Managing Petabytes of data with iRODS

Jean-Yves Nief

CC-IN2P3

France

Talk overview

- Data management context.
- Some data management goals:
  - Storage virtualization.
  - Virtualization of the data management policy.
- Examples of data management rules.
- Why choose iRODS ?
- What is iRODS ?
- iRODS usage examples.
- Propects: scalability, data protection.

Data management context

- Data centers like CC-IN2P3 (Lyon, France) works for international scientific collaborations.
- Examples:
  - High Energy Physics: CERN (Fr/Switzeland), SLAC (USA), Fermilab (USA), BNL (USA) etc…
  - Astroparticle physics / astrophysics: Auger (Argentina), HESS (Namibia), AMS (Int. Space Station).

➔ ***Distributed environment: experimental sites, data centers, collaborators spread around the world.***

Experimental sites and collaborating Data Centers.

Data management context

- Multidisciplinary environment:
  - High Energy and nuclear physics.
  - Astrophysics.
  - Biology.
  - Biomedical applications.
  - Arts and Humanities.
  - Private partners.

➔ ***Various constraints, various needs for data management.***

Data management context

- Data stored on various sites.
- Heterogeneous storage:
  - Data format: flat files, databases, data streams…
  - Storage media, server hardware: disks , tapes.
  - Data access protocols, information systems.
- Heterogeneous OS on both clients and servers side.

➔ ***Needs to federate all this in a homogeneous way.***

Data management context

- Data deluge.
- Eg: storage needs follows the Moore's law so far for our Mass Storage System (14 PBs now, could be 52 PBs in 2015 ???).
  - E.g. CERN : 4 PBs / year.
- Future science projects like LSST (astro), even bigger: ~ 10 PBs / year of raw data.
  - derived products: **Exabytes scale** !!! (SuperNovae search)

Some data management goals

- *Not in the scope here:*
  - *Intensive parallel I/O for data analysis.*
- In the scope here:
  - Data preservation (replication, consistency …).
  - Data access distributed over different sites.
  - Data life cycle (file format transformation, data workflows, interactions with various info systems).
- Need for virtualization of the storage:
  - Logical view and organization of the data.
  - ➔ Data migration to new hardware/software transparent to the end clients tools: no view of the physical location of the data and underneath technologies.
  - ➔ Logical view of the data unique to all the users independently of their location.
  - Virtual organization (VO) of the users:
    - Unique id for each user.
    - Organization by groups, role (simple user, sysadmin etc…).
    - Access rights to the data within the VO.

Some data management goals

- Storage virtualization not enough.
- For client applications relying on these middlewares:
  - No safeguard.
  - No guarantee of a strict application of the data preservation policy.
- Real need for a data distribution project to define a coherent and homogeneous policy for:
  - data management.
  - storage resource management.

➔ Crucial for massive archival projects (digital libraries …).

➔ No data grid tool had these features until 2006.

Virtualization of the data management policy

- Typical pitfalls:
  - No respect of given pre-established rules.
  - Several data management applications may co-exist at the same moment.
  - Several versions of the same application can be used within a project at the same time.

  ➔ *potential inconsistency.*
- Remove various constraints for various sites from the client applications.
- Solution:
  - Data management policy virtualization.
  - Policy expressed in terms of rules.

