 doi: 10.15496/publikation-29062

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/` (visisted 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.

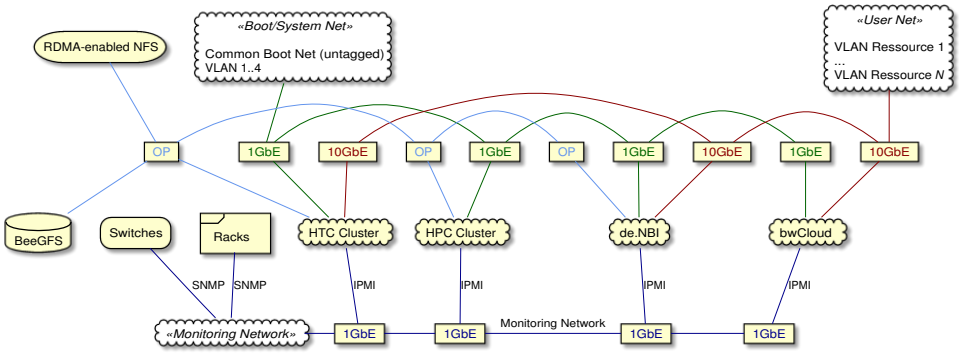

Figure 1: Basic network configuration of distinct Ethernet and Omni-Path infrastructures

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).

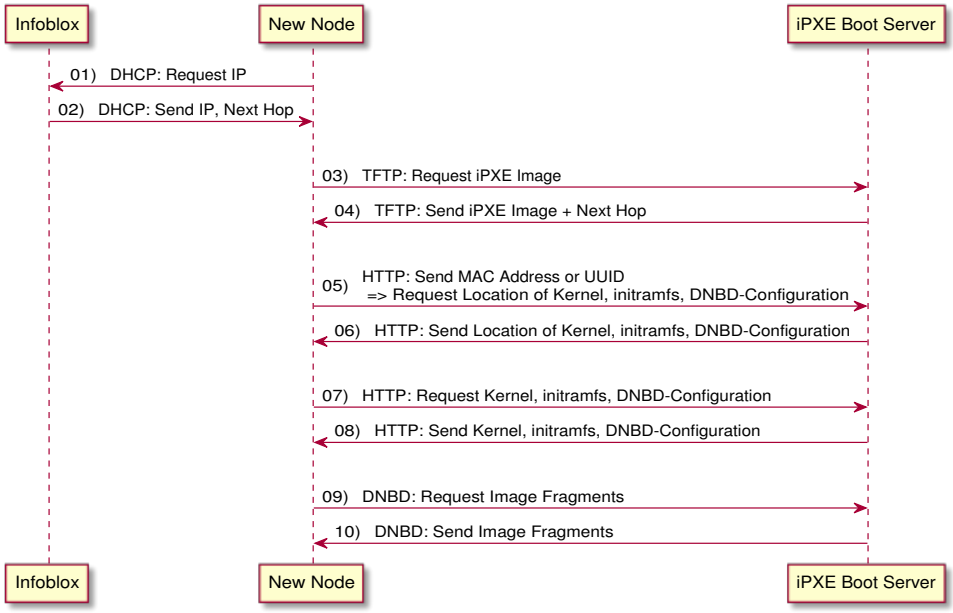

Figure 2: Boot Sequence including Boot Selection Server

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.