A national approach for storage scale-out scenarios based on iRODS

Christine Staiger
SURFsara
Science Park 140,
Amsterdam, The Netherlands
christine.staiger@surfsara.nl

Ton Smeele
Utrecht University
ITS/RDM
Heidelberglaan 8,
Utrecht, The Netherlands
a.p.m.smeele@uu.nl

Rob van Schip
Utrecht University
ITS/RDM
Heidelberglaan 8,
Utrecht, The Netherlands
r.j.a.vanschip@uu.nl

ABSTRACT

The Dutch Universities and associated Medical Centers are developing research data management environments built on iRODS to support their scientists. The underlying storage is currently primarily located on the premises and under the control of said institutes. However, some local storage systems offer too little capacity. Moreover, there is a need for a variety of storage systems to offer efficient and cost effective data storage solutions that may differ per use case. Because requirements towards the storage backend between single research institutes overlap, a national approach can add significant value. We present a proof of concept study how such a scenario can be supported using iRODS. In our use case scenario SURFsara, the national high-performance compute (HPC) and data centre, provides storage resources connecting local data to European infrastructures such as EUDAT, EGI and PRACE. We highlight the infrastructural aspects and which data policies can be supported. The scenarios are substantiated with performance tests executed with the underlying transfer protocol to the different storage systems.

Keywords

iRODS, Storage scale-out, Infrastructure, Policies, Research data management (RDM) platforms.

INTRODUCTION

The Dutch Universities are developing data management platforms to assist researchers to safely store and collaborate on data during and after their research, to account for data generated and processed in a research project and to facilitate reuse of such data. The way how data needs to be treated and stored, i.e. the data policies, can vary per university, per faculty and per research project.

We base our study on data management platforms built on iRODS drawing on the following advantages:
Data policies can be system-enforced, targeted to specific data types and use cases and maintained efficiently. Moreover, iRODS allows to integrate heterogeneous storage solutions accounting for different requirements and lowering costs for storage by combining cheap and expensive storage media. Management, distribution and migration of data files across locations and vendor storage solutions is performed transparent to the user and automated through and directed by applicable data policy rules.

Additional cost reduction can potentially be achieved through adopting a cloud storage delivery model that serves storage for several projects, institutes and universities lowering overhead costs as well as employing expertise on different storage systems from a dedicated third party rather than fostering and maintaining expertise and hardware at the single Universities and institutes.

Research projects are dealing with sensitive data. Those data are subject to strict legal regulations, e.g. such data must be managed by University staff and may not be transferred across national or European Economic Area (EEA)

iRODS UGM 2017 June 13-15, 2017, Utrecht, Netherlands

borders. The data management platforms need to support such strict data policies which are very hard to put into practice when combining those platforms with storage from commercial storage providers.

As the collaborative ICT organisation for Dutch education and research, SURF [1] is part of the Dutch research landscape and part of the European research infrastructures such as PRACE [2], EGI [3] and EUDAT [4]. Thus, SURFsara [5], as part of SURF, can serve as a trusted storage provider and is well-positioned to support the scenario above. We will investigate the opportunity to provide a cloud storage solution as a service managed by SURFsara that integrates with each university's iRODS data management platform.

Such a cloud storage solution needs to support a replication and a storage scale-out scenario. In a replication scenario universities outsource secondary copies of data to storage provided by SURFsara to serve as fall back copies for disaster recovery. In a scale-out scenario, however, universities store active data on such a storage system, i.e. scientists work directly with these data. In fact, in both cases the same infrastructure can be used. In our investigation we focus on the scale-out scenario, since this is the most demanding scenario with respect to performance requirements.

We present a proof-of-concept study that can support the above-mentioned scenarios. We provide the technical setup for both scenarios and we test in particular the scale-out with respect to performance and user experience and whether the local data policy [6] requirements can be met.

Note, out of scope of our study are use cases that benefit greatly from using storage directly attached to a workstation because they have been designed to take advantage of low latency disk read/write operations.

