From rootd to xrootd, from PFN to LFN

(experience on accessing and managing distributed data)

Jérôme Lauret[1] **Pavel Jakl**[2] Andrew Hanushevsky[3]
Arie Shoshani[4] Alex Sim[4]

[1]Brookhaven National Laboratory, United states of America

[2]Nuclear Physics Institute, Czech Republic

[3]Stanford Linear Accelerator Center, United states of America

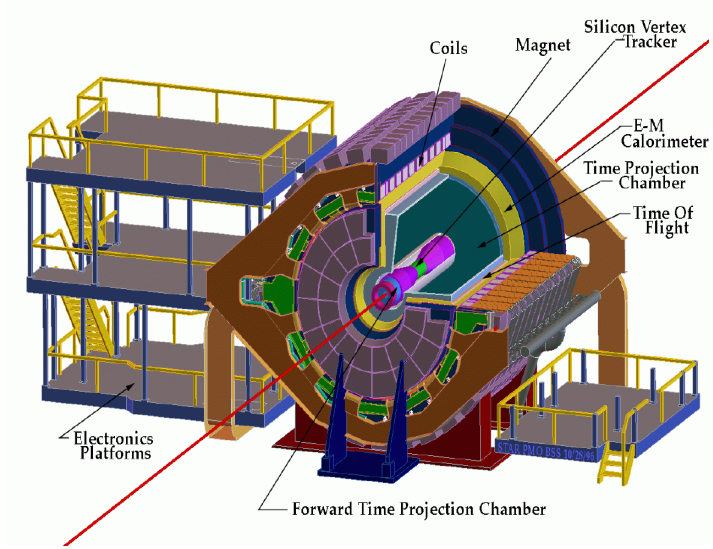
[4]Lawrence Berkeley National Laboratory, United states of America

Computing in High energy and Nuclear Physics 2006,
Mumbai, India
15.2.2006

Outline

STAR Experiment at a glance

- detector located at the 6 o'clock position at the Relativistic Heavy Ion collider ring at BNL (USA)
- STAR is designed to study the behavior of strongly interacting matter at high energy density and to search for signatures of Quark Gluon Plasma (QGP) formation
- a PByte scale experiment overall (raw, reconstructed events) with several millions of files



RHIC Computing facility

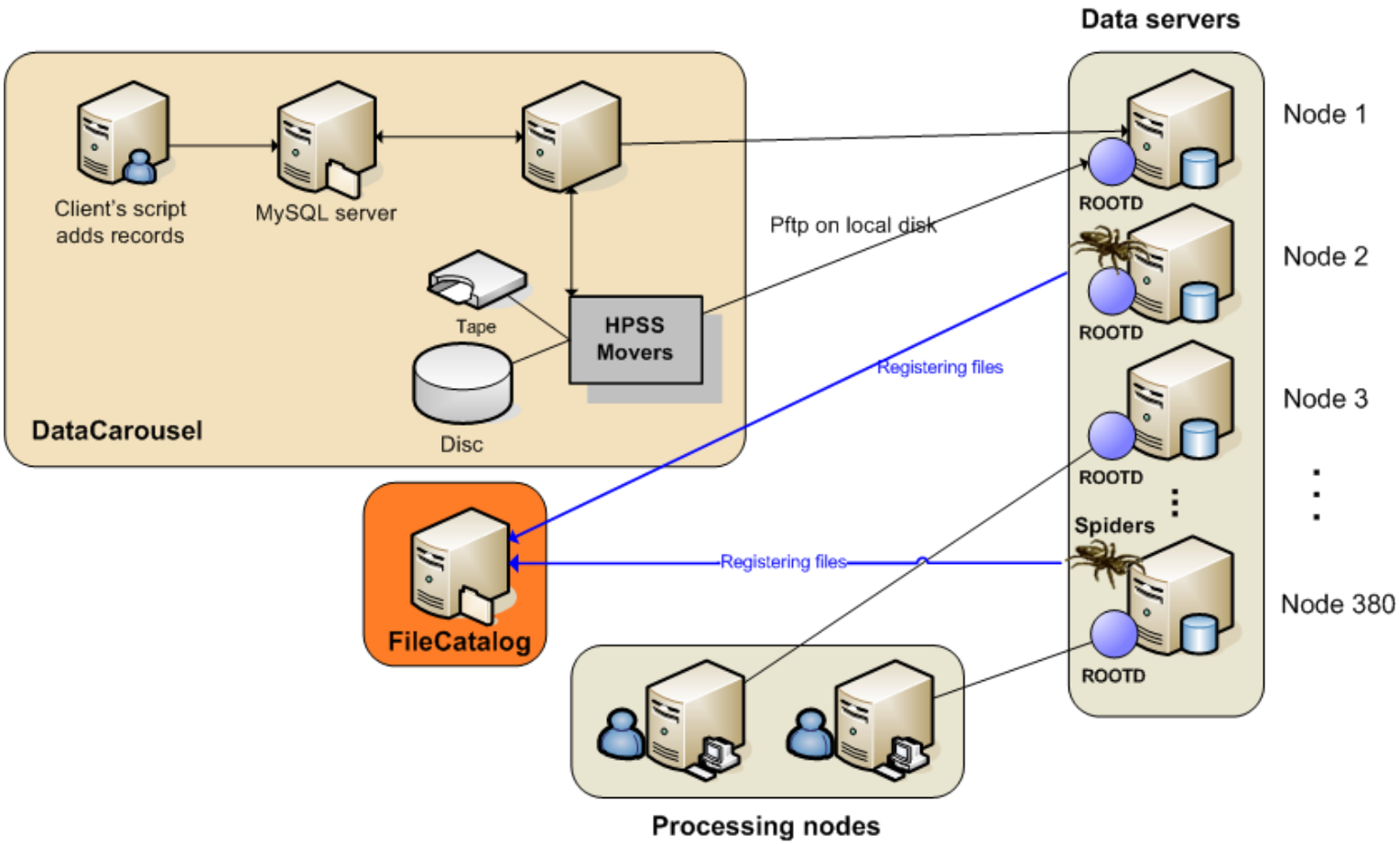
- STAR computing resources are allocated:
  - **STAR CAS Linux cluster** - cluster for analyses of users, about 320 nodes and 130TB of disk space
  - **STAR CRS Linux cluster** - cluster designated for the purpose of raw data reconstruction, about 200 nodes
- 3 storages for data population:
  1. **HPSS** - 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

