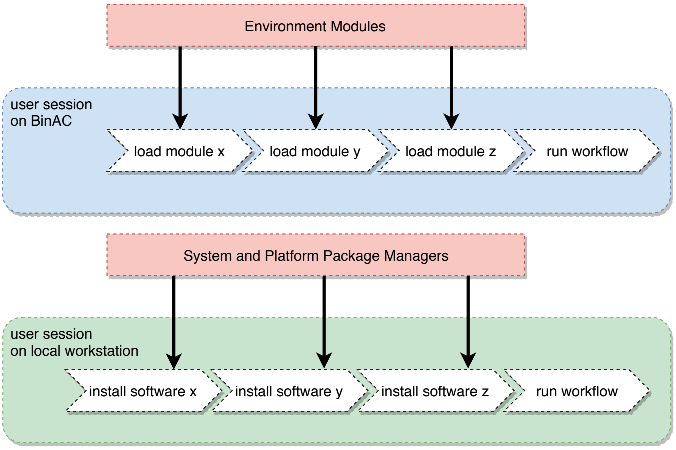


Figure 1: The process of providing software modules required for a computational workflow. On BinAC these modules are loaded via the Environment Modules system. Identical modules might not be available on other HPC clusters. On a local workstation the software modules are installed using the system package manager or platform specific platform managers, e. g. pip for Python libraries. It cannot be guaranteed that the exact same libraries as on the HPC cluster are available via these package managers.

On HPC clusters like BinAC, users integrate module load operations in their job scripts submitted to the scheduling system of the computing cluster. A command to load Keras, TensorFlow and Python in specific versions on the BinAC cluster would look like the following:

```
module load cs/keras/2.1.0-tensorflow-1.4-python-3.5
```

Loading another module, which would provide any of these applications in a different version, is then prevented by the Environment Modules system. This is to prevent conflicts with already loaded modules. Each of the available modules is described in a so-called `modulefile`, which is executed upon loading and contains every step needed to provide an application or software library for the current user. Modules are available system-wide, and thus loadable by all users of this shared environment.

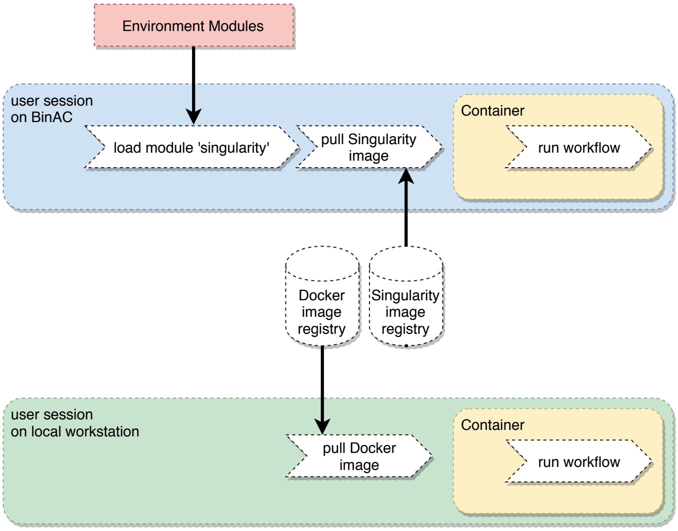


Figure 2: The process of providing a containerized runtime environment for a computational workflow. For HPC environments a Singularity image is pulled from a Singularity image registry. The Singularity system must still be provided on the HPC cluster in any way, e. g. via Environment Modules. In a non-HPC environment Docker might be used as a container platform. Here a Docker image is pulled and executed as a container. The containers on both sides provide an identical application stack, as they both are based on the same `Dockerfile`.

In this paper, we assume an exemplary use case with the goal of training an Artificial Neural Network (ANN) using the Keras library on top of the TensorFlow framework, coupled with Python in their most recent versions. There is no `modulefile` defined on BinAC, which provides this exact combination of the libraries required. These libraries are also not loadable as independent single modules. Under these conditions, we cannot execute our model training code on BinAC in its current setup. There are several solutions to this problem. As a first option, the system administrator could provide a new module with the required software versions. A second solution is to rewrite the Python code for it to be compatible with the software stack that is available now. Also, a local build of these modules in the user's home directory can be performed. These solutions come with manual effort, with the particular downside of an ever-growing number of modules which need to be maintained by an administrator or user. The approach of providing the required software stack via Environment Modules is therefore not well-suited for the described use case. This is particularly true considering the high frequency of TensorFlow releases (TensorFlow, 2018).

Another option for this problem is to let the user define the required software stack in the form of a container and run the computational workflow within it. This would not only reduce manual effort for the system administrators, but, as previously explained, also improve the reproducibility of the computational results across different environments outside the HPC cluster. The process of providing containerized software stacks in both, HPC and non-HPC environments is displayed in Figure 2.

The following sections describe the setup of a container environment we used to circumvent this specific problem faced on the BinAC cluster. Additionally, we describe how the setup allows the execution of such computational workflows not only on HPC clusters, but also in many other container-enabled environments, in a reproducible manner.

3 Operating Singularity Containers on BinAC

This section highlights how we provide the necessary libraries for the previously described use case as a user-defined software stack and how to execute the resulting container in the BinAC HPC cluster environment. We also demonstrate how the approach of using a Docker image as a starting point, rather than Singularity, allows to provide the software stack in non-HPC environments, such as Windows or Mac workstations, where Singularity is not easily available. While Docker does require a virtualization layer on such non-Linux host systems, it is arguably more comfortable to use Docker on these systems. The virtualization layer is directly provided by Docker for Windows and macOS host systems and does not require the user to install any virtualization software separately.

Implementing a scientific computational workflow for HPC environments does not only involve remote interaction with HPC cluster machines, e. g. via SSH, but also writing code on the users local workstation. Users feel certainly more comfortable developing locally on their workstation, having their own editors, development environments, and other tools of choice at hand. This is especially true for debugging purposes. A user's common process might therefore involve local development, debugging, and testing of a computational workflow, eventually submitting it as a job to an HPC cluster. This scenario involves two entirely different software and hardware environments, expected to deliver identical results upon execution. Taking this scenario as a premise, we must extend our previous goal of providing a reproducible software stack for the execution on BinAC by the requirement of serving an identical stack for arbitrary user workstations. Singularity is, however, only available for Linux OS's, requiring the installation of a VM hypervisor on the majority of desktop workstations, as they mostly run on Windows or macOS. Singularity offers the capability of automatically migrating Docker images to its own image format. By this means, a promising solution is to provide a Docker image for workflow execution in non-HPC environments, as it is available for a range of operating systems including Windows and macOS. In a subsequent step, this Docker image can be used as the basis for a Singularity image, which can be executed on BinAC. It is then possible to execute a computational workflow within a defined and reproducible environment, having an identical software stack available across different environments, be it HPC clusters or local workstations.

3.1 Defining and Distributing a Docker Image

As mentioned before, we use Docker images as the basis for building equivalent Singularity images. Docker is used in version 18.03.1 and Singularity in version 2.4.1. The contents of Docker images are described by a so-called `Dockerfile`. This is a text file containing the name of the base image to start from, steps to install additional software on top of that image, and an entry point for the container to start upon execution. Dockerfiles can contain more information to define an image, e. g. directories to mount from the host system or network ports to expose to the host during execution. A comprehensive guide can be found in the online documentation provided by the Docker project (Docker Inc., 2018).

As described in Section 2, we provide a container with certain applications and libraries in specific versions. We start with a publicly available Docker base image, provided by the TensorFlow project. This already includes TensorFlow and Python in the required versions. As this image is based on an Ubuntu Linux base image, we can use Ubuntu's `apt-get` package manager to obtain additional system software. Python libraries, like Keras, can be installed via its own package manager `pip`. We use these two commands to install Keras and additional dependencies on top of the base image, eventually making all of the required software available in a running container. Listing 1 shows a `Dockerfile` providing the software stack described in Section 2. It can be built into an image on any machine with Docker installed by issuing the `build` command. The resulting image can then be executed as a container with the `run` command.

```
FROM tensorflow/tensorflow:1.8.0-gpu-py3
RUN pip install keras==2.1.6
ENTRYPOINT ['/usr/bin/python', 'main.py']
```

Listing 1: `Dockerfile` for the provisioning of Python 3.5, TensorFlow 1.8, and Keras 2.1.

After building the Docker image, we are able to run the computational workflow within the container on a local workstation, under the only premise of having Docker installed. To distribute the image, i. e., making it available to other users, it is necessary to publish it to a container registry. This can either be the official public registry, Docker Hub, or any privately hosted instance. Pre-built images can then be pulled and executed as containers from any remote machine with access to the repository. Listing 2 provides an example of how a `Dockerfile` can be built and executed. To push an image to a registry, it is necessary to authenticate to that registry, i. e. Docker Hub or any privately hosted registry.

```
docker build . -t tensorflow-keras-py3:latest-gpu
docker push schanzel/tensorflow-keras-py3:latest-gpu
docker run schanzel/tensorflow-keras-py3:latest-gpu
```

Listing 2: Shell commands to build and publish a docker image, finally executing the image as a container. The publishing `push` command can be omitted if the image is only needed locally. The period in the `build` command denotes for Docker to search for a `Dockerfile` in the current working directory.

3.2 Building a Singularity Image for HPC Environments

In a following step, we want to run the same container on BinAC, where Docker is not available, but Singularity is. Therefore, it is necessary to build a Singularity image with the identical software stack. Singularity has, as mentioned earlier in this section, the ability to migrate Docker images into its own image format. This conversion is usually done by providing the location and name of a Docker image to Singularity's `build` command. Building an image yields a `.img` file, representing the executable Singularity image in its own format. This file can then be distributed by copying it to a desired location on the file system of the target machine (i. e. a shared file system on BinAC in our case), or by pushing it to a Singularity repository, making it more accessible to other users. Comprehensive documentation about the usage of the Singularity container platform is available in the original publications (Kurtzer et al., 2017; V. V. Sochat et al., 2017; V. Sochat, 2017) as well as in the online user guide (Sylabs Inc., 2018b). Listing 3 provides an example of how to build a Singularity image from the previously published Docker image and executes the resulting image directly from the local file system.

```
singularity build tensorflow-keras-py3.img \
    docker://schanzel/tensorflow-keras-py3:latest-gpu
singularity run --nv tensorflow-keras-py3.img
```

Listing 3: Shell commands to build and run a Singularity container based on the previously built Docker image. The `docker://` prefix on the Docker image URL denotes for Singularity to pull the Docker image from Docker Hub. By invoking the `run` command with the `--nv` flag, Singularity's NVIDIA GPU support is enabled, which is required to leverage the accelerated hardware of BinAC.

Running this container on BinAC requires the Singularity executable to be available on the cluster machines. This is the only prerequisite, and since Singularity is tailored for HPC environments, can be regarded as a given. On the BinAC cluster, Singularity is provided as a loadable module via the Environment Modules system. After the execution of `module load devel/singularity/2.4.1` BinAC is able to pull and run Singularity containers.

It is then possible execute the ML use case, as described in Section 2, in a user-defined software stack on any HPC cluster where Singularity is available. Additionally, this setup ensures that a workflow is executed in an identical software stack, independently from the host environment. This ultimately minimizes environmental influences on the computational results and allows for a better reproducibility as containers are easily distributable via registries or even in a textual form as a `Dockerfile`. It should be noted here, that while Singularity is currently able to directly pull and run Docker containers, this practice should be avoided in this case, due to unpredictable changes on upstream Dockers images, as stated by the Singularity documentation (Sylabs Inc., 2018a).

3.3 A Continuous Delivery Pipeline

To not only improve the reproducibility of computational workflows by the use of containers, we would also like to maintain the reproducible provisioning of containers themselves. The reproducibility of build and deployment steps of containers, as for any other software, can be improved by maximizing the degree of automation in this process and hereby minimizing undocumented manual steps. The use of a Continuous Integration and Continuous Delivery (CI\CD) pipeline is used to automate such build and deploy processes (Humble et al., 2010). This is an aspect which often receives very little attention during research work (Volpert et al., 2018). This section describes how we automate the process of building and providing Docker and Singularity images with such a pipeline.

To continuously build and publish container images, we define a common process which executes every step needed for this purpose. As common in CI\CD, this process is automatically triggered when a relevant change is detected by the pipeline system. This could be a change to a `Dockerfile` being pushed to a version control system (VCS), e. g. Git.

There are two possible starting points for the process as we either want to

(i) build an own Docker image based on a `Dockerfile` or

(ii) use an existing Docker image already provided in a registry.

Case (i) involves building a Docker image from a `Dockerfile` and publishing it to a Docker registry first. In case (ii) there is a Docker image already provided in a registry. Here the process would just use an existing image, e. g. from Docker Hub, and convert this to a Singularity image, omitting the Docker-related build and publish steps. Either case involves building and publishing a Singularity image. The use case described previously matches with case (i) of the process. Case (ii) is suitable for applications that do not need additional libraries on top of an existing Docker base image.

The triggers for this process to be executed continuously also differ between the two cases. In case (i) there is a `Dockerfile` managed by a VCS which can easily trigger the CI\CD process upon change detection. For case (ii), things are slightly more difficult as it requires the detection of changes on an image in a Docker repository which might be owned by some other entity. Docker Hub allows to send push notifications, so-called Webhooks, to arbitrary HTTP endpoints after a new image version was published. This mechanism could be used to trigger the process of building and publishing a Singularity image based on the Docker image that has just received the update. However, such a Webhook must be configured by the owner of the Docker Hub repository. An alternative approach, not relying on the source repositories owner, is to poll the Docker repository for changes on a regular basis. Figure 3 illustrates this process with the described differences between the identified cases.

Having such a pipeline in place, it is possible to automatically and reproducibly build and publish containers for both Docker and Singularity. This further improves the interoperable provisioning of software stacks for computational workflows. A continuous delivery pipeline not only manages containers for both, HPC and non-HPC environments, but also makes sure that the two images are always published in the same version.

4 Conclusions and Future Work

In this paper, we have shown several approaches of deploying containers on local environments as well as clusters for an exemplary machine learning use case. Our approach is easily transferable or extendable and requires little to no knowledge about the underlying software. This solves the problem of running arbitrary software on clusters without administrator privileges, with Singularity being the sole prerequisite provided by the cluster administration.

The proposed solution does not take into account performance implications of containers, yet. Currently, the exact performance drop introduced by the use of containers is being further investigated but initial results indicate low impact for the tested single node jobs (Sweeney et al., 2018). Additionally, the intercommunication of containers with each other was not considered in this paper. A more visible performance impact is expected in this case. ML frameworks like TensorFlow offer

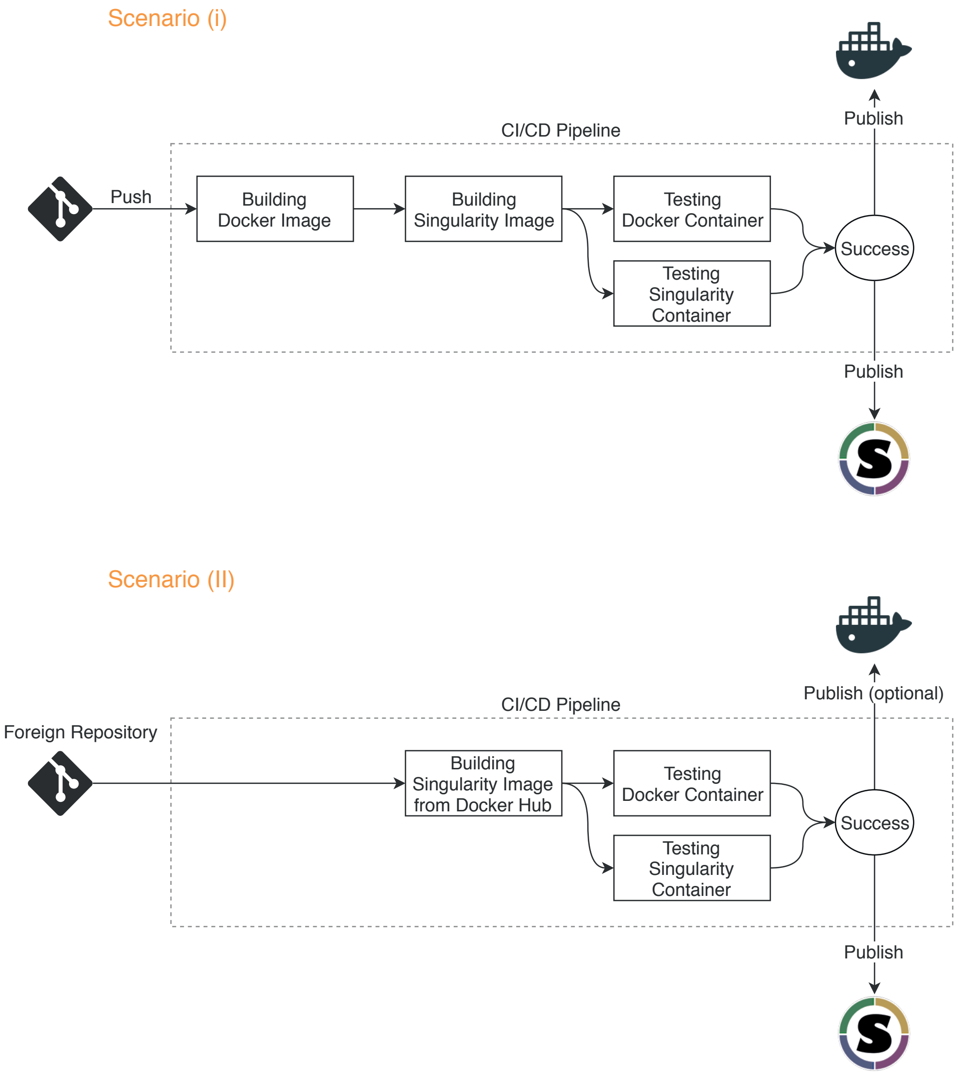


Figure 3: The proposed CI\CD architecture to automate the provisioning of both, Docker and Singularity images. Scenario (i) considers building an own Docker image, while in Scenario (ii) an existing image is pulled from Docker Hub. In either scenario, the Docker image is converted to a Singularity image and published to a registry. Since the image is already being pulled from Docker Hub, the Docker image can be optionally pushed to a private repository.

the possibility of distributed learning. Hereby, a coordinator node subdivides the data onto several nodes, decreasing the required training time. However, there is no clear indication whether the communication overhead introduced by such a setup can be efficiently supported by container solutions. For multi-node jobs, additional tests are needed.

Acknowledgements

The authors acknowledge the support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG.

The research leading to these results has received funding from the EC's Framework Programme HORIZON 2020 under grant agreement number 732667 (RECAP).

Corresponding Author

Mark Leznik: mark.leznik@uni-ulm.de
Institute of Information Resource Management,
Ulm University, Germany

References

Bartusch, F., M. Hanussek and J. Krüger (2018). »Containerization of Galaxy Workflows Increases Reproducibility«. In: *Proceedings of the 4th bwHPC Symposium.* DOI: 10.15496/publikation-25200.

Boettiger, C. (2015). »An Introduction to Docker for Reproducible Research«. In: *SIGOPS Oper. Syst. Rev.* 49.1, pp. 71–79. ISSN: 0163-5980. DOI: 10.1145/2723872.2723882.

Docker Inc. (2018). *Docker Documentation.* URL: https://docs.docker.com/ (visited on 13. 06. 2018).

Gartner Inc. (2016). *Gartner Says Worldwide Server Virtualization Market Is Reaching Its Peak.* URL: https://www.gartner.com/newsroom/id/3315817 (visited on 29. 08. 2018).

Humble, J. and D. Farley (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.* Upper Saddle River, NJ: Addison-Wesley. ISBN: 978-0-321-60191-9.

Krüger, J. et al. (2017). »Bioinformatics and Astrophysics Cluster (BinAC)«. In: *Proceedings of the 3rd bwHPC-Symposium.* DOI: 10.11588/heibooks.308.418.

Kurtzer, G. M., V. Sochat and M. W. Bauer (2017). »Singularity: Scientific Containers for Mobility of Compute«. In: *PLOS ONE* 12.5, e0177459. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0177459.

Portworx Inc. (2017). *2017 Annual Container Adoption Survey: Huge Growth in Containers.* URL: https://portworx.com/2017-container-adoption-survey/ (visited on 29. 08. 2018).

Sandve, G. K., A. Nekrutenko, J. Taylor and E. Hovig (2013). »Ten Simple Rules for Reproducible Computational Research«. In: *PLoS Computational Biology* 9.10. Ed. by P. E. Bourne, e1003285. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003285.

Sochat, V. (2017). »Singularity Registry: Open Source Registry for Singularity Images«. In: *The Journal of Open Source Software* 2.18, p. 426. ISSN: 2475-9066. DOI: 10.21105/joss.00426.

Sochat, V. V., C. J. Prybol and G. M. Kurtzer (2017). »Enhancing Reproducibility in Scientific Computing: Metrics and Registry for Singularity Containers«. In: *PLOS ONE* 12.11, e0188511. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0188511.

Sweeney, K. M. D. and D. Thain (2018). »Efficient Integration of Containers into Scientific Workflows«. In: *Proceedings of the 9th Workshop on Scientific Cloud Computing.* ScienceCloud'18. New York, NY, USA: ACM, 7:1–7:6. ISBN: 978-1-4503-5863-7. DOI: 10.1145/3217880.3217887.

Sylabs Inc. (2018a). *Singularity and Docker — Singularity Container 2.6 Documentation.* URL: https://www.sylabs.io/guides/2.6/user-guide/singularity_and_docker.html#the-build-specification-file-singularity (visited on 30. 08. 2018).

— (2018b). *User Guide — Singularity Container 2.6 Documentation.* URL: https://www.sylabs.io/guides/2.6/user-guide/ (visited on 29. 08. 2018).

TensorFlow (2018). *Computation Using Data Flow Graphs for Scalable Machine Learning: Tensorflow/Tensorflow.* URL: https://github.com/tensorflow/tensorflow (visited on 30. 08. 2018).

Volpert, S., F. Griesinger and J. Domaschka (2018). »Continuous Anything for Distributed Research Projects«. In: *Dependability Engineering.* InTech. DOI: 10.5772/intechopen.72045.

Integration of NEMO into an existing particle physics environment through virtualization

Felix Bührer[*] Anton J. Gamel[*†] Benoît Roland[*]
Benjamin Rottler[*] Markus Schumacher[*] Ulrike Schnoor[*§]

[*]Institute of Physics, University of Freiburg, Freiburg, Germany
[†]Computing Center, University of Freiburg, Freiburg, Germany
[§]Now at CERN, Geneva, Switzerland

With the ever-growing amount of data collected with the experiments at the Large Hadron Collider (LHC) (Evans et al., 2008), the need for computing resources that can handle the analysis of this data is also rapidly increasing. This increase will even be amplified after upgrading to the High Luminosity LHC (Apollinari et al., 2017). High-Performance Computing (HPC) and other cluster computing resources provided by universities can be useful supplements to the resources dedicated to the experiment as part of the Worldwide LHC Computing Grid (WLCG) (Eck et al., 2005) for data analysis and production of simulated event samples. Computing resources in the WLCG are structured in four layers – so-called Tiers. The first layer comprises two Tier-0 computing centres located at CERN in Geneva, Switzerland and at the Wigner Research Centre for Physics in Budapest, Hungary. The second layer consists of thirteen Tier-1 centres, followed by 160 Tier-2 sites, which are typically universities and other scientific institutes. The final layer are Tier-3 sites which are directly used by local users. The University of Freiburg is operating a combined Tier-2/Tier-3, the ATLAS-BFG (Backofen et al., 2006). The shared HPC cluster »NEMO« at the University of Freiburg has been made available to local ATLAS (Aad et al., 2008) users through the provisioning of virtual machines incorporating the ATLAS software

DOI: 10.15496/publikation-29053

environment analogously to the bare-metal system at the Tier-3. In addition to the provisioning of the virtual environment, the on-demand integration of these resources into the Tier-3 scheduler in a dynamic way is described. In order to provide the external NEMO resources to the user in a transparent way, an intermediate layer connecting the two batch systems is put into place. This resource scheduler monitors requirements on the user-facing system and requests resources on the backend-system.

1 Introduction

Compute clusters shared between many users at the same time cannot follow a simple first-in-first-out scheme for deciding, how resources are allocated to the work packages submitted by the users, but require sophisticated scheduling algorithms. These algorithms can take into account a wide array of parameters, both concerning the requirements of the job to be scheduled and ensuring a fair use of the provided resources.

The details of how the scheduling is done as well as the hardware being used depend on the general purpose of the cluster. In the case of targeting a High-Throughput Computing (HTC) setup, the aim is to get as much absolute compute power as possible, whereas a High-Performance Computing (HPC) setup is built in order to get results quickly using multi-node parallel-processing.