USE CASES

The data management platforms and thus the underlying infrastructure are built for scientists to maintain their data during and after the research process. We will discuss and test the following use cases where performance plays a paramount role:

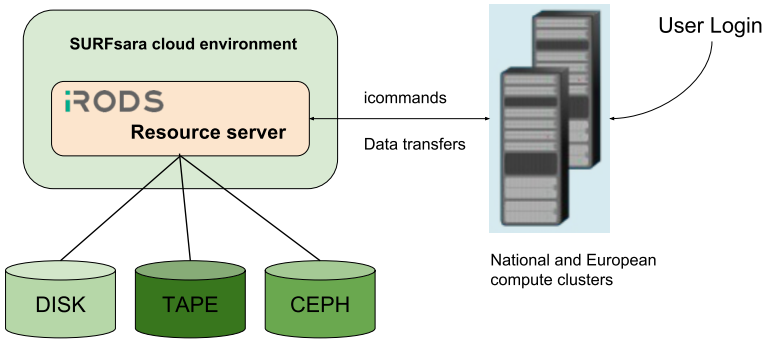

Figure 1. Usage of data managed by iRODS from compute systems.

- **Users mount iRODS to their workstations to up and download data and to work on the data directly.** This is accomplished by Davrods [7] and allows users to drag and drop data between the iRODS file system and their local file system. Data can be stored on local storage or on a scale-out resource server at a different site. For programmatic data transfers users employ the *icommands* to put and get data to and from iRODS.

- **Users manage data in iRODS and analyse data on HPC infrastructures (Figure 1).** To this end an HPC cluster at an HPC centre such as SURFsara is used. SURFsara hosts the national supercomputer [8] that is part of PRACE (Partnership for Advanced Computing in Europe Research Infrastructure) and the national compute cluster [9]. The use of a storage service close to the HPC infrastructure can improve transfer speed.

The HPC cluster needs to accommodate an iRODS client e.g. the icommands with which the user can move data between the iRODS instance and the HPC cluster.

For another use case we will discuss the technical setup:

- **Long-term archiving of data** can be accomplished by storing data on tape and labeling it for later reference. Data will be migrated to cheap, high-latency media as tape. Here a replica or copy is created on the iRODS resource server.

PROOF OF CONCEPT ARCHITECTURE

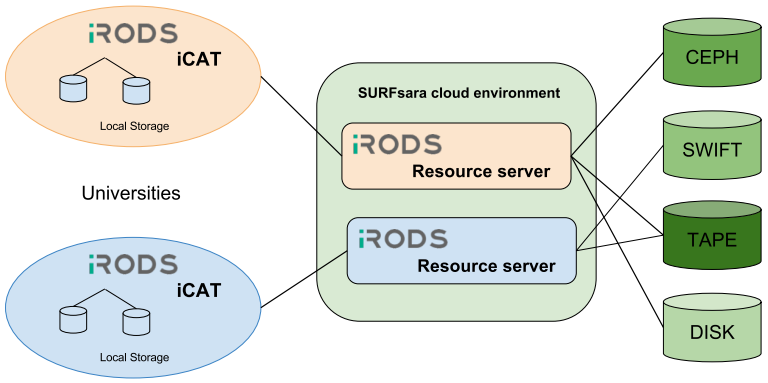

Figure 2. High-level overview of the infrastructure. Universities get access to storage infrastructure via iRODS resource servers which are attached to the University's iRODS zone. The resource servers are hosted on virtual machines running on a SURFsara cloud environment.

To facilitate the access to the national storage systems and integrate them with the Universities' iRODS platforms we deploy iRODS resource servers on the SURFsara HPC cloud (see Figure 2). Storage can be attached to these resource servers as first order resource or as compound resource depending on the backend storage system. Universities get access to storage infrastructure via iRODS resource servers which are attached to the University's iRODS zone. This guarantees that all data is subject to the Universities' data policies although located at a third-party storage provider. In the following paragraph we will describe how scientists and data managers can make use of the underlying infrastructure.

