Prospects and needs for iRODS at CC-IN2P3

Jean-Yves Nief
Pascal Calvat
Yonny Cardenas
Brief status of iRODS usage at CC-IN2P3:
- Users.
- Architecture.

What is needed or missing for us in iRODS?

Where are we heading with iRODS?
- Future plans.
- Scalibility.
What is CC-IN2P3?

- **IN2P3:**
  - one of the 10 institutes of CNRS.
  - 19 labs dedicated to research in high energy, nuclear physics, astroparticles.

- **CC-IN2P3:**
  - computing resources provider for experiments supported by IN2P3 (own projects and international collaborations).
  - resources opened both to French and foreign scientists.
Researchers of various disciplines:
- Data sharing, management and distribution.
- Data processing.
- Data archival.

- Physics:
  - High Energy Physics
  - Nuclear Physics
  - Astroparticle
  - Astrophysics
  - Fluid mechanics
  - Nanotechnology

- Biology:
  - Genetics, phylogenetics
  - Ecology

- Biomedical:
  - Neuroscience
  - Medical imagery
  - Pharmacology (in silico)

- Arts and Humanities:
  - Archeology
  - Digital document storage
  - Economic studies

- Computer science
Prospects and needs for iRODS at CC-IN2P3 – iRODS user meeting 2015, Chapel Hill
23 zones.

41 groups.

469 users:
- Maximum of 800k connections per day.
- Maximum of 6.4m connections per month.

80 millions of files.

9700 TBs of data as of today:
- Up to +30 TBs growing rate per day.
Local or external batch systems. WAN and LAN (thousands).

iRODS Server side architecture

15 Data Servers (DAS): 840 TBs

Database cluster: Oracle 11g RAC

ccirods (DNS alias)

iCAT Server

HPSS

100 Gbps

iCAT Server

clients
Connection control:
- Very important as client activity not under control.
- Can go wild (especially memory usage):
  - Usage of CCMS: mainly protecting the iCATs.

Improvements needed:
- Better to queue the client requests instead of rejecting them immediately.
- If connection limit reached, even privileged user cannot login.
Rule management:

- Up to 10 of thousands of delayed rules in queue (eg: replication onto tapes).
- Some rules (like admin rules) have higher priority (eg: disk cleanup).

Scheduling priority needed: no need for fancy scheduling.

- Adding a name stick to rule id: easier to manage (for iqdel etc…).
- Rule information stored on local filesystem? Could be in the database? (easier to have several irodsReServer in the same zone).
On the client side.

- **Windows support.**
  - icommands v3.0 still being used.

- **Should remain multiplatform as it is (various Unix flavors used).**

- **GUIs:**
  - Old Windows explorer still being used.
  - Unmaintained php browser still being used.

- **APIs:**
  - Support of PHP APIs.

- **Bulk download:**
  - Extremely important for collections with small files.
GUI example (developed at CC-IN2P3)
On the fly compression for upload/download (like what has been made for myirods with snappy lib).

File versionning:
- Extra value for small sites: could be a solution for sites needing a backup solution.

Automatic replication of « hot » files between two physical resources.

Monitoring/accounting modules added to iRODS (for small sites):
- Everybody cannot have Nagios, Elastic Search etc…
Volume increase: linear growth now.

Reaching 10 PBs very soon.

Massive migration to v4.

Provide a REST interface to our storage systems through iRODS.

Medium term archival service build on iRODS?

Scalability: throttle the clients requests, avoid overwhelming the servers.

Feeling the rising competition with cloud technologies:
  ◦ Even though they do not provide the same services or just a subset of what iRODS provides.