HPC, as realized in NEMO, requires a fast interconnect, a fast cluster filesystem, homogeneous machine types and no hyperthreading of the CPUs. In contrast, for HTC local storage or file caches and heterogeneous machine types may be sufficient and the network requirements are significantly reduced. Usually, hyperthreading is activated. On the operating system (OS) side, most general-purpose Linux systems can easily serve both HPC and HTC setups. The used batch schedulers on the other hand are more specifically chosen for the desired working model. An overview of many of the current cluster scheduling systems can be found in (Reuther et al., 2017). In general, HPC clusters can be used more easily for HTC-like workflows rather than vice versa. Most of the available schedulers offer possibilities to do some kind of dynamic extension or reduction of the available computing resources. These can either be only temporarily available local or remote resources.

The task at hand is to link together two independent clusters, the ATLAS-BFG and NEMO cluster, each with their own resources and two separate batch managers.

However, there are no standard interfaces or abstraction layers in place, and the cross-linking of clusters with different HPC/HTC setups, different resource managers or different login schemes is therefore not a straight-forward task. Accepting workloads from secondary schedulers or delivering basic monitoring information that can be passed through, are not features readily available today.

In order to achieve the on-demand scheduling of resources on one cluster due to requirements on a different cluster, an intermediate layer, or resource scheduler, is put in place.

From its primary concept, the NEMO HPC cluster was designed to provide a full virtualization solution with OpenStack (OpenStack Foundation, 2010) that enables users to spawn virtual machines (VMs) with a pre-configured image – so-called virtual research environments (VREs) (Suchodoletz et al., 2017). These VMs are requested by sending a wrapper-job to the NEMO batch-system (MOAB), which is queued in the same way as other jobs by NEMO users. When the wrapper-job starts, a virtual machine is spawned on the OpenStack instance. The lifetime of the VM is defined by the walltime of the MOAB job. After start-up and some initial checks, the VM is incorporated in the front-end scheduler on the ATLAS-BFG as an additional resource. This resource is in turn used to run the jobs that triggered the start of the VM in the first place. All of these mechanisms are completely transparent to the user of the ATLAS-BFG (Gamel et al., 2017).

In the following, we describe how these virtual resources are integrated into the ATLAS Tier2/Tier3 (ATLAS-BFG) cluster. NEMO and the ATLAS-BFG are utilizing different batch systems. On the user-facing (or frontend) side, SLURM (Jette et al., 2002) is used as a scheduler, while the NEMO cluster (backend) runs a combination of MOAB & Torque (Adaptive Computing, 2014). We use ROCED (Erli et al., 2017), developed at the Karlsruhe Institute of Technology (KIT) to schedule resources. ROCED is used already to integrate NEMO resources into the HTCondor (University of Wisconsin – Madison, 2018) system at KIT. Due to the modular architecture of ROCED, it can also be used for connecting the two systems in Freiburg. To do so, a new component monitoring the SLURM queue on the frontend has been developed.

The system described is running very stable and is in use by the local ATLAS researchers since fall 2017. The performance of the system has been measured using several different benchmarking programs. These benchmarks are also used to quantify the modification of the performance due to changes in the configuration of the resource scheduler and of the virtual machines being spawned. They will also be part of a future continuous monitoring effort in order to be able to detect changes in the submitted workloads. This monitoring and tuning effort will ensure a robust but also dynamic and efficient setup, that reflects changes in user workflows and requirements.

2 Challenges

The ATLAS research groups in Freiburg have very specific requirements to the operating system as well as the installed software. This is to ensure reliable scientific results across all grid sites of the WLCG.

Virtualization has been found to be a technology that can simplify the challenge to provide a specific environment on a range of different heterogeneous and changing platforms, especially in the context of particle physics (Buncic, Aguado Sánchez et al., 2011).

Being only one of multiple user groups on a shared HPC system, especially the choice of operating system has to take into account considerations from all user groups as well as from the party operating the cluster. A fully virtualized environment, independent of the choices made on the HPC cluster itself, will give the best possible scope to implement a system, that looks and behaves in the same way as the non-virtualized ATLAS-BFG cluster. This consistency between the two systems would also make it possible in the future to redirect ATLAS grid jobs submitted remotely to either NEMO or any other opportunistic resource as long as the resource provides the needed infrastructure to run the VM images. The VM images which are made available to OpenStack on NEMO have to be created and updated easily in an automatic procedure and have to fulfil the following requirements:

- Scientific Linux 6 (Fermilab et al., 2011) – current OS on the ATLAS-BFG cluster
- Access to ATLAS software via the CERN virtual file system CVMFS
- User environment from ATLAS-BFG
- Access to both grid-aware datasets on the distributed storage system dCache (Millar et al., 2014) and the local NEMO parallel filesystem BeeGFS (ThinkParQ et al., 2014)

Since the VMs are completely self-contained, all features needed to monitor and benchmark the machine are independent of the two schedulers that are involved and can either be implemented on the VM itself or offloaded to the resource scheduler. In the future, this information will also be used for continuous monitoring of the robustness and performance of the system.

Since the virtualized environment provides access to all resources in the same way as the ATLAS-BFG system, the users of the frontend system do not have to be registered as users of the backend system providing the resources.

2.1 Generation of the virtual machines

The VM template is generated with packer (HashiCorp, 2013), using a Scientific Linux 6 netinstall image as base. The customization and configuration of the template is done with puppet (Puppet, 2005), which is also used for the contextualisation of the non-virtualized worker nodes on the ATLAS-BFG cluster. Changes in configuration are automatically picked up by both systems. The output of this procedure is a static image that can be uploaded to the OpenStack server and is directly available.

In order to simplify software configurations across grid sites, most commonly-used software packages for the HEP-workflow are distributed using CVMFS (Blomer et al., 2015). The software distributed on CVMFS is managed centrally by the experiments, hosted on web servers and transferred to the worker nodes on demand.

The VMs do not contain a predefined CVMFS cache and do not utilize the hard disk as persistent cache. Instead, a RAM disk is being filled on demand from the local frontier squid (Blumenfeld et al., 2008) proxies. The RAM disk is adequately sized to cache the software used in a common ATLAS analysis. This circumvents an on-disk CVMFS installation on the host, which would lead to a heavy usage of the SSDs from mainly unused data. This is a different approach to other solutions relying on access to software from CVMFS and using an on-disk cache like CernVM (Buncic, Aguado-Sanchez et al., 2011).

2.2 Connection of front and backend batch systems

The biggest challenge for a smooth operation is the interconnection between the two different batch systems – SLURM on the frontend (ATLAS-BFG) and MOAB on the backend (NEMO) through a resource scheduler.

The workflow is as follows: the resource scheduler monitors the frontend scheduler to which users send their workload. The requirements are then compared to a list of available configurations on the NEMO HPC and the resource scheduler decides on the number of virtual machines for each configuration needed to fulfil the requirements. The appropriate number of batch jobs are sent to the backend scheduler, each spawning a virtual machine of the chosen type. The jobs that are used to start VMs in the OpenStack environment are regular user jobs on NEMO, which are started according to the availability of resources and the fair share of the NEMO user used to reproduce the fair share of the project. After startup of the VMs, they are integrated as additional resources into the SLURM scheduler and can then be used to process the user jobs queued in the frontend scheduler.

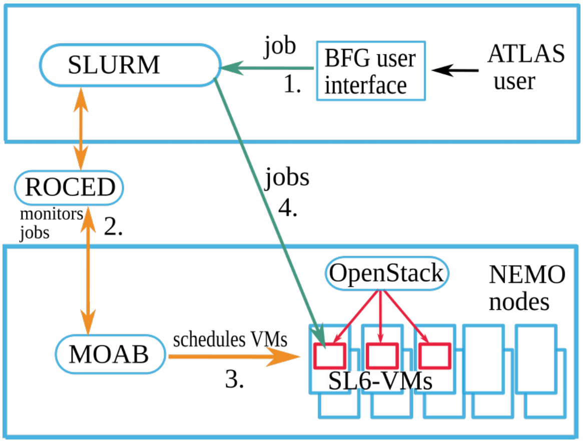


Figure 1: Schematic view of how user jobs submitted to the frontend scheduler SLURM trigger jobs to start VMs on the OpenStack instance at NEMO.

Figure 1 shows the general mode of operation. When submitting the contribution, SLURM in the ATLAS-BFG cluster is set up with separate partitions that reflect the user's affiliation to one of the ATLAS working groups. Each working group is in turn represented by a single user on NEMO, which is used to queue the jobs starting the VMs. By this mechanism, the fair shares for the different areas of research using NEMO are incorporated into the workflow.

ROCED monitors these partitions and requests the start of a VM after the user's job submission. As long as resources are requested and available on NEMO, additional virtual machines can be started. This mechanism leads to a dynamic extension of the amount of job slots available for physics analyses on the frontend system. Before VMs are integrated into SLURM, a diagnosis-check is done to see whether all needed resources are available. After a successful check the VM is set to online in SLURM and jobs can be submitted to the resources.

3 Benchmarks

To understand the performance losses introduced by going to a fully virtualized environment, different benchmarks have been run. All benchmarks are carried out on the same hardware and the results obtained on the virtualized research environment are compared to the results running directly on hardware (»bare metal«) on both the ATLAS-BFG and the NEMO cluster as well, to also assess the impact of different operating systems on the benchmark results.

In addition to the legacy HEP-SPEC06 (HS06) benchmark (HEPiX Benchmarking Working Group, 2006), the evaluation of the performance of the compute resources makes use of three benchmarking programs available in the CERN benchmark suite (Alef et al., 2017):

1. Dirac Benchmark 2012 DB12 (Graciani et al., 2012)
2. Whetstone benchmark (Curnow et al., 1976)
3. Kit Validation benchmark KV (Salvo et al., 2010).

The DB12 and Whetstone programs are evaluating the performance of the CPU through floating-point arithmetic operations. One of the main differences between the two benchmarks resides in the variables used as input to the arithmetic operations: DB12 uses random numbers generated according to a Gaussian distribution, while Whetstone utilizes variables with predefined values.

The KV benchmark runs the ATLAS software ATHENA (Calafiura et al., 2005) to simulate and reconstruct the interactions of muons in the detector of the ATLAS experiment. As our primary target is to measure performances of CPUs in the context of High Energy Physics (HEP) applications, the KV benchmark constitutes a realistic payload, more suited to our goal than the DB12 and Whetstone software. The DB12 benchmark is measured in units of HS06 and therefore can be compared directly to the results from the HEPS-SPEC06 benchmark. The Whetstone scores are expressed in Million of Whetstone Instructions Per Second (MWIPS), and the KV output provides the number of events produced per second. The different benchmarks are used to evaluate the performance of identical 20 cores Intel Xeon E5-2630 CPUs on the two different clusters: the Tier2/Tier3 cluster (ATLAS-BFG) and the shared HPC cluster (NEMO). The performance has been evaluated on three different configurations; the Tier2/Tier3 and NEMO HPC clusters running both on bare metal and the virtual machines running on the NEMO HPC cluster – except for the KV benchmark, which currently cannot be run on NEMO bare-metal nodes.

On the Tier2/Tier3, hyperthreading (HT) technology is activated and the number of cores that can be used is higher by a factor of two with respect to the physical number of CPU cores available. For the virtual machines, an arbitrary number of CPU cores can be requested. The operating system used is Scientific Linux 6 in both cases. The NEMO bare metal has no HT activated due to the more general use case of the system, and uses CentOS7 (CentOS Project, 2017) as operating system. The scores of the HEP-SPEC06, DB12, Whetstone and KV benchmarks have been determined for these three configurations as a function of the number of cores actually used by the benchmarking processes. This number ranges from 2 to 40 for the Tier2/Tier3 bare metal and for the VMs running on the NEMO cluster, for which HT is enabled, and from 2 to 20 for the NEMO bare metal, for which HT is not implemented. The results have been determined by step of two core units. The benchmarks have been run 20 times for each core multiplicity value, and the means and root-mean-squares (RMS) of the corresponding distributions have been extracted.

The scores per CPU and the total scores are presented in Figures 2 and 3 respectively, for the four benchmarks and the three configurations considered, except for the KV software for which the NEMO bare-metal results are not yet available.

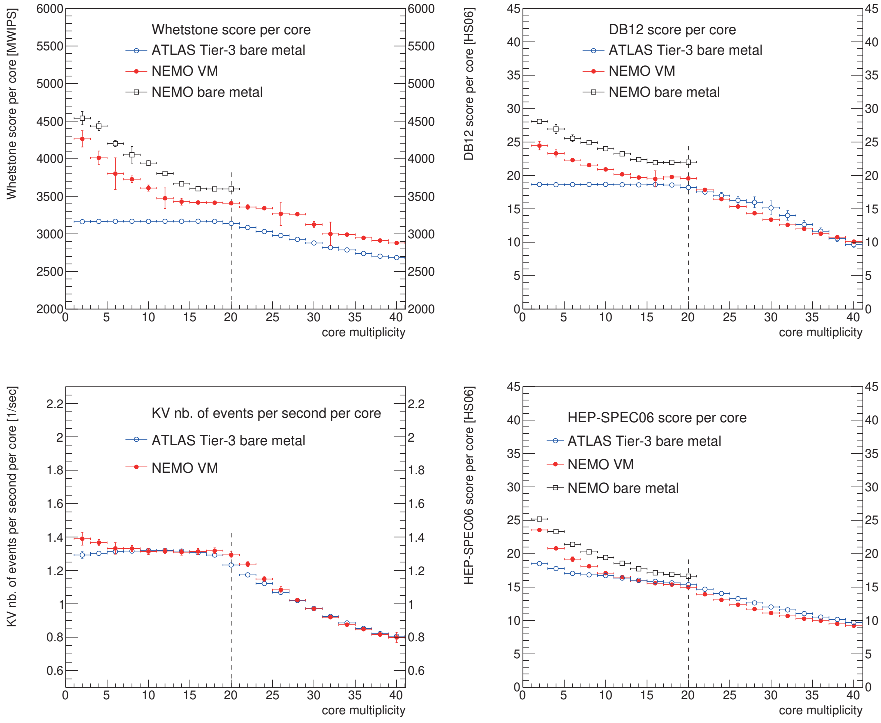


Figure 2: Score per CPU as a function of the core multiplicity for the Whetstone (top left), DB12 (top right), KV (bottom left) and HEP-SPEC06 (bottom right) benchmarks for the Tier2/Tier3 (ATLAS-BFG) running bare metal (blue open circles), the NEMO VMs (red full circles) and the NEMO bare metal (black open squares). The data points represent the average values of the benchmarks for each core multiplicity, and the vertical bars show the associated root-mean-squares. Horizontal error bars are only drawn for visibility and do not represent an uncertainty. The dotted vertical lines at a core multiplicity of 20 indicate the maximum number of physical cores.

While a constant CPU performance is expected per physical core, an increase of the score per CPU is observed in Figure 2 when going towards low values of the core multiplicity for the Whetstone, DB12 and HEP-SPEC06 benchmarks in the NEMO VMs and NEMO bare-metal configurations. Such a behaviour could be explained by a dynamic overclocking of the CPU cores in the system if not all cores are allocated. This is currently under investigation.

For the Tier2/Tier3 bare metal, the scores per CPU remain constant until the maximum number of physical cores is reached, and only start to decrease in the region where hyperthreading is active. The KV results exhibit a similar behaviour

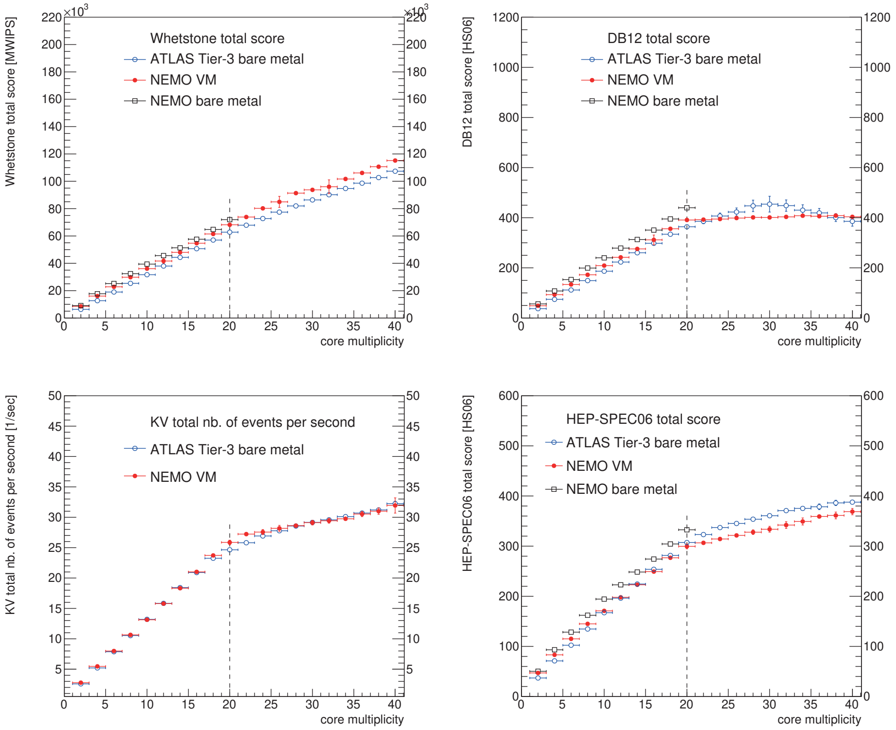


Figure 3: Total score as a function of the core multiplicity for the Whetstone (top left), DB12 (top right), KV (bottom left) and HEP-SPEC06 (bottom right) benchmarks for the Tier2/Tier3 (ATLAS-BFG) running bare metal (blue open circles), the NEMO VMs (red full circles) and the NEMO bare metal (black open squares). The data points represent the average values of the benchmarks for each core multiplicity, and the vertical bars show the associated root-mean-squares. Horizontal error bars are only drawn for visibility and do not represent an uncertainty. The dotted vertical lines at a core multiplicity of 20 indicate the maximum number of physical cores.

for both the Tier2/Tier3 bare metal and the NEMO VMs, with a constant number of events produced per second per CPU below the maximum number of physical cores and a decrease of the performance afterwards. The Whetstone score per CPU at a core multiplicity of 20, the maximum number of physical cores available, is considered as an illustrative example of the benchmark behaviours on the three different configurations. An increase of the CPU performance by the order of 5% is observed when going from the Tier2/Tier3 bare metal to the NEMO VMs, while going from the NEMO VMs to the NEMO bare metal leads to a further increase of performance of the order of 5% as well.

A continuously increasing total score is observed in Figure 3 for the Whetstone benchmark on the three different configurations, while the DB12, KV and HEP-SPEC06 results are characterized by a flattening increase or a constant behaviour once the maximum number of physical cores has been reached. The Whetstone benchmark provides higher CPU performances in the HT region in comparison to the scores obtained with the three other benchmarks. The scores obtained with the KV and HEP-SPEC06 benchmarks indicate an increase of the CPU performance by 15 to 20% when going from the maximum number of physical cores to the upper edge of the HT region, while the Whetstone scores exhibit a larger increase of the order of 60%. The Tier2/Tier3 bare metal and the VMs running on the NEMO cluster share the same configuration in terms of hardware, operating system and hyperthreading. A given benchmark should therefore exhibit a similar behaviour for both configurations and show the effect caused by the virtualization as an offset. The KV benchmark, besides being the more realistic estimator of the CPU performance in the context of HEP applications, is the only benchmark for which this expectation is observed. The behaviours of the different benchmarks still need to be studied in more detail, in order to fully understand the impact of the operating system, hyperthreading and virtualization on the CPU performances.

4 Summary

The HPC cluster NEMO has successfully been integrated into the workflow of local users in Freiburg running ATLAS data analysis jobs. This has been achieved in a transparent way using full virtualization on NEMO and utilizing the resource scheduler ROCED for the integration of the virtual resources into the frontend scheduler. The system is in production since fall 2017.

First performance tests using different benchmarks show some degrading in performance of the virtual machines compared to running on bare metal. The differences are within the expected ranges when using different operating systems and are significantly reduced when going from pure CPU benchmarks like Whetstone or DB12 to benchmarks more closely related to high energy particle physics analysis like HS06 or KV.

The continuous benchmarking effort will ensure a stable and efficient environment. Integrating the results of the benchmarks into the resource-scheduling process can enable cluster administrators to test different configurations, leading to a more efficient usage of the provided resources.

Acknowledgements

The research is supported by the Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg in the project MWK 34-7547.221 »ViCE – Virtual Open Science Collaboration Environment« and by the Bundesministerium für Bildung und Forschung in the project 05H15VFCA1 »Higgs-Physik und Grid-Computing für das ATLAS-Experiment am LHC«.

Corresponding Authors

Felix Bührer: felix.buehrer@physik.uni-freiburg.de
Anton J. Gamel: anton.gamel@physik.uni-freiburg.de
Institute of Physics, University of Freiburg, Freiburg, Germany

ORCID

Felix Bührer https://orcid.org/0000-0002-9274-5004
Anton J. Gamel https://orcid.org/0000-0002-7044-8324
Benoît Roland https://orcid.org/0000-0003-3397-6475
Benjamin Rottler https://orcid.org/0000-0002-6762-2213
Markus Schumacher https://orcid.org/0000-0002-1733-8388
Ulrike Schnoor https://orcid.org/0000-0002-2237-384X

References

Aad, G. et al. (2008). »The ATLAS Experiment at the CERN Large Hadron Collider«. In: *JINST* 3, S08003. DOI: 10.1088/1748-0221/3/08/S08003.

Adaptive Computing (2014). *MOAB HPC SUITE*. URL: http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-suite-grid-option/.

Alef, M. et al. (2017). »Benchmarking cloud resources for HEP«. In: *J. Phys. Conf. Ser.* 898.9, p. 092056. DOI: 10.1088/1742-6596/898/9/092056.

Apollinari, G., O. Bruening, T. Nakamoto and L. Rossi (2017). »High Luminosity Large Hadron Collider HL-LHC«. In: *CERN Yellow Report CERN-2015-005*, pp. 1–19. DOI: 10.5170/CERN-2015-005.1. arXiv: 1705.08830.

Backofen, R. et al. (2006). »A Bottom-up approach to Grid-Computing at a University: the Black-Forest-Grid Initiative«. In: *Praxis der Informationsverarbeitung und Kommunikation* 29, pp. 81–87. DOI: 10.1515/PIKO.2006.81.

Blomer, J. et al. (2015). »The Evolution of Global Scale Filesystems for Scientific Software Distribution«. In: *Computing in Science and Engineering* 17.6, pp. 61–71.

Blumenfeld, B., D. Dykstra, L. Lueking and E. Wicklund (2008). »CMS conditions data access using FroNTier«. In: *Journal of Physics: Conference Series* 119.7, p. 072007. URL: http://stacks.iop.org/1742-6596/119/i=7/a=072007.

Buncic, P., C. Aguado Sánchez, J. Blomer, A. Harutyunyan and M. Mudrinic (2011). »A practical approach to virtualization in HEP«. In: *The European Physical Journal Plus* 126.1, p. 13. ISSN: 2190-5444. DOI: 10.1140/epjp/i2011-11013-1.

Buncic, P., C. Aguado-Sanchez, J. Blomer and A. Harutyunyan (2011). »CernVM: Minimal maintenance approach to virtualization«. In: *Journal of Physics: Conference Series* 331.5, p. 052004. URL: http://stacks.iop.org/1742-6596/331/i=5/a=052004.

Calafiura, P., W. Lavrijsen, C. Leggett, M. Marino and D. Quarrie (2005). »The Athena Control Framework in Production, New Developments and Lessons Learned«. In: *Computing in High Energy Physics and Nuclear Physics 2004*. DOI: 10.5170/CERN-2005-002.456.

CentOS Project (2017). *CentOS Linux release 7.4.1708 (Core)*. URL: https://www.centos.org/.

Curnow, H. and B. Wichman (1976). »A Synthetic Benchmark«. In: *Computer Journal* 19, pp. 43–49.

Eck, C. et al. (2005). *LHC computing Grid: Technical Design Report. Version 1.06 (20 Jun 2005)*. Technical Design Report LCG. Geneva: CERN. URL: https://cds.cern.ch/record/840543.

Erli, G. et al. (2017). »On-demand provisioning of HEP compute resources on cloud sites and shared HPC centers«. In: *Journal of Physics: Conference Series* 898.5, p. 052021. URL: http://stacks.iop.org/1742-6596/898/i=5/a=052021.

Evans, L. and P. Bryant (2008). »LHC Machine«. In: *JINST* 3, S08001. DOI: 10.1088/1748-0221/3/08/S08001.

Fermilab and CERN (2011). *Scientific Linux release 6.8 (Carbon)*. URL: http://www.scientificlinux.org/.

Gamel, A. J., U. Schnoor, K. Meier, F. Bührer and M. Schumacher (2017). *Virtualization of the ATLAS software environment on a shared HPC system*. Tech. rep. ATL-SOFT-PROC-2017-070. Geneva: CERN. URL: `https://cds.cern.ch/record/2292920`.

