

# OSG Public Storage and iRODS

---

## 1 Problem Statement

One of the goals of the OSG is to provide the Virtual Organizations (VOs) with opportunistic usage of grid resources. One of the essential resources is storage. OSG initiated production-scale *Opportunistic Storage* provisioning and usage on all OSG sites. The major problem for all stakeholders is that the OSG doesn't provide efficient means to manage this storage. There are common issues raised by multiple VOs, such as Engage, SBBGrid, Dzero, NEES, SCEC and others. The essential complains are listed below:

- most of the sites do not support dynamic storage allocation and do not have tools for automatic management;
- the VOs that rely on opportunistic storage have difficulties finding an appropriate storage, verifying its availability and monitoring its utilization;
- a VO needs tools to manage metadata, OSG doesn't offer any common solution;
- the involvement of a Production Manager, Site Admins and VO support personnel is required to allocate or rescind storage space.

## 2 Overview of Proposed Solution

In this document we propose to use iRODS as resource management and data movement services for the OSG public storage. The iCAT-enabled IRODS is capable to register and set quota on public resources per VO as well as per group, and per user. It allows to define resource allocation policies as a set of Rules. Rules trigger a chain of actions (micro-services) that are executed when resource allocation or resource availability has been changed. A chain of actions may include recovery from failures and notification.

In the current proposal iRODS is considered as SaaS. It is deployed on a central node along with iCAT catalog. iCAT catalog contains information about resources, resource usage, quotas and users. It also serves as metadata catalog for the VOs data collections.

## 3 Design Details

### 3.1 Architecture

A high level architecture is shown in the Figure 1. An iRODS instance is hosted on a central node. It is configured with multiple compound resources (remote public storage on sites A, B, and C) and sizeable disk cache that is used as an intermediate hop for data uploads. A Production Manager is able to perform resource allocation per VO. A VO manager is allowed to do resource sub-allocation relevant to his own VO. The authorized VO users may upload files if they are operating within pre-set quota.