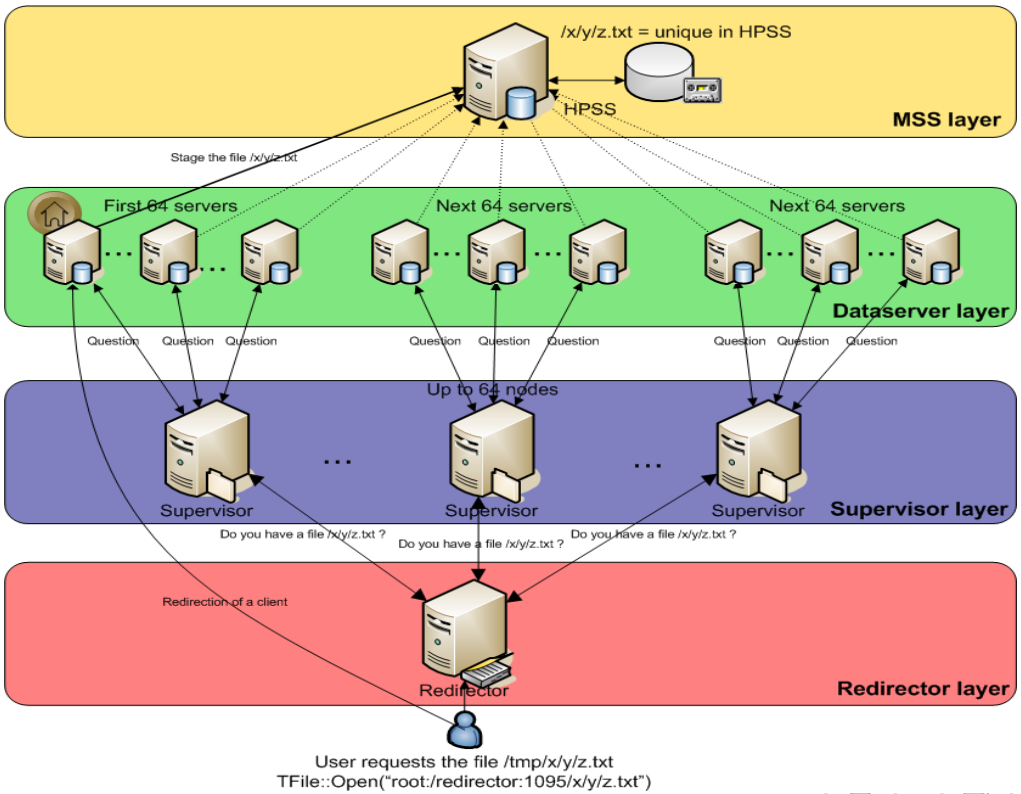
1. ROOTD knows only PFN
   - rootd doesn't know where the data are located -> data needs to be cataloged and kept up-to-date
2. Overloaded and not responding node
   - rootd connection will expire after defined time and job will die
3. Job start time latency
   - catalog is not updated accordingly when 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