Storage system implementations

In the following sections we will describe the technical setup for several storage systems. Our tests are based on an implementation using a CEPH storage system. In our setup all resource servers are run on SURFsara's HPC cloud environment based on OpenNebula. From there the connection to other storage media is made.

Storage systems that support POSIX compliant random access file operations can be attached directly as a *unix-filesystem* type resource. Other storage systems such as object stores and tape archival systems typically need to be configured as a compound resource. The compound resource adds a POSIX compliant cache resource in front of an archive resource. Exchange of data objects between cache resource and the actual storage system is executed through vendor specific drivers that perform data transfer protocol translations.

CEPH

The HPC cloud infrastructure uses a CEPH cluster to support virtual machines with extra storage. CEPH partitions are attached as an extra file system to the virtual machines directly. Such storage can be integrated in iRODS as iRODS *unixfilesystem* resource (see Figure 3, left side). Most other resources like archive and SWIFT will be accessed via a compound resource as we will see in the following Sections.

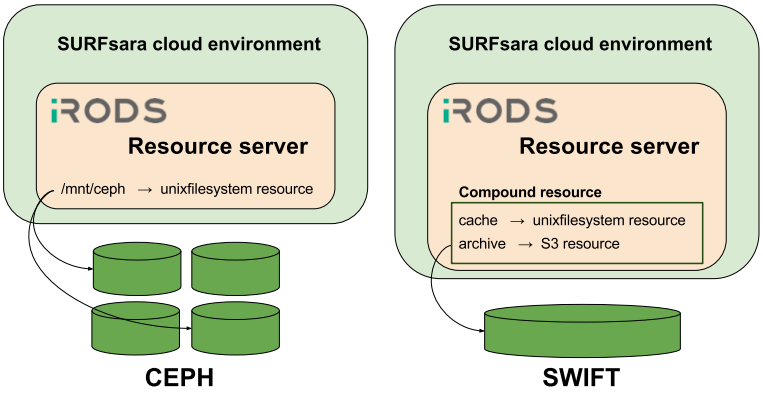

Figure 3. Storage scale-out by either as first order resource or via a compound resource. Left: By attaching file systems to the resource server these file systems can be used by an iRODS *unixfilesystem* storage resource. Right: The connection to an OpenStack SWIFT cluster can be made via iRODS compound resources employing the iRODS S3 resource type as archive resource. Green: Storage and infrastructure managed by SURFsara; Orange: iRODS servers managed by the universities.

Openstack SWIFT

In contrast to CEPH storage, iRODS can only make use of OpenStack SWIFT or other S3 compatible storage types via a compound resource (see Figure 3, right side). The connection between the cache resource and the OpenStack SWIFT archival resource is made via the S3 plugin. The plugin uses a login on the OpenStack SWIFT cluster in form of an AWS key-pair, i.e. all data on this storage will be owned by this account no matter which iRODS user ingested the data into the resource.

Archiving data in a Tape library

To account for the need of cheap storage that supports long-term archiving of data we integrated the resource server with an storage environment based on tape. This environment can be accessed from iRODS via a compound resource as we saw with OpenStack SWIFT. However, in this case the iRODS distribution does not provide a native plugin to facilitate the communication between the cache resource and the archive resource. The communication between the cache and archive resource is defined by a universal MSS interface script that implements the functions *syncToArch, stageToCache, mkdir, chmod, rm, mv* and *stat*. Based on the general universal MSS interface [10] SURFsara provides such universal MSS interface scripts to connect to tape environments using either *gridFTP* [11] and *rsync* [12].

Federation as an alternative for storage system implementations

Opposed to the previous architecture where we extended the storage under one iRODS instance by directly attaching resources, one can also make use of iRODS federations to give access to the underlying storage infrastructures. Federations are not a solution for storage scale-out, yet federations support replication scenarios. We will briefly outline an example of such an architecture below.

