# iRODS at CC-IN2P3

*Jean-Yves Nief, Pascal Calvat, Yonny Cardenas, Pierre-Yves Jallud, Thomas Kachelhoffer*
CC-IN2P3, CNRS USR 6402, Villeurbanne, France

## Abstract

In this paper, we will show how iRODS is being used at CC-IN2P3, the future plans, code development, and also SRB to iRODS migration.

## Introduction

CC-IN2P3 [1] is a national computing centre located in Lyon (France) which is dedicated to high energy physics, nuclear physics, astrophysics and now is involved in Arts and Humanities projects as well as biology and biomedical applications. It provides computing, storage resources and other services to the French and international scientific community.

iRODS, like its predecessor SRB, is a key service for CC-IN2P3 as it provides the ability to manage large amounts of data which can be distributed across other data centers. These data, produced by instruments or computing simulations, can be accessed and shared from anywhere when scientists within the same experiment or project are spread around the world.

In this paper, we will show how iRODS is used in production and how its usage will evolve in the near future. We will also describe the participation of CC-IN2P3 in the iRODS code development as well as a Java explorer. As SRB is still heavily used in production for several experiments and projects, we will describe plans for the SRB to iRODS migration.

## 1. iRODS in Production

### 1.1 Hardware and software setup

The iRODS service at CC-IN2P3 is supported by 10 servers. It includes:

- 2 servers used to host the iCAT servers: Linux boxes running Scientific Linux 4 and 5 operating system.
- 6 servers are used as non-iCAT servers which are hosting the data: These are Sun X4540 servers on Solaris 10 operating system, ZFS is used for the file system storing files in iRODS. There is a total of 200 TB of disk space available.
- 2 Linux boxes are used to host the Oracle 11g database cluster which is hosting the iCAT databases.

Each project or experiment has its own iRODS instance running on a given port number. Therefore the hardware is shared by all the users of the iRODS service.

The iCAT server is vital to the service. It is a single point of failure and therefore redundancy is needed. To mitigate this we have duplicated the iRODS iCAT servers on two machines which are seen under a unique name DNS alias called “ccirods”: this DNS load-balanced alias is based on software developed at SLAC. Client applications connect to the service through this DNS load-balanced alias.

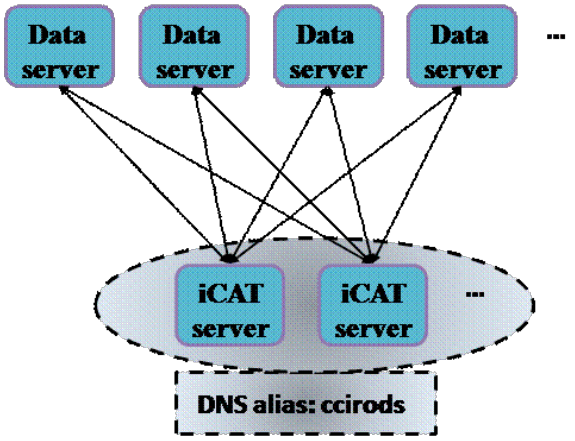


Figure 1: Hardware setup.

We are planning to host more data than can be accommodated by the data servers, and part of the files registered in iRODS will end up on tapes. Therefore the iRODS data servers will be interfaced with HPSS, our Mass Storage System, using the universal Mass Storage driver developed at CC-IN2P3. The transfer protocol used for data migration between iRODS and HPSS is RFIO.

For some projects, parts of their data are unique and must not be lost under any circumstances. We have decided to use Tivoli Storage Manager to do the backup of these files. Other copies of these files are double copied on tapes located in our computing facility as well as another building on campus. We could have used iRODS data replication functionalities with servers located in the other building, but this solution would have been more expensive and was not required, as data can be restored quickly with our current system.

Servers’ health is checked every 30 minutes using Nagios probes which test whether the iRODS instances are responding properly to iRODS connection attempts. In case of server reboot or

daemons disappearance, the servers are restarted automatically by cron tasks.

A report of iRODS usage (number of files, amount of storage space) by users, groups, and experiments is made on a daily basis. The results are reported in our MRTG system and available to all iRODS authenticated users.

### 1.2 Usage examples

iRODS has been in production since 2008 for a few large projects that we will describe briefly in this section.