Graciani, R. and A. McNab (2012). *Dirac benchmark 2012*. URL: `https://gitlab.cern.ch/mcnab/dirac-benchmark/tree/master`.

HashiCorp (2013). *packer*. URL: `https://www.packer.io/`.

HEPiX Benchmarking Working Group (2006). *HEP SPEC06 (HS06) benchmark*. URL: `https://w3.hepix.org/benchmarking.html`.

Jette, M. A., A. B. Yoo and M. Grondona (2002). »SLURM: Simple Linux Utility for Resource Management«. In: *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*. Springer-Verlag, pp. 44–60.

Millar, A. P. et al. (2014). »dCache: Big Data storage for HEP communities and beyond«. In: *Journal of Physics: Conference Series* 513.4, p. 042033. URL: `http://stacks.iop.org/1742-6596/513/i=4/a=042033`.

OpenStack Foundation (2010). *OpenStack*. URL: `https://www.openstack.org/`.

Puppet (2005). *puppet*. URL: `https://puppet.com`.

Reuther, A. et al. (2017). »Scalable System Scheduling for HPC and Big Data«. In: *Journal of Parallel and Distributed Computing*. DOI: `10.1016/j.jpdc.2017.06.009`. arXiv: `1705.03102`.

Salvo, A. D. and F. Brasolin (2010). »Benchmarking the ATLAS software through the Kit Validation engine«. In: *Journal of Physics: Conference Series* 219.4, p. 042037. URL: `http://stacks.iop.org/1742-6596/219/i=4/a=042037`.

Suchodoletz, D. von, B. Wiebelt, K. Meier and M. Janczyk (2017). »Flexible HPC: bwForCluster NEMO«. In: *Proceedings of the 3rd bwHPC-Symposium*. (2016). Heidelberg: heiBOOKS. DOI: `10.11588/heibooks.308.418`.

ThinkParQ and ITWM (2014). *BeeGFS*. URL: `https://www.beegfs.io`.

University of Wisconsin – Madison (2018). *HTCondor*. URL: `https://research.cs.wisc.edu/htcondor/`.

doi: 10.15496/publikation-29062

de.NBI Cloud Storage Tübingen

A federated and georedundant solution for large scientific data

Benjamin Gläßle Maximilian Hanussek Felix Bartusch
Volker Lutz Ulrich Hahn Werner Dilling Thomas Walter
Jens Krüger

High Performance and Cloud Computing Group, University of Tübingen, Tübingen, Germany

The »German Network for Bioinformatics Infrastructure«, or in short »de.NBI«, is a national research infrastructure providing bioinformatics services to users in life sciences research, biomedicine and related fields. At five sites across Germany, cloud sites were established to host the bioinformatics services. In Tübingen an extension of the storage capabilites of the cloud was planned, implemented and brought into production. We here report about the motivation, requirements, design decisions and experiences which might serve as inspiration for other large-scale storage endeavours in the academic domain.

1 Introduction

In this paper we describe the implementation of the OpenStack-based[1] scientific de.NBI Cloud[2], focusing on the storage solution for the deplyoment in Tübingen. First we give a short description of the de.NBI network, its relation to the European ELIXIR project, and the de.NBI cloud in general. In Section 4 we will give an overview of existing storage solutions that can be used in cloud computing. High network traffic in an OpenStack Cloud will be produced by many components and tasks such as storage, deployment and management. To assure good performance, a sophisticated network design is mandatory and will be discussed in Section 5. For a cloud

[1] https://www.openstack.org/
[2] http://www.denbi.de/cloud

doi: 10.15496/publikation-29054

storage system, the integration with VM deployment and management facilities is an important issue. The cloud hosts the collaborative work of different research groups, therefore, an access control for the shared storage is mandatory. To satisfy all these diverse requirements with different products will cause a high management overhead, hence, it is desirable to have a common management interface for all components. In Section 7 we will show how we use the software defined storage solution Quobyte[3] to solve these problems. As common for all bioinformatics applications, I/O-performance is crucial. Several benchmarks for OpenStack Cinder volumes and also for mounted Quobyte volumes are presented in Section 8.

2 de.NBI Cloud

The »German Network for Bioinformatics Infrastructure« (de.NBI) provides high-quality bioinformatics services to users in life sciences research and biomedicine. These services are offered by eight service centers, each focusing on one specific field in life sciences. In 2016 five of these service centers (Tübingen, Bielefeld, Gießen, Heidelberg, and Freiburg) started the de.NBI Cloud. The de.NBI Cloud is an academic cloud federation, providing compute and storage resources free of charge for academic users with research questions in bioinformatics.

Each de.NBI Cloud site operates an OpenStack infrastructure (Ismail et al., 2015; Mullerikkal et al., 2015). A cloud federation concept integrates all instances into a common cloud computing platform (Villegas et al., 2012; Goiri et al., 2010; Celesti et al., 2010). The de.NBI Cloud Portal[4] guides the researchers to a suitable cloud instance that fulfills the researchers' needs. The cloud portal and the OpenStack instances are accessible through single sign-on (SSO), which is based on the ELIXIR Authentication and Authorization Infrastructure (ELIXIR-AAI) (Elixir, 2018; Peter Belmann, 2018). This ensures the connectivity and sustainability in the international context.

In order to get access to the cloud, the researchers have to apply for cloud resources by proposing a project and describing required resources. After approval by a scientific commitee of de.NBI Cloud members, the project is created in the de.NBI Cloud Portal and project resources are allocated at one of the five cloud sites. The

[3] https://www.quobyte.com

[4] https://cloud.denbi.de/

principal investigator of such a project can add colleagues, start VMs, and use the assigned cloud storage.

3 Data Challenge

Each project and its researchers working in the de.NBI cloud site Tübingen have different needs regarding storage resources. Some researchers just want a virtualized hard disc to store their data and access it from their VM. This is typically handled by the block storage component Cinder in OpenStack. It allows the researcher to create so-called volumes in the OpenStack Dashboard and attach them to their VMs. The Quota for this block storage can be set for each project seperately by OpenStack. Thus, the cloud storage system has to provide an integration of the OpenStack Cinder service and additionally fulfill the following requirements.

- Shared usage
- User authentication
- Object storage
- Redundancy
- Scalability
- Management

Because a Cinder volume can be mounted by only one VM at a time (OpenStack Ocata), it is not a suitable solution for data storage in collaborative projects. Data should be shared and used by all contributing researchers in the project. Thus, the cloud storage system should offer the possibility to create a storage volume that can be mounted by several VMs at the same time. A challenge for the storage system is secure multi-tenancy, which means that data of this storage volume is accessible only by researchers of the corresponding project.

Oftentimes, fine granular file permissions are needed for a project-wide storage solution. In some projects valuable primary data should only be writable for a group of the researchers, for example if this primary data has been obtained by expensive wet lab experiments. However, a larger group of people should be able to have read-only access to the data to work with it. This cannot be provided by OpenStack, thus, the cloud storage system has to authenticate and authorize the correct user within the VM for access of the project storage.

The challenge for cloud storage described in this paper so far has to be considered with respect to traditional file systems. Another storage solution that has become more popular in recent years is the so-called object storage (Mesnier et al., 2003), which allows simple, key-based access to files. Instead of working with file systems, files, and file hierarchy, the user stores data objects. These data objects and additional metadata are accessible by a unique ID. A cloud storage system should offer a possibility to use object storage. We envision achieving a considerable integration improvement through automatisation and usage of workflows, directly interacting with objects residing within a corresponding storage.

Another important prerequisite for reliable storage systems is redundancy. A broken hard disc or the outage of a whole storage node must not result in data loss. There are several feasible methods to prevent data loss, like error correction codes, any kind of RAID, synchronous or asynchronous replication, and geographical replication to achieve redundancy. The storage management software should also provide information about the status of the devices. In case of a device error, the automatic recreation of the affected data has to be possible, taking advantage of the data redundancy of the system.

As the OpenStack cloud is up and running, the storage system should integrate as seamlessly as possible with the OpenStack infrastructure and services. The storage system should be scalable by capacity and performance, meaning that the addition of new storage shelves or nodes is possible.

A storage solution should also provide an easy-to-use management software that can perform basic tasks like setting quotas, creating or deleting volumes.

4 Federated Cloud Storage

The term cloud storage is frequently used in various contexts and in a rather indiscriminate fashion, often without clearly specifying for what purpose and which technologies referred to. The use cases range from a remote Dropbox-like storage[5] to highly performant parallel file systems. For the academic de.NBI Cloud at hand, federated network or clustered file systems are of interest. These kind of systems offer POSIX compliant file systems or object storages to store, process and share data. These volumes or objects are accessible from VMs residing in the de.NBI Cloud.

[5] https://www.dropbox.com

The following list of technologies and providers gives a brief overview of currently available solutions and is far from being exhaustive.

Ceph represents an object storage residing on a distributed cluster of storage devices[6]. Per design, it aims at a fully distributed mode of operation avoiding a single point of failure. There is no intrinsic scaling limit, theoretically allowing to assemble a Ceph storage on the exabyte scale. Ceph is able to expose block storage volumes as a thin-provisioned block device.

Compuverde[7] is a software-defined storage solution, basically able to run on heterogenous storage hardware. It relies on a decentralized and symmetric architecture, avoiding special purpose nodes and consequently single points of failure. Linear scaling in terms of capacity and performance is achieved due to this architecture. Compuverde provides block, file system or object based access to data.

BeeGFS[8] is a parallel file system clearly focusing on speed and availability, mainly intended for high-performance computing environments. It separates the metadata from user data, enabling scalability similar to other federated storage solutions while maintaing an outstanding performance. A variant of BeeGFS is available, called BeeOND[9], making a parallel file system available ad hoc over multiple (virtual) machines.

NetApp[10] offers hybrid cloud data services. The portfolio covers a whole range of cloud and data mangement related storage solutions, including object storage, all flash storage and backup strategies. NetApp not only offers technologies but also acts as service provider.

A similar spectrum is covered by HPE[11]. As classic hardware supplier which also offers a broad range of services and solutions addressing data transfer between multiple cloud sites, hybrid storage solutions and data deduplication, among other aspects.

HDS[12] offers solutions for hybrid flash storage and cloud object storage. The hybrid storage solution is a technical basis for data tiering, combining high performance through solid state disks and capacity by conventional hard disks.

[6] `https://ceph.com/`

[7] `http://compuverde.com/`

[8] `https://www.beegfs.io`

[9] `https://www.beegfs.io/wiki/BeeOND`

[10] `https://www.netapp.com`

[11] `https://www.hpe.com`

[12] `https://www.hitachivantara.com`

Similar to its competitors, Dell/EMC[13] offers a broad spectrum of storage solutions, ranging from all-flash, over hybrid and software-defined to object storage.

XtreemFS (Hupfeld et al., 2008) is an academic project as well as a federated file system. It is being developed mainly at the Zuse Institute Berlin and was supported through multiple EU and national grants. XtreemFS claims to be versatile, reliable and scalable, which is achieved through seperation of data and metadata services in combination with a smart replication scheme. Quobyte is conceptionally based on XtreemFS.

The MoSGrid science gateway (Grunzke et al., 2012; Krüger et al., 2014) has used XtreemFS for the past 8 years as the basis for its federated storage concept and the handling of simulation data, including its annotated metadata.

5 Network Layout

The current setup at the University of Tübingen consists of two independent storage clusters. While cluster I is located at the primary data center »MS24«, cluster II serves as a replication target and resides at a secondary in a different part of town, »WAE76«. Both are connected by (i) an isolated 10/40 Gbit/s backbone for data replication. Consequently, only cluster I interfaces with OpenStack over various networks: (ii) All storage nodes provide a 40 GBit/s uplink into the OpenStack network layer managed by Neutron, allowing direct storage access from VMs. (iii) In addition, some storage nodes are equipped with FDR Infiniband, serving Openstack Glance and Cinder via IP over IB. (iv) Finally, a standard 1 GBit/s network is used for basic services such as DNS/DHCP, NTP, node provisioning, node monitoring, and access to the baseboard management controller, BMC (see Fig.1).

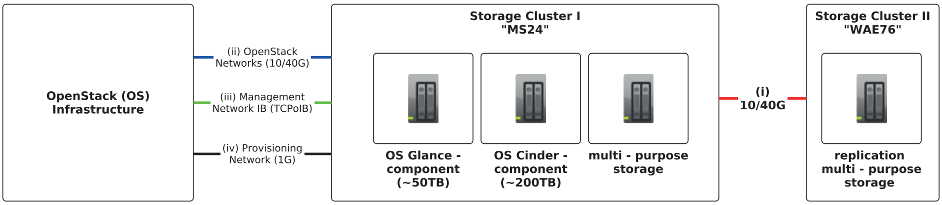

OpenStack (OS) Infrastructure
(ii) OpenStack Networks (10/40G)
(iii) Management Network IB (TCPoIB)
(iv) Provisioning Network (1G)
Storage Cluster I "MS24"
OS Glance - component (~50TB)
OS Cinder - component (~200TB)
multi - purpose storage
(i) 10/40G
Storage Cluster II "WAE76"
replication multi - purpose storage

Figure 1: Storage cluster replication network and integration into OpenStack.

[13] https://www.dellemc.com/en-us/storage/data-storage.htm

A more detailed representation of the overall storage cluster network and its integration into OpenStack can be found in Figure 2. The 10/40 Gbit/s replication network (red lines) is fully isolated and thus inaccessible from OpenStack. Its 40 Gbit/s switch backbone and the 10 Gbit/s uplinks provide a node-to-node bandwidth of 9.4 Gbit/s and latencies between 25 µs (cluster internal) and 80 µs round trip time (rtt) across the clusters.

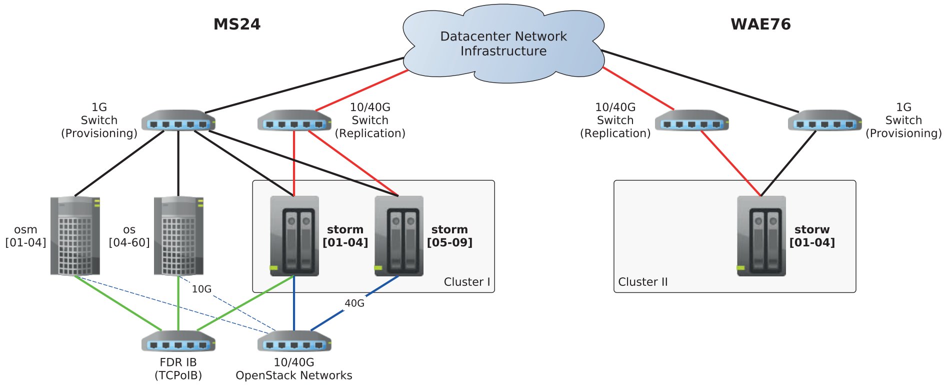


Figure 2: *Left:* Storage cluster I with cluster nodes storm[01-09], integrated into OpenStack. *Right:* Storage replication cluster II with cluster nodes storw[01-04].

The 40 Gbit/s OpenStack IPv6 network (Fig.2, blue line) not only allows access of the OpenStack VMs to storage cluster I, but is also utilized for internal storage communication, i. e. for data striping or erasure coding. We measured the corresponding inter-node network bandwidth and latency to be 37 Gbit/s and 20-30 µs, respectively. Given the large number of hypervisor nodes, each attached via a single 10 Gbit/s network interface (Fig.2, dashed blue line), and the expected heterogeneous data traffic to occur, additional network benchmark tests were not performed.

Native Infiniband protocols are currently supported neither in OpenStack nor by the Quobyte Storage Appliance (see below in Section 6.2) itself. Further limitations of the underlying hardware[14] leave IP over IB (IPoIB) as the only usable network layer. As a result, this network (Fig.2, green line) is used almost exclusively for OpenStack management tasks and for accessing the OpenStack Cinder service on storm[01-04].

[14] To the best of our knowledge, Mellanox ConnectIB network interface cards cannot be reconfigured to support Ethernet protocol.

The resulting trade-off manifests itself most clearly by comparing performance characteristics of the native Infiniband with the corresponding IPoIB layer. While the native Infiniband on storm[01-04] consistently reaches 48 Gbit/s with a latency of 0.95 µs, IPoIB bandwidths seem lower and less stable, varying between 36 and 44 Gbit/s (latencies are measured between 20 and 34µs).

6 Hardware Setup

Our cloud storage setup consists of 13 servers, each equipped with 20 CPU cores (Intel Xeon E5-2640 v4) distributed over 2 sockets and 64 GB of RAM. Most noteably, all storage servers offer 90 disks slots in 4 U. Adding hard disks with a capacity of 12 TB results in a very dense and space-efficient overall storage capacity of 13.9 PB. 4 out of the 13 servers are equipped with 16 SSD drives, with a capacity of 3.8 TB each, replacing some of the 12 TB hard disks. Each storage server is equipped with 4x 10 Gbit/s, 2x FDR-IB, 2x 1 Gbit/s network devices (see Section 5). The 13 servers are distributed over two areally separated server-rooms with a distance of 3 kilometers in between. 9 servers form the core working site and are directly attached to our OpenStack cloud infrastructure. The other 4 remaining servers are set up in the server room of our central computing building and will be used for geo-replication purposes.

7 Software Implementation

The storage hardware is complemented by the distributed filesystem and management software Quobyte[15]. As a distributed filesystem, Quobyte distinguishes between file data and metadata. Both types of data are striped over a number of storage server nodes, allowing parallel read and write operations of many devices at the same time. Quobyte is a software-defined storage system and management software relying on so-called registry, metadata, data, S3 and web services. The whole amount of available storage space is combined into a single storage pool. This storage pool can be divided into different Quobyte volumes as desired. Specifically for our cloud we have set up a Quobyte volume for the OpenStack Glance service to store all images and snapshots of our users and a separate Quobyte volume for the

[15] `https://www.quobyte.com/whitepaper`

OpenStack block device service Cinder. Both volumes require different setups which can be handled by Quobyte. The Glance Quobyte volume, for example, consists only of fast accessible SSDs with a size of ~50 TB. Whereas the Cinder Quobyte volume is built up only of HDDs with a capacity of ~200 TB. These examples show the flexibility which a software defined storage can offer. Beyond that, you can change a diverse setup of properties to tune the volumes and their behavior to your needs. It is possible to restrict the size of a volume by setting different quotas.

An even more interesting feature is to set read and write permissions on a per-user-base and per volume. Especially in the field of bioinformatics this plays an important role, as some data are sensible patient data which has to be protected thoroughly. Access to a Quobyte volume can be restricted by different mechanisms. First, it is possible to limit the access to a volume to clients within a specified IP range, which results in the fact that clients can only access a volume if they have access to the same network. The second mechanism is to provide certificates for specific volumes. The usage of certificates allows to set read and write permissions on a per-user-base which has already been used in production for a cloud project working on sensible patient data in the context of neuronal diseases. Quobyte can create a new X.509 Certificate Authority (CA) or import an existing CA and private key. Subsequently, one can create and distribute a certificate for each to grant access to the specific volume. Users without a certificate cannot access this volume. For each user with a certificate, the administrator explicitly grants read and/or write access to this specific volume. Thirdly, Quobyte allows, in conjunction with the certificate mechanism, to prevent the access for root. As such, accidental or intended abuse of sudo rights or root privileges in combination with mounted volumes can be prevented. In practice, this disables users without write permission to change or delete data, even if they are root in their VM in which the volume is mounted.

Beneath the services on block storage level, Quobyte also offers a native S3 interface to use the underlying hardware as part of an S3 object storage, which provides even more flexibility. The current setup consists of Cinder and Glance volumes. The remaining storage can be used as multipurpose storage, for example as an S3 object storage, as a shared native Quobyte volume mounted directly into a VM of the cloud or as a repository for biological reference data and databases. Especially for such valuable reference data or large datasets in the range of petabytes we offer the service of mirroring such data to the second server room to keep them for disaster recovery purposes. Even if the main server room were lost, it would be possible to resume working on the georeplicated data at the second server room. Besides the mirroring mechanisms, Quobyte offers a set of mechanisms to be as fault tolerant as possible. Due to different concepts like replication or erasure coding the loss of single hard drives will not be a problem. If one of the four registry servers is lost, the system will just change to one of the three remaining. The other five machines are purely for keeping data and metadata.

8 Performance

In this section we present I/O benchmarks obtained by running »iozone« and other utilities on Openstack instances against the storage cluster via a high-bandwidth network (see Section 5). However, please note that especially data on the object storage described in Section 3 is often accessed from remote locations, in which case performance is limited by the typically much lower network bandwidth of the remote end, making the benchmarks presented below irrelevant for the remote access use case. Remote location in this case means that access takes place from a network outside of the computing centre. An example would be an access on the object storage from an institutional network from another continent. The storage integration currently provides two methods to access block storage within a VM. The first is to mount OpenStack Cinder volumes, which relies on a Quobyte volume as backend to store the actual data, whereas the second means to mount Quobyte volumes directly. Be aware that the first method utilizes our FDR IB network and only 4 Quobyte servers, whereas the second one uses the 10 Gbit/s Ethernet network and all 9 servers. Additionally, Quobyte volume configuration options such as replication or erasure coding mode, striding width, lock and cache settings can have a large influence on the performance, especially if multiple processes concurrently work on the same files. Due to the additional complexity of multiple processes, we restrict ourselves to single thread benchmarks on one VM.

Single node IO benchmarks are performed with `iozone` version 482 (Norcott, 2018) on a CentOS-7 VM occupying a complete hypervisor. A Cinder volume is attached to the VM, formatted as `xfs` and mounted. Differently configured Quobyte volumes are mounted as well, by adding the VM to an internal 10 Gbit/s storage network and using the Quobyte client. On such a mount point and for comparison on the root partition of the VM residing on a local SSD of the hypervisor, iozone performs write, rewrite, and random write as well as the corresponding read operations for various file and block sizes. Additionally, manual timings and tests with iperf, cp, scp and rsync were performed to corroborate the iozone results.

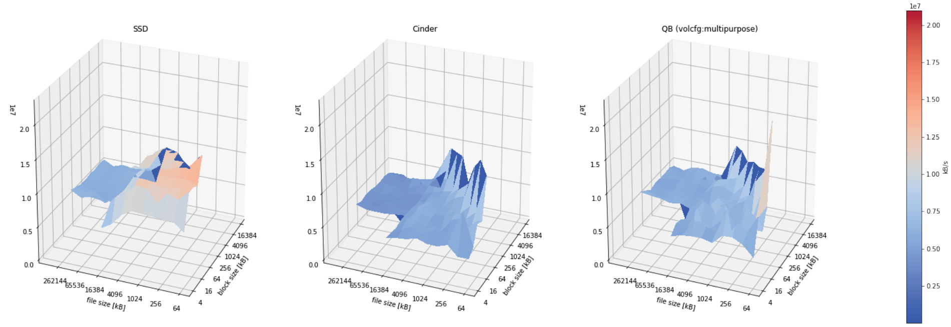


Figure 3: Read bandwidths

The IO performance of small files is dominated by caching effects. Reading, as shown in Fig. 3, performs equally well on local SSDs, Cinder volumes and directly mounted Quobyte volume and achieves artificial read rates as high as 20 GByte/s. Read bandwidths being larger than the underlying network bandwidth indicate cache effects on the node, e. g. of the VM kernel and in the case of Cinder of the host kernel and its xfs filesystem driver. Write performance is illustrated in Fig. 4. Writing on directly mounted Quobyte volumes does not utilize the page-cache of the guest kernel and therefore is significantly slower for small blocksizes than writes on a local SSD or the Cinder volume. Larger file and especially block sizes lead to a regime where the performance of the Quobyte volume seems to be limited by the network bandwidth, whereas Cinder volumes might be limited by the underlying storage architecure, such as the aggregate bandwidth of the hard discs the files are actually stored on.

IO on large files up to 8 GBytes shows more stable bandwidths and revealed a large dependence on the Quobyte volume configuration. Reading from a Quobyte multipurpose (5+3 error coded) volume still exceeds the network bandwidth with up to 6 GBytes/s, whereas a Cinder volume only achieves 4.5 GBytes/s and a 3× replicated Quobyte volume reaches only 0.8 GBytes/s. The performance of the multipurpose volume indicates that the VM kernel is still able to cache. Write benchmarks reveal a significantly different picture. Cinder volumes achieve 1.8 GBytes/s, whereas the 3× replicated volume is slightly better with 0.9 GBytes/s than the multipurpose volume with 0.7 GBytes/s, which might be due to the error coding overhead.