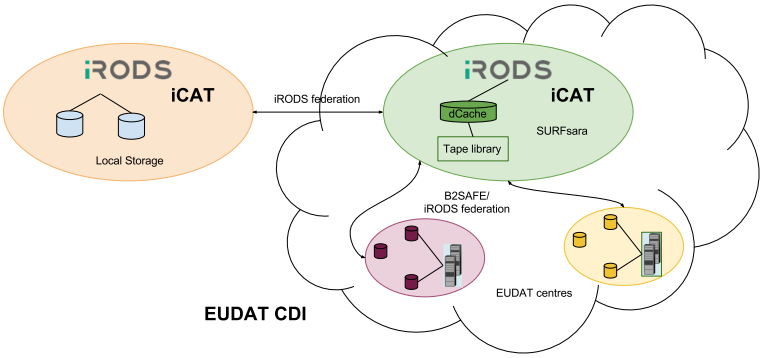

Figure 4. Archiving via a federation. SURFsara hosts an own iRODS instance which is coupled to the tape library via the dCache environment. The same iRODS instance is part of the EUDAT CDI and implements the B2SAFE service with which data can be replicated to other EUDAT centres.

SURFsara hosts an own iRODS instance. This instance is part of the EUDAT B2SAFE [13] network and part of EUDAT's collaborative data infrastructure (CDI). B2SAFE is EUDAT's service for safe data replication between EUDAT centres. The service integrates iRODS with persistent identifiers to keep track of data and its replicas. Hence, Universities can use the B2SAFE service to create persistent identifiers for the replicas for identification and citation; and use SURFsara's iRODS instance as an entry point to the EUDAT CDI as indicated in Figure 4.

RESULTS AND DISCUSSION

Evaluation of the implementation

In the evaluation of the proof of concept implementation we focused on the usability of the CEPH resource under the iRODS resource server.

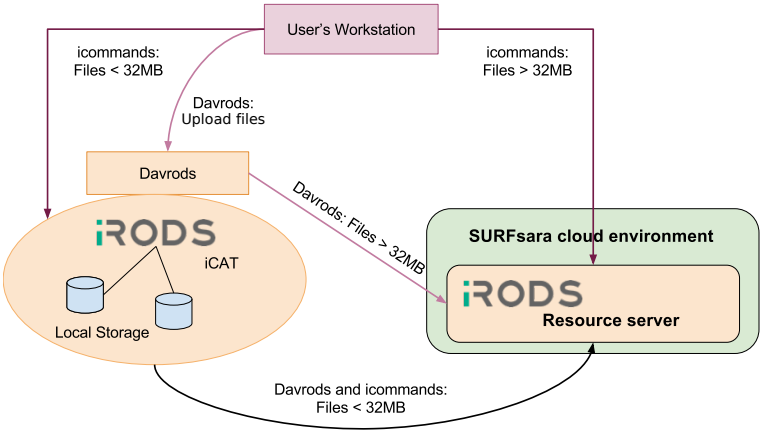

Figure 5. Testing scenario for interacting with the iRODS resource server from a user's workstation. User connects with Davrods to the iCAT enabled server (light purple arrow). For file transfers less than 32 MB the iCAT server acts as a hub in between Davrods and the resource server (black arrow). Transfer of larger files is performed via peer-to-peer connection between Davrods and the resource server. As an alternative to Davrods, *icommands* can be deployed on the user's Linux workstation (dark purple arrows). In that case larger files are transferred peer-to-peer between workstation and resource server.

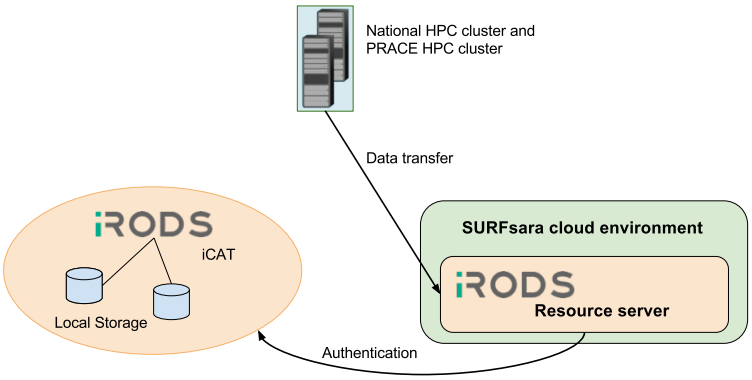

Figure 6. Testing the up and download of files to the resource server with the *icommands* from the HPC clusters. The user connects directly to the iRODS resource server which will in turn connect to the iCAT database to authenticate the user. The data is directly up and down loaded to and from the iRODS resource server.