Examples of data management rules

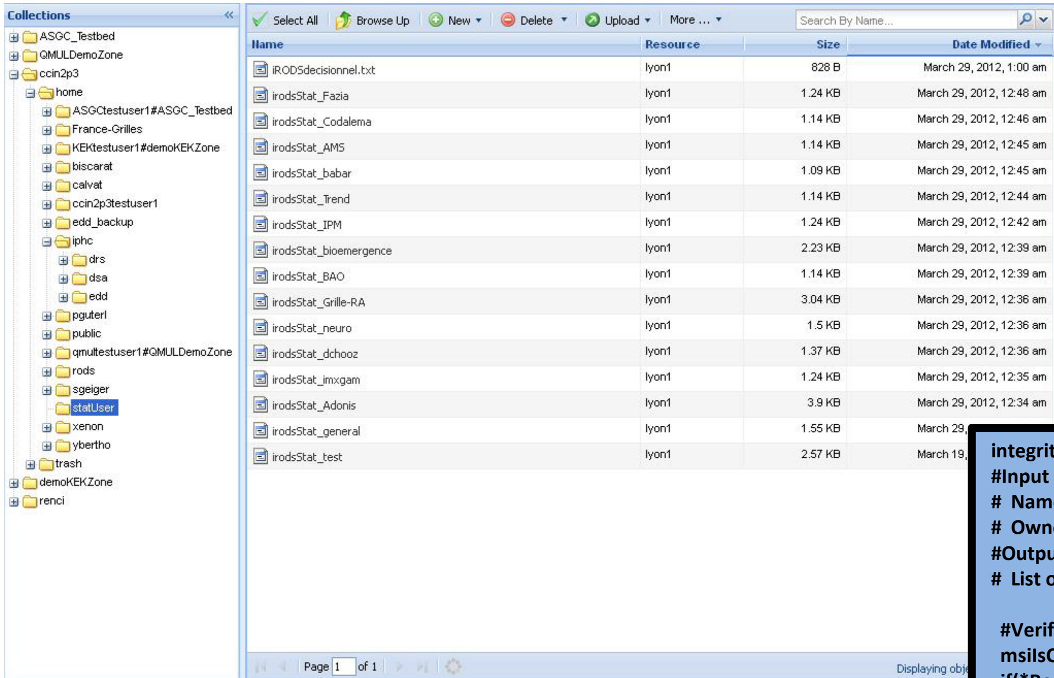
- Customized access rights to the system:
  - Disallow file removal from a particular directory even by the owner.
- Security and integrity check of the data:
  - Automatic checksum launched in the background.
  - On the fly anonymization of the files even if it has not been made by the client.
- Metadata registration:
  - Automated metadata registration associated to objects (inside or outside the iRODS database).
- Small files aggregation before migration to MSS.
- Customized transfer parameters:
  - Number of streams, stream size, TCP window as a function of the client or server IP.
- … up to your needs …

Why choose iRODS ?

- Provide a solution to the above requirements.
- SRB (iRODS predecessor) has been used so far:
  - Data virtualization.
  - But no policy rule based mechanisms.
- In 2007, no « grid » tools except iRODS could provide data management policies based on rule.
- Scalable.
- Can be customized to fit a wide variety of use cases.

What is iRODS ?

- **iR**ule **O**riented **D**ata **S**ystems (DICE team: UNC, San Diego):
  - started in 2006.
  - open source.
- In a « zone » (administrative domain):
  - One or several several servers connected to a Centralized Metacatalog (RDBMS) with files metadata, user informations, data locations etc… ➔ Logical view of the data in a given ***zone***.
  - Data servers spread geographically within a zone.
- Possibility to have different ***zones*** (separate administrative domains) interconnected.
- Data management policies expressed with rules in a « C-like » language:
  - Can be triggered automatically for various actions (put, get, list, rename….).
  - Can be run manually.
  - Can be run in batch mode.
  - Rules versioning.
- Client interactions with iRODS:
  - APIs (C, Java, PHP, Python), shell commands, GUIs, web interfaces.



Logical name space (example of a web interface)

