Swedish National Storage Infrastructure for Academic Research with iRODS

Ilari Korhonen
KTH Royal Institute of Technology
SE-100 44 Stockholm, Sweden
ilarik@kth.se

Dejan Vitlacil
KTH Royal Institute of Technology
SE-100 44 Stockholm, Sweden
vitlacil@kth.se

Janos Nagy
Linköping University
SE-581 83 Linköping, Sweden
fconagy@nsc.liu.se

Krishnaveni Chitrapu
Linköping University
SE-581 83 Linköping, Sweden
krishnaveni@nsc.liu.se

Ilker Manap
KTH Royal Institute of Technology
SE-100 44 Stockholm, Sweden
manap@kth.se

ABSTRACT

The Swedish National Infrastructure for Computing (SNIC) has decided to invest in a large scale distributed iRODS-based storage infrastructure for complementing its national storage offering for academic research. Currently the SNIC Swestore national storage service is relying on dCache as its storage solution while it has grown into a petascale operation. However, many users and research groups have expressed interest in different access methods and functionalities (such as the use of multiple different user authentication methods and metadata management) than what can be easily accommodated with dCache. This prompted the investigation of extending the national storage service with the use of different storage technologies. A project was commenced by SNIC for an iRODS-based distributed scalable storage service complementing Swestore. The SNIC iRODS project has now been concluded and the resulting system is being installed into a production environment and integrations are being set in place with other SNIC services. Our accomplishments including but not limited to: a model for the deployment of a geo-replicated iRODS iCAT over two administrative domains with a DNS-based failover mechanism; a model for the deployment of existing and future distributed storage resources within SNIC with iRODS; a novel iRODS interface for tape resources written against the IBM Spectrum Protect (TSM) API; improvement for iRODS logging capabilities with a syslog forwarder; contributions for iRODS user authentication via an alternative PAM authenticator; automated provisioning of iRODS grids and associated services with Ansible, optionally provisioning clusters of VM's with Vagrant for testing; integration of the SNIC iRODS storage service with the SNIC User and Project Repository (SUPR) for provisioning of iRODS users and groups for approved proposals. This solution enables the easy integration of local HPC storage solutions as well as EUDAT, which delivers data to the HPC, HTC and cloud services in Europe.

Keywords

Research data, metadata management, infrastructure.

INTRODUCTION

Swedish National Infrastructure for Computing (SNIC) is a national initiative in Sweden responsible for the financing of most of the High Performance Computing (HPC) and related data storage activities in Sweden. SNIC is being funded by Vetenskapsrådet (Science Council), which itself is a governmental agency in Sweden, tasked to guarantee a high level of research in all fields of science.

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

SNIC distributes funding between the major Swedish HPC centers, of which currently there are six: PDC in KTH Royal Institute of Technology (Stockholm), NSC in Linköping University (Linköping), UPPMAX in Uppsala University (Uppsala), LUNARC in Lund University (Lund), C3SE in Chalmers University (Gothenburg) and HPC2N in Umeå University (Umeå).

SWEDISH NATIONAL STORAGE INFRASTRUCTURE

SNIC coordinates and provides high end computing and storage capacity for Swedish academic research and education. For this purpose, SNIC provides a set of resources to meet the needs of researchers from all scientific disciplines and from all Swedish universities, university colleges and research institutes. Swestore is a National Research Data Storage Infrastructure operated by SNIC.

The resources provided by Swestore are made available through open procedures such that the best Swedish research is supported and new research is facilitated. Prioritization and allocation of resources must be done in a clear and transparent manner, based on scientific quality, scientific need and technical feasibility of using the requested resources efficiently.

The purpose of Swestore allocations, granted by Swedish National Allocations Committee (SNAC), is to provide large scale data storage for 'live' or 'working' research data, also known as active research data.