#### 1.2.1 TIDRA

TIDRA stands for «Traitement Informatique Distribué en Rhône-Alpes». This is the Rhône-Alpes area data grid which federates computing resources in five laboratories spread across Lyon, Grenoble, and Annecy campus, with CC-IN2P3 as the main data center. TIDRA provides computing resources for the Rhône-Alpes scientific community, and iRODS is a key component for the storage and data management part.

It is used in biology applications (phylogeny): jobs are submitted on the grid and access data from iRODS and store job output back into iRODS at a high rate. The number of connections on the iRODS cluster has reached 60,000 per day, with aggregate network activity up to 2 Gbits/s: no obvious limitations have been noticed and therefore I/O activity can be increased without any anticipated problems.

It is also used in biomedical applications such as animal imagery (mice) and human data (heart and lung studies). For DICOM files, the extraction of the header metadata is being done using DCMTK [2], a DICOM toolkit. The metadata are then registered into iRODS in a bulk mode. All these steps are included in a Rule that is triggered automatically on the iRODS server side every time a DICOM file is registered into the system. This allows researchers to search for a data subset based on some metadata criteria.

Other users are expected to use iRODS in the near future, such as researchers from the synchrotron facility in Grenoble (ESRF [3]). There are already 15 users and 3 million files registered in the catalog and we expect to host 20 TB of data by the end of 2010.

#### 1.2.2 Adonis

Adonis [4] is a French national funded project which aims to federate and provide a platform of computing services for the Arts and Humanities national community. Adonis is also connected to European projects like DARIA.

Various projects within Adonis are already using iRODS, which is one of the key services. There are various needs: archival of documents from the Middle Age: data access from batch processing farms (riverbed studies in geography, movie simulations of ancient monuments). iRODS will be also used for data access through web sites, easing web site code development and avoiding having to modify legacy web applications which will be hosted at CC-IN2P3. Fuse-iRODS allows mounting iRODS collection trees as a regular file system on the servers, therefore it is well suited in the present case to adapt web applications to data access through iRODS, without having to change a single line of code.

One of the main Adonis project is to provide a platform to make long-term preservation of data produced by research laboratories spread across France into CINES [5], a national computing facility in Montpellier, and then provide online data access through Fedora-Commons [6] at CC-IN2P3. In the example below (Fig. 2), audio files are produced by researchers from CRDO (Paris). The data are pushed to CINES where they are archived. The files that belong to the same object (i.e. the same family) are pushed in tar balls to CC-IN2P3 using iRODS. On the iRODS servers in Lyon, the files in the tar ball are extracted and registered automatically in Fedora-Commons using an iRODS Rule triggered automatically. The Fedora-Commons storage back end is iRODS: this is achieved using a Fuse-iRODS mount point on the Fedora-Commons server. It was the most obvious way of interfacing iRODS with Fedora-Commons as the present Java interface between the two does not have all needed functionalities. In a future version of the workflow, it is foreseen to also have an iRODS server in CINES, so that once the data produced by any research lab is pushed into iRODS, even the second step of the workflow, i.e. data preservation, would be triggered automatically using iRODS Rules.

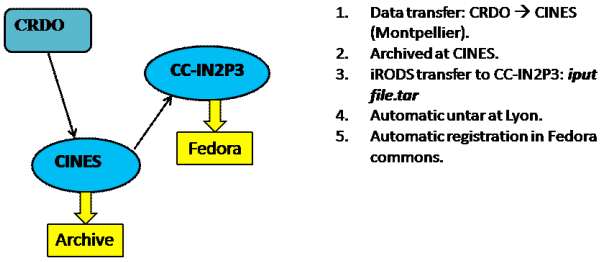


Figure 2: Adonis preservation and publication workflow.

There are already 20 TB of data stored in iRODS, representing 2 million files. This will increase to more than 100 TB by the end of 2010.

### 1.3 Prospects for 2010

iRODS is being adopted by many projects in various fields hosted at CC-IN2P3. They have already started to use iRODS or will start soon. Here is a non-exhaustive list:

- Biology: phylogeny.
- Biomedical applications: animal imagery, cardiology, neuroscience using

magnetoencephalography, positron emission tomography, fMRI, X-ray and gamma ray imagery.