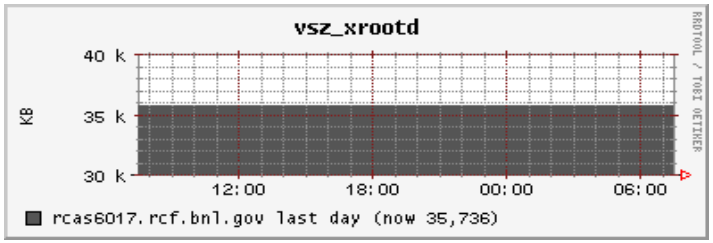
XROOTD configuration/auto-configuration

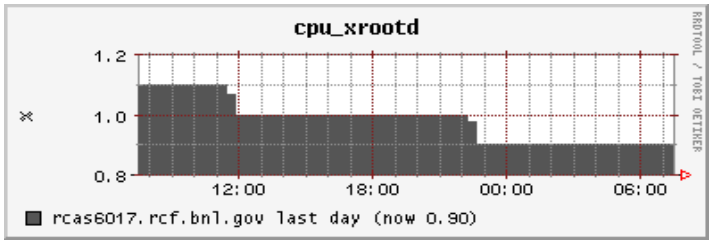
- preparation of the configuration file containing configuration of load balancing, authentication and MSS plugin
- implementation and testing of xrootd daemons managing tools



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



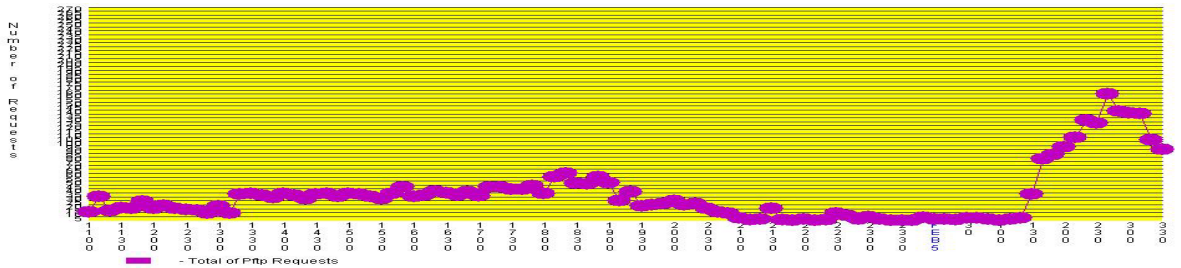


Problems and repairs/contribution:

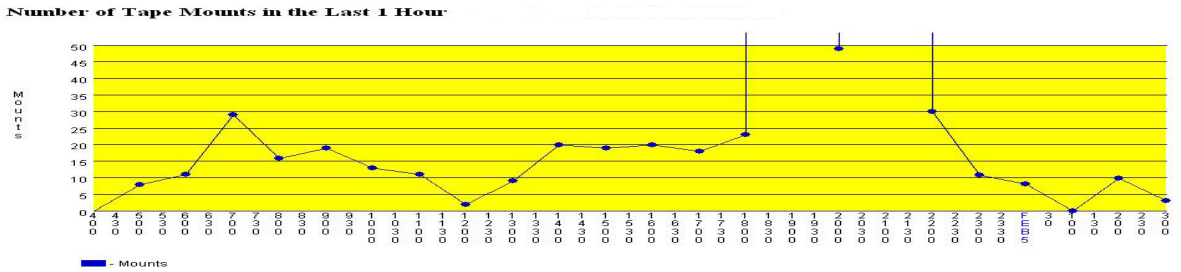
1. Needed to wait for the 64 node limitations removal (reported in February 2005, available in April/May 2005)
2. Different security model:
   - we were beta testers
   - shaky initial implementation and documentation
   - provided a bug fix and possibility to authenticate a user as other user
3. ROOTD 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

HPSS access pattern consequence

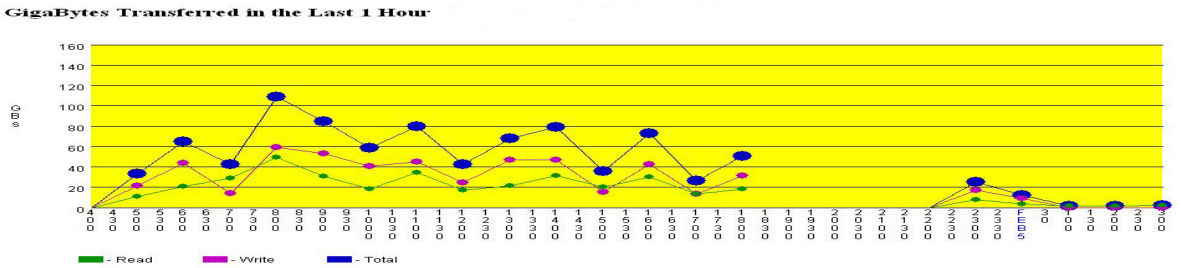
- requests to HPSS are not coordinated:
  1. increase number of requests



  2. increase tape mounts to maximum



  3. decrease I/O Rate to zero



Motivation . . .