To gain insight into the overall user experience, we tested typical workflows from a local workstation using the CEPH resource (see Figure 5). Furthermore, we measured up and download speeds of file transfers to and from the resource server

- from a local workstation with the *icommands* (see Figure 5)
- from the national and European HPC clusters employing the *icommands* (see Figure 6)

User experience tests

To test end user experience we executed several office application workflows using two different client workstations and three different storage locations. Our client environments include both Windows7 and Linux client workstations to identify potential impact of client operating system on the user experience. Our storage locations are 1) workstation locally attached disk drive 2) a storage resource directly managed by and attached to the university's iRODS iCAT server and 3) the CEPH storage partition located at SURFsara in Amsterdam managed via a resource server. The iCAT server communicates with the resource server via the internet.

Upon each test our client workstation connects to the iRODS iCAT server via Davrods and mounts the iRODS home collection as a network drive. We worked with files stored locally on the workstation and compared this experience with working with files stored on the network drive. As for the network drive, we varied the default resource configuration of Davrods to select either the resource on the iCAT-enabled iRODS instance or the resource on the iRODS resource server.

We used the MS-Office suite on Windows 7 and LibreOffice suite on Linux to manipulate text documents and spreadsheets. We also used accessory tools such as an ascii-text editor and a web browser to browse ascii text, JPG and PDF files. The files varied in size between 15 kilobytes and 17 megabytes.

We found that open and save operations that access a file stored locally on the workstation are slightly faster than using similar operations to access a file stored on the iCAT server resource. In nearly all workflows the response times remained below a second and as such they are within acceptable user experience ranges. In odd cases the response time in the iCAT server resources configuration could amount to 2-3 seconds. Interestingly, the response time remains about the same when we change our configuration to access a file stored on the resource server.

The choice of the client workstation operating system did not influence the user experience in all major operations

that we tested with one significant exception: it took upto 10 seconds to mount the network drive for the first time using the native MS Windows drivers (Windows Security). Alternatively when we opened a connection on the same workstation using Cyberduck drivers the operation completed within 3 seconds.

On the whole this experiment shows that the performance in all configurations will support the user in working with his/her data in an adequate way.

File transfer tests

We tested file transfers to and from the iRODS resource server using *icommands* in two settings: 1) from a Linux workstation and 2) from two HPC compute clusters. In the first setting the user connects to the iCAT-enabled iRODS server but stores data on a CEPH resource located at the iRODS resource server (Figure 5). In the second setting (Figure 6) the user connects from the HPC clusters directly to the iRODS resource server and stores data on the respective CEPH resource.