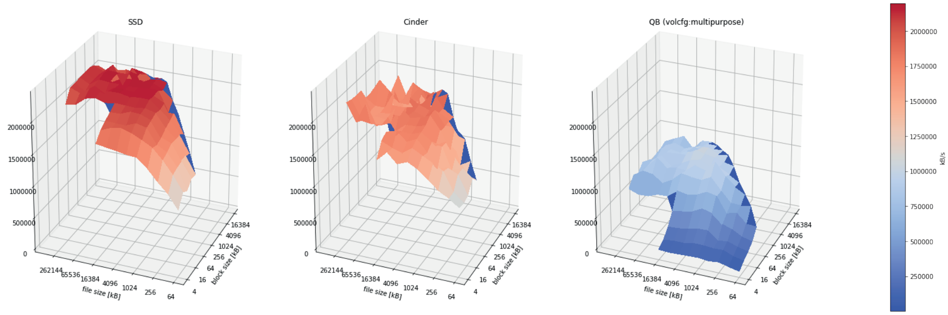


Figure 4: Write bandwidths

Parallel `dd` write activity on a single Quobyte volume mounted by all 9 Quobyte servers was able to saturate the 40 Gbit/s storage network. Quobyte volumes further provide the possibility of file-name-based prefetching, which could substantially improve certain applications, e. g. machine learning.

9 Future Development

The scalable and federated de.NBI Cloud Storage has proven to be a solution for the needs of the de.NBI community. The build-in flexibility of the solution allows for customization of the storage system to be used in further scientific disciplines and collaborations. Therefore, as part of the cyber valley initiative, an adaptation of the storage solution will be implemented to meet the requirements of the machine learning community. Furthermore, the federated approach of the de.NBI Cloud Storage predestines the use of the storage solution to implement services within the emerging BaWü data federation (»BaWü-Datenföderation«, Hartenstein et al., 2013). If legal and political conditions permit, de.NBI cloud storage can even become part of the data federation. With connection to or participation in the BaWü data federation, the services offered by other participants in the data federation will also open up for the de.NBI cloud. This will significantly simplify the implementation of e. g. archiving or research data management within the de.NBI cloud.

10 Conclusion

We were aiming at implementing a storage solution for the de.NBI Cloud Tübingen, providing a broad range of features and functionalities. The full integration with the existing Openstack installation was a must. Building on Quobyte, we were able to install a multipurpose storage solution, offering block and object storage for the bioinformatics community.

Acknowledgement

The authors acknowledge the support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG. Part of the work presented here was also supported through BMBF funded project de.NBI (031 A 534A) and MWK Baden-Württemberg funded project CiTAR (»Zitierbare wissenschaftliche Methoden«).

Corresponding Author

Jens Krüger: jens.krueger@uni-tuebingen.de
High Performance and Cloud Computing Group,
University of Tübingen, Tübingen, Germany

ORCID

Benjamin Gläßle https://orcid.org/0000-0002-4679-979X
Maximilian Hanussek https://orcid.org/0000-0002-2598-4398
Felix Bartusch https://orcid.org/0000-0003-0711-5196
Volker Lutz https://orcid.org/0000-0003-0787-4600
Ulrich Hahn https://orcid.org/0000-0003-4471-9263
Werner Dilling https://orcid.org/0000-0001-6962-416X
Thomas Walter https://orcid.org/0000-0002-8656-2340
Jens Krüger https://orcid.org/0000-0002-2636-3163

References

Celesti, A., F. Tusa, M. Villari and A. Puliafito (2010). »How to Enhance Cloud Architectures to Enable Cross-Federation«. In: *2010 IEEE 3rd International Conference on Cloud Computing*. IEEE, pp. 337–345. ISBN: 978-1-4244-8207-8. DOI: 10.1109/CLOUD.2010.46.

Elixir (2018). *Elixir Handbook of Operations*. URL: https://www.elixir-europe.org/sites/default/files/documents/elixir-handbook-operations.pdf (visited on 21. 06. 2018).

Goiri, I., J. Guitart and J. Torres (2010). »Characterizing Cloud Federation for Enhancing Providers' Profit«. In: *2010 IEEE 3rd International Conference on Cloud Computing*. IEEE, pp. 123–130. ISBN: 978-1-4244-8207-8. DOI: 10.1109/CLOUD.2010.32.

Grunzke, R. et al. (2012). »A data driven science gateway for computational workflows«. In: *UNICORE Summit 2012, Proceedings*. Vol. 15. ISBN: 9783893368297.

Hartenstein, H., T. Walter and P. Castellaz (2013). »Aktuelle Umsetzungskonzepte der Universitäten des Landes Baden-Württemberg für Hochleistungsrechnen und datenintensive Dienste«. In: *PIK - Praxis der Informationsverarbeitung und Kommunikation* 36.2, pp. 99–108. ISSN: 1865-8342. DOI: 10.1515/pik-2013-0007.

Hupfeld, F. et al. (2008). »The XtreemFS Architecture - A Case for Object-based File Systems in Grids«. In: *Concurrency and Computation: Practice and Experience* 20.17, pp. 2049–2060.

Ismail, M. A., M. F. Ismail and H. Ahmed (2015). »Openstack Cloud Performance Optimization using Linux Services«. In: *2015 International Conference on Cloud Computing (ICCC)*. IEEE, pp. 1–4. ISBN: 978-1-4673-6617-5. DOI: 10.1109/CLOUDCOMP.2015.7149648.

Krüger, J. et al. (2014). »The MoSGrid science gateway - A complete solution for molecular simulations«. In: *Journal of Chemical Theory and Computation* 10.6. DOI: 10.1021/ct500159h.

Mesnier, M., G. Ganger and E. Riedel (2003). »Storage area networking - Object-based storage«. In: *IEEE Communications Magazine* 41.8, pp. 84–90. ISSN: 0163-6804. DOI: 10.1109/MCOM.2003.1222722.

Mullerikkal, J. P. and Y. Sastri (2015). »A Comparative Study of OpenStack and CloudStack«. In: *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*. IEEE, pp. 81–84. ISBN: 978-1-4673-6993-0. DOI: 10.1109/ICACC.2015.110.

Norcott, W. D. (2018). *Iozone Filesystem Benchmark*. URL: http://www.iozone.org/ (visited on 21. 06. 2018).

Peter Belmann, M. P. (2018). *ELIXIR Webinar: de.NBI Cloud Integration to ELIXIR AAI.* URL: `https://www.elixir-europe.org/events/elixir-webinar-denbi-cloud-integration-elixir-aai` (visited on 21. 06. 2018).

Villegas, D. et al. (2012). »Cloud federation in a layered service model«. In: *Journal of Computer and System Sciences* 78.5, pp. 1330–1344. ISSN: 0022-0000. DOI: `10.1016/J.JCSS.2011.12.017`.

A Sorting Hat For Clusters

Dynamic Provisioning of Compute Nodes for Colocated Large Scale Computational Research Infrastructures

Jonathan Bauer[*] Manuel Messner[*] Michael Janczyk[*]
Dirk von Suchodoletz[*] Bernd Wiebelt[*] Helena Rasche[†]

[*]eScience Department, Computer Center, University of Freiburg, Freiburg, Germany
[†]Department of Bioinformatics, University of Freiburg, Germany

Current large scale computational research infrastructures are composed of multitudes of compute nodes fitted with similar or identical hardware. For practical purposes, the deployment of the software operating environment to each compute node is done in an automated fashion. If a data centre hosts more than one of these systems – for example cloud and HPC clusters – it is beneficial to use the same provisioning method for all of them. The uniform provisioning approach unifies administration of the various systems and allows flexible dedication and reconfiguration of computational resources. In particular, we will highlight the requirements on the underlying network infrastructure for unified remote boot but segregated service operations. Building upon this, we will present the Boot Selection Service, allowing for the addition, removal or rededication of a node to a given research infrastructure with a simple reconfiguration.

1 Motivation

Efficient procurement, deployment, administration, and operation of digital research infrastructures is a central task for computer centres at scientific facilities. The eScience department of the computer centre of the University of Freiburg is responsible for the provisioning of reasonably scaled research infrastructures, such as cloud, storage, and especially HPC, to cater to the needs of various scientific communities.

DOI: 10.15496/publikation-29055

Ideally, these infrastructures are optimally matched to the requirements of their respective users during their entire life cycle and offer the best return on investment possible. By unifying cloud and HPC nodes into a common provisioning and monitoring environment, a more flexible and easily extensible research infrastructure can be provided: researchers can pool their project funds to be quickly translated into the appropriate compute services.

Operating numerous, large research infrastructures, in particular with a small team of administrators, requires significant standardization in hardware and software. Cluster management systems like xCAT[1] facilitate the management of compute nodes by providing tools for fast provisioning and retiring of systems and configuring core services like DHCP and DNS servers – these help administrators to focus on more relevant challenges. Traditionally, every individual large scale computational system uses its own dedicated provisioning infrastructure. While this achieves a clear separation of tasks between operators, it also duplicates the administrative efforts to manage different instances of similar services. Consolidating the various infrastructures by sharing a single base infrastructure is a logical step towards a unified operating model.

2 Provisioning Methods

Orchestrating the provisioning of various types of machines in a shared infrastructure has it challenges, however. Established bare-metal provisioning techniques are typically either stateful or stateless. The stateful approach involves disk imaging techniques like xCAT or Kadeploy (Jeanvoine et al., 2012) or live provisioning tools like Foreman and puppet (Lehrbach et al., 2017).[2] An alternative approach is stateless setups, like xCAT's diskless NFS-based or RAM-based implementation. While widely adopted as a technique to bootstrap minimal environments to install an operating system on the local hard drives for stateful operations (Stirenko et al., 2013), remote boot is not as popular for stateless bare-metal provisioning. Unlike in stateful installations where nodes can reboot in their exact previous state, stateless nodes can quickly change to another mode of operation by simply rebooting, facilitating node replacement.

[1] `https://xcat.org/` (visited on 10. 06. 2018).

[2] `https://www.theforeman.org/`; `https://puppet.com/` (visited on 03. 01. 2019).

There exists a long tradition of operating numerous large and purpose-built infrastructures within the computer centre of Freiburg. In order to ensure the uniformity of the software running on those systems, PCs in pools (Trahasch et al., 2015) as well as all HPC compute nodes are being booted through PXE (Schmelzer et al., 2011). We use the OpenSLX booting project[3] as a core to create stateless bootable Linux environments distributed via Distributed Network Block Devices (DNBD3) – an internally developed NBD variant replicating on and distributing images from multiple servers to alleviate image synchronization and network bottleneck problems often present in similar PXE boot architectures (Rettberg et al., 2019). Base root filesystem images are complemented with configuration flavors applied at boot time depending on machine-specific attributes. This approach avoids the »personalization« of the machines and makes nodes easily replaceable and interchangeable. Once our iPXE[4] booting method is applied to a class of compute resources, the affected compute nodes become part of a pool and their individual operational profile can be changed easily. This idea led us to the development of a centralized Boot Selection Service (BSS) orchestrating the commissioning of new hardware resources, reducing the time and efforts required between their acquisition and their operationality.

3 Base Infrastructure

Experience with the different user communities and an analysis of the actual requirements of their computing power needs showed that the variance in hardware of existing systems is rather limited. This similarity in underlying hardware allows for the simplification of new hardware procurement, operation, administration, and monitoring of the whole installation, as common techniques and services can be reused for all infrastructure pools. Our group procured more than 1000 compute nodes in the last two years for cloud and cluster projects like de.NBI cloud, bwCloud SCOPE, bwForCluster NEMO (HPC) and ATLAS Tier2/Tier3 (HTC) compute resources. More are expected to be added to that list. Fortunately, it was possible to acquire highly similar systems sharing the same base hardware configuration. Only a few deviations from the one-node-fits-all-purposes exist, such as the operation of some high memory nodes, GPGPU machines, or nodes having an additional 10 GbE

[3] `https://github.com/OpenSLX` (visisted on 14. 06. 2018).
[4] `https://ipxe.org` (visited on 04. 01. 2019).

card installed. This limitation in the variance of machine configurations and vendors simplifies tasks, e. g. IPMI remote management or defect handling.

All the machines share the same redundant Ethernet switch infrastructure and uplink for a uniform network connectivity. Network isolation between the different projects is achieved through individual VLANs that are, atypically, available on all switch ports throughout the network and later enforced within their respective operating systems. This obviously requires a mutual trust relationship between the cluster operators, since nothing prevents administrators from joining other VLANs available in the network.

3.1 Designing the Network Infrastructure

The main goal of the existing common network infrastructure was to separate tasks like HPC or cloud operation of a certain flavour into distinct subnets. This was statically configured at the switch level. However, we used a common network (either VLAN or directly attached) for the machine health and hardware monitoring over IPMI and further components like switches and racks (Figure 1). The user traffic (either for high speed resource access or external traffic) is handled within a dedicated user network. As before, different resources, e. g. NFS, dCache shares or traffic of the user instances from the clouds, are separated into different VLANs. For booting and machine filesystem provisioning, all systems used a 1 GbE interface. In the proposed network layout, we select the appropriate subnet according to the operation mode during the boot procedure. Only the boot process uses plain Ethernet traffic. We define a VLAN for each operation mode: HPC, HTC and the two clouds. Special-purpose RDMA parallel filesystem or MPI traffic is kept within the Omni-Path infrastructure.

3.2 Flexible Preboot Environment

Large hardware installations require significant coordination between services to function as a unit. Some services are essential for base operations like DHCP servers for network connectivity, TFTP/HTTP servers to deliver the preboot environments, and, in our case, DNBD3 servers to provide the main operating system images as remote block device to the bare-metal nodes. Other services like monitoring and inventory management are optional, though they are also often employed for management purposes. The proper cooperation of these services is key to achieving the nodes' expected behaviour throughout the infrastructure – creating or changing node configurations quickly becomes a hassle.

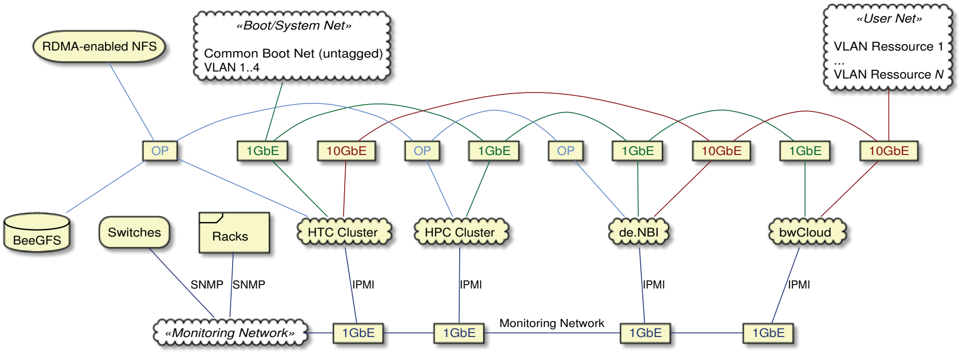


Figure 1: Basic network configuration of distinct Ethernet and Omni-Path infrastructures

Here iPXE, a fork of gPXE, shows significant improvements over its older, Etherboot derived predecessor (Anvin et al., 2008). The combination of the minimal scripting language, the ability to chain scripts, and the HTTP(S) interface provides a high degree of flexibility during the otherwise very static preboot phase of plain PXE setups. Even though client-specific configuration was possible in PXE, it could only be applied based on a client's UUID, its MAC address or its IPv4 address or subnet. Serving different boot entries based on other machine-specific attributes and/or on information stored on external services could not be elegantly implemented. iPXE scripts, however, can implement such decisions based on certain information gathered by the firmware (architecture, model, machine UUID), on the network parameters received by the DHCP server and by accessing external resources. Rewriting DHCP options can be handy, e. g. overwriting the option 66 (next-server) to reroute to another PXE server instance for availability reasons. Moreover, the configuration of network interfaces is more extensive. Multiple interfaces can not only be configured independently via DHCP but also statically, DNS support enables the use of FQDNs when connecting to remote hosts and VLANs can be configured early to gain access to these networks and their resources.[5]

[5] E. g. boot files like kernel and initramfs or other PXE servers.

Chaining of iPXE scripts with HTTP requests provide unique opportunities. A web application can receive a client's requests, evaluate client properties and metadata, and then trigger further actions specific to that client. During this step, the web application can access APIs of other services to include additional information in its decision making process. Similarly, information gathered by the iPXE firmware can be propagated to other services, or even used to configure these services directly, e. g. to create a DHCP reservation. Finally, iPXE also supports cryptographic features like TLS, HTTPS, the use of private root SSL certificates to secure web communication as well as code signing to verify the integrity of downloaded files.

4 The Magic of Booting

In a standard PXE setup, two components are involved: a DHCP server and a TFTP server with PXE images. Upon the initialization of the PXE ROM, an IP address is requested from the DHCP server which issues a lease based on the node's MAC address or machine UUID, and then points to the TFTP server and the PXE image to retrieve from it. Those images initially load a configuration file, again based on MAC or UUID, containing boot entries pointing to kernels and initial RAM disks located on a remote file server. The PXE phase ends by loading and executing the kernel.

This process is not only static due to the PXE configuration: changing the next-hop address and PXE image names traditionally involves reconfiguring the DHCP server. This process is also error-prone: all files are transferred with TFTP via UDP which is known to be unreliable, especially in highly loaded or multi-hop networks. This can potentially lead to boot failures leaving nodes in an unpredictable state. The network uncertainties can be mitigated by using TCP powered HTTP for file transfers, supported in newer versions of PXELINUX or iPXE. However, overcoming the static character of the setup is a bigger challenge that requires a new component: Boot Selection Service (BSS, Figure 2).

BSS is an internally developed service which dynamically responds with custom iPXE scripts depending on the requesting machines' attributes (MAC, UUID) and on the projects (HPC, cloud, PC pool, service and testing environment) they are associated with. After the initial handshake with the DHCP server (e. g. Infoblox, steps 01-02), machines download a generic iPXE image from the next-hop server (03-04) containing an embedded script automatically chaining to the BSS' web API, including its MAC address and UUID as `GET` parameters (05). The BSS then determines its project affiliation from these machine-specific attributes and responds with the custom iPXE script (06) to boot that project's operating system from a DNBD3 remote block device (07-10).

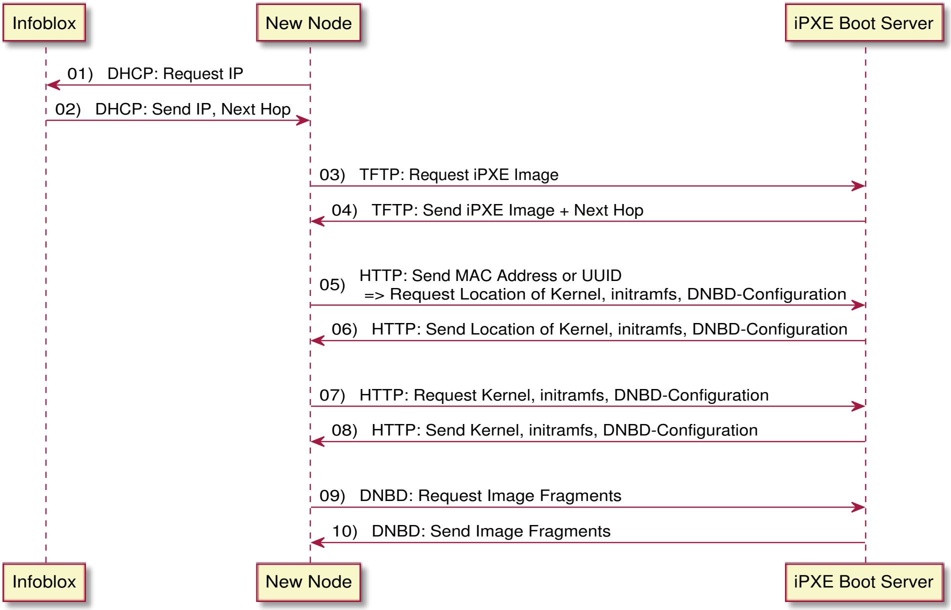

Infoblox
New Node
iPXE Boot Server
01) DHCP: Request IP
02) DHCP: Send IP, Next Hop
03) TFTP: Request iPXE Image
04) TFTP: Send iPXE Image + Next Hop
05) HTTP: Send MAC Address or UUID
=> Request Location of Kernel, initramfs, DNBD-Configuration
06) HTTP: Send Location of Kernel, initramfs, DNBD-Configuration
07) HTTP: Request Kernel, initramfs, DNBD-Configuration
08) HTTP: Send Kernel, initramfs, DNBD-Configuration
09) DNBD: Request Image Fragments
10) DNBD: Send Image Fragments
Infoblox
New Node
iPXE Boot Server

Figure 2: Boot Sequence including Boot Selection Server

In its current form, the BSS has two configuration files: one to define projects and their script template to deliver to matching clients and another to assign MAC addresses and/or UUIDs to projects. Changing a machine's boot behaviour or configuring VLAN within iPXE becomes as easy as editing the relevant configuration files. Work is in progress to develop a web frontend and an API to allow convenient access to the configuration stored in a database for administrators from different projects.

However, since the BSS is the first step of the preboot process of every node, it represents a new single point of failure. Any availability issues of the BSS, from server or network segment outages, would result in a failure to boot and could potentially affect the whole installation. This can be mitigated in various ways. Taking advantage of the DNS support of iPXE, the initial script can chain to the BSS using its FQDN instead of an IP address, and retry in case of failure. Multiple BSS instances deployed in different network locations, coupled with DNS round robin, can then provide further protections against a single-host and network segment outages.

5 First Tests and Evaluation

A pragmatic approach to optimise services usage will be to deploy a unified operating model to partition the worker nodes into their respective service domains (cloud, HPC, classroom) for dedicated longer time periods, e. g. several months, and monitor their usage profiles. During an auditing phase, the partitions can be adjusted, taking resource usage and funding constraints into account. In between the auditing and adjustment points, load balancing between the service domains can be accomplished by various means. On the one hand, HPC services could be able to start additional worker nodes in the cloud and the cloud services could be able to start additional VM instances inside the HPC system (Mateescu et al., 2011; He et al., 2010; Gamel et al., 2017; Meier et al., 2017).[6] On the other hand, the BSS could be extended with an additional cross-domain monitoring service analysing workloads, scheduling information and automatically rebalancing the nodes' partitions when needed. In all cases, governance and funding issues need to be considered.

At the time of writing, only preliminary results can be reported. Applying the new provisioning concept was required only for a fraction of the infrastructures as the HPC-node booting was simply updated to the new scheme. The cloud setup followed with the production start of the bwCloud SCOPE infrastructure. The new infrastructure aided in the smooth migration of several nodes from the ATLAS Tier2/Tier3 environment into the NEMO environment. Additional machines acquired in the meantime were likewise easily integrated into the new environment. In general, the level of granularity with which nodes can be moved around is defined by the size of an Omni-Path island, or reasonable fractions thereof. It is non-trivial to migrate nodes on the single node level. Switchover between modes will most probably occur on a monthly or weekly rather than hourly basis, as draining (at least in HPC) takes time.

[6] This was explored in more depth in the Virtual Research Environments project ViCE (Meier et al., 2017; Bauer et al., 2019).

There were several factors which significantly improved the results of the initial tests. The VLAN configuration was already partially in place and only needed to be extended to a couple of additional switches. It was to our advantage to have research infrastructure plus ViCE and de.NBI cloud project staff bundled in a team, shortening communication paths and reusing previously established concepts and technologies. Additionally, tight cooperation and coordination with the network department within the computer centre helped. After a couple of weeks of operation, we are optimistic that we will be able to operate a significantly larger number of machines with fewer people.

5.1 Security Considerations

System and network security is a concern as large scale computational infrastructures with high bandwidth uplinks are always a target worth attacking, either to consume compute power, to launch network attacks, or to generate massive DoS packet floods. Independent of the actual developments of a unified system and operating model, the individual infrastructures were already exposed to the Internet to a certain degree. Compared to the moderately sized user base of HPC clusters, the number of cloud users is significantly higher.[7] On the physical side, there will be an increased »mixture« of nodes within one chassis or rack and on the switches present. Bare-metal users with access to network interface configuration might discover additional networks visible to them. In normal operations, cloud users never have access to the hypervisor's network interface level.

We have identified several risk mitigation strategies. The bundling of resources unfortunately prevents the duplication of previous firewalling strategies. The distribution of VLANs, however, is limited to a well-defined section of the physical network. VLANs alone are not sufficient for network segregation in our case, though. Configuring every VLAN on every switch ports exposes the network to VLAN hopping attacks. These can be averted by the deconfiguration of unneeded VLANs from switch ports using Software Defined Network (SDN) strategies (Fang et al., 2012). Many switches offer an API for automated port configuration: the BSS could access these to reconfigure VLANs, depending on the corresponding nodes' current operating mode. In parallel, improved monitoring could help to detect unwanted network

[7] Both the bwHPC and the bwCloud SCOPE projects cater to users from both Freiburg University and from within the state of Baden-Württemberg.

activity. Even in a secured computer centre environment, rogue DHCP servers or man-in-the-middle attacks during the download of the iPXE binary could become an issue. Trusted computing technologies like TPM might become an answer to these threats by verifying the integrity of the various components downloaded in the early phase of remote boot (iPXE binaries, kernel, initramfs and configuration files) thus ensuring an untampered boot. While ubiquitous for business like PCs or laptops, TPM chips are only available for certain server platforms but are not widely installed yet.