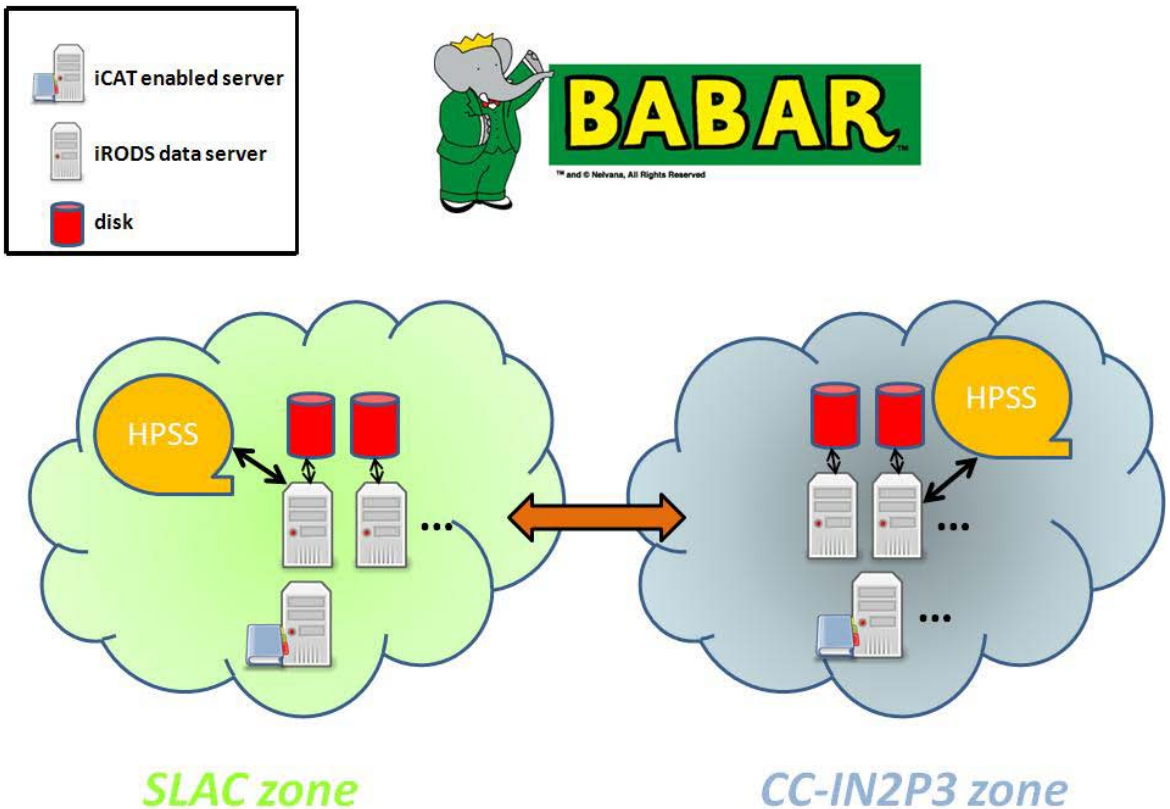
Rule example (e.g.: *Are all the files in a given collection all owned by a given user ?*)

```
integrityFileOwner {
#Input parameter is:
#  Name of collection that will be checked
#  Owner name that will be verified
#Output is:
#  List of all files in the collection that have a different owner

 #Verify that input path is a collection
 msiIsColl(*Coll,*Result, *Status);
 if(*Result == 0) {
  writeLine("stdout","Input path *Coll is not a collection");
  fail;
 }
 *ContInxOld = 1;
 *Count = 0;

 #Loop over files in the collection
 msiMakeGenQuery("DATA_ID,DATA_NAME,DATA_OWNER_NAME","COLL_NAME =
'*Coll'",*GenQInp);
 msiExecGenQuery(*GenQInp, *GenQOut);
 msiGetContInxFromGenQueryOut(*GenQOut,*ContInxNew);
 while(*ContInxOld > 0) {
  foreach(*GenQOut) {
   msiGetValByKey(*GenQOut,"DATA_OWNER_NAME",*Attrname);
   if(*Attrname != *Attr) {
    msiGetValByKey(*GenQOut,"DATA_NAME",*File);
    writeLine("stdout","File *File has owner *Attrname");
    *Count = *Count + 1;
   }
  }
  *ContInxOld = *ContInxNew;
  if(*ContInxOld > 0) {msiGetMoreRows(*GenQInp,*GenQOut,*ContInxNew);}
 }
 writeLine("stdout","Number of files in *Coll with owner other than *Attr is *Count");
}
INPUT *Coll = "/tempZone/home/rods/sub1", *Attr = "rods"
OUTPUT ruleExecOut
```