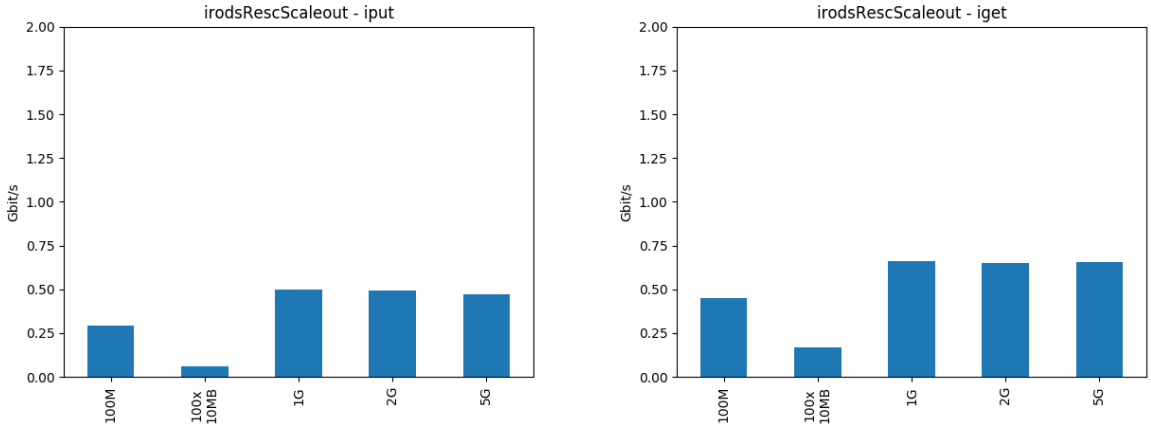

Figure 7. Mean performances of iput (left) and iget (right) from a user's Linux workstation to the iRODS resource server via the iCAT-enabled server.

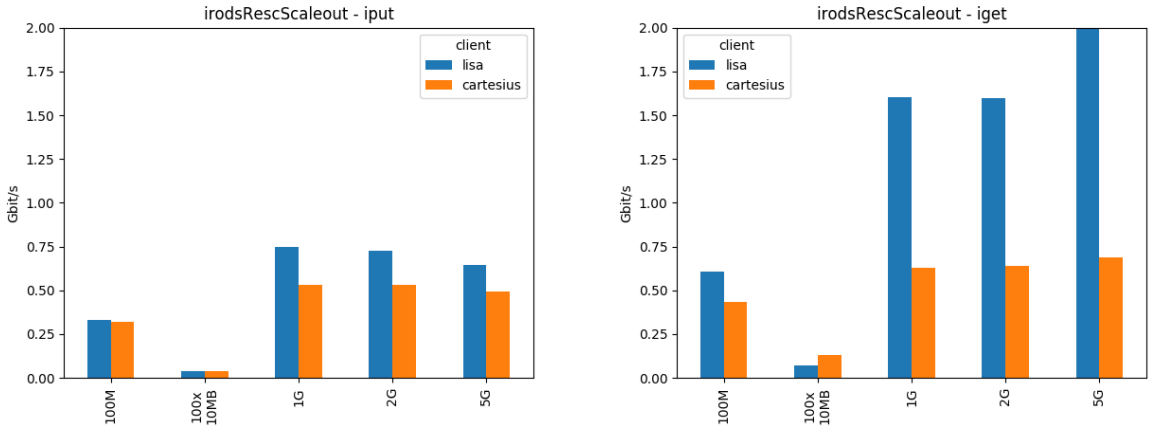

Figure 8. Mean performances of iput (left) and iget (right) from the national supercomputer (cartesius) and from the national compute cluster (lisa). The difference in transfer speeds can be explained by the different internet network connections on the two infrastructures.

Figure 7 and 8 show the performance of put and get operations on the iRODS resource server from the respective HPC clusters and the workstation. We tested the up and download of single files of size 100MB, 1GB, 2GB and 5GB and the transfer of a collection containing 100 files of size 10MB. The tests show that the data transfers to and from the CEPH resource work slightly faster when the client connects directly the resource server. Notably, up and downloading collections with a lot of small files (100x10MB) is much slower than downloading the same amount of data in a single file (1G). The low performance is due to setting up connections for each single file and look-ups and entry creation in the iCAT database. Hence the experiment shows that the iRODS resource server configuration efficiently supports use cases where larger files are transferred. Use cases that involve transfer of batches of smaller files will require additional measures to counter protocol overhead (e.g. bundle files prior to transfer).

Other findings

Impact on network configurations

Connectivity tests have shown that iRODS expects iCAT and resource servers in the data grid zone to be accessible using their fully qualified domain name (FQDN). Network configurations where resource servers are addressed via a proxy server such as a load balancer are not fully supported.

This limitation is a result of the iRODS parallel transfer protocol implementation which by default kicks in on transfers of files that exceed 32 MB in size. Suppose a client connects to server A. Now consider a scenario where the file needs to be transferred to or from a resource not managed by server A. In this case server A will use the ICAT database to locate server B that manages the resource. It opens a server-to-server connection to server B. Server B provides its own hostname (FQDN) and the TCP ports to be used for data transfer. Server A communicates this information to the client so that the client can open ports to server B. Subsequently the data flows directly between the client and server B. Note that in this scenario the client must be able to connect directly to server B using B's FQDN which could fail if server B is behind a proxy.