- Astrophysics: LSST [7], JEM-EUSO.
- High Energy Physics: dChooz [8] (neutrino experiment).
- Arts and Humanities: Adonis.

We estimate that iRODS services at CC-IN2P3 will host at least 300 TB of data by the end of 2010. This does not include some projects which are using SRB at the moment and that will migrate to iRODS this year. The petabyte scale will be reached in the near future. Some issues with file names which can potentially contain accented letters must be solved in the area of Arts and Humanities.

## 2. Code Development

In this section, we describe our participation in iRODS code development (author: Jean-Yves Nief) and also a Java-based GUI interface called JUX which can be used to browse iRODS, among other protocols (author: Pascal Calvat).

### 2.1 Scripts

A script has been developed to test iCommand functionalities and ensure that they are behaving as expected: it allows checking that no obvious bug shows up before a new iRODS software version is released.

Another script has been written to do stress tests by launching in parallel a certain number of iRODS operations and measuring the time response of the system.

All these scripts need to be updated as the number of iRODS features has increased significantly in the recent past.

### 2.2 Micro-services and iCommands

A set of Micro-services has been written for several purposes:

- **Access control**: This is a flexible firewall that can be tuned using a configuration file located in *server/config*. It prevents iRODS connections from any set of machines and for any user or group of users that has been specified in the configuration file. It can be triggered simply by using the *acChkHostAccessControl* hook in *core.irb*.
- **Tar file creation and tar file extraction**: these Micro-services can create a tar file from a given output collection and register it automatically into iRODS, and they can extract files from a tar ball registered into iRODS and put the content of the tar archive into a given output collection.
- **Access rights setting**: Sets the access rights on a given input collection or a file.
- **Resource Monitoring system Micro-services**: these will be described in more detail in the subsection below.

An iCommand called iscan has been created: it checks if a local data object or a local directory content is registered in iRODS. This tool is intended for administrators to look for orphan objects on iRODS data servers (i.e. objects not registered in iRODS).

### 2.3 Universal Mass Storage System driver

The goal of this driver is to interface iRODS with any kind of Mass Storage System or any other storage system, using the communication protocol of the administrator's choice (e.g: pftp, rfio, gridftp, hpss etc.) based on the shell commands they are already using. It is an easy way to quickly interface an existing storage system with iRODS without having to use the storage system APIs. This can save time in code development and more importantly users can continue to access this Mass Storage System using the same tools they have been using for direct access, therefore allowing users to maintain a homogeneous way of accessing their MSS system.

This is very flexible and highly configurable on the system and can be configured to provide, for example, HPSS access in a similar manner as the built-in HPSS driver (which uses client libraries to interface). In both HPSS cases, files will be cached between iRODS and HPSS: using compound resources to handle MSS resources is mandatory when using this driver as no direct access to the MSS for the iRODS client is allowed.

### 2.4 Resource Monitoring System

The resource monitoring system has two goals:

- It provides a monitoring system of the servers' activity for a given federation of iRODS servers: it measures the load of each server at a given frequency. The measured quantities are the CPU load, runq load, memory usage, swap memory usage, paging I/O activity, network activity, and disk occupancy on the file systems used by the iRODS physical resources.
- It provides a load balancing system: it gives a measure of the load of each server based on the information extracted above. This information can subsequently be used to choose one physical resource among others for put/get operations.

For monitoring server activity (Fig. 3), a Rule is being executed at a given frequency (say every 10 minutes) and starts a Micro-service called *msiServerMonPerf*. This Micro-service will trigger execution of the script *irodsServerMonPerf* on all (or a subset of) the iRODS servers having physical resources declared: the action is launched on all the target servers at the same time. The *irodsServerMonPerf* script will measure the quantities described above (e.g. CPU load, memory usage etc.):

each quantity is an integer with a value between 0 and 100 (e.g.: CPU load = 0 means that no CPU is used, CPU load = 100 means that 100% of the CPU is used). Once the script has finished, all measurements are stored in a dedicated iCAT table and are used subsequently by other Micro-services described below.