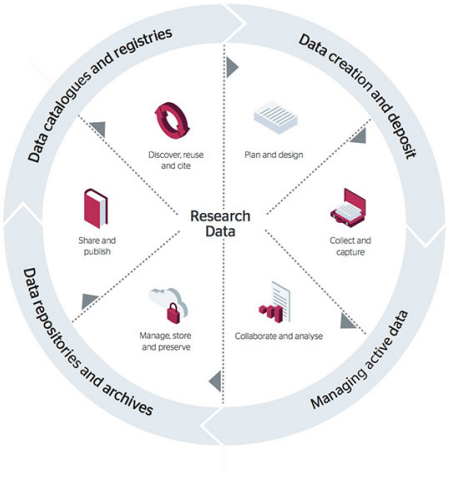

Figure 1. Research Data Lifecycle (©Jisc and Bonner McHardy [CC BY-NC-ND])

Individuals and groups, eligible to apply, can apply for allocations on the available SNIC resources. The entire process is managed by SUPR (SNIC User and Project Repository), which is the self-service portal for users at all SNIC centres. It is the SNIC database used to keep track of persons, projects, project proposals and more.

Once an application has been approved, a project will be established on the resources on which allocations are granted and users are informed. Initial authentication and authorisations are set accordingly. The applicant will be responsible (PI) for the project and must keep an overview of the usage of the allocation(s).

When the research project is over the SNIC users are required to remove their research data to more suitable data services and resources (e.g long-term preservation and archiving, data repositories, data catalogues and registries).

For the past 10 years, SNIC has been engaged in a national storage service for scientific data, called Swestore. While Swestore has grown into a petascale system, supporting many researchers in different fields of science, many new users have requested additional methods of access and user authentication as well as metadata management. The current Swestore service is based on dCache with GSI authentication. This technology was brought to Sweden alongside the

Nordic Tier-1 Storage Infrastructure for WLCG (Worldwide LHC Computing Grid).

With dCache Swestore is able to provide users with a reliable distributed storage service for their scientific datasets. However, for certain more advanced use cases such as (automated) metadata management, other solutions such as iRODS would be more suitable. With the deployment of iRODS, to complement existing Swestore service, these functional limitations of the current Swestore offering could be overcome. This was the motivation for the SNIC iRODS project, which is now in the process of being deployed into production. Within the project, spearheaded by Stockholm KTH PDC and Linköping NSC, we were able to build a scalable and distributed storage system with customisable features.

We developed methods for geo-replication of and failover of the iCAT database as well as methods for for deploying distributed storage resources within Sweden with iRODS. We have integrated the iRODS-based storage environment into the SNIC services side-by-side with the existing SNIC Swestore storage service. We also made some contributions to the iRODS ecosystem, namely: a novel tape interface for iRODS written against the IBM Spectrum Protect (TSM) API; s syslog forwarder which can be used with iRODS logs; an alternate PAM authentication interface with debugging features; and an automated provisioning environment to build iRODS grids with optional provisioning of (disposable) virtual machines for testing.

REDUNDANT GEO-REPLICATED ICAT

For resilience our resource hierarchies are configured for replication of the data and to make the grid redundant, the iCAT server is also (geo) replicated. The iCAT servers are running local PostgreSQL database servers, and using streaming replication to a hot standby which also has a warm standby iCAT server. The entire failover process is scripted, albeit it is manual on intent, since distributed systems are complicated and in this case we thought it is better for the system administrator to be in control, to be able to verify every recovery step when a disaster should happen.

For the iCAT server / PostgreSQL database we use a set of small scripts to initiate the failover. To switch over to the standby servers `dnsupdate` is used to change the IP addresses of the respective machines. We have a subdomain delegated for the iRODS services and we update the addresses in this zone. The zone has a slave in Stockholm to take over. The secondary PostgreSQL server can also be used for read-only queries such as accounting or detailed usage statistics.

In case of failure we trigger the PostgreSQL failover to promote the secondary to primary and we start the standby iCAT server. A monitoring script sets the unavailable resources offline so they will be served from the available replica resources. Finally we update the DNS entries to refer the takeover hosts.

MODEL FOR STORAGE RESOURCE DEPLOYMENT

Because of the distributed nature of SNIC services being coordinated over several HPC centers, SNIC storage services are also deployed with a distributed model. Currently the Swestore service is being operated jointly by several centers with NSC and HPC2N providing the core services and several other centers run dCache storage pools.

With the deployment of iRODS, we follow the same model with KTH PDC and Linköping NSC providing the core services and replicated storage pools in the start of operations. Later more centers can be included with their prospective storage pools. The main principle is to provide high availability of services with geo replication.