Impact of compound resources

In the previous sections we have shown that other complex storage systems can only be made available to iRODS via a compound resources. This has impact on the data workflows in such a storage system. When uploading data directly to a resource (e.g. a *unixfilesystem* resource) the user can be sure that the data is stored correctly after the transfer finished. In case of uploading data to a compound resource the user can only be sure that the data is stored safely on the cache resource and will eventually - depending on the system configuration - be moved to the archive resource. This poses two risks:

1. If the connection to the archive resource does not work as expected, the cache resource is filled up and no data is further transferred, clogging the system for other users and not keeping the data safe.
2. The user himself has to either rely on the system configuration to delete the replica located on the cache resource as soon as possible or he has to do that himself, which in turn requires users to be familiar with the underlying infrastructure and the command line tool options for the *icommands*.

The usage of the tape environment is possible in two ways: either directly using a compound resource or indirectly using a federated zone that employs the compound resource. Attaching the tape environment as a compound resource to iRODS allows Universities to integrate this resource seamlessly into their environment and manage access with their data policies. Alternatively, federations allow for complex configurations across administrative domains.

CONCLUSIONS AND FUTURE WORK

We demonstrated that a national cloud storage service can be used as a seamless extension of iRODS-based data management platforms hosted by the Dutch Universities and research institutes. Read and write performances remain within acceptable user experience ranges except that transferring batches of small files is relatively slow. Deployment

of a national cloud storage service in a scale-out scenario requires that the Universities' iRODS servers are directly accessible from the internet.

In our work we did not test performances in a real-life setting, i.e. many users, large amounts of files. In the future we plan to explore these scalability aspects. Moreover, specific service configurations need to be tested e.g. what are the performance characteristics and limitations when using a compound resource. We will also investigate in which ways the cloud storage service model can be complemented by other service models based on zone federations rather than zone extension.

REFERENCES

[1] SURF. Collaborative organisation for ICT in dutch education and research. [Online]. Available: https://www.surf.nl/en

[2] PRACE research infrastructure. [Online]. Available: http://www.prace-ri.eu/

[3] EGI. Advanced computing for research. [Online]. Available: https://www.egi.eu/about/

[4] EUDAT. The EUDAT collaborative data infrastructure. [Online]. Available: https://www.eudat.eu/eudat-cdi

[5] SURFsara. High performance computing & data infrastructure for science and industry. [Online]. Available: https://www.surf.nl/en/about-surf/subsidiaries/surfsara/

[6] YODA. University Utrecht iRODS rule set. [Online]. Available: https://github.com/UtrechtUniversity/irods-ruleset-uu

[7] T. Smeele and C. Smeele, "Davrods, an apache webdav interface to iRODS,âĂŹâĂŹ in *Proceedings iRODS User group meeting 2016*. https://irods.org/uploads/2016/12/irods_ugm2016_proceedings.pdf, 2016, p. pp 41.

[8] Cartesius: The Dutch supercomputer. [Online]. Available: https://userinfo.surfsara.nl/systems/cartesius

[9] The Dutch national compute cluster: The lisa system. [Online]. Available: https://userinfo.surfsara.nl/systems/lisa

[10] J.-Y. Nief. Universal mss interface. [Online]. Available: https://github.com/cookie33/irods-compound-resource/blob/master/scripts/univMSSInterface_generic.sh

[11] R. Verkerk. iRODS composable compound resource description at SURFsara. [Online]. Available: https://github.com/cookie33/irods-compound-resource

[12] C. Staiger. iRODS compound resource training. [Online]. Available: https://github.com/EUDAT-Training/B2SAFE-B2STAGE-Training/tree/master/ExampleTrainings/iRODS-SysAdmin-Training

[13] EUDAT. B2SAFE. [Online]. Available: https://eudat.eu/services/userdoc/b2safe