6 Outlook

The increased complexity of scientific workflows, the rising demands of researchers on compute power, and the sheer amount of servers to monitor and administer demand for new operational models. Optimally, such models help to apply proven business models for efficient hardware utilization. Business models in the public sector for HPC and cloud research operations have to be different from commercial ones. Our approach spans the dichotomy of cloud and cluster, and gains flexibility which is otherwise not achievable in strongly isolated setups. On the one hand, the effort to install automated switching of node modes (i. e. from HPC to cloud) is not trivial in the first place and needs an extra management layer to be deployed. On the other hand, the joint view and responsibilities of the administration team encourages joint issue management and may trigger new concepts and ideas. The gain in flexibility of the installation and the usage of the infrastructure allows for a better allocation of freshly gained funds for further improvement or rolling-updates of the hardware base. Intelligent rededication or reconfiguration of machines for optimal use benefits the whole scientific computing community.

This new form of procurement allows better utilization of resources, but definitely raises discussions on the feasibility within existing funding schemes and frameworks: the flexible reconfiguration based on load is surely a selling point, but it has a big political constraint. Funding agencies might not accept that resources are used for competing projects even if resources are traded back in a later period, i. e. in form of CPU hours. The ongoing discussion calls for an adapted shareholder model –

financial contributions are to be translated into CPU or memory hours available within the whole system, depreciating unused CPU hours over time.[8]

Several BSS extensions are planned for future development. An ongoing student's master team project at the University of Freiburg is analysing the requirements for multi-tenancy concepts, time- and location-based event mechanisms (e. g. for periodic hardware tests) and workflows for automatic registration of unknown clients to DHCP servers and inventory management systems. Closely related is a master's thesis focusing on securing the preboot phase with TPM and Secure Boot to provide an initial trust anchor in remote boot scenarios and, in coordination with the other project, how to handle the initial TPM configuration within the BSS' client registration process.[9] Moreover, we want to analyse the technical viability of automatically rebalancing nodes partitions for HPC and cloud.

Having a concept for boot selection and flexible resource provisioning in place, the presented approach could get extended to PC pool operation.[10] The cluster nodes were »well-provisioned« with enterprise class components like 10 GbE. This made our considerations easy. Additional costs of extra hardware without direct primary need to fulfil the requirements for the approach must pay off. The same applies to the extra efforts of draining and time spent on rebooting. This process – at least from today's perspective – will remain a manual process to a certain degree.

Acknowledgements

The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 39/963-1 FUGG (bwForCluster NEMO).

The bwCloud SCOPE resources and the ViCE projects are funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg.

The de.NBI resources and ATLAS resources were co-funded by the DFG and the BMBF.

[8] Extension of the cluster fair-share model as discussed in (Wiebelt et al., 2016).

[9] TPM configuration requires some tools which could be provided by a special registration boot image.

[10] For many administrators an obvious use case is the deployment of PC pools to compute tasks during off-peaks.

Corresponding Author

Jonathan Bauer: jonathan.bauer@rz.uni-freiburg.de
eScience Department, Computer Center, University of Freiburg
Hermann-Herder-Str. 10, 79104 Freiburg, Germany

ORCID

Jonathan Bauer https://orcid.org/0000-0002-5624-2055
Michael Janczyk https://orcid.org/0000-0003-4886-736X
Dirk von Suchodoletz https://orcid.org/0000-0002-4382-5104
Bernd Wiebelt https://orcid.org/0000-0003-2771-4524
Helena Rasche https://orcid.org/0000-0001-9760-8992

References

Anvin, H. P. and M. Connor (2008). »x86 Network Booting: Integrating gPXE and PXELINUX«. In: *Linux Symposium.* Citeseer, pp. 9–18.

Bauer, J., D. von Suchodoletz, J. Vollmer and H. Rasche (2019). »Game of Templates. Deploying and (re-)using Virtualized Research Environments in High-Performance and High-Throughput Computing«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 245–262. DOI: 10.15496/publikation-29057.

Fang, S., Y. Yu, C. H. Foh and K. M. M. Aung (2012). »A loss-free multipathing solution for data center network using software-defined networking approach«. In: *APMRC, 2012 Digest.* IEEE, pp. 1–8.

Gamel, A. J., U. Schnoor, K. Meier, F. Bührer and M. Schumacher (2017). *Virtualization of the ATLAS software environment on a shared HPC system.* Tech. rep. ATL-SOFT-PROC-2017-070. Geneva: CERN. URL: https://cds.cern.ch/record/2292920.

He, Q., S. Zhou, B. Kobler, D. Duffy and T. McGlynn (2010). »Case study for running HPC applications in public clouds«. In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing.* ACM, pp. 395–401.

Jeanvoine, E., L. Sarzyniec and L. Nussbaum (2012). *Kadeploy3: Efficient and Scalable Operating System Provisioning for HPC Clusters.* Research Report RR-8002. INRIA. URL: https://hal.inria.fr/hal-00710638.

Lehrbach, J. et al. (2017). »ALICE HLT Cluster operation during ALICE Run 2«. In: *Journal of Physics: Conference Series.* Vol. 898. 8. IOP Publishing, p. 082027.

Mateescu, G., W. Gentzsch and C. J. Ribbens (2011). »Hybrid computing—where HPC meets grid and cloud computing«. In: *Future Generation Computer Systems* 27.5, pp. 440–453.

Meier, K., B. Grüning, C. Blank, M. Janczyk and D. von Suchodoletz (2017). »Virtualisierte wissenschaftliche Forschungsumgebungen und die zukünftige Rolle der Rechenzentren«. In: *10. DFN-Forum Kommunikationstechnologien, 30.-31. Mai 2017, Berlin, Gesellschaft für Informatik eV (GI)*, pp. 145–154.

Rettberg, S., D. von Suchodoletz and J. Bauer (2019). »Feeding the Masses: DNBD3. Simple, efficient, redundant block device for large scale HPC, Cloud and PC pool installations«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 231–243. DOI: 10.15496/publikation-29056.

Schmelzer, S. et al. (2011). »Universal Remote Boot and Administration Service«. In: *7th Latin American Network Operations and Management Symposium (LANOMS 2011)*, pp. 42–47.

Stirenko, S., O. Zinenko and D. Gribenko (2013). »Dual-layer hardware and software management in cluster systems«. In: *Proc. Third Int. Conf. »High Performance Computing« HPC-UA*, pp. 380–385.

Trahasch, S., D. von Suchodoletz, J. Münchenberg, S. Rettberg and C. Rößler (2015). »bwLehrpool: Plattform für die effiziente Bereitstellung von Lehr- und Klausurumgebungen«. In: *DeLFI 2015 - Die 13. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI), München, 1.-4. September 2015*, pp. 291–297. ISBN: 978-3-88579-641-1. URL: http://subs.emis.de/LNI/Proceedings/Proceedings247/article14.html.

Wiebelt, B. et al. (2016). »Strukturvorschlag für eine bwHPC-Governance der ENM-Community«. In: *Kooperation von Rechenzentren Governance und Steuerung – Organisation, Rechtsgrundlagen, Politik.* Ed. by D. von Suchodoletz, J. C. Schulz, J. Leendertse, H. Hotzel and M. Wimmer. de Gruyter, pp. 343–354. ISBN: 978-3-11-045888-6. DOI: 10.1515/9783110459753-029.

Feeding the Masses: DNBD3

Simple, efficient, redundant block device for large scale HPC, Cloud and PC pool installations

Simon Rettberg Dirk von Suchodoletz Jonathan Bauer

eScience Department, Computer Center, University of Freiburg, Freiburg, Germany

In computer center operations many sites operate large PC lecture pools or HPC clusters which can require similar or identical operating system images and software packages. Booting over the LAN allows instantaneously usable systems but requires the efficient provisioning of the root file system. Traditionally, general purpose file systems like NFS are used, but read-only Network Block Devices like the presented DNBD3 provide a range of attractive features, which can outperform alternatives across a range of situations. DNBD3 not only allows for caching and proxying at various levels, but it comes with a built-in performance monitor, versioning, and failover functionality. DNBD3 has been under development at Freiburg University for the past few years. It is released under a GPLv2 license, and consists of a Linux kernel module for the clients, and a user space executable for the servers. It is running in production for two highly heterogeneous use cases: as a distributed setup of campus-wide computer pools with more than 400 connected machines, and in the 1000+ node compute cluster backing the Freiburg HPC and Clouds. Aggressive local caching might even allow the use of mobile clients on WLAN infrastructures in stateless Linux operation.

1 Motivation

Management of large scale desktop computer pools, HPC clusters, and private clouds all usually incur high administrative costs or operational expenses that can easily surpass the figures for capital expenditures such as hard- and software purchases. One strategy to reduce the Total Cost of Ownership (TCO) is to employ

doi: 10.15496/publikation-29056

the use of remote boot options utilizing pre-existing high bandwidth local networks. While per-GB cost of centralized reliable storage is usually higher, maintaining such a system often requires less manual work, especially if the storage is shared with other projects and scaled accordingly.[1]

Exporting the root file system (rootfs) of many clients via NFS has been shown to be an inefficient solution due to significant file system overhead and serialized round trips for access operations. While iPXE initial network boot is very efficient (Bauer, Messner et al., 2019), highly concurrent access to a central NFS server from many machines booting simultaneously can result in bandwidth bottlenecks resulting in poor end-user experience (Stirenko et al., 2013).

Both the ability to provide failover in the case of a server outage and to provide load-balancing is a requirement in large scale setups of 100+ machines. With NFS versions prior to 4.0, providing NFS server redundancy was cumbersome[2] and active load-balancing impossible. This improved marginally in version 4.1, which includes Parallel NFS. More promising results were produced with Network Block Devices (NBDs): During initial development, only a compressed squashfs file system image was distributed by NBD (Schmelzer et al., 2014). Since NBDs work only at the block level, they have significantly less overhead compared to NFS.

Both in PC lecture pools and compute clusters (HPC, Clouds), the machines share a broad software base, which is ideally highly redundant between the use cases for operator convenience. Stateless operation, and thus easy exchange of clients between operating roles, is significantly more important than the individualization of single clients.

2 Previous experiments and alternate developments

Block devices besides traditional local disks and optical storage started to emerge by the end of the 1990s in the Linux kernel.[3] After the original NBD, derived variants like GNBD or DRBD emerged (Kim et al., 2001). iSCSI could be seen as

[1] In case of computer pools, compare swapping a disk in your redundant central storage vs. locating a workstation somewhere on campus and replacing its only disk.

[2] NFSv3 is stateless, so it allowed one server in »hot standby« to detect another server's disappearance and simply take over its IP address. However, this is rather abusing the statelessness of NFSv3 than an elegant integrated solution, often resulting in client-side hangs approaching 10+ seconds.

[3] The Linux Network Block Device was developed in 1997 by Pavel Machek and first included into the standard Linux kernel with version 2.1.101

falling into a similar category but is not made for highly concurrent access.[4] As a result of iSCSI being implemented largely by wrapping SCSI commands in TCP/IP, it is comparatively inefficient due to the additional overhead. This led to various extensions to solve the underlying issue, such as multipath routing, which uses several TCP/IP connections to parallelize operations against multiple endpoints, or iSER, which can greatly speed up bulk data transfers by leveraging RDMA on suitable transports, like InfiniBand. Since iSCSI supports read and write operations for disk targets, the implementation is more complex than required for the use case envisioned. Our goal was to fully saturate a traditional 10GbE link without requiring excessive resources or special hardware on either end, apart from the network card and fast storage or enough RAM to deliver data at that speed.

DRBD, like iSCSI, implements a read/write block device, but it is focused on live replication of written data amongst secondary DRBD hosts. It focuses on providing quick failover of a service to another host rather than providing redundancy to a multitude of clients, making it unsuitable for the use case. Previous projects with similar goals (ENBD, ANBD, GNBD) appear to be dead and are no longer maintained.

The first iteration of the Distributed Network Block Device (DNBD1) implemented multicast-like features to allow access to a read-only NBD over shared media such as WLANs.[5] To cope with the intrinsic communications overhead of wireless networks, DNBD1 aimed to avoid sending out the same data over and over again to every single client. This was achieved by connecting wireless clients and servers in a multicast network on the IP layer. DNBD1 clients take advantage of this overhead by pre-caching data requested by other DNBD1 clients. In ideal situations, all redundant requests can be prevented, greatly reducing bandwidth consumption on the shared medium. The development of the superseding versions of DNBD2 and DNBD3 catered to the fact that the wireless future didn't arrive as quickly as envisioned and brought replication and scalability features to the standard unicast LAN environment.[6]

The latest version of the Distributed Network Block Device, DNBD3, exclusively focuses on providing a redundant and fast, read-only block device which greatly

[4] Setting a read-only target allows multiple clients to connect, though.

[5] Since in 2006, it was assumed that soon everybody would exclusively use laptops or similar mobile devices.

[6] See `https://lab.openslx.org/attachments/download/19/ba-dnbd2-dileo.pdf` (visited on 10. 01. 2019).

reduces implementation and operational complexity, especially on the server side. The kernel module compiles with all major Linux kernels. The server is a user land executable written in C, mostly compatible to standard POSIX environments.[7] DNBD3 is a complete rewrite of previous iterations, the main objective of which is to combine the advantages of the standard NBD with caching, versioning, and wide area distribution features.[8] The optimal utilization of the available bandwidth is achieved through strong parallelization and efficient handling of I/O intensive operations within the Linux kernel. Resilience is realized through deploying redundant servers and allowing clients to distribute across the available servers. The clients independently search for the fastest server and respond dynamically to bottlenecks, ensuring the best possible performance even when used in wide area networks.

3 The DNBD3 architecture

Since DNBD3 was designed to suit the use case of many read-only clients, a great deal of complexity supporting distributed read-write access could be completely omitted. Emphasis was instead laid on supporting quick failover in case of connectivity issues, and load-balancing in case of apparent network congestion (Figure 1). Since the operational requirements for this project included not only providing the root file system to workstations but ensuring quick response times, i. e. no multiple second long client hangs while reconnecting to another server, it was made of utmost importance that the protocol incurs as little overhead as possible, especially regarding connection establishment.

3.1 Server and client setup

A typical setup involves a primary server, which exports one or more images from a configurable file system location. Images are addressed by their relative path within the exported directory. Additional DNBD3 servers function as proxies, pointed at the primary server, initially not storing any images or data. In this setup, a client can connect to any of these servers and request an image. When a client requests an image from a proxy that the proxy server does not know about, it will transparently

[7] Currently tested on Linux and FreeBSD, could be rather easily ported to similar systems, too, given they have a comparable mechanism to sendfile(2).

[8] The source is available at `https://git.openslx.org/dnbd3.git/` (visited on 10. 01. 2019).

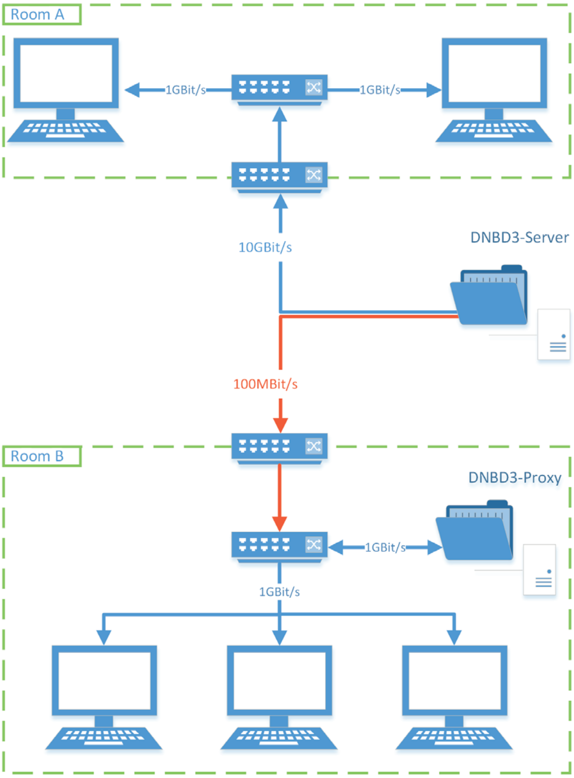


Figure 1: Distributed cache and proxy cascade for performance to deal with low bandwidth links.

forward the requests to the primary server, relaying all replies to the client. Simultaneously, it will cache those replies locally for future requests. Proxy servers will automatically evict the least recently used image from the cache when the proxies run out of space.

To (dis)connect a DNBD3 client device, the user space binary `dnbd3-client` is executed. It can be passed one or more server addresses to work with, and requires the desired image name and revision number, which are then passed to the kernel module via `ioctl`. Once the kernel module has established a connection to a server, it will request a list of suggested servers[9] to use for failover and load balancing. This means the server list used by the client can be pre-populated when initializing the client (e. g. embedded in initramfs and randomized on each boot) and optionally

[9] From the client's viewpoint servers and proxies are functionally the same.

extended later on by querying servers for additional addresses. During operation, DNBD3 does not do any additional caching on the client, but merely acts as a plain old block device. Since the general use case means ultimately mounting a file system from the device, caching will eventually happen higher up the stack, using the global page cache.

3.2 Load balancing and failover

Load balancing is kept fairly simple in DNBD3; each client individually measures and decides which of the available servers to use for requesting blocks. While it is tempting to design an algorithm where the involved servers coordinate distributing clients among them according to their NIC speed, this approach falls short as soon as the network infrastructure between servers and clients is any more complex than one or two switches. Individual clients need to accurately track bottlenecks that only affect clients in certain subnets for globally optimal performance. Simply distributing clients evenly across servers will certainly not be sufficient, especially under fluctuating network loads.

Instead of going down a rabbit hole of designing a highly sophisticated algorithm to master this task, we opted for clients to periodically measure the speed of every known server. If another server's response time is lower by 33% compared to the current server, the client will make a switch. To account for occasional spikes, the average of four such measurements is used when comparing servers.

A server's speed is determined by establishing a connection, requesting the image currently in use by the client, and reading the first 4096 bytes of that image.[10] This ensures enough network round trips to produce a meaningful result without consuming too much bandwidth. Requesting the currently used image from the server instead of having a synthetic benchmark feature in the protocol has two advantages: first, this ensures that the server being measured is actually able to serve the image in question, and second, since the full handshake has already happened after the measurement is done, there is zero networking overhead when deciding to actually switch over to that server, other than having to re-queue any in-flight requests that the old server didn't reply to yet. Switching to another server merely requires closing the current connection and then re-initializing the sending and

[10] Those first bytes will almost certainly reside in the server's cache in memory and thus omit the delays of actual disk reads, but it is generally a good enough proxy for the requirements of the protocol.

receiving threads with the new socket descriptor already established during the measurement.

After startup, clients measure every four seconds for the first 30 seconds in order to quickly switch to another server, should the initially selected server be overloaded. If no switch occurs within 30 seconds, the current server is assumed to be a good choice, and clients lower measurements to every 20 seconds. This approach, combined with the averaging over four measurements, results in a pretty stable distribution of clients among servers in both, the comparably spread out topology of bwLehrpool, and the much larger but more homogeneous network infrastructure of the HPC/NEMO setup.

3.3 sysfs interface

DNBD3 exports some simple statistics via sysfs: the currently connected server's address, the list of known servers, their measured response times, the current image, and its revision number.

```
# ll /sys/block/dnbd0/net/
total 0
drwxr-xr-x 2 root root    0 Jun 27 14:38 ./
drwxr-xr-x 9 root root    0 Jun 27 08:38 ../
-r--r--r-- 1 root root 4096 Jun 27 14:38 alt_server_num
-r--r--r-- 1 root root 4096 Jun 27 14:38 alt_servers
-r--r--r-- 1 root root 4096 Jun 27 14:38 cur_server_addr
-r--r--r-- 1 root root 4096 Jun 27 14:38 cur_server_rtt
-r--r--r-- 1 root root 4096 Jun 27 14:38 image_name
-r--r--r-- 1 root root 4096 Jun 27 14:38 rid
```

Example output of known server list:

```
# cat /sys/block/dnbd0/net/alt_servers
10.4.128.240,5003,426,0
10.16.0.22,5004,252,0
10.8.8.88,5003,452,0
10.23.4.2,5003,1071,0
```

This shows the list of servers known to serve the image currently in use. The columns are IP address, port, response time in µs and a counter which tracks how many times this server was consecutively found inaccessible.

3.4 Versioning

Versioning is an optional feature available in DNBD3. It allows for multiple versions of the image files to co-exist on the server by appending .r[0-9]+ to the image name, representing the revision number of the image. Virtually, the image is exported without this suffix, and clients can indicate which revision of an image they want when connecting to a server. The revision number 0 has a special meaning in this context – it refers to the highest revision number. This way, a client already running can explicitly request the revision number of an image that it is currently running on, while a freshly booted client can simply request revision 0 to get the latest version.

4 Client-side copy-on-write for r/w access

Creating a writable file system for the client OS on top of DNBD3 can be achieved in two ways. Up until recently, the source file system got packed with squashfs before it got exported as an image. On the client, the squashfs was mounted in read-only mode and then a tmpfs top layer was added via AUFS[11] to make the file system virtually writable. Squashfs offers great compression ratios and since copy-on-write (CoW) happens on the file system layer, the size of the rootfs is determined individually on the client side through the size of the tmpfs. Nevertheless, adding a single byte to a large file meant fully re-copying it into the tmpfs layer.

The second approach is exporting a disk image in any container format, e. g. qcow2, and then providing a CoW backing on the client with qemu-nbd on the block level. It is also possible to use the Linux kernel device mapper to create a writable layer for the image, but this requires exporting a raw image file which has all the unused space preallocated. The advantage of this approach is that it is possible to directly export the disk image of a virtual machine without the tedious squashfs packing process. Unfortunately, since this operates on the block layer, the size of the root file system will be predetermined during image creation, so using a

[11] Special unioning file system which never made it into the kernel because of significant complexities and resulting issues. Compared to OverlayFS (which wasn't part of the mainline kernel when bwLehrpool started), it offers much more flexibility, e. g. adding/removing layers after mounting.

small virtual disk during image creation would make it possible to run out of disk space on the client, even if plenty of space were available for the CoW file.[12]

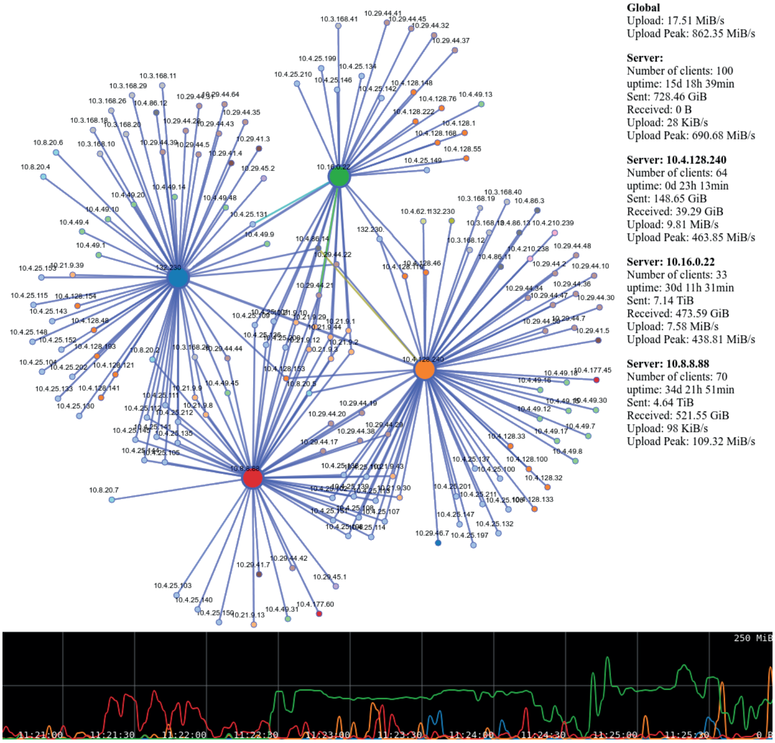
5 Practical experience

DNBD3 has been powering the bwLehrpool infrastructure since late 2013 (Figure 2). bwLehrpool (Suchodoletz et al., 2014) is a stateless Linux remote boot project offering various hypervisors and containerization on top to host »user created content«. It enables the flexible and efficient deployment of virtual teaching and laboratory software environments in computer lecture rooms. bwLehrpool offers instructors at cooperating educational institutions the possibility to quickly, simply, and independently create teaching environments for a wide range of courses. In bwLehrpool both the base operating system, as explained above, and, additionally, the various VM images are provided through DNBD3. Since it is desirable that the environment looks exactly the same every time the students start the VM complementing a lecture, any changes that happen on the block layer are temporarily written to a copy-on-write backing file on the client and discarded when the individual session ends.

Using DNBD3 for providing disk images to clients enables risk-free updating of DNBD3 servers and proxies without any service interruption. It even makes it possible to experiment with different server features and configurations, with the advantage of always being able to test against a real world workload.