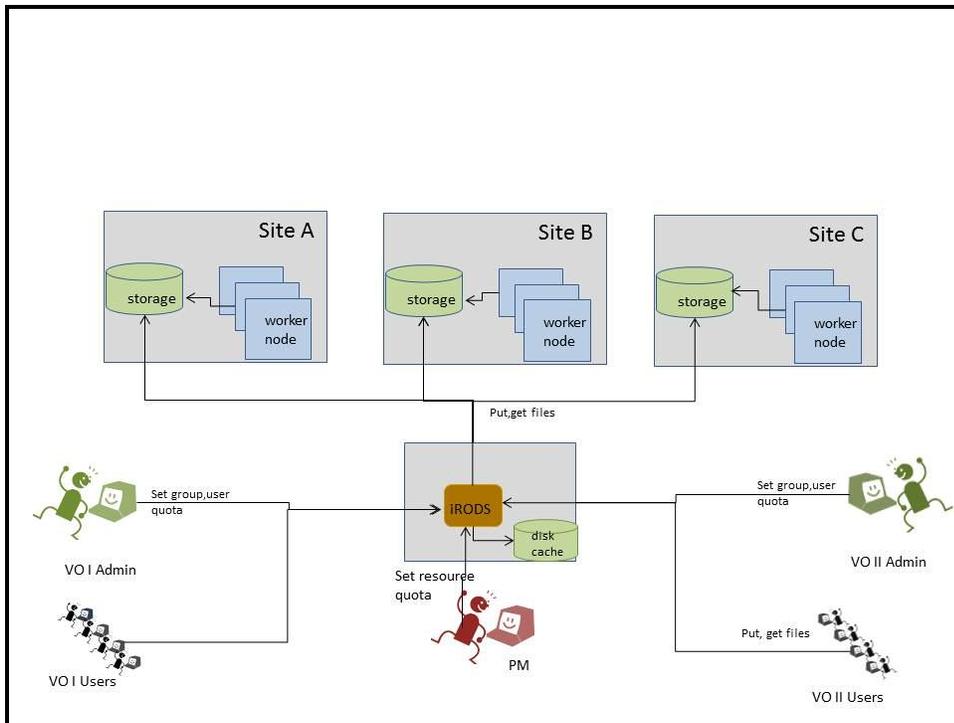


Figure 1 High Level Architecture

The design of a iRODS dedicated node is shown in the Figure 2. iCAT-enabled IRODS is installed on this node. It is configured with customized Rules and micro-services that define the OSG policies on resource allocation and usage . iRODS will use the OSG storage driver for uploading and downloading files to/from the OSG resources. The auxiliary software that automatically registers and validates OSG resources and users may be also installed on this node. The users information is pulled from a relevant VOMS instance . The storage resources could be discovered by querying BDII.

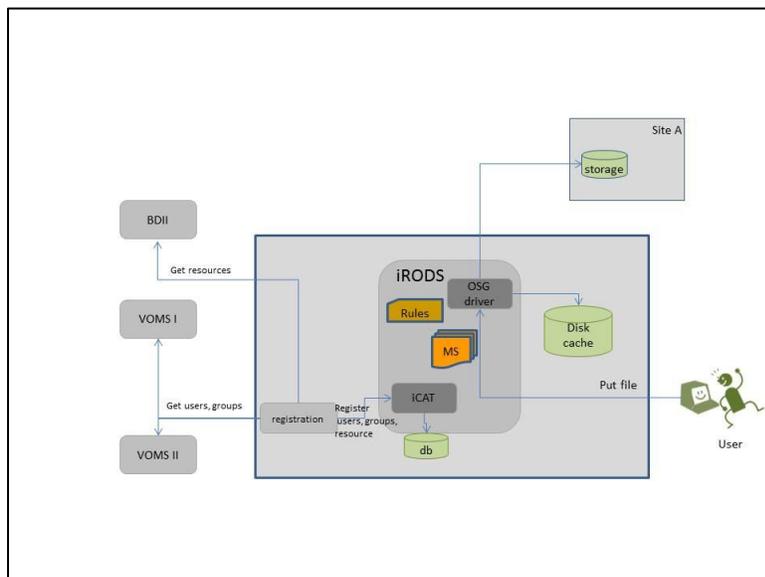


Figure 2 iRODS node layout

The file access for read and write from a job running on the worker node is shown on Figure 3 and 4 respectively. iRODS shouldn't be used if a job needs access a file that has been already pre-staged to a local SE and its location is known (A). In this case a job can either access the file directly through NFS, fuse, etc. or use native storage command to download file into a temporary working area (B). If a file physical location is unknown then the iRODS client command should be executed by user's script to find an appropriate storage resource. If a file needs to be copied into a local storage then a replication request has to be executed via iRODS (C). We are planning to provide a script (e.g using condor transfer plugin mechanism) to facilitate this action.

