Xrootd status and ongoing/future work

(Status report)

Pavel Jakl

S&C meeting

19th of April 2006

Outline

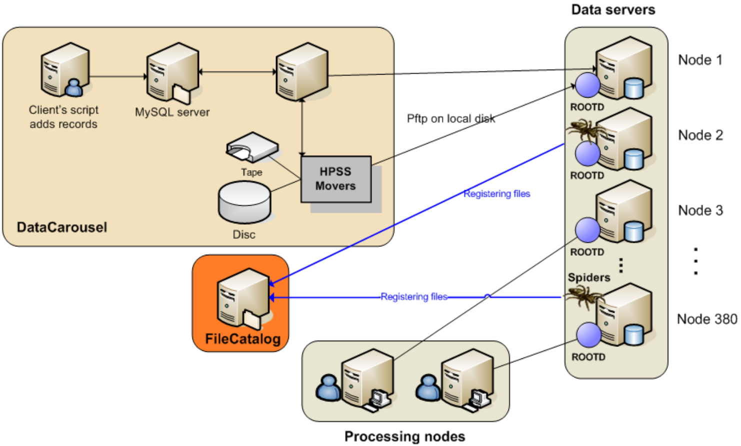
1. Motivation at the beginning
2. Introducing XROOTD solution and deployment
3. Ongoing/future work
4. Summary

RHIC Computing facility

- 3 storages for data population:
  1. **HPSS** - all data (raw, reconstructed) are stored there, each PFN is unique
  2. **NFS area** - about 75 TB of free space, is often overloaded, therefore lots of disruptions and not reliable
  3. **Distributed disk** - about 130TB of free space decomposed on about 320 nodes, not possible to manage it with NFS
- Question: How to best utilize the storage space on nodes?
- Solution: **ROOTD** - daemon which provides ROOT-based access to remote files

Introduce static model of ROOTD

STAR distributed data model: " Started with very homemade and very **static** model "



Problems with ROOTD model

1. ROOTD knows only PFN
   - rootd doesn’t know where the data are located -> data needs to be cataloged and kept up-to-date
2. Overloaded and not responding node
   - rootd connection will expire after defined time and job will die
3. Job start time latency
   - catalog is not updated accordingly when node is down for maintenance
   - job dies when requested files are deleted between the time "a" job is submitted and starts
4. Static data population
   - human interaction is needed to populate data from HPSS to distributed area
   - datasets need to be watch (datasets gets "smaller" in case of disk reset/format)
5. Write access and authorization issue
   - everyone in rootd is "trusted" user (missing authorization)

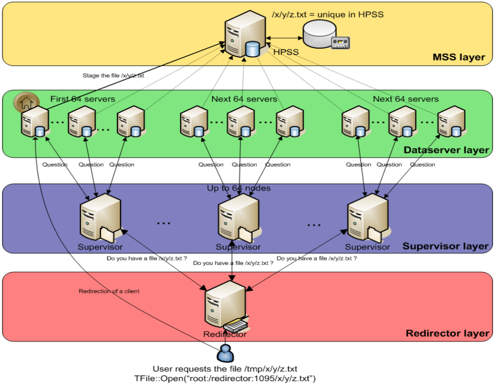
Solve rootd problems with xrootd features

- **XROOTD** - file server which provides high performance file-based access(scalable, secure, fault-tolerant . . .)

1. ROOTD knows only PFN ->XROOTD knows "LFN"
   - data are located within xrootd process and no need to be catalogized
2. Overloaded and not responding node ->Load balancing
   - xrootd determines which server is the best for client’s request to open a file
3. Job start time latency -> Fault tolerance feature
   - missing data can be again restored from MSS
4. Static data population ->Mass storage system plugin
   - movement from **static** population of data to **dynamic**
5. Write access(authorization) issue -> Authorization plugin
   - resolve "trusted/untrusted" user for write access

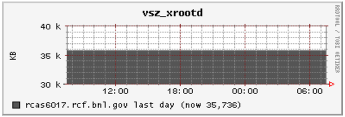
XROOTD configuration/auto-configuration

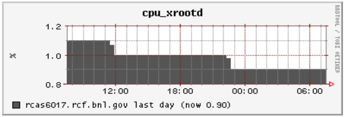
- preparation of the configuration file containing configuration of load balancing, authentication and MSS plugin
- implementation and testing of xrootd daemons managing tools



Integration into STAR

- integration with current framework - as for example new features into SUMS (Star Unified Meta Scheduler)
- conversion of all PFNs (already placed files on STAR distributed disk) into XROOTD "LFNs"
- script for monitoring: using the Ganglia cluster toolkit





Problems and repairs/contribution:

1. Needed to wait for the 64 node limitations removal (reported in February 2005, available in April/May 2005)
2. Different security model:
    - we were beta testers
    - shaky initial implementation and documentation
3. ROOTD does only PFN, Xrootd cannot do both PFN and LFN
    - it is a question of **how** to convert a request to a PFN
    - LFN->PFN is now done in a fix way("one choice fits all")
    - provide a plugin would be more flexible (discussed in July 2005, interface available in January 2006)
4. non-functional script for meassuring the load of servers repair was sent to xrootd development team
5. un-coordinated requests to HPSS (in 20 jobs the HPSS crashed) -> solution is to use DataCarousel (in progress)

Ongoing/future work

- need additional work and improvements on DataCarousel solution
  - discs are sometimes filled up to 100% -> bad decomposition of requests among xrootd cluster
  - need to set up and test purging policies for unprompted cleaning of filled space
- need to test in large scale (not only 2 users), even without HPSS plugin
- set-up monitoring system of xrootd cluster (measure data movement on the farm)
- bytes/sec measurement of NFS/XROOTD compare to number of running jobs
- fault-tolerance measurement to compare number of died jobs ROOTD/XROOTD
- Long-term: Integration with SRM (Storage resource manager)

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)
- modulo few fixes in year 2005 the system looks stable and ready to use in production mode!!!
    - When ? end of this week (beginning of the next week)
    - without HPSS plugin - still need to work more on DataCarousel solution
    - HPSS plugin will be available end of this month
- no need to change anything in user's macros -> new "xrootd" fileList syntax already in SUMS
- load balancing and handshake with MSS make the system resilient to failures
- the monitoring of XROOTD behavior in large scale scale and over long period of time haven't shown significant impact on CPU on nodes