While the virtualizers' own mechanisms are still used to realize the CoW layer for VM images, a writable layer was required for the rootfs images provisioning bare-metal machines. Since AUFS was never officially supported by the Linux kernel, it was cumbersome to patch its code to compile against newer kernel versions. The qcow2 format proved to be a good alternative to AUFS to implement CoW directly on the block layer, even though this relies on user-space tools to handle the format. Recent experiments with the device mapper framework were promising and have inspired us to work towards realizing the CoW layer entirely in the kernel-space.

[12] Actually, the block device can be enlarged when creating the CoW layer, and given a suitable partitioning and file system choice inside the container, the file system of the rootfs could be extended on the client, but it is usually not worth the effort. Making the base image large enough is usually much simpler and sufficient.



Global
Upload: 17.51 MiB/s
Upload Peak: 862.35 MiB/s

Server:
Number of clients: 100
uptime: 15d 18h 39min
Sent: 728.46 GiB
Received: 0 B
Upload: 28 KiB/s
Upload Peak: 690.68 MiB/s

Server: 10.4.128.240
Number of clients: 64
uptime: 0d 23h 13min
Sent: 148.65 GiB
Received: 39.29 GiB
Upload: 9.81 MiB/s
Upload Peak: 463.85 MiB/s

Server: 10.16.0.22
Number of clients: 33
uptime: 30d 11h 31min
Sent: 7.14 TiB
Received: 473.59 GiB
Upload: 7.58 MiB/s
Upload Peak: 438.81 MiB/s

Server: 10.8.8.88
Number of clients: 70
uptime: 34d 21h 51min
Sent: 4.64 TiB
Received: 521.55 GiB
Upload: 98 KiB/s
Upload Peak: 109.32 MiB/s

Figure 2: Example of the bwLehrpool DNBD3 distribution structure. The four big nodes represent DNBD3 servers, small nodes are connected clients. Some clients are currently using more than one image, so they can be connected to multiple servers at the same time. Blue edges are idle links, green edges represent network traffic. The graph at the bottom shows aggregated egress traffic for all four servers over the last 20 minutes.

Another DNBD3 use case, in production since the start of the new HPC cluster in 2016, is the provisioning of the root file system of NEMO in Freiburg. Over 1000 nodes are booted simultaneously while two DNBD3 servers provide their root file system. During the boot process, the 10 GbE network interfaces of both servers are mostly saturated while generating minimal CPU load, achieving short boot-times and stable operation. Updates are run gradually without interrupting the cluster operation: Node are drained and rebooted into a new version of the rootfs. The old version is provided in parallel to the new one as long as nodes continue to request it.

Without delving into the details, both the bwLehrpool and NEMO rootfs images are generated with a combination of packer templates to install the base OS from installation ISOs and ansible playbooks to install and configure the project's respective software stack (Bauer, Suchodoletz et al., 2019). The resulting qcow2 images are exported via DNBD3 and prepared for network boot with the same set of tools. This approach has proved to be flexible enough to cater for the different use cases.

6 Conclusion and outlook

DNBD3 has been in production for several years and has gracefully survived multiple server failures without interruption of service for connected clients. Thus, DNBD3 is not just another network block device, but fills a gap in the Linux block device selection as it combines features which are not provided by pre-existing designs. In the short term, some adaptations to newer kernel versions were necessary[13]. Compatibility with new kernel releases is currently maintained by the bwLehrpool team. In the long term, it is planned to do major cleanups and checks whether it still conforms to modern block device designs, which should eventually make it possible to suggest DNBD3 for upstreaming into the official kernel.

To allow a simpler and more efficient updating of user provided base and VM images, the use of a local CoW image on the client will be explored for seamless updating. So, aside from simply discarding the snapshots for stateless operations, the resulting diff-file from the CoW layer could be further processed to handle incremental updates on base images, both for server-side image versioning and even to implement client-side persistence. This would allow copying only the different blocks instead of the entire image whenever changes should be made to the served image. To allow DNBD3 operation in low or fluctuating bandwidth environments, a client-side disk caching feature will be explored in further experiments in order to enable bwLehrpool operation in mobile setups.

Acknowledgement

The work presented in this publication was part of the bwLehrpool project sponsored by the Ministry of Science, Research and the Arts Baden-Württemberg, Germany. NEMO is part of the state-wide HPC research infrastructure, which is supported

[13] As of writing, DNBD3 is known to work on all kernel versions from 3.0 to 4.19.

by both the DFG and the state of Baden-Württemberg. The support is gratefully acknowledged.

Corresponding Author

Simon Rettberg: simon.rettberg@rz.uni-freiburg.de
eScience Department, Computer Center, University of Freiburg
Hermann-Herder-Str. 10, 79104 Freiburg, Germany

ORCID

Dirk von Suchodoletz https://orcid.org/0000-0002-4382-5104
Jonathan Bauer https://orcid.org/0000-0002-5624-2055

References

Bauer, J., M. Messner et al. (2019). »A Sorting Hat For Clusters. Dynamic Provisioning of Compute Nodes for Colocated Large Scale Computational Research Infrastructures«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 217–229. DOI: 10.15496/publikation-29055.

Bauer, J., D. von Suchodoletz, J. Vollmer and H. Rasche (2019). »Game of Templates. Deploying and (re-)using Virtualized Research Environments in High-Performance and High-Throughput Computing«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 245–262. DOI: 10.15496/publikation-29057.

Kim, K., J.-S. Kim and S.-I. Jung (2001). »GNBD/VIA: a network block device over virtual interface architecture on Linux«. In: *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM.* IEEE.

Schmelzer, S., D. von Suchodoletz, M. Janczyk and G. Schneider (2014). »Flexible Cluster Node Provisioning in a Distributed Environment«. German. In: *Hochleistungsrechnen in Baden-Württemberg. Ausgewählte Aktivitäten im bwGRiD 2012.* Beiträge zu Anwenderprojekten und Infrastruktur im bwGRiD im Jahr 2012. Ed. by J. C. Schulz and S. Hermann. KIT Scientific Publishing, Karlsruhe, pp. 203–219. ISBN: 978-3-7315-0196-1. DOI: 10.5445/KSP/1000039516. URN: urn:nbn:de:0072-395167.

Stirenko, S., O. Zinenko and D. Gribenko (2013). »Dual-layer hardware and software management in cluster systems«. In: *Proc. Third Int. Conf. »High Performance Computing« HPC-UA*, pp. 380–385.

Suchodoletz, D. von et al. (2014). »bwLehrpool – ein landesweiter Dienst für die Bereitstellung von PC-Pools in virtualisierter Umgebung für Lehre und Forschung«. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 37.1, pp. 33–40. DOI: 10.1515/pik-2013-0046.

doi: 10.15496/publikation-29062

Game of Templates

Deploying and (re-)using Virtualized Research Environments in High-Performance and High-Throughput Computing

Jonathan Bauer[*] Dirk von Suchodoletz[*] Jeannette Vollmer[*]
Helena Rasche[†]

[*]eScience Department, Computer Center, University of Freiburg, Freiburg, Germany
[†]Department of Bioinformatics, University of Freiburg, Germany

The Virtual Open Science Collaboration Environment project worked on different use cases to evaluate the necessary steps for virtualization or containerization especially when considering the external dependencies of digital workflows. Virtualized Research Environments (VRE) can both help to broaden the user base of an HPC cluster like NEMO and offer new forms of packaging scientific workflows as well as managing software stacks. The eResearch initiative on VREs sponsored by the state of Baden-Württemberg provided the necessary framework for both the researchers of various disciplines as well as the providers of (large-scale) compute infrastructures to define future operational models of HPC clusters and scientific clouds. In daily operations, VREs running on virtualization or containerization technologies such as OpenStack or Singularity help to disentangle the responsibilities regarding the software stacks needed to fulfill a certain task. Nevertheless, the reproduction of VREs as well as the provisioning of research data to be computed and stored afterward creates a couple of challenges which need to be solved beyond the traditional scientific computing models.

1 Motivation

The exponential growth of computational power in the past decades has greatly contributed to scientific advances in all fields. One of the key success strategies in science is to recognize recurring patterns and exploit them via templates. First, find out which part of a problem is static or invariant – this becomes the template. Then iterate over the variable part of the problem to search for the solution.

The development of hardware virtualization for the x86 platform in the last two decades and the cloud revolution also triggered a paradigm shift for university computer centers. The way IT resources are provided and which services should accompany them is changing. The ubiquitous use of digitalized workflows and the Fourth Paradigm in science demand an ever-increasing amount and variety of IT-based research infrastructures. To avoid handing over sizeable proportions of infrastructure-providing activities to the commercial domain – for reasons ranging from privacy and security to expertise considerations – computer centers have to find new ways to offer a significant range of infrastructures in an efficient way. It should provide comparable offerings regarding features and pricing[1] as well as to avoid overextending existing personnel resources when scaling up. Demands for hardware often come up on short notice and for project periods well below the cost-amortization period of five to six years that is typical for digital equipment. Having decentralized and often duplicated personnel to select, procure and operate all the various research infrastructure components is expensive.

Further challenges of university computer centers and faculty IT units are rooted in the very diversity of scientific communities and their broad set of demands with respect to software, tools or scientific workflows. This creates varied and often contradicting demands regarding software environments. Facilitating virtualization can help to separate the different requirements. As many resources in research infrastructures are underutilized for certain time periods, tapping into cloud strategies can help to significantly save on investment and hardware resources. A welcomed by-product would be savings on rackspace and energy.

2 Project objectives and related work

The Virtual Open Science Collaboration Environment (ViCE) project – sponsored by the state of Baden-Württemberg under the umbrella of the eScience initiative – brought together researchers from various science domains and infrastructure providers (computer centers from different universities). Its goal was to facilitate the

[1] The term »pricing« is used in a wider sense here, as it is necessary to consider different models in basic free services, cost recovery or extension of infrastructure by bringing in project money.

exchange of ideas, concentrating on the separation of the responsibilities of infrastructure providers from core scientific tasks and vice versa. Virtualized Research Environments (VRE) are a core concept to achieve a separation of tasks while preserving flexibility on the sides of both the users and providers.

Research projects should enjoy a quick start without tedious workflows to procure and set up the necessary IT infrastructure. Especially compute resources need to scale up and down following the demands of the individual project progress. At the same time, students and research assistants need to be integrated efficiently into research workflows. Virtual Machines (VM) can help by allowing prepared software environments to be copied, avoiding setting up the complete hardware, operating system, and application stack including configuration. Additionally, individual researchers and workgroups should gain more flexibility to set up their own derived versions of research environments and workflows.

The ViCE project aimed to loosen the originally tight connection matrix between research, administration, hardware, and software. A less-static environment invalidates a couple of traditional assumptions: NFS IP-based authentication becomes less an option with the dependence on local authentication frameworks. If VMs are really moved around to be used on different platforms and on different sites, deployment workflows have to be adapted. This will provide new means for experimenting and exchanging ideas and (complete) digital workflow environments.

The term Virtual(ized) Research Environment appears in the context of eResearch and eResearch infrastructures. VREs were introduced to foster cooperation in distributed projects and the exchange of complete software stacks. The shared resources not only mean shared data but additionally shared scientific workflows. Different VREs may focus on different aspects like versioning or large scale distribution and may come in different forms of representation. The author of (Allan, 2009) gives a wide-ranging definition of VREs in various forms by describing them in terms of intended capabilities.[2] Many projects saw VREs focused on web-based access to resources, though. »myExperiment« (De Roure et al., 2009) is a large public repository of scientific workflows created e. g. with Taverna (Oinn et al., 2006) or Galaxy (Afgan et al., 2018). It offers a collaborative environment where researchers can share publish and cite published scientific workflows. Workflows can be packed with digital objects to be swapped, sorted and searched. Taverna and Galaxy are pro-

[2] See p. 11f. VRE description includes different modes of operation from desktop to servers, talks of means of accessibility and usability, workflows and the focus on Open Source.

jects that enable users to graphically compose bioinformatics (and other) workflows exploiting several web services and tools spread all over the world. It allows the design, development and execution of scientific workflows over an existing compute and storage (science grid) infrastructure. The workflows are written in SCUFL and have a particular XML schema. The FreeFluo workflow engine, which is coupled with Taverna, manages the execution of the workflows. While Taverna and Galaxy focus on high-level workflows and existing execution environments, VREs in the context of ViCE focus on the software environment powering the workflows (Meier et al., 2017). The approach of using a VM consisting of the full software stack is suitable to allow a wide range of VRE variants of different disciplines, as workflows can be represented by more than just web-enabled tools.

VREs are packaged software tools for data analysis and computing which together with the research data processed through them represent the complete scientific workflows. To make them findable like data sets, a repository and registry are needed. These could be used as well for versioning or for exchange of VREs among researchers.

To make VREs more common and to attract a wider user base, different hosting platforms should be enabled to accept VREs. For various purposes in research and teaching, those platforms range from the desktop environment used in preparation of workflows and for interactive teaching, to HPC and cloud infrastructures. The relevant platforms used throughout the project were primarily the bwForCluster NEMO, the bwCloud and bwLehrpool.[3] Thus, various container and virtualization technologies such as Singularity, Docker or OpenStack were enabled and evaluated on the relevant science infrastructure platforms.[4]

Further objectives of the project were to evaluate frameworks for the exchange of VREs between the computer centers of the state universities. It should allow joint teams to develop, test and deploy VREs of various forms and provide necessary provisioning workflows. Further, measures to provide secure computing environments for sensitive data were to be explored. Additionally, the findings of the project were to be communicated by various workshops and trainings.

[3] NEMO (`http://www.hpc.uni-freiburg.de/nemo`), bwCloud (`https://www.bw-cloud.org`) and bwLehrpool (`https://www.bwlehrpool.de`) are federated state-wide infrastructure projects co-financed by the Ministry of Science, Research and the Arts, Baden-Württemberg in different configuration of partners and scope (visited on 05. 01. 2019).

[4] See `https://singularity.lbl.gov`, `https://www.docker.com` and `https://www.openstack.org/` (visited on 11. 01. 2019).

3 Steps to create a VRE

Building on the purpose and findings from previous projects, VREs can support highly differentiated goals, from virtualizing environments which were meant to be installed on compute clusters to packaging certain tools to be accessed via web services. They are usually tailored to the scientific application and can take the form of templates which are adapted by single researchers or cloned for massive parallelization. VREs can help to manage complexity by splitting workflow steps into distinct machines with clearly defined purposes which can later be chained or re-used in different workflows. The complexity of steps to complete will vary depending on the purpose of the VRE:

1. Audience: Depending on the intended purpose, VREs are meant to be created and run per scientist, per scientific workgroup, or even per scientific field.[5] Options of authentication depend on the actual users to log-on to the machine, if any. The computation of sensitive data e. g. person-related information might be restricted on certain infrastructures as these might not comply with all requirements.
2. Define the amount of resources: The requested resources for scientific workflows might differ significantly and may require grid systems, resource brokers, portals, knowledge systems or (large) data collections to be included.
3. Technical environment: The origin technical platforms may include desktop, cloud or HPC resources defined by the intended use e. g. for tool and workflow development, teaching and learning or non-interactive massive computation. Access to special hardware resources might be required.
4. External dependencies of filesystems, identity management systems or, if required, of license servers and the like needs to be resolved.
5. Planning of setup and maintenance: Optimally, instances can be created by automatic procedures like Packer, Ansible, Puppet and similar or can be cloned from templates.[6] Long-running VREs might require updates and older versions may need to be stored for reproducibility.

[5] In the scope of the ViCE project different variants were evaluated.

[6] See `https://packer.io/`, `https://www.ansible.com/`, `https://puppet.com` (visited on 12. 01. 2019).

4 Experiments and findings

During the project several use cases were evaluated ranging from bioinformatics VREs in the form of Galaxy services, two different types of particle physics workflows (Meier, 2017; Bührer et al., 2018), and a range of unique VREs created for English language studies, economics, microsystems technology or neuroscience. The focus for the particle physics VREs rested on the complete reproduction of the well-defined software stack needed for the analyses run in the CME and ATLAS experiments based on the CERN VM, which is itself a Scientific Linux version 6. The VRE ensured the complete control over all software components including kernel and base libraries required for large-scale distributed experiments. Thus, a solution including full virtualization was required and containerization was not an option because of Linux kernel dependencies. The Galaxy VREs are meant to run bioinformatic workflow tools and are less dependent on the basic software layer. To use the same deployment workflows as for the particle physics environments, full virtualization was used in the beginning of the project. ViCE helped to enable containerization on HPC and the PC pool environments and make special container VMs available in the cloud. This allowed smooth migrations from one environment into the other and the same VRE could be demonstrated and interactively used in teaching lessons as well as non-interactively for mass computing as shown in Figure 1.

The ATLAS and CMS VREs as well as the workflow VMs are created for potentially massive parallelization and not meant to be accessed directly. Special users are created which run the relevant tools and offer the necessary interfaces to interact with them. The VREs are meant to be created or cloned and thrown away after use. For accounting purposes, they are assigned to the user who started them either via HPC job control or through the cloud API. The VREs created for the language studies was meant to be used by students and lecturers throughout the state of Baden-Württemberg and utilize Shibboleth authentication backed by the bwIDM federation.[7]

[7] Shibboleth based identity federation, `https://www.bwidm.de` (visited on 14. 01. 2019).

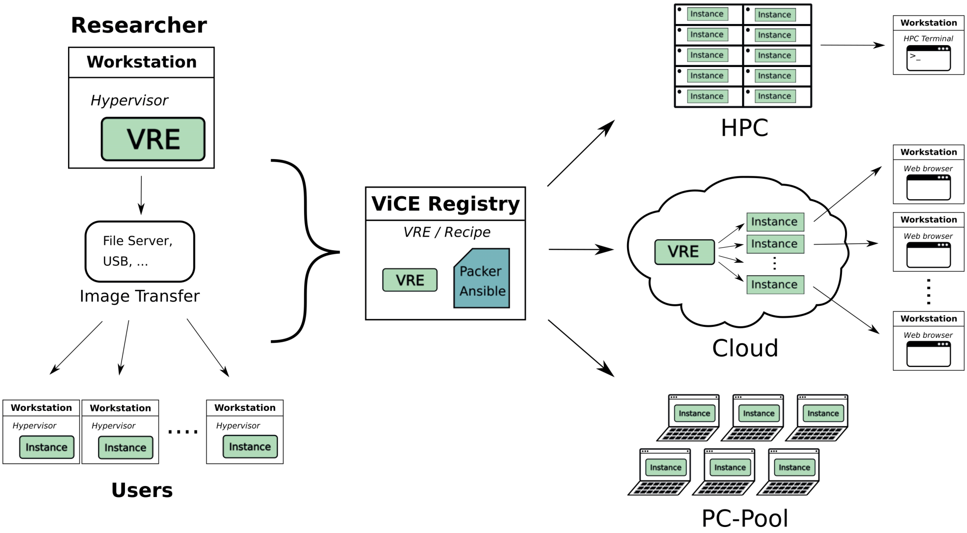


Figure 1: Reproducible VRE templates for flexible distribution.

4.1 Security considerations

The computation of sensitive data becomes even more a challenge in a VRE. The complete software stack including the hypervisor needs to be protected against compromise (Lombardi et al., 2011). While the topics of data ownership and quality of service are less of a concern in the compute environments considered during the project, confidentiality, integrity, data mobility and data protection remain significant challenges (Zissis et al., 2012; Shahzad, 2014). Encryption of data can provide a solution to secure storage when flexible access, scalability in key management and efficient user revocation are properly implemented (Li et al., 2013; Wang et al., 2012). But, necessary cryptographic operations lead to an additional complexity in cloud environments compared to traditional bare-metal environments. It is due to control of systems on which both the key management system and protected resources are located as well as difference in data owners and service providers (Chandramouli et al., 2014). Many of the challenges named require additional auditing and certificate infrastructure, rigorous processes by the infrastructure provider and (external) certification (Zissis et al., 2012).

To enforce data security and privacy different encryption techniques are deployed such as full disk encryption or fully homomorphic encryption. While the first encrypts the entire disk, the latter encrypts particular functions and is used to secure data from exploitation during computation (Zhao et al., 2014). The first option was implemented as a baseline measure by encrypting the storage holding either the virtual machine images in the bwCloud or local scratch space in bwLehrpool. The implementation of concepts like homomorphic encryption is much more expensive to set up and requires additional computation (Tari et al., 2015). It was not further considered as it lay well beyond the primary scope of the ViCE project. Up to now, trusted computation and storage can not be guaranteed as the requirements can only partly be met in the existing infrastructure provided by NEMO or bwCloud at the Freiburg site.

4.2 Technical environment

After defining the intended purpose of a VRE, the technical specifications must be set: The software environment representing a certain scientific workflow may require a specific system environment such as kernel and system libraries with clearly defined versions. While full system virtualization using hypervisors reproduces complete machines, containerization tools such as Docker or Singularity use concepts like namespaces to separate environments. The overhead of e. g. CPU and IO virtualization and thus the potential loss in performance of the former case is heavier. In the latter case, the software of the VRE runs directly on the host kernel and is thus dependent on its version and capabilities. Full virtualization abstracts core hardware components and peripherals. Special purpose hardware like GPGPUs, Infiniband or Omni-Path infrastructures are not easily virtualized and not easily available from inside the VRE. They cannot be trivially shared among VREs running on a single host system, although there exist a couple of ways to dedicate such resources to single VM instances. Nevertheless, a fully virtualized VRE is less dependent on the existence of hardware components and thus easier to share and move across different host systems. Further challenges arise for tasks such as (remote) visualization of data as envisioned for a microsystem technology VRE.

To allow the sharing of GPU resources within the NEMO HPC cluster, a Docker or Singularity container was created which allows direct access to the necessary hardware and to the parallel file system at the same time. PCI passthrough is one of the options to allow VMs to access hardware in the host system, but exclusively. Nevertheless, it can help to share a well-equipped GPU node among completely different users and their software environments. Up to now, the experiments with

Docker and Nvidia GPU demonstrated a couple of kernel and driver challenges as software versions need to be tightly matched in the host and Docker environments, reintroducing dependencies meant to be overcome by virtualization.

4.3 External dependencies

Software and infrastructural dependencies become explicit if deployed in a VRE. One of the resources a research or teaching project might need is storage. Often source and destination shares, e. g. home directories, or software module collections are mounted from a central resource and secured by defining IP ranges to which an export is allowed. If used in a VRE, especially on top of different resources at different sites or if meant to be shared among different colleagues in a distributed group, this option is no longer suitable. A similar problem arises from latencies if the shared resource is not available from within the hosting site but a couple of network hops away. Moreover, higher latencies with jitter usually hurt the performance of traditional file systems.

SDS@hd (Baumann et al., 2017) is one service offering storage for projects in Baden-Württemberg and was tested to see if it was useful as a solution to the dependency problem. SDS@hd offers storage statewide to scientists at public higher education institutions.[8] The service specifies that the storage is intended for data in active use, not for long term storage or backups. This means that it could be useful in exactly the cases outlined above – cloned or parallel projects requiring access to shared data or a space to save their results.

Conveniently, SDS@hd also offers an existing test project that can quickly and easily be connected to in order to determine if the service will work for a specific project or in the given infrastructure. For a productive use of this service an entitlement must be granted: first an entitlement by the institute, then a request for a specific amount of storage with justification must be submitted; after receiving provisional approval, a contract must be signed and submitted before the allocation can be approved. This process has to be completed once for every storage project, but once it is done the project owner can easily invite other users to the partition.

Once approved, the storage project could be accessed by various methods – SSHFS access was easy and instantly available using a password of one's own choosing; NFSv4 access required human interaction (providing personal data and information

[8] Subsidised by the university for researchers in Heidelberg, at a fee for external users.

regarding the machine that would be used to make the connection) in order to generate a keytab for access; SMB is also a connection option, but was not tested in the course of the ViCE Project. Having heard complaints of slow data transfers with SSHFS as a potential negative outweighing the ease of connection, several tests were run to compare performance. While initial results confirmed the assumption that NFS would be faster, further tests were run using different ciphers, resulting in comparable results using both connection types. Tests showed that from bwCloud to the SDS@hd storage project, NFS was able to handle writes faster than SSHFS, and the inverse was true for reads. Thus, a good understanding of the usage patterns for each project and some preparation at setup time can pay off in the longer term for a project with intensive reads or writes.

4.4 Setup and maintenance of VREs

VREs are intended to exist over a long period of time, and to allow for reproducibly running software within the environments. Provisioning different types of VREs with their respective software stack while providing a generic deployment process to make the resulting images VM or container compliant requires a structured workflow. As such, the long term development and maintenance can become a significant concern. In order to combat these issues, VREs should be well-defined environments, from the base image to any changes applied to them. Any solution identified should provide a flexible base infrastructure provisioning, allowing for easy adaptation to any future scientific workflows. The challenge is to standardize as much of the process as possible, minimizing the efforts required to realize various software environments.

