 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).

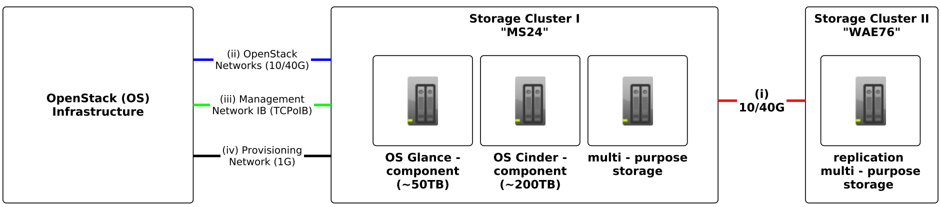

Figure 1: Storage cluster replication network and integration into OpenStack.

[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.

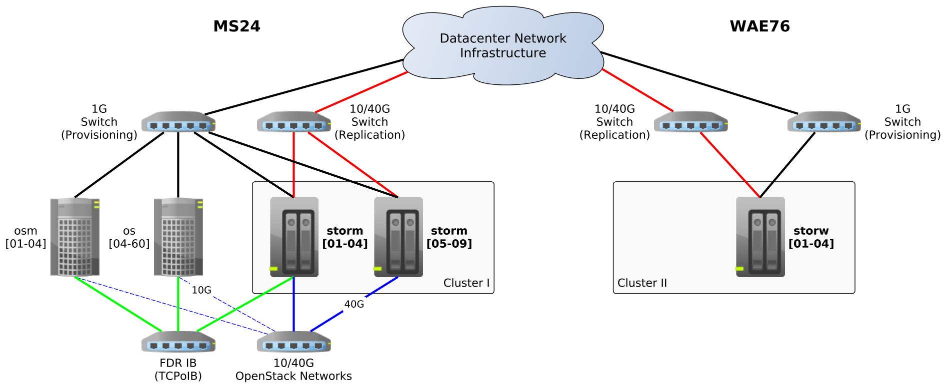

Figure 2: *Left:* Storage cluster I with cluster nodes storm[01-09], integrated into OpenStack. *Right:* Storage replication cluster II with cluster nodes storw[01-04].

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.

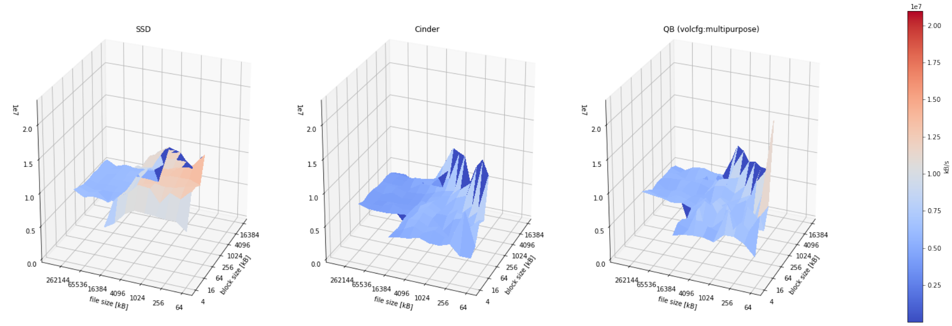

Figure 3: Read bandwidths

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 $3\times$ 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.

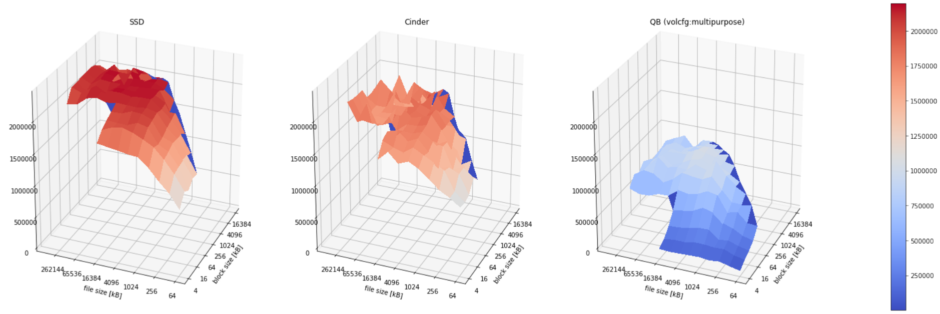

Figure 4: Write bandwidths

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.