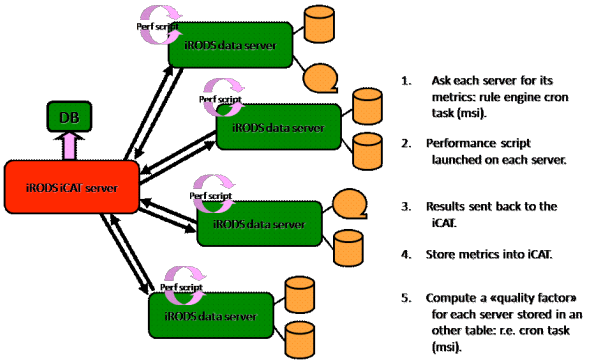


Figure 3: RMS in action.

For the load-balancing system, a Rule is executed at a given frequency (say every 10 minutes) and starts a Micro-service called *msiDigestMonStat* that computes a load factor for each server. This load factor has a value between 0 and 100: a higher number corresponds to a higher server load. This load factor is computed based on the measured quantities above:

$$load = (\alpha \times CPUload + \beta \times MEMload + \gamma \times RUNQload + \delta \times SWAPload + \varepsilon \times PAGEIOload + \theta \times NETload + \mu \times DISKused)/7$$

where α, β, γ, δ, ε, θ and μ must be between 0 and 1 and must be set by the administrator. For instance, if one wants to choose servers based on the CPU load and the network load criteria, then α=0.5 and θ=0.5; all the other factors have to be set to 0. The load factor is stored into a dedicated iCAT table. It can then be used directly by hooks like *acSetRescSchemeForCreate* to pick out the least loaded physical resource within a list of resources.

The remote monitoring system also updates some physical resource metadata such as the disk space available and the resource status, i.e. if it is up and running, or down because it is unreachable.

### 2.5 JUX

JUX [9] stands for "Java Universal eXplorer". The main purpose of JUX is to provide a single Graphical User Interface written in Java to access data stored on different kind of data grids. JUX is intuitive and easy-to-use for non-expert users. Its uncluttered interface uses contextual menus and features like "drag and drop", and is close to widely used explorers such as the Windows Explorer. There are similar tools to JUX such as Hermes [10] developed by James Cook University (Australia) and VBrowser [11] developed by the Virtual Lab for e-Science (Netherlands) which are based on Apache Commons VFS.

JUX is based on the Java implementation of the SAGA specifications called JSAGA [12] and developed by Sylvain Reynaud at CC-IN2P3. JSAGA provides a data management layer as well as security mechanisms. It allows JUX to connect with many different protocols such as iRODS, SRB, gsiftp, SRM, http, sftp, zip, and local file systems using security mechanisms such as login/password or X509 certificates. An iRODS plugin to JSAGA had to be written using Jargon APIs. Files can be copied from one system such as SRB to another such as iRODS in a single "drag and drop". With JUX, it is also possible to display the content of a file (ascii, pictures, audio files) as well as iRODS metadata attached to it (fig. 4). It will be soon possible to search files based on metadata criteria.

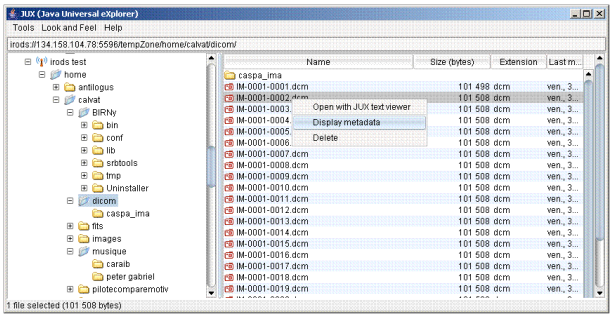


Figure 4: Example of JUX display.

## 3. SRB to iRODS migration

The SRB to iRODS migration is an important topic for CC-IN2P3 as SRB is still heavily used.

### 3.1 SRB usage at CC-IN2P3

Since 2003 the Storage Resource Broker has been used by more than 10 projects and experiments, from high energy physics and nuclear physics, to astroparticle physics, biology and biomedical applications. SRB is a key component of these international projects as CC-IN2P3 plays a major role for them, whether as the central repository or mirror site: SRB is the main repository and means of data access and management for them.

SRB handles more than 2 PB of data on both disk and tape, hundreds of thousands of connections per day, a daily network traffic that can reach 15 to 20 TB. SRB clients span from laptops to supercomputers (e.g. IBM BlueGene), on a wide range of Operating Systems (Windows, Mac OSX, Linux, AIX, Solaris). In order to access the SRB, they use either Scommands (equivalent to iCommands), Java APIs (Jargon), or web services. Connections to SRB come from all over Europe and from as far as Hawaii and Australia. SRB is still growing and will reach 3 PB of data managed by the end of this year.