Leveraging infrastructure-as-code is a solution to this problem; by defining VREs as a base image and a set of provisioning steps, managed in a git repository and output as a bootable machine image, the process of developing and deploying reproducible, re-usable infrastructure is significantly simplified.

The basic VM installation is handled by Packer using an appropriate source image (e. g. minimal core distribution ISO or a cloud image). Complete use-case-specific software stack installation is then performed using plug-in software provisioning tools (e. g. Ansible or Puppet) creating the different image variants which will be used by downstream compute infrastructure as system images.

The building of these images for bare-metal and cloud consumption is done automatically via a Jenkins[9] continuous integration server. When changes are pushed to their respective git repositories, builds are automatically triggered. The resultant bootable images are either self-contained in the cloud use case or in addition to the kernel and generated using Dracut[10] initramfs for the bare-metal use case, are then deployed to iPXE boot servers or directly uploaded to OpenStack, ready for deployment.

Ansible and Packer were explored to allow for easily building and re-building of VREs as requirements evolved and bugs were discovered over time. Packer was used to boot an image, run provisioning steps within the VM, and save the output as a new image. The choice of Packer provided easy flexibility for deployment of varied software stacks on top of the base VMs; it permitted a choice in virtualisation method (QEMU, VMware, numerous cloud providers), and a choice in provisioning method (shell scripts, Puppet, Ansible). Many pre-existing workflows leveraged Ansible playbooks to configure bare-metal nodes. Packer allowed for re-use of these existing provisioning workflows to directly create bootable machine images identical to their bare-metal counterparts.

A side benefit of this reproducible build of machine images is that, due to infrastructure existing as code in a git repository, it becomes possible to track the history and to reproduce old versions of images, which can be rebuilt and redeployed as needed.

The final result of this system is a clean division of labor where network boot and cluster administrators can each focus on their own tasks, independently of each other. With the automated image building and deployment process, we achieve the ability to rapidly produce VREs targeting the needs of specific users and communities with only small changes in configuration.

The base image template is utilized by various Ansible playbooks for different image flavors. The burden of VRE development is reduced by the playbooks which can be collaboratively used for basic tasks such as installing Singularity, updating packages, and managing services. Any future workflows that require additional software or services can be accommodated by forking the base image playbooks and by

[9] See `https://www.jenkins.io/` (visited on 01. 02. 2019).

[10] See `https://dracut.wiki.kernel.org/index.php/Main_Page` (visited on 05. 02. 2019) extended by a customized network boot module to enable stateless operations. Refer to (Schmelzer et al., 2014) for HPC stateless deployment in general.

implementing the new requirements. Any historical workflows can be reproduced by deploying an old image from the registry.

4.5 VRE registry

The experience with different scientific communities during the ViCE project has shown that software stacks or tool chains required for particular tasks are often similar across different workgroups within the same research field. As such, collaboration and exchange of VREs needs to be promoted to avoid individual groups creating similar VREs. Instead, the focus should be on sharing, improving and reusing existing VREs. To this end, scientists should have a way to search through available environments to find suitable VREs for their workflows. Finally, VREs should become part of Research Data Repositories (Pampel et al., 2013).

Gathering various categories of metadata is key to cataloging any kind of data and make it findable by others. From a user perspective, attributes like the specific research field, the operating system and the available software stack help identify relevant VREs for a particular purpose. From a technical perspective, minimal virtual hardware requirements, disk image or container format, are required to properly deploy them in various execution environments. Finally, from an operational perspective, management metadata like the researcher and the affiliated research project are useful for accounting and VRE life-cycle purposes. Before publication of scientific findings, archiving the employed software environment in order to reproduce the results is just as important as archiving the data sets utilized. This is essential for the long-term perspective of reproducible and citable research (Rechert et al., 2017).

To fill in these gaps, the idea of a central collaborative platform for users to manage, exchange and publish VREs was envisioned. A proof-of-concept implementation, the ViCE Registry, was quickly developed. As a first step, interfaces to OpenStack cloud infrastructures and PC pool infrastructure[11] were integrated and allowed users to import and export VREs from one system into the other and to search through those available in the registry (Hauser et al., 2017). While the prototypical implementation was promising at first, it was soon clear that the completion of all its features as well as its maintenance over time would not be sustainable in

[11] Orchestrated by the network booting bwLehrpool project (Suchodoletz et al., 2014).

the long-term. As such, it was discontinued and alternative concepts to realise an exchange and collaboration platform for VREs needed to be evaluated.

Some concepts in research data management, like the FAIR[12] principle, identified requirements similar to those of the ViCE registry (Wilkinson et al., 2016). From its early conception, the ViCE registry focused on those four requirements. Classifying metadata into organisational, cataloging and scientific attributes is essential for indexing purposes. Further, interoperability between computational research infrastructures allow researchers to access and reuse VREs in a diverse manner. Finally, reliable storage of the curated data and its metadata achieves its long-term preservation and accessibility. This led to a new ViCE registry concept melding into existing data management concepts, only focusing on the particular challenge of migrating VREs between different execution platforms.

A first evaluation of the data management software iRODS was promising (iRODS Consortium, 2017). Beyond the core storage functions and the large support of storage systems that enables custom hierarchical storage management, its ruling engine seems particularly attractive. Automatic gathering of organizational metadata on data ingestion depending on its origin, enforcing a set of metadata before publication of the data in a collaborative exchange area (facilitating the indexing thereof) or automation of VRE imports to and exports from computational research infrastructures are but a few examples of applications of iRODS rules (Rajasekar et al., 2010).

As an early experiment, the local PC pool's internal image exchange mechanism was adapted to use iRODS as a storage backend. Descriptive and technical metadata were inferred from bwLehrpool's internal data structure. Using iRODS metadata query language, third-party applications can search through the available images by criteria. For example, an OpenStack application using the iRODS API to search for VREs by technical metadata and then import compatible VREs into the cloud image repositories. Going a step further, the cloud image repository could use iRODS as a backend directly – then using a PC pool VRE in the cloud infrastructure would only require creation of a cloud image using image stored in iRODS, removing the need to actively transfer the image from one storage backend to another. These experiments were promising and confirmed the validity of an iRODS-based approach to create a VRE exchange platform. However, the documentation of the

[12] **FAIR** stands for **F**indability, **A**ccessibility, **I**nteroperability and **R**eusability.

iRODS API was lackluster. As such, it impeded the development of the interface to bwLehrpool's infrastructure and it can be expected that it will hinder its integration in the workflows of scientific communities. There are alternatives though, for example the object storage protocol S3 which offers multiple convenient interfaces to access its resources, such as a REST API and even s3fs [13], a third-party POSIX-like filesystem based on FUSE. Future work should evaluate an S3-based approach to realize a VRE registry and the extent to which an S3 storage layer can be complemented with a higher level iRODS layer for metadata management and automation, combining the best of both worlds.

5 Conclusion

The increased complexity of scientific workflows, the rising demands of researchers on compute power, and the sheer number of servers to monitor and administer demand new operation models. The workflows to run VREs are rather complex and took a while to mature. The first versions of VRE were in production pretty much since the official start of the NEMO cluster in mid 2016. Since then, a couple of improvements were implemented, but envisioned features like mapping Moab commands to OpenStack API allowing the pausing, hibernation and resumption of the virtual machine for preemption or maintenance instead of killing a job are still waiting to be tackled.

Limitations still exist for generalizing the use of VREs. The use cases considered featured embarrassingly parallel High-Throughput-Computing (HTC) workloads. These do not require high-speed low-latency networks for interaction between cluster nodes at all. Concentrating on such VREs simplified the setup and operation of virtual machines as special direct hardware access could be ignored. As further use cases like remote visualization emerge which require access to special hardware from inside a VRE, direct hardware access will be reconsidered in future activities. Experiments on containerization with Singularity show encouraging results for extending the scope of VRE deployment. A couple of bwHPC clusters plan to include Singularity by default in the near future. Having VREs in place opens up future paths like cloud-bursting.

[13] See `https://github.com/s3fs-fuse/s3fs-fuse` (visited on 10. 02. 2019).

A further challenge arises from the need to access data from within VREs. Traditional parallelized high-performance storage often does not provide the necessary security concepts to be directly accessed from within clearly defined security perimeters of a HPC system. If users become root in their VREs then the traditional means of privilege separation will no longer work. Another challenging issue arises from the intended heightened mobility of VREs. While in static cluster configurations, IP-based security even with its limitations made sense; the steps necessary to mount remote network filesystems into a VRE meant to run in more than one location no longer do. Even modern implementations like NFSv4 using account-based security face limitations. If a researcher moves on to another workgroup the account may be disabled and the access to the share becomes impossible. In the highly volatile environment of research institutions the chances for long-term stable user account-based access methods are dim. VREs become truly independent of location when the ties to traditional network and parallel file systems can be overcome e. g. by object storage solutions and access management gets moved on access tokens as well as global identities for researchers.

Depending on the way virtualization or containerization is orchestrated, the scheduling setup for the compute clusters has to be aware of the more dynamic nature of resources. HPC schedulers have to be aware of virtualized resources which, as a VM or Singularity container, are partially opaque to them. Further on when pre- and post-processing tasks in modern workflows are considered, these often profit from interactive handling instead of batch-driven automatic processing. Here, VREs could help to use the same working environment for cloud (pre-, post-processing and visualization) and HPC systems (main computational task). Containerization and VRE open the way to achieve Certified Research Environments in the sense that states of actual workflows could get frozen and archived in a consistent state. It would be beneficial to provide a platform and registry for researchers to get an overview of existing VREs including their relevant metadata. An ongoing challenge is the handling of sensitive data on shared resources like HPC and cloud. The requirements of the implemented data protection ruling are to be honored. Nevertheless, further research into versioning and long-term access to previous versions is still needed.

On the operational side, VREs allow the simple and convenient redistribution of tasks between the researchers focusing on the application side and the computer centers focusing on providing scalable research infrastructure. It extends the chosen path of the successful centralization and specialization of HPC resources. It helps to easily provision additional hardware resources brought in by a third party. For the next generation bwHPC cluster in Freiburg, the HPC team will reconsider the options to reduce the complexity of the VRE scheduling. For the upcoming cluster, a distinct cloud partition for HTC-focusing VREs is planned. Additionally, new models for scaling and configuring the computational resources are evaluated (Bauer et al., 2019).

Acknowledgement

The work presented in this publication was a part of the ViCE project sponsored by the Ministry of Science, Research and the Arts, Baden-Württemberg, Germany. The support is gratefully acknowledged.

Corresponding author

Jonathan Bauer: `jonathan.bauer@rz.uni-freiburg.de`
eScience Department, Computer Center, University of Freiburg
Hermann-Herder-Str. 10, 79104 Freiburg, Germany

ORCID

Jonathan Bauer https://orcid.org/0000-0002-5624-2055
Dirk von Suchodoletz https://orcid.org/0000-0002-4382-5104
Jeannette Vollmer https://orcid.org/0000-0001-5190-2942
Helena Rasche https://orcid.org/0000-0001-9760-8992

References

Afgan, E. et al. (2018). »The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update«. In: *Nucleic Acids Research* 46.W1, W537–W544. DOI: 10.1093/nar/gky379.

Allan, R. N. (2009). *Virtual research environments: From portals to science gateways.* Elsevier.

Bauer, J. et al. (2019). »A Sorting Hat For Clusters. Dynamic Provisioning of Compute Nodes for Colocated Large Scale Computational Research Infrastructures«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 217–229. DOI: `10.15496/publikation-29055`.

Baumann, M., V. Heuveline, O. Mattes, S. Richling and S. Siebler (2017). »SDS@hd–Scientific Data Storage«. In: *Proceedings of the 4th bwHPC Symposium October 4th, 2017, Alte Aula Eberhard Karls Universität Tübingen.* Ed. by J. Krüger and T. Walter, pp. 32–36. DOI: `10.15496/publikation-25204`.

Bührer, F. et al. (2018). »Dynamic Virtualized Deployment of Particle Physics Environments on a High Performance Computing Cluster«. In: *Computing and Software for Big Science.* arXiv: `1812.11044 [physics.comp-ph]`.

Chandramouli, R., M. Iorga and S. Chokhani (2014). »Cryptographic key management issues and challenges in cloud services«. In: *Secure Cloud Computing.* Springer, pp. 1–30.

De Roure, D., C. Goble and R. Stevens (2009). »The design and realisation of the myExperiment Virtual Research Environment for social sharing of workflows«. In: *Future Generation Computer Systems* 25.5, pp. 561–567. DOI: `10.1016/j.future.2008.06.010`.

Hauser, C. B. and J. Domaschka (2017). »ViCE Registry: An Image Registry for Virtual Collaborative Environments«. In: *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 82–89. DOI: `10.1109/CloudCom.2017.11`.

iRODS Consortium, ed. (2017). *iRODS User Group Meeting 2017 Proceedings – 9th Annual Conference Summary.* 83 pp. URL: `https://irods.org/uploads/2017/irods_ugm2017_proceedings.pdf`.

Li, M., S. Yu, Y. Zheng, K. Ren and W. Lou (2013). »Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption«. In: *IEEE transactions on parallel and distributed systems* 24.1, pp. 131–143.

Lombardi, F. and R. D. Pietro (2011). »Secure virtualization for cloud computing«. In: *Journal of Network and Computer Applications* 34.4. Advanced Topics in Cloud Computing, pp. 1113–1122. ISSN: 1084-8045. DOI: `10.1016/j.jnca.2010.06.008`.

Meier, K. (2017). »Infrastrukturkonzepte für virtualisierte wissenschaftliche Forschungsumgebungen«. PhD thesis. Albert-Ludwigs-Universität Freiburg im Breisgau.

Meier, K., B. Grüning, C. Blank, M. Janczyk and D. von Suchodoletz (2017). »Virtualisierte wissenschaftliche Forschungsumgebungen und die zukünftige Rolle der Rechenzentren«. In: *10. DFN-Forum Kommunikationstechnologien, 30.-31. Mai 2017, Berlin, Gesellschaft für Informatik eV (GI)*, pp. 145–154.

Oinn, T. et al. (2006). »Taverna: lessons in creating a workflow environment for the life sciences«. In: *Concurrency and Computation: Practice and Experience* 18.10, pp. 1067–1100.

Pampel, H. et al. (2013). »Making Research Data Repositories Visible: The re3data.org Registry.« In: *PLOS ONE* 8.11. DOI: 10.1371/journal.pone.0078080.

Rajasekar, A. et al. (2010). »iRODS primer: integrated rule-oriented data system«. In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2.1, pp. 1–143.

Rechert, K. et al. (2017). »Preserving Containers«. In: *E-Science-Tage 2017: Forschungsdaten managen*. Ed. by J. Kratzke and V. Heuveline. Heidelberg: heiBOOKS, pp. 143–151. DOI: 10.11588/heibooks.285.377.

Schmelzer, S., D. von Suchodoletz, M. Janczyk and G. Schneider (2014). »Flexible Cluster Node Provisioning in a Distributed Environment«. German. In: *Hochleistungsrechnen in Baden-Württemberg. Ausgewählte Aktivitäten im bwGRiD 2012*. Ed. by J. C. Schulz and S. Hermann. KIT Scientific Publishing, Karlsruhe, pp. 203–219. ISBN: 978-3-7315-0196-1. DOI: 10.5445/KSP/1000039516. URN: urn:nbn:de:0072-395167.

Shahzad, F. (2014). »State-of-the-art survey on cloud computing security Challenges, approaches and solutions«. In: *Procedia Computer Science* 37, pp. 357–362.

Suchodoletz, D. von et al. (2014). »bwLehrpool – ein landesweiter Dienst für die Bereitstellung von PC-Pools in virtualisierter Umgebung für Lehre und Forschung«. In: *PIK – Praxis der Informationsverarbeitung und Kommunikation* 37.1, pp. 33–40. DOI: 10.1515/pik-2013-0046.

Tari, Z., X. Yi, U. S. Premarathne, P. Bertok and I. Khalil (2015). »Security and privacy in cloud computing: Vision, trends, and challenges«. In: *IEEE Cloud Computing* 2.2, pp. 30–38.

Wang, C., Q. Wang, K. Ren, N. Cao and W. Lou (2012). »Toward secure and dependable storage services in cloud computing«. In: *IEEE transactions on Services Computing* 5.2, pp. 220–232.

Wilkinson, M. D. et al. (2016). »The FAIR Guiding Principles for scientific data management and stewardship«. In: *Scientific data* 3. DOI: 10.1038/sdata.2016.18.

Zhao, F., C. Li and C. F. Liu (2014). »A cloud computing security solution based on fully homomorphic encryption«. In: *Advanced Communication Technology (ICACT)*. 16th International Conference on Advanced Communication Technology (ICACT). IEEE, pp. 485–488.

Zissis, D. and D. Lekkas (2012). »Addressing cloud computing security issues«. In: *Future Generation computer systems* 28.3, pp. 583–592.

doi: 10.15496/publikation-29062

Storage infrastructures to support advanced scientific workflows

Towards research data management aware storage infrastructures

Dirk von Suchodoletz[*] Ulrich Hahn[†] Bernd Wiebelt[*]
Kolja Glogowski[*] Mark Seifert[*]

[*]eScience Department, Computer Center, University of Freiburg, Freiburg, Germany
[†]High Performance and Cloud Computing Group, University of Tübingen, Tübingen, Germany

The operators of the federated research infrastructures at the involved HPC computer centers face the challenge of how to provide storage services in an increasingly diverse landscape. Large data sets are often created on one system and computed or visualized on a different one. Therefore cooperation across institutional boundaries becomes a significant factor in modern research. Traditional HPC workflows assume certain preliminaries like POSIX file systems which cannot be changed on a whim. A modern research data management aware storage system needs to bridge from the existing landscape of network file systems into a world of flexible scientific workflows and data management. In addition to the integration of large scale object storage concepts, the long term identification of data sets, their owners, and the definition of necessary meta data becomes a challenge. No existing storage solution on the market meets all of the requirements, and thus the bwHPC-S5 project must implement these features. The joint procurement and later operation of the system will deepen the cooperation between the involved computer centers and communities. The transition to this new system will need to be organized together with the scientific communities being shareholders in the storage system. Finally, the created storage infrastructures have to fit well into the growing Research Data Repositories landscape.

doi: 10.15496/publikation-29058

1 Motivation

Modern research has become increasingly digital, leveraging a wide variety of hardware, data collection instruments, and software to gather, process, and visualize data in multitudes of ways. Such digital workflows are getting more complex and the volume of data processed or created is ever rising. From the perspective of a researcher, the typical workflow traditionally is executed on a local machine and, when the amount of data and computation exceeds local resources, handled by a larger system like an (external) HPC cluster. With the availability of new tools and options to process and view data, more systems will become involved in workflows.

Data Intensive Computing (DIC) involves big data or methods like deep learning to provide new perspectives on existing data (Schneider et al., 2019). This would require to bring data and compute resources to a common location efficiently. Depending on the type of data and workflow envisioned special resources like GPGPU are required which are not present at every site. Within Baden-Württemberg both the compute systems like the bwForClusters and the bwCloud as well as the LSDFs are distributed over different physical locations. The HPC systems on various tiers are complemented by e. g. bwCloud compute capacities to allow pre- or post-processing runs which would be a waste of resources on HPC systems. Remote visualization facilities – special systems sitting near the data – become relevant to render data and stream the results without the need to copy large data sets to the local machine of the user.[1] Additionally, requirements of reproducible science, the better understanding of the value of research data, the objective of open data publication all change the definition of data management.

Modern data management should extend beyond the traditional data-handling performed by a single scientist. Researchers often do not standardize metadata, making interoperability and sharing difficult. Data curation, the selection of data sets of relevance, and the removal of irrelevant data is often not a formalized step in the workflow. It often takes place when files must be copied across across systems, or whenever quotas were exceeded.[2] A storage system designed with research data management in mind should at least provide multiple ways to both automate workflows over the data lifecycle.

[1] The bandwidth consumed to stream visualization is usually much lower than to copy terabytes of data in reasonable time.

[2] This is not the proper trigger to achieve a high quality of data sets as valuable data might get thrown away because of size limitations.

The text is structured as follows: It gives an overview of the current state of data management and its challenges in HPC. The limitations of state-of-the-art of file systems used in HPC related scientific workflows will be explored. The requirements stemming from today's and future workflows of HPC and DIC user communities will be discussed. Further, it explores options to extend and optimize existing scientific workflows and local compute infrastructure setups. From this discussion it tries to provide a coherent conceptual framework for the design of a research data management aware large scale data facility. Storage-for-Science (bwSFS) is a DFG and state supported research infrastructure project to provide joint storage and research data management functionality for various research groups in Tübingen, Freiburg, Ulm and Stuttgart. The presented article extends upon the discussion provided in the paper published for the DFN-Forum 2017 (Meier et al., 2017) and outlines the framework for the intended design.

2 Scientific data on the tier-3 HPC environment

When scientists use a storage system, their satisfaction largely depends on how easy it is to access and on whether it is available in all the various usage scenarios. Usually, the optimization of one characteristic might degrade others. E. g. a local home directory on the scientist's laptop offers the easiest form of access to the data but is usually more severely capacity limited compared to home directories or data shares provided e. g. via NFS or SMB from specialized storage appliances. Network file systems require a common authentication service and properly mapped identities and face limitations regarding performance in wide area network operation. Data can be shared among colleagues to a certain degree from a networked file system provided by a storage appliance but can not easily extend beyond institutional boundaries.

All file systems (as well as their POSIX completeness and their performance) rely on the operating system support of the machine they are deployed on. If different systems involved in a scientific workflow do not feature the same operating system additional challenges will arise. These include the availability features a file system provides.[3]

[3] E. g. some metadata and additional information is lost when copying data from a Mac OS X HFS to Linux or Windows or vice versa. Another problem is the availability of a certain network file system in an operating system or version of it (French et al., 2007).

In the typical HPC world specialized local or near local storage in the geographical sense is attached to each compute infrastructure. Most bw*Clusters share these storage characteristics.

Fast parallel distributed file systems like e. g. BeeGFS, Lustre or GPFS in HPC are meant to provide a short-term high-performance storage to cater to many concurrent, parallel jobs.[4] BeeGFS and Lustre have free-of-charge[5] or open-source(-like) licenses but generally should be operated with a support contract.[6] GPFS, now marketed under the name Spectrum Scale, is another option but proprietary and rather expensive.[7] All are POSIX compliant and require a client-side kernel module. Parallel file systems are designed for capacity plus speed and are typically configured with a low level of hardware redundancy. The HPC system in Freiburg e. g. is set up with with two metadata daemons configured with two 1200 GB NVMe mirrored disks each running on one physical machine. The data is stored on four rack mount storage containers with sixty 4 TB disks each controlled by four machines. Each machine provides six RAID 6 storage targets with roughly 32 TB of capacity adding up to 768 TB of total usable parallel storage capacity.[8] These file systems usually offer large volume and are available during the time of actual computing, but not necessarily provided for long term storage of valuable data sets. The storage space even if vast is limited as it is shared among many users. Parallel file systems in HPC are usually not backed up.

In an era dominated by spinning magnetic disks, parallel storage offered an attractive compromise between speed, capacity and costs. However, depending on built-in redundancy, additional SSD- or NVMe-caching, chosen filesystem, and solution provider it can be quite costly in some configurations. It usually offers the largest user and group quotas available on an HPC system. Often it is possible to allocate several tens of terabytes per group. To manage concurrent use the parallel file systems offer means to set quotas e. g. on the amount of space or inodes taken. Nevertheless, the huge quantities of storage provided produces new challenges for users, operators, and system tools.

[4] For discussion of various storage technologies, future development and parallel file systems refer to the survey (Lüttgau et al., 2018) and the report (Brinkmann et al., 2017).

[5] See the end-user license agreement at `http://www.beegfs.io/docs/BeeGFS_EULA.txt`.

[6] Longer outages of a crucial part of an HPC system are costly as the whole system is affected.

[7] Aside project prices a per-socket license is charged.

[8] The costs per terabyte of the BeeGFS in NEMO run below 450 €.

Traditional system tools for listing files or accessing file metadata become unusably slow when run with significant amounts of files, degrading performance for other users. To properly manage scarcity of resources and create a kind of fairshare mechanism for storage a workspace regime is deployed for the bw*Clusters.[9] Workspace tools allow for automatic cleaning of unused data after a predefined amount of time. On HPC cluster NEMO users can allocate a workspace for a maximum of 100 days. After that period the workspace gets deleted automatically if the duration is not extended manually by the creator of this workspace. Users can extend their workspaces a configurable number of times. They can have multiple workspaces that are only limited by their quota. Workspaces provide a mechanism to clean old and unused data automatically if no manual interaction is made. This is alleviates the common problem of old inactive accounts occupying significant resources which could be better allocated to active projects.