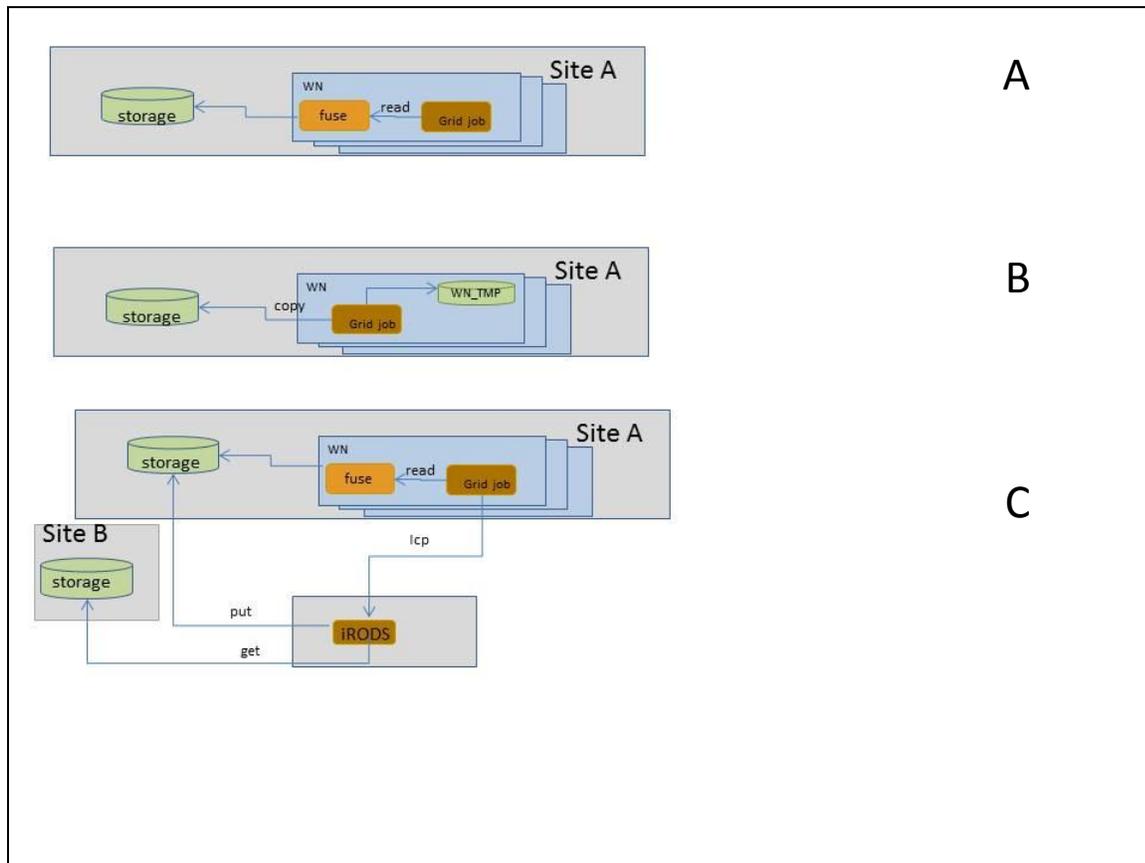


Figure 3 Read access to a file form a worker node

A user's job that is running on a worker node may need to upload data either to a local storage or to a remote storage. In both cases a user's job needs to use a plugin developed by OSG ( e.g the condor transfer plugin mechanism may be used). In the first case (A) iRODS will be contacted after a file is copied successfully to a local storage. The file metadata and the resource used space will be updated. In the second case (B) uploading of user's file to a remote location (e.g home institution) is done via iRODS.