iRODS usage @ CC-IN2P3

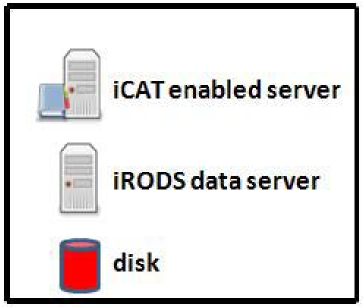
- Used for data distribution, archiving, integration in analysis or data life cycle management workflows.
- Interfaced with:
  - Mass Storage Systems: HPSS.
  - External databases or information systems: RDBMS, Fedora Commons.
  - Web servers.
- **High energy and nuclear physics**:
  - BaBar: data management of the entire data set between SLAC and CC-IN2P3: total foreseen 2PBs.
  - dChooz: neutrino experiment (France, USA, Japan etc…): 400 TBs.
- **Astroparticle and astrophysics**:
  - AMS: cosmic ray experiment on the International Space Station (280 TBs).
  - TREND, BAOradio: radioastronomy (170 TBs).
- **Biology and biomedical applications**: phylogenetics, neuroscience, cardiology (50 TBs).
- **Arts and Humanities**: Adonis (46 TBs).

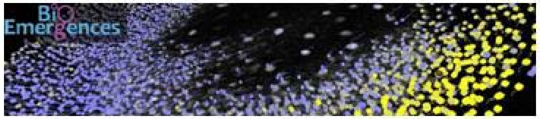
iRODS usage examples

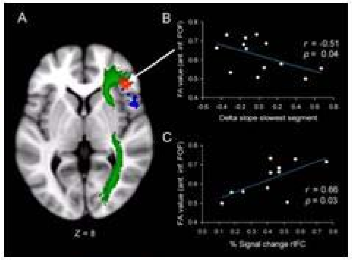


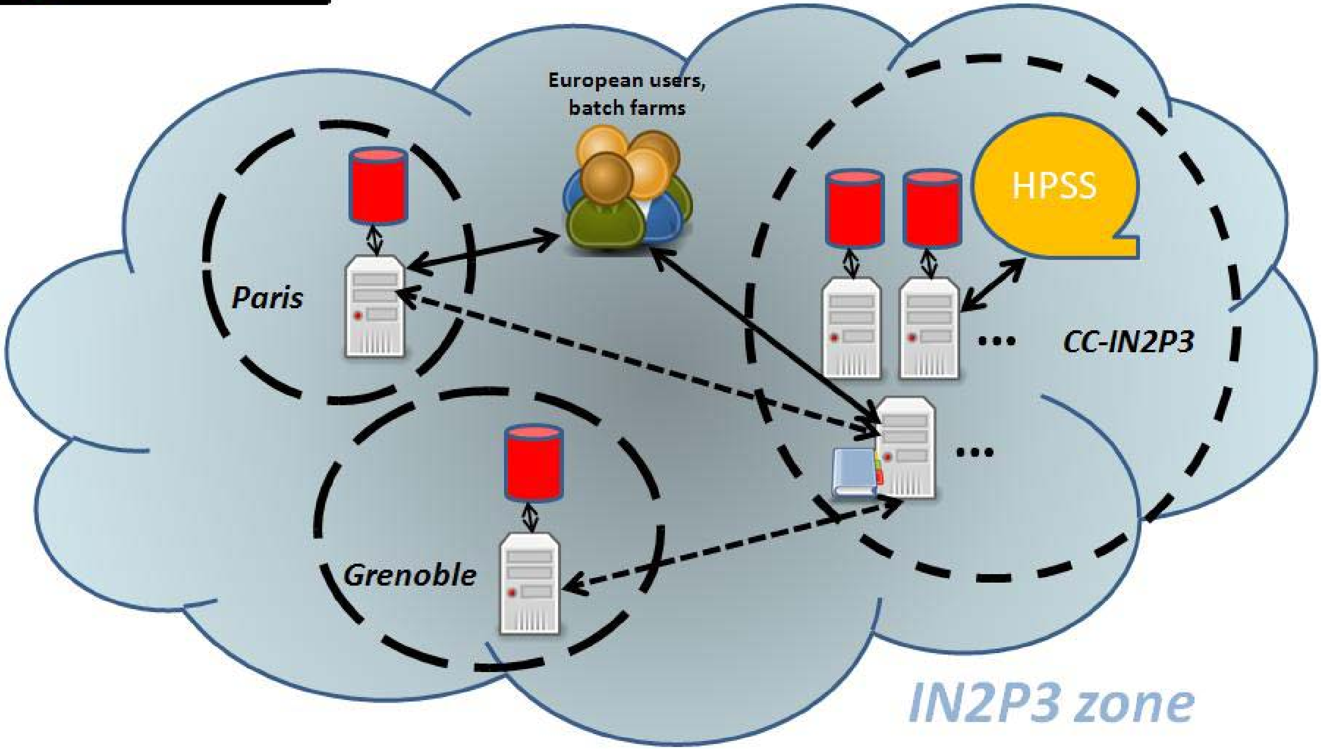
- archival in Lyon of the entire BaBar data set (total of 2 PBs).
- automatic transfer from tape to tape: 3 TBs/day (no limitation).
- automatic recovery of faulty transfers.
- ability for a SLAC admin to recover files directly from the CC-IN2P3 zone if data lost at SLAC.