Among the biggest users are:

- BaBar [13]: a High Energy Physics experiment based at SLAC (Stanford). Data analysis is being performed both at the SLAC computing centre and also at CC-IN2P3. In order to have full data access to end users in Lyon, we designed a two zone system, with automatic synchronization of data between the two sites, allowing receiving newly produced data from the BaBar detector and simulation data at CC-IN2P3 within 24 to 48 hours. Once this system was set up, it has been able to transfer up to 5 TB of data per day from SLAC tapes to CC-IN2P3 tapes, with a minimal amount of manpower and maintenance work. Three million files, corresponding to more than 1 PB of data have been transferred in this way.
- Lattice QCD: this field of theoretical physics produces vast amounts of data. CC-IN2P3 hosts the largest QCD repository in Europe with more than 1 PB stored. These data are produced and accessed from several computing centers in France, Germany, and the Netherlands.

The scalability and reliability of SRB has proven to be extremely important in fulfilling the needs of these experiments. SRB has been a key point for their success.

### 3.2 Migration Plan to iRODS

SRB is a central tool for these experiments, it is heavily used on a daily basis. Therefore, the migration from SRB to iRODS in this production environment must be handled carefully as we need to have minimal disturbances in the process. For projects using Jargon APIs for their client applications, it will be fairly easy and require a minimum amount of work. But the other projects are using shell, Perl, or Python scripts using the Scommands. This will require more work as these scripts can be spread through various parts of the software tools written for the project, and also in the code of the end users. In order to make this migration less painful we are planning to write a small utility that will parse user scripts and detect lines where Scommands need to be replaced by iCommands.

We will begin this migration process this year with BioEmergences [14] (60 TB of data by the end of 2010). We will continue with other projects in the next two years, and hope to finish the migration by the end of 2012. We will not do the migration for projects that have already finished taking data within the last two years: this is the case of BaBar and Supernovae Factory [15].

## 4. Conclusion

SRB has proven to be a powerful data management tool that can be easily adapted to many different needs. It is highly scalable and robust. It is an important requirement for scientific data management as the amount of data and metadata increase at a huge rate. We are now at the scale of the Petabyte, and in just a few years will reach the Exabyte level.

iRODS, with its ability to handle complex data workflows goes far beyond the functionalities of the SRB and any other grid middleware tool. The iRODS Rule mechanism offers a wide range of solutions for data management and great flexibility to adapt to any needs: it can interact in a transparent manner with a large amount of third party software and data storage systems. Its unique features are very appealing for many users, and their feedback so far has been extremely positive. We are also confident that iRODS is well-suited for projects handling hundreds of petabytes and hundreds of millions of files. iRODS is very easy to install on any platform and requires very little maintenance as it is a robust tool. This is a major requirement for CC-IN2P3: a manpower-consuming tool could be an important show stopper for our projects.

We expect to quickly reach the Petabyte scale for iRODS within a year, with an increased number of projects using it. Other developments are envisaged, especially for the Resource Monitoring System and in other areas.

## 5. Acknowledgment

We wish to thank the DICE team for their support and feedback. We also thank T. Kachelhoffer and P-Y. Jallud for their contribution on the Adonis project and Yonny Cardenas for his contribution on TIDRA project.

## 6. References

[1] http://cc.in2p3.fr/
[2] http://dicom.offis.de/dcmtk.php.en
[3] http://www.esrf.eu/
[4] http://www.tge-adonis.fr/
[5] http://www.cines.fr/
[6] http://www.fedora-commons.org/
[7] http://www.lsst.org/lsst
[8] http://doublechooz.in2p3.fr/Public/public.php
[9] https://forge.in2p3.fr/wiki/jux
[10] http://wiki.arcs.org.au/bin/view/Main/HermeS
[11] http://staff.science.uva.nl/~ptdeboer/vbrowser/
[12] http://grid.in2p3.fr/jsaga/
[13] http://www.slac.stanford.edu/BFROOT/
[14] http://www.bioemergences.eu/
[15] http://snfactory.lbl.gov/