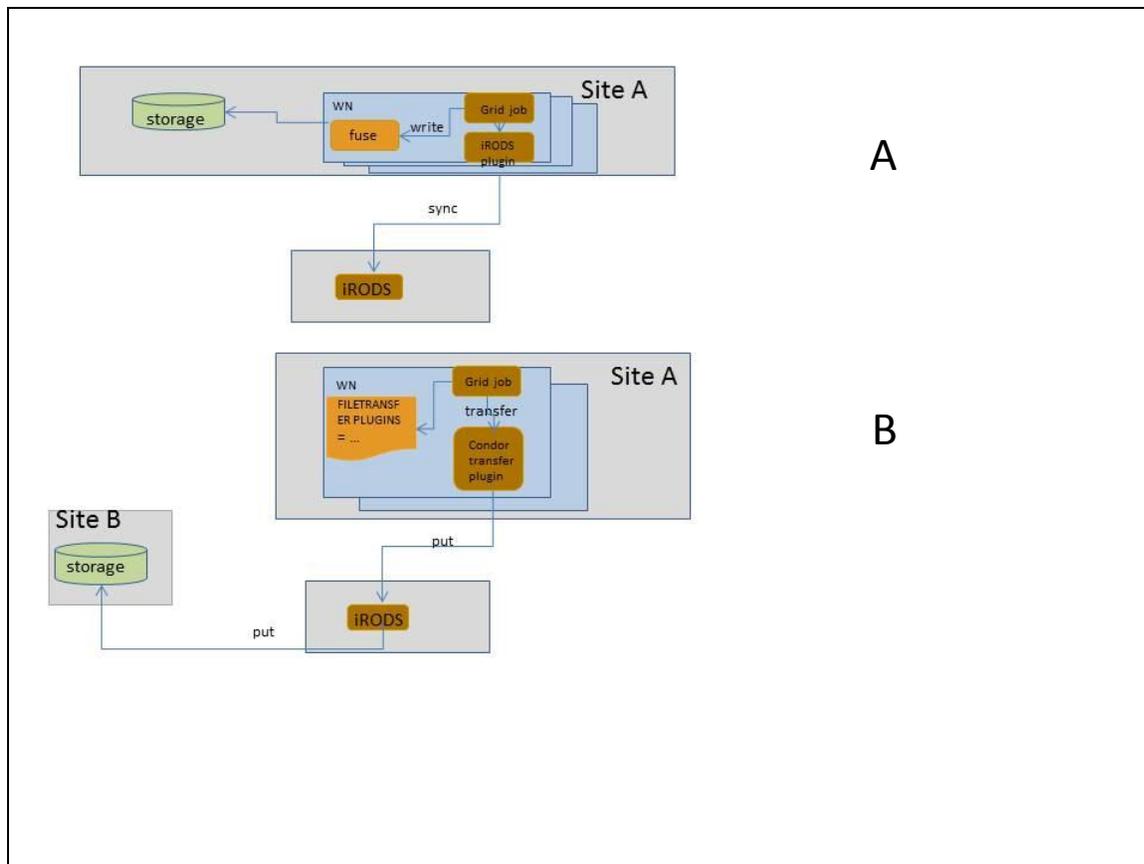


Figure 4 File upload from a worker node

Currently there is no mechanism that could prevent users from uploading files outside of iRODS framework. This action decreases the amount of available space allocated for a VO without adjusting this information in iRODS. The change of space utilization may obviously cause unexpected failures of legitimate file uploads done via iRODS. We propose to run a WatchDog service to monitor all the storage resources in order to minimize consequences of these actions. A WatchDog service is capable of detecting unaccounted files, deleting them from storage, and notifying VO administrators of the affected VO. If need this service could be extended to perform synchronization between a storage resource and iCAT.

## 3.2 Technical Plan

In order to implement the proposed solution the following changes are needed in the OSG infrastructure:

### 3.2.1 iRODS dedicated node:

- A central node run by Operation
- This node should have significant amount of disk that will be used as iRODS disk cache (~tens of TBs)
- At least 2GB Memory for postgres database (iCAT)

- Operation should provide a HA service. Otherwise VOs will not be able to successfully run grid jobs that require files transfers from/to the storage resources
- Installed Software:
  - iRODS
  - iCATS
  - The custom Rules/micro-services . We will need to be in contact with iRODS community in order to get access to existing Rules and micro-services. It is highly probable that we will need to develop some of them.
  - The various scripts to deal with automatic users' registration, resources discovery, registration and validation. These scripts should be developed by OSG. They can be based on tools available from the OSG software stack.
  - The custom driver to access the OSG storage resources. This driver should be developed with help from iRODS developers.

### 3.2.2 Production Manager, VO Submission or User Node

- iRODS client should be installed on all nodes that need to access iRODS

### 3.2.3 OSG sites

- The site admin must allocate a separate partition for all VOs that are using public storage via iRODS.
- iRODS client should be installed on the worker node. It could be done via pilot job or by installing OSG WN stack(see 3.2.4)
- The user with iRODS credential (with VO-iRODS service certificate) should have write permission to all files that belong to users of the same VOs.
- It is possible that WatchDog service should run on the site. It could be submitted as one of VO grid jobs.

### 3.2.4 OSG Software

- iRODS client should be included in the OSG repo . Work on creating rpm that follows OSG software guidelines is required.
- It could be beneficial to include iRODS client in WN rpm

## 3.3 Interfaces supported external to OSG

In order to account for files that are transferred from the XSEDE sites to an OSG storage resource the iRODS client has to be installed on XSEDE worker node and users jobs has to use client's commands to upload file to an OSG resource. Failure to do so will cause files deletion by a WatchDog service.