Home directories are usually not identical to traditional homes in a multi-user environment like at a typical research institution, especially in bwHPC as users from different universities share domain specialized clusters for their research.[10] In the NEMO cluster environment the user home is provided via NFS from a enterprise grade storage appliance.[11] The system offers several levels of redundancy and snapshots to allow the user to go back to earlier versions of files. The space provided is intended to host the relevant research related files and data. Nevertheless, it is not the best location to store large amount of research data, as the size of disk quota provided is rather low. The per gigabyte costs are driven up by the various levels of redundancy and number of snapshots taken. Data can be stored for the lifetime of an account active for the particular cluster. Because of performance implications both on the storage appliance and on the speed of remote file access in the cluster users are strongly discouraged to run (parallel) HPC jobs out of their homes.

Scratch space is typically a fast local file system in each node which is node-exclusive with no redundancy and not shared among other nodes within the cluster. Most installations in the bwHPC federation deploy local scratch space, which is

[9] Workspace tools on GitHub: `https://github.com/holgerBerger/hpc-workspace` (visited on 13. 02. 2019)

[10] The situation is slightly different at the other bwHPC sites as some operators mount additional shares for local users or deploy parts of the parallel file system to be used as home directories.

[11] The per terabyte costs run at about 1200 €.

usually set into the relation to the number of CPU cores in the compute node. It is meant to allow processes to write temporary results in a fast fashion or to take debugging and logging output. All modern systems have solid state disks installed, but with a rather moderate capacity of in between 200 and 500 gigabytes per node. To overcome these capacity-limitations, special block device extensions featuring a concatenation of the local disk with network block device part backed by the parallel file system have been developed by the operators of the JUSTUS cluster in Ulm (Neuer et al., 2016).

A method to create an on-demand parallel file system backed scratch space is BeeOND (Brinkmann et al., 2017). It uses local scratch space combined with high speed networks like InfiniBand and Omni-Path to create larger storage capacities for a certain parallel jobs. The costs of scratch space directly correlates with the costs of the individual disks in each node. The big advantage is that the local scratch space usually consists of SSDs and therefore file access and write and read performance can be better than on the central storage which can still have a traditional hard disk setup for the storage targets. With BeeOND the local SSDs is utilized as a central storage for parallel jobs, permitting use of storage which otherwise would be idle and unused. That way the load on the central storage is reduced and misbehaving jobs with high IO do not affect other jobs. BeeOND is not configured with any levels of (hardware) redundancy which applies to the worker nodes as well. Data can be stored for the duration of the job lifetime but is expected to be moved away to a workspace at the end of the job. The on-demand scratch space is destroyed at the end of the job and data which has not been saved at this point is lost.

2.1 Limitations of current setups

There are a couple of limitations for modern data management in today's solutions deployed in the various bwHPC clusters. They are driven by a combination of factors. Storage in HPC systems is not meant to safeguard data for longer periods. The publication of data sets is outside the scope of HPC clusters and only partially answered by projects like bwDATAarchive. Not only the quota in the dimension of time or capacity is limited but also the level of redundancy and data backup. Most

HPC systems are set up as »closed boxes«, the available storage is not exposed outside the cluster itself and data must be copied in and out.[12]

The amount of data regarding file sizes and number of files play a significant role. Copying of data is acceptable for a certain quantity of files and size but becomes unbearable if a single operation exceeds a couple of hours. Researchers face versioning and synchronization challenges as the same data sets may exist on different systems at the same time. They need to apply stringent data management to avoid conflicts or to run into quota limitations. Additionally, long preparation times might require proper data staging to avoid long startup delays of scheduled HPC jobs. To circumvent inefficient use of HPC systems, preprocessing, post processing or visualization of data is expected to take place somewhere else.

All mentioned storage options adhere to the POSIX standard which is convenient for application development but might restrict parallel performance (Lüttgau et al., 2018). For performance and simplicity reasons many systems apply IP based security which is acceptable within tightly controlled physical installations and networks. Parallel file systems usually skip certain security checks and do not record access time. The latter would be relevant e. g. for hierarchical storage management systems.

Further limitations were experienced when experimenting with Virtual Research Environments (VRE). VREs abstract scientific workflows – a packaged software stack with certain configuration – in a virtual machine or a container (Bauer, Suchodoletz et al., 2019). To allow simpler sharing of environments or to run in widely distributed setups VREs are made independent of the underlying physical hardware. When moved from one cluster to another or to a cloud resource (Heidecker et al., 2017; Bührer et al., 2018) the task of a VRE remains the same but the accessibility of data sources and sinks poses a challenge as they break out of the walled garden of simple IP based access control and security.

Sharing of datasets becomes more relevant as cooperation of (geographically distributed) research groups, as well as the demand for re-using existing data, grows (Tenopir et al., 2011). With the predominant POSIX file systems the limited export options for data sharing are typically confined within a single HPC cluster or work group. Further limitations apply to sophisticated ACL settings or different views on

[12] The HPC systems in Karlsruhe and Heidelberg provide an exemption as they offer the mounting of local file system from local user groups or provide direct access to the SDS@hd service running on the Large Scale Data Facility (LSDF) (Bauer, Suchodoletz et al., 2019; Baumann et al., 2017).

a collection of files. In a traditional file system everything is hierarchically organized in directories, different views in the form of an alternatively structured directory tree for different users are not available.[13] Assigning rights to some subdirectory or share exceeding those of the top level directory are possible. But, it becomes increasingly difficult to monitor if everyone requiring access has the proper rights and if the rights are completely updated, when a person changes its role.

3 Rethinking scientific storage – towards bwSFS

A next generation scientific storage system should include features to support research data management in the sense that data could stay within the same storage system over the complete data life cycle – spanning from data acquisition over the various stages of computation, visualization to long term archiving and publication (Tenopir et al., 2011; Demchenko et al., 2012; Meier et al., 2017). The system should provide tools and services to support researchers in their data management tasks and various workflow designs. With the new bwHPC-S5 project started mid 2018, data management in HPC and DIC will get added to the development and support activities (Barthel et al., 2019). Federated services like EUDAT can provide guidance on which services are to be provided an how the several challenges are tackled (Lecarpentier et al., 2013; Ardestani et al., 2015).

The HPC sites of BinAC in Tübingen and NEMO in Freiburg[14] plan to complement the compute infrastructure by a research data management aware storage system also focused on their scientific cluster communities. The system is planned to run in a cooperated, federated fashion spanning both locations. After multiple rounds of discussion with future users during the grant application process and after the requested sum was approved a couple of characteristics and abstract features of a future research data management system can be summarized:

- The system needs to scale well: regarding both the total capacity and the size of individual files.

[13] Links, possible in some file systems, are not an alternative as they are not updated, if a file gets relocated.

[14] More information on the individual clusters and bwHPC in general can be found at `http://www.bwhpc.de/ressourcen.php`.

- It should offer a good compromise of price per terabyte, performance, and capacity over the whole system. Currently a hierarchical storage solution is planned. IO metrics of individual workflows will help to determine the requirements for the according hierarchy levels.
- The system should provide various levels of redundancy including geographically distributed locations.
- For trusted, reproducible scientific workflows the archived data should be immutable.
- Definable service classes for different qualities of storage should be provided. A user should be able to declare, that a certain data set should be kept geographically redundant or that it may not be copied to a location outside a given campus.
- The system should provide (high-performance) interfaces to other research infrastructures like HPC clusters and cloud systems.
- It should allow the automation of workflows by providing appropriate interfaces like REST APIs allowing asynchronous operation so that users do not need to wait until a certain storage related process is finished.
- The system needs to implement an identity mapping for users and long term data owners and data objects abstracted from the local IDM to federated IDM systems.
- Copying of large data sets should be avoided whenever possible. Many storage systems solve this by using references to data sets and update these, when changes occur. The future planning of computational infrastructures will be more and more influenced by the physical location of data.
- Monitoring and accounting features are required to provide administrators with enough insight to detect and avoid bottlenecks both in performance and system usage.

Beside the technical specifications the non-technical characteristics should adhere to the following ideas:

- The implemented solution should support at least in some abstract form the FAIR principles (Wilkinson et al., 2016) and the data life cycle.
- It should support the implementation of workflows or should provide the relevant stubs to deal with the various stages in the data life cycle:

  - Registration of storage capacity (of a certain kind, for a project with time limits).
  - Curation of data sets (e. g. certain files are automatically removed after a certain period because of a tag, provide a UI for interactive use, API).
  - Enrichment of data sets with metadata in the various stages of the life cycle.
- Workflows should be automated as far as possible by the system. Manual intervention should nevertheless be possible.
- Avoid vendor and technology lock-ins as the system is intended to exceed the short living technological life cycles.
- Provide various (discipline defined) interfaces for data publication via external third party services to blend into the growing Research Data Repositories landscape (Pampel et al., 2013).
- An interesting feature of a research data management repository would be to allow authenticated third party user comments on the data sets.[15]
- A sustainable and quality assured research data management will require certification of the system and the involved workflows. They are various certification options available, see e. g (RLG-NARA Task Force, 2007; DINI, Working Group Electronic Publishing, 2011; CCSDS Secretariat, 2011; ESF & EUROHORCs (2011), 2011).

The ongoing consultation with the stake- and shareholder of the federated research management system will further complete the matrix of the characteristics of the data sets like ACLs, size, value, share-ability. Alternatives to the up to now prevalent network file systems, like the S3 protocol or alike for object store systems, will be evaluated to gradually change workflows to better suit them to a sustainable research data management.

[15] Challenges remain how to distinguish between (il-)legitimate posts, though (Golbeck, 2018).

3.1 Security and privacy implications

Beside technical considerations legal implications will drive the requirements of a shared, multi-user storage systems. Funding agencies require retention of data for defined periods, commercial partners might want to restrict access to data sets and sensitive data like from clinical trials, surveys should not be disclosed to third-parties, or some projects require embargos for defined timespans. Certain datasets should not get copied outside the home institution even if redundancy would require it. Secure storage is not only achieved through local encryption of disks and filesystems but also through access management and secure transportation. Such challenges are discussed in the connection with cloud storage (Li et al., 2013; Wang et al., 2012). Security and privacy matters could be part of certification processes.

3.2 Mapping of identifiers

The provisioning of storage infrastructures in a broader sense regarding potential users and storage periods requires proper handling of user identities and identifiers for the objects. The identifiers for users will differ from traditional user accounts provided by the IDM of the local institution. This is already the case for the existing cluster and cloud logins. Unified credentials might be desirable for the network file systems which could in turn create challenges for the storage systems and proper user mapping. Switching to object store oriented operation access tokens could provide a more versatile solution than complex ACLs. The management of the tokens can be shifted more easily to the user side than traditional role management within a file system.

The need to identify data sets is completely different for the system and the users. Identifiers of data sets need to be mapped between the system and the interfaces for data management e. g. for indexing, citation, or access by external infrastructures. Metadata of various types[16] is crucial here as implicit information stored within a POSIX file system like access time or ownerships might get lost when files are moved.

[16] Ranging from common structures to describe projects, runtimes and owners to community specific information structures.

4 Managing transition

In Freiburg, the existing local compute services of both the HPC clusters of NEMO and the ATLAS experiment, as well as the bwCloud SCOPE and de.NBI cloud become more tightly integrated. For efficiency and manageability reasons they share the relevant base infrastructures like server racks, high speed Ethernet connectivity and hardware monitoring. Workflows which require the processing of data in different facilities like preprocessing in the cloud and parallel computing in NEMO should not require the copying of data sets as the relevant storage systems get more easily accessible within the conglomerate.

Learning from the results of the bwVisu project (Schridde et al., 2017) visualization servers could be added to an HPC cluster or a visualization cloud or cluster could be set up close to a large storage system for scientific data using cloud operation or container models to share the hardware. Services like CVMFS get deployed in a way to be available in all kinds of infrastructures ranging from the HPC clusters to clouds and VREs. Furthermore the provisioning of LSDF resources like SDS@hd (Baumann et al., 2017) from cloud, HPC and VREs got explored and enabled. Local storage of the various research groups should be available as well. This results in a matrix of accessible shares.

As large scale compute infrastructures become commonplace for most researchers and workflows get more diverse future data services should keep these developments in mind. In Freiburg a step by step approach is envisioned to provide truly flexible workflows. It works on different stages. As a compromise to the preexisting security and access control models of network file systems VLANs are used to virtually separate the installed infrastructures. As a flexible boot model defines into which operation mode a server is deployed the IP networks are assigned dynamically (Bauer, Messner et al., 2019). Thus, each infrastructure can mount resources depending on the operation model and workflow in use. Resources from the other infrastructures are not visible to the nodes because of the logical network separation.

To include cloud resources into a workflow, e. g. for pre and/or post processing a new rather flexible storage option becomes available. It provides a completely independent storage environment: Ceph block storage which can get configured and attached to VMs in various (user defined) ways for a wide range of different capacities. It adds another cost effective solution for temporary data. In further steps new approaches like the inclusion of S3 object storage will be evaluated. Suitable workflows are evaluated together with various research groups. A couple of *tiger teams* – an ad hoc group formed of scientists and practitioners to work on a particular issue – around data management were created within the context of bwHPC-S5 (Barthel et al., 2019).

To better support future data management data sets are to be tagged and/or described when created or processed, so that information becomes available and get added to metadata automatically. Optimally individual scientific workflows attach metadata already available or generated by them.

5 Outlook

The definition and design of a research data management system is an ongoing process and only tentative results can be given here. After securing the initial funding the next steps include the definition of required features and the decision on their realization. Not all requirements are met in systems available on the market yet. Hierarchical storage management systems offer a good yield on capacity, performance and prices per terabyte. Nevertheless, they need to be complemented by workflows helping both researchers and system providers to implement sustainable research data management. Here, the extended bwHPC-S5 project can provide the necessary resources to implement missing bits and pieces in-house and support the various communities in their data management needs. The joint procurement of the system both in Tübingen and Freiburg helps to gain a significant scaling-up and features like geographical redundancy. The sizing of the system allows to include more communities beside the core HPC cluster users of NEMO and BinAC. Upon a generic system core of standard technical features community specific options like specialized metadata and data services can be added. A similar »growth and upgrade path« as established with the HPC clusters should be implemented for the storage system as well: It should be possible that research groups, institutes etc. can pool-in financial resources and get equivalents in storage capacities of the requested quality level.

The transition from ad-hoc data handling today to sustainable data management is to be organized together with the scientific communities and institutions being shareholders in the storage system. On the technical layer a couple of steps were already implemented by the dynamic provisioning of HPC and cloud nodes via remote boot stateless provisioning (Bauer, Messner et al., 2019) or (Schmelzer et al., 2014) including the dynamically configured access to the relevant storage systems. Governance like already implemented globally and locally for the HPC systems (Wesner et al., 2017; Suchodoletz et al., 2017) needs to be in place as well. Monitoring and accounting need to provide the relevant input for decision making and the sustainable re-financing of the system in the long run. Finally, the automation of workflows from the definition of project requirements in a Data Management Plan to the deployment in the storage system would be an attractive feature (Bakos et al., 2018).

Acknowledgement

The research infrastructures described in this publications are part of the bwHPC project sponsored by the Ministry of Science, Research and the Arts, Baden-Württemberg (MWK) and the German Science Foundation. The same applies to the work presented from the ViCE project which got supported by the MWK as part of it's eScience initiative. The support is gratefully acknowledged.

Corresponding author

Dirk von Suchodoletz: dirk.von.suchodoletz@rz.uni-freiburg.de
eScience Department, Computer Center, University of Freiburg
Hermann-Herder-Str. 10, 79104 Freiburg, Germany

ORCID

Dirk von Suchodoletz https://orcid.org/0000-0002-4382-5104
Ulrich Hahn https://orcid.org/0000-0003-4471-9263
Bernd Wiebelt https://orcid.org/0000-0003-2771-4524
Kolja Glogowski https://orcid.org/0000-0002-1361-5712
Mark Seifert https://orcid.org/0000-0002-1042-6107

References

Ardestani, S. B. et al. (2015). »B2share: An open escience data sharing platform«. In: *2015 IEEE 11th International Conference on e-Science (e-Science)*. IEEE, pp. 448–453.

Bakos, A., T. Miksa and A. Rauber (2018). »Research Data Preservation Using Process Engines and Machine-Actionable Data Management Plans«. In: *International Conference on Theory and Practice of Digital Libraries*. Springer, pp. 69–80.

Barthel, R. and J. Salk (2019). »bwHPC-S5: Scientific Simulation and Storage Support Services. Unterstützung von Wissenschaft und Forschung beim leistungsstarken und datenintensiven Rechnen sowie großskaligem Forschungsdatenmanagement«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 17–28. DOI: 10.15496/publikation-29039.

Bauer, J., M. Messner et al. (2019). »A Sorting Hat For Clusters. Dynamic Provisioning of Compute Nodes for Colocated Large Scale Computational Research Infrastructures«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 217–229. DOI: 10.15496/publikation-29055.

Bauer, J., D. von Suchodoletz, J. Vollmer and H. Rasche (2019). »Game of Templates. Deploying and (re-)using Virtualized Research Environments in High-Performance and High-Throughput Computing«. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 245–262. DOI: 10.15496/publikation-29057.

Baumann, M., V. Heuveline, O. Mattes, S. Richling and S. Siebler (2017). »SDS@hd–Scientific Data Storage«. In: *Proceedings of the 4th bwHPC Symposium October 4th, 2017, Alte Aula Eberhard Karls Universität Tübingen.* Ed. by J. Krüger and T. Walter, pp. 32–36. DOI: 10.15496/publikation-25204.

Brinkmann, A., K. Mohror and W. Yu (2017). *Challenges and Opportunities of User-Level File Systemsfor HPC.* Tech. rep. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States). DOI: 10.2172/1424647. URL: https://www.osti.gov/servlets/purl/1424647.

Bührer, F. et al. (2018). »Dynamic Virtualized Deployment of Particle Physics Environments on a High Performance Computing Cluster«. In: *Computing and Software for Big Science.* arXiv: 1812.11044 [physics.comp-ph].

CCSDS Secretariat (2011). *Audit and certification of trustworthy digital repositories. Recommended Practice.* Recommendation for Space Data System Practices. Version CCSDS652.0-M-1. Space Communications and Navigation Office: Council of the Consultative Committee for Space Data Systems. URL: https://public.ccsds.org/pubs/652x0m1.pdf.

Demchenko, Y., Z. Zhao, P. Grosso, A. Wibisono and C. de Laat (2012). »Addressing Big Data challenges for Scientific Data Infrastructure«. In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pp. 614–617. DOI: 10.1109/CloudCom.2012.6427494.

DINI, Working Group Electronic Publishing (2011). *DINI-Certificate Document and Publication Services 2010 [March 2011]*. Deutsche Initiative für Netzwerkinformation (DINI). DOI: 10.18452/1494.

ESF & EUROHORCs (2011) (2011). *Basic Requirements for Research Infrastructures in Europe*. URL: http://www.dfg.de/download/pdf/foerderung/programme/wgi/basic_requirements_research_infrastructures.pdf.

French, S. M. and S. Team (2007). »A New Network File System is Born: Comparison of SMB2, CIFS and NFS«. In: *Linux Symposium*. sn, p. 131.

Golbeck, J. (2018). »Data We Trust—But What Data?« In: *Reference & User Services Quarterly* 57.3, pp. 196–199.

Heidecker, C., M. Giffels, G. Quast, K. Rabbertz and M. Schnepf (2017). »High precision calculations of particle physics at the NEMO cluster in Freiburg«. In: *Proceedings of the 4th bwHPC Symposium October 4th, 2017, Alte Aula Eberhard Karls Universität Tübingen*. Ed. by J. Krüger and T. Walter, pp. 28–31. DOI: 10.15496/publikation-25203.

Lecarpentier, D. et al. (2013). »EUDAT: a new cross-disciplinary data infrastructure for science«. In: *International Journal of Digital Curation* 8.1, pp. 279–287. DOI: 10.2218/ijdc.v8i1.260.

Li, M., S. Yu, Y. Zheng, K. Ren and W. Lou (2013). »Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption«. In: *IEEE transactions on parallel and distributed systems* 24.1, pp. 131–143.

Lüttgau, J. et al. (2018). »Survey of Storage Systems for High-Performance Computing«. In: *Supercomputing Frontiers and Innovations* 5.1, pp. 31–58. DOI: 10.14529/jsfi180103.

Meier, K., B. Grüning, C. Blank, M. Janczyk and D. von Suchodoletz (2017). »Virtualisierte wissenschaftliche Forschungsumgebungen und die zukünftige Rolle der Rechenzentren«. In: *10. DFN-Forum Kommunikationstechnologien, 30.-31. Mai 2017, Berlin, Gesellschaft für Informatik eV (GI)*, pp. 145–154.

Neuer, M. et al. (2016). »Motivation and Implementation of a Dynamic Remote Storage System for I/O Demanding HPC Applications«. In: *International Conference on High Performance Computing*. Springer, pp. 616–626.

Pampel, H. et al. (2013). »Making Research Data Repositories Visible: The re3data.org Registry.« In: *PLOS ONE* 8.11. DOI: 10.1371/journal.pone.0078080.

RLG-NARA Task Force (2007). *Trustworthy Repositories Audit & Certification: Criteria and Checklist.* Report. Version 1.0. RLG- National Archives and Records Administration Digital Repository Certification Task Force. URL: http://bibpurl.oclc.org/web/16713.

Schmelzer, S., D. von Suchodoletz, M. Janczyk and G. Schneider (2014). »Flexible Cluster Node Provisioning in a Distributed Environment«. German. In: *Hochleistungsrechnen in Baden-Württemberg. Ausgewählte Aktivitäten im bwGRiD 2012.* Beiträge zu Anwenderprojekten und Infrastruktur im bwGRiD im Jahr 2012. Ed. by J. C. Schulz and S. Hermann. KIT Scientific Publishing, Karlsruhe, pp. 203–219. ISBN: 978-3-7315-0196-1. DOI: 10.5445/KSP/1000039516. URN: urn:nbn:de:0072-395167.

Schneider, G. et al. (2019). »Umsetzungskonzept der Universitäten des Landes Baden-Württemberg für das High Performance Computing (HPC), Data Intensive Computing (DIC) und Large Scale Scientific Data Management (LS^{2}DM)«. Gekürzte Fassung. In: *Proceedings of the 5th bwHPC Symposium. HPC Activities in Baden-Württemberg.* Freiburg, September 2018. 5th bwHPC Symposium. Ed. by M. Janczyk, D. von Suchodoletz and B. Wiebelt. TLP, Tübingen, pp. 3–16. DOI: 10.15496/publikation-29040.

Schridde, D., M. Baumann and V. Heuveline (2017). »Skalierbare und flexible Arbeitsumgebungen für Data-Driven Sciences«. In: *E-Science-Tage 2017: Forschungsdaten managen.* Ed. by J. Kratzke and V. Heuveline. Heidelberg: heiBOOKS, pp. 153–166. DOI: 10.11588/heibooks.285.377.

Suchodoletz, D. von, B. Wiebelt and M. Janczyk (2017). »bwHPC Governance of the ENM community«. In: *Proceedings of the 3rd bwHPC-Symposium.* (2016). Ed. by S. Richling, M. Baumann and V. Heuveline. Heidelberg: heiBOOKS. DOI: 10.11588/heibooks.308.418.

Tenopir, C. et al. (2011). »Data Sharing by Scientists: Practices and Perceptions«. In: *PLOS ONE* 6.6, pp. 1–21. DOI: 10.1371/journal.pone.0021101.

Wang, C., Q. Wang, K. Ren, N. Cao and W. Lou (2012). »Toward secure and dependable storage services in cloud computing«. In: *IEEE transactions on Services Computing* 5.2, pp. 220–232.

Wesner, S., D. von Suchodoletz, B. Wiebelt, G. Schneider and T. Walter (2017). »Overview on governance structures in bwHPC«. In: *Proceedings of the 3rd bwHPC-Symposium: Heidelberg 2016.* (2016). Ed. by S. Richling, M. Baumann and V. Heuveline. Heidelberg: heiBOOKS. DOI: 10.11588/heibooks.308.418.

Wilkinson, M. D. et al. (2016). »The FAIR Guiding Principles for scientific data management and stewardship«. In: *Scientific data* 3. DOI: 10.1038/sdata.2016.18.

Proceedings of the 5th **bwHPC Symposium**

In modern science, the demand for more powerful and integrated research infrastructures is growing constantly to address computational challenges in data analysis, modeling and simulation. The bwHPC initiative, founded by the Ministry of Science, Research and the Arts and the universities in Baden-Württemberg, is a state-wide federated approach aimed at assisting scientists with mastering these challenges.

At the 5th bwHPC Symposium in September 2018, scientific users, technical operators and government representatives came together for two days at the University of Freiburg. The symposium provided an opportunity to present scientific results that were obtained with the help of bwHPC resources. Additionally, the symposium served as a platform for discussing and exchanging ideas concerning the use of these large scientific infrastructures as well as their further development.

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

TÜBINGEN
LIBRARY PUBLISHING

9 783946 552253