iRODS usage examples (biology, neuroscience)









Rule examples: biomedical data

- Human and animal data (fMRI, PET, MEG etc…).
- Usually in DICOM format.
- Main issue for human data:
  - Need to be anonymized !
- Need to do metadata search on DICOM files.

➔ Rule:

1. Check for anonymization of the file: send a warning if not true.
2. Extract a subset of metadata (based on a list stored in iRODS) from DICOM files.
3. Add these metadata as user defined metadata in iRODS.

- Archival and data publication of audio files for Arts and Humanities:
  - Tar ball registered in a archive.
  - Pushed into iRODS and « untarred » automatically.
  - Published and registered automatically into Fedora Commons.

Other rule examples

- Mass Storage System integration:
  - Using compound resources: iRODS disk cache + tapes.
  - Data on disk cache replication into MSS asynchronously (1h later) using a delayExec rule.
  - Recovery mechanism: retries until success, delay between each retries is doubled at each round.
- ACL management:
  - Rules needed for fine granularity access rights management.
  - Eg:
    - 3 groups of users (**admins**, **experts**, **users**).
    - ACLs on /<zone-name>/*/rawdata => **admins** : *r/w*, **experts** + **users** : *r*
    - ACLs on all others subcollections => **admins** + **experts** : *r/w*, **users** : *r*

Prospects: scalability

- 1.7 PBs managed by iRODS so far.
- 5 PBs expected until the end of 2012 (include migration from SRB to iRODS).

➔ No scalability issues foreseen.

- Pitfalls:
  - Metadata scalability ? (billions of entries in the catalog ?).
  - Control of the number of simultaneous connections to be enforced (like for Apache servers): needed in a wide opened environment.

Prospects: data protection

- Medical data records:
  - Medical data anonymization (*disallow registration of non anonymized data outside the hospitals infrastructure*): will be implemented for Multiple Schlerosis database (32,000 patients, more than 20 hospitals).
  - Encrypted connections, site-to-site VPN between hospitals and data centers ?
- Sensitive data for private business:
  - Data encryption and/or secured connections ?

Acknowledgements

- DICE team.
- CC-IN2P3:
  - Pascal Calvat.
  - Yonny Cardenas.
  - Rachid Lemrani.
  - Thomas Kachelhoffer.
  - Pierre-Yves Jallud.
- SLAC:
  - Wilko Kroeger.

References

- iRODS: https://www.irods.org/index.php/Main_Page
- CC-IN2P3: http://cc.in2p3.fr/
- BaBar: http://www.slac.stanford.edu/BFROOT/
- LSST: http://www.lsst.org/lsst/