### 3.4 Performance Goals

A Service Provider (TBD) has to deliver a High Availability Service to prevent grid jobs failures for multiple VOs. It is crucial to measure iRODS scalability and make sure that multiple VOs can upload files via iRODS. If scalability of iCAT becomes an issue it is possible to have multiple iRODS installations and integrate them into federation. This will distribute load to multiple servers and improve performance.

It is very important to understand network requirements in order to prevent iRODS to become a bottleneck for file uploads. It should be always sufficient if bandwidth on a dedicated node is equal to the sum of all available opportunistic VOs bandwidth on all the OSG sites.

It is hard to estimate the current transfer rates for non-owner VOs. Only a few sites are running the transfer probes and many of them don't have reliable data due to misconfiguration of the probes.

According to OSG Gratia accounting non-owner VOs have transferred about ~ 2TB of data during last 30 days to USCD CMS Tier-2 site. This corresponds to ~ 6.5 Mb/s transfer rate. If to assume that about 80 sites would provide public storage then the required throughput rate on the dedicated node should be about 0.5 Gb/s to achieve similar results. Obviously, this is an upper bandwidth threshold. Additional studies of transfer patterns would be very desirable. We also have to account and plan for peak needs.

### 3.5 Known limitations or issues

- When a resource quota is decreased by an administrator the changes in actual resource utilization don't propagate right away. iRODS invokes Rules that define what files should be deleted and then executes a micro-service that performs deletion. If some of the actions fail iRODS will send notification:

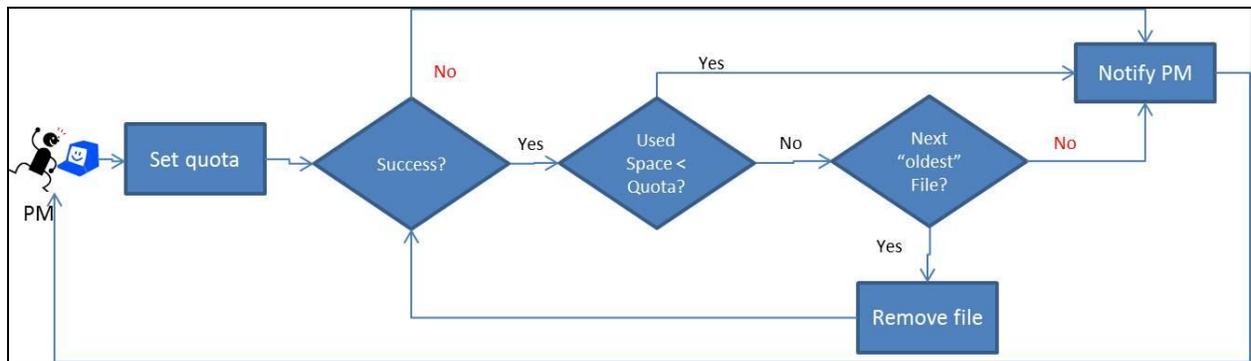


Figure 5 Change of quota example

- The status of a completed grid job cannot be fully trusted. For example if a WatchDog service determines that this an "illegal" file, or iRODS needs to clean up resource to comply to a new quota, the file could be deleted immediately after a job is finished with a successful status.
- The OSG site pool (disk) failure cannot be determined by iRODS, so the used space on this disk will be counted in the resource allocation. We need to come up with some mechanism of purging non-available files from iCAT.
- The additional space added to a resource can be discovered if it is properly updated in bdi but quota needs to be changed by a Production Manager.