XROOTD could be extended:

- does not bring files over from other space management systems
- always bring files from MSS, not from neighboring cache
- in large scale pools of nodes, clients could ALL ask for a file restore: lack of coordination or request "queue"
- no advanced reservation, no extended policies per users or role based
- other middleware are designed for space management. Leveraging on other projects and targeted re-usable components ?

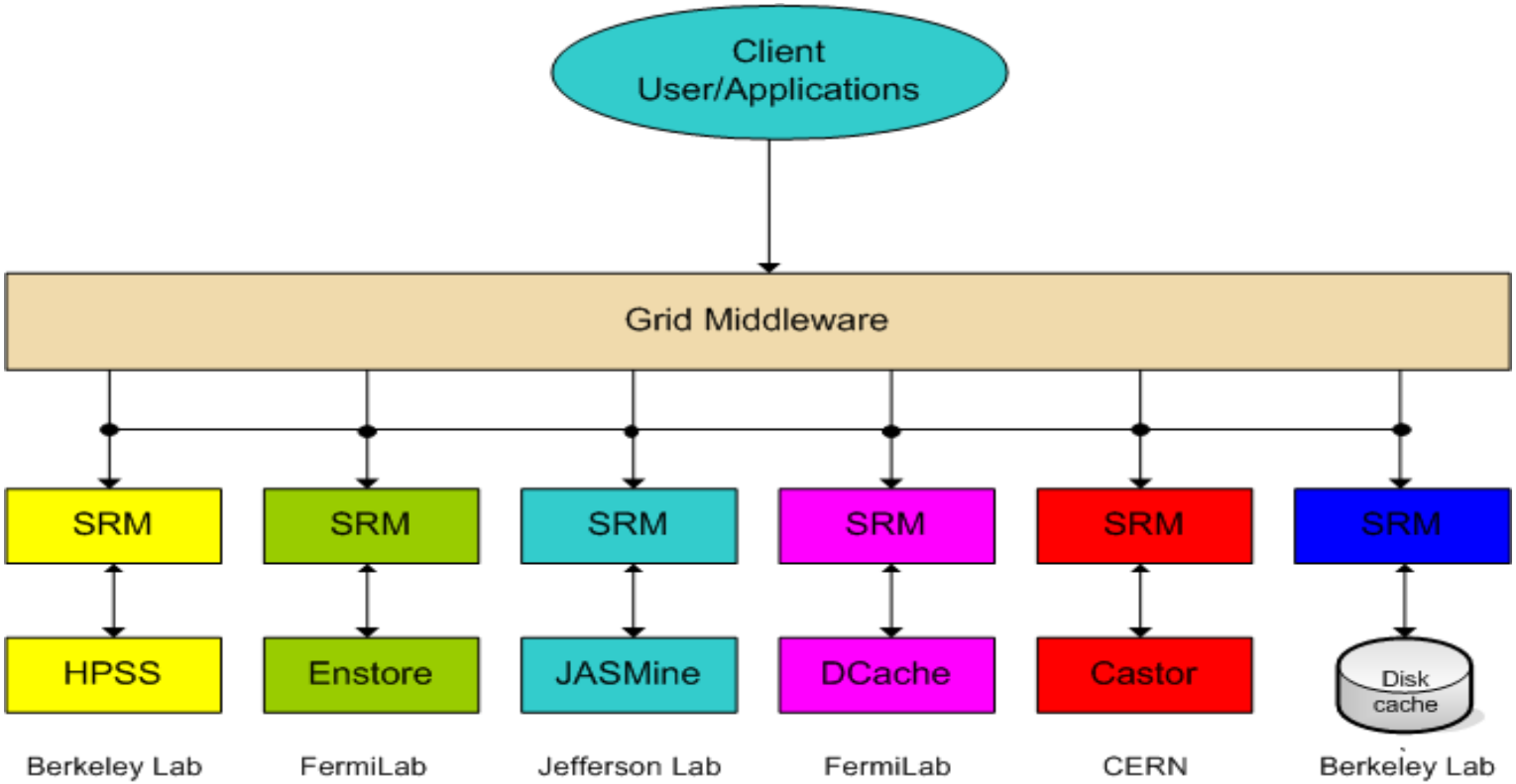
SRM funcionality

- **SRM:** the grid middleware component whose function is to provide dynamic space allocation and file management on shared distributed storage systems
  - Manage