We currently rely on iRODS built in replication (coordinating resource) to ensure dual copies of data objects between the two storage pools. This provides an easy way to ensure the successful replication of data. However, this method comes with a drawback, the performance penalty of synchronous replication. Since iRODS handles replication in sequence instead of in parallel, receiving the data object on one of the replicated resources first and then replicating the data object while the client waits for completion of the put operation. With a very fast network connection between the resource servers this affects the overall throughput less, but in our case it is noticeable. Hopefully the

upcoming 100 Gbit/s upgrade of both the KTH PDC network backbone and SUNET connectivity would mitigate this.

Additionally, we have considered using asynchronous replication by implementing the replication as iRODS rules with delayed execution via the rule exec queue. This method would increase the overall throughput of the system with a tradeoff in resiliency. Finally, data locality remains an issue we need to solve. To maximise throughput and minimize network traffic over SUNET, the clients should push the first replica of the data always into the resource which is in the same subnet, and then replicate over SUNET.

In addition to the resources available in the SNIC zone, via federation against EUDAT and KTH PDC iRODS zones, we enable the use of remote zone resources for distinct users granted access. For example a KTH researcher might want to have copies in the SNIC zone and PDC zone as well, since the PDC iRODS resources are in the process of being made visible to the KTH PDC compute clusters via local InfiniBand access. This will be elaborated on in the last 2 sections of this paper.

TAPE LIBRARY ACCESS WITH TSM INTERFACE

Tape mass storage access is a frequent requirement for large scale storage systems, and is even more vital for an archival system. We have long traditions of using tape libraries at both PDC and NSC. The tape libraries are used for backups and archives via the use of IBM TSM (currently called Spectrum Protect, but referred as TSM in this paper).

Since iRODS provides the Universal Mass Storage Interface driver to define a resource accessed via external shell scripts (or executables), we developed a simple utility which can be used with the MSS driver to provide tape storage access using the TSM API. It is implemented in C and it compiles into one executable. It requires a TSM client to be installed and configured, with the TSM API packages also installed and TSM credentials in place.

It implements the MSS operations, which are syncToArch, stageToCache, mkdir, chmod, rm, mv and stat. There are additional operations for housekeeping. It is prepared to handle a list of files to access them in the optimal order, however unfortunately iRODS issues the access requests only one-by-one. To create a tape resource we need a resource hierarchy with a cache resource in front. Then we simply define an MSS resource with the `dsmarc` executable as the driver in the context string of the resource.

The sources are available at `https://github.com/KTH-PDC/irods-dsmarc`.

PAM AUTHENTICATION

An alternate implementation of the iRODS `PamAuthCheck` module was also developed at our project, with more robust error handling and debugging features. PAM authentication provides extreme flexibility which makes the external PAM authentication module is a very useful feature in iRODS. But, for the very same reason PAM authentication can also become rather complicated. To facilitate troubleshooting in the PAM configuration, more verbose debugging and logging functionalities have been added in our implementation.

An additional feature has been implemented to enable per user choice of PAM service files for authentication. With an extra config file instead of the default `irods` PAM service, a different PAM service can be assigned to the user. This provides more choices for the authentication, i.e. one user can use traditional password authentication, while an another a hardware token and so on.

The source distribution is very small and simple, only one C source file and a makefile. With some additional work further enhancement could be possible to provide LDAP integration, so that extra LDAP group membership could decide which PAM service a given user should be associated with.

The sources are available at `https://github.com/KTH-PDC/irods-pamauth`.

FORWARDING OF IRODS LOGS

In a similar fashion to the previous utilities, a simple utility has been developed which watches a set of log files and forwards the new log entries to `syslog` as they are being appended by the application. The utility is implemented in C and it runs as a daemon in the background. The messages are sent to the local `syslog` daemon which will process and forward them if configured to do so.

The sources are available at `https://github.com/KTH-PDC/irods-logforw`.

AUTOMATION

We are consolidating our configurations in Git repositories, currently hosted in GitHub, in our `KTH-PDC` namespace. To later enable continuous integration and for us to be able to test and verify our configurations, we developed an Ansible environment for deploying entire iRODS grids with a built-in replicated iCAT and all services deployed. The package is called `irods-provisioner` and it contains a set of idempotent Ansible *roles* for deploying a certain service or function. Together these form a fully functional iRODS grid.