### 3.6 Implementation Plan

The implementation plan will include the following steps:

- Acquire a dedicated node with enough memory and reasonable amount of disk for testing
- Install and configure iRODS, iCAT etc
- Develop missing software modules
- Perform various tests to understand iRODS performance and scalability using ITB resources
- Identify a VO that could immediately benefit from get access to managed public storage
- Identify couple of site that are willing to participate
- Install iRODS client on VO/users machine and worker nodes of participating sites.
- Work with a Production Manager to test resource allocation.
- Work with VO admin to test resource sub-allocation
- Help VO to modify their workflow to incorporate needed changes
- Monitor and measure the results of workflow execution
- Simulate resource failure and recovery

## 4 Operational Plan

### 4.1 Impact on sites, VOs (user side) and peers (e.g. XSEDE)

Impact on the OSG sites should be minimal. The OSG sites that offer access to public storage should provide a separate partition for all the participating VOs. VOs should be able to install iRODS client and access public storage via iRODS. The client should be also available on all the XSEDE Service Providers sites if the OSG public storage will be accessed from the jobs that are running on that sites.

### 4.2 Impact on Operations and interfaces to OSG-ET and Council

The Operations will have to carry out the majority of responsibilities in running iRODS and related software. It should be run as a High Availability Service. As a Service Provider the Operations have to cover software support services, including software installations and upgrades. The service upgrades should be announced and performed according to the TBD policy.

## 5 Proposed time-line and resource needs (staff, budget)

We propose to divide the execution of the project into at least three phases. During Phase I, we need to acquire the essential missing software modules and test basic functionalities. The successful outcome of Phase I will allow us to expand the service and test its usability for the real use cases of a selected VO. If

the results satisfy this VO, we will continue with Phase III. The final goal is to deliver a production service that satisfies the need for resource allocation management for the OSG Production Manager and opportunistic VOs, without increasing the responsibilities of the OSG site administrators. The involvement of the IRODS developer team and the establishment of a close relationship with the iRODS community will be crucial for the successful implementation of this plan. Early on in the project we will need to identify a VO that could get immediate benefits by utilizing this service.

## 5.1 Phase I

During Phase I we propose to do the following:

- Acquire a working iRODS osg storage driver. The help from iRODS developers is essential.
- Acquire/Implement iRODS rules and micro-services that allow to do the following:
  - Set quota per resource, vo, group, user
  - Prevent users from uploading files if quota is reached
  - Delete files in order to comply with changed quotas
  - Send notification about success/failure

The help from iRODS developers and access to existing Rules/MS is essential.

- Get access to FermiCloud VM that satisfy memory and storage requirements for test installation
- Install and Configure iRODS
- Register Engage VO, and several users with iRODS
- Register couple of sites (UCSD & Nebraska)
- Perform functionality tests:
  - A PM can set/modify quota
  - A VO admin can set/modify quota
  - User can upload/download files
  - User can not write via iRODS if quota is reached
  - iRODS can delete files from storage if needed
  - iRODS can send notification
- Verify accuracy of resource allocation and usage accounting and monitoring

## 5.2 Phase II

If the results of Phase I are adequate we will continue with Phase II. This Phase consists of the following milestones:

- Implement condor transfer plugin for file transfer from a WN with help of iRODS developers.
- Perform scalability tests
  - Pre-staged data (~100GB?) from a user laptop
  - Upload data from the worker node to remote storage and update iRODS metadata
- Test basic failure condition and recovery.
- Test resource allocation and management with two VOs and two sites. Modify rules if needed
- Identify a VO, users that can benefit from access to public storage via iRODS
- Negotiate with two OSG sites, so a separate partition is allocated and configured
- Help the selected user adopt a new workflow
- If the results are suitable identify another VO

### 5.3 Phase III

Upon successful completion of Phase II we should be ready to involve Operation and migration IRODS from test installation to the appropriate machine. The following steps should be done before this could become a production service:

- Work with iRODS developer team to
  - Implement/test/deploy automatic VO user registration with iRODS
  - Implement/test/deploy automatic resource discovery and verification
  - Implement/test/deploy a WatchDog service
- Negotiate with the OSG sites allocation of X% of storage for public storage via iRODS
- Provide documentation for Operation, Production and VO teams
- Provide reasonable packaging of software
- Work with OSG Software to add iRODS client, related scripts to OSG and WN clients