The deployment of a fully configured iRODS grid into a cluster of virtual machines in VirtualBox using Vagrant can be done as follows with `irods-provisioner`.

```
$ git clone -b 4-1-stable https://github.com/KTH-PDC/irods-provisioner.git
$ cd irods-provisioner
$ vagrant up
$ ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -b -i hosts-test irods-cluster.yml
```

The sources are available at `https://github.com/KTH-PDC/irods-provisioner`.

INTEGRATION INTO SNIC SERVICES

To integrate the new iRODS storage into the existing SNIC Swestore services, we integrated SUPR (SNIC User and Project Repository) into iRODS and FreeIPA, which we use as an Identity Management (IdM) solution.

SUPR and iRODS Integration

SUPR is an online user management and project application/approval system for SNIC. Synchronising user and project data from SUPR to iRODS enables us to have a master source for users and PIs to manage their data and projects both from iRODS and other systems. Prior to this synchronisation, the project request and approval was carried out through support tickets, which can be tough to track for changes.

The SUPR-iRODS synchronisation script uses the Python iRODS Client API supported by the iRODS Consortium. It connects to SUPR periodically and checks for new iRODS projects and users, pulls the user data, creates the user id associated with the user name and creates users, groups and collections for the user(s) in the SNIC iRODS.

SUPR and FreeIPA Integration

FreeIPA is a identity management system (IdM). We use it for SNIC iRODS (and later most likely the entire Swestore) centralised user and password management, to enable the users to manage their user accounts and passwords for both dCache and iRODS. We were able to integrate SUPR and FreeIPA so that the iRODS users from SUPR are synchronised to FreeIPA. The users can then log in to FreeIPA and can set the password for dCache or iRODS.

FEDERATION

The SNIC iRODS grid is federated with the Swedish EUDAT zone (also operated by KTH PDC) as well as the KTH PDC local iRODS zone, for enabling certain users the access of additional resources. This way we are able to deploy an iRODS path from local parallel filesystems at the HPC resources via national resources to European resources.

HPC Integration

At KTH PDC we are deploying a local iRODS grid for HPC users. The most important objective of this iRODS deployment thus is performance. The storage resources at the `pdc.kth.se` iRODS grid are to be accessible via the local InfiniBand fabric via IPoIB. We are developing proof-of-concept models for deploying high performance storage for iRODS resource servers in a scalable and cost-effective fashion. We are aiming at high performance transfers in and out of our 5 PB Lustre filesystem to offload the cluster filesystem storage.

We developed an effective proof-of-concept solution with InfiniBand SRP and ZFS. We are able to access multiple ZFS pools via multipathing over redundant links and separate IB fabrics. With this scalable approach we are able to deploy more JBOD storage to the fabric(s) and by increasing the number of ZFS/iRODS servers more throughput will be available.

Next step in our testing will to introduce the ZFS resource servers to our 100 Gbps EDR InfiniBand fabric at our pre/postprocessing cluster. We will be testing IPoIB performance with iRODS as well as bare IP over 100 Gbps Ethernet parallel transfer performance. So far we were able to reach at maximum ≈ 3,300 MB/s iRODS parallel transfer throughput. This is however achieved with incomplete TCP and hardware tuning (iperf ≈ 42 Gbps max).

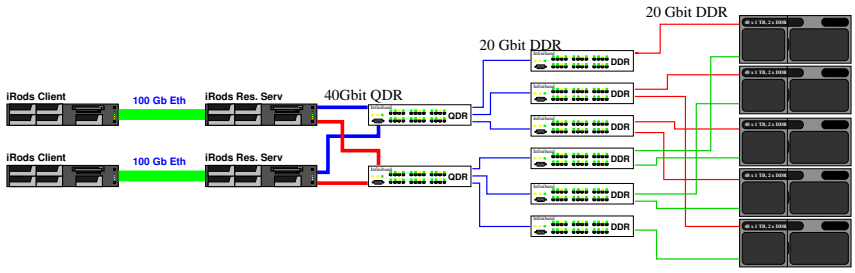

Figure 2. KTH PDC iRODS over 100 Gbps Ethernet testing environment with an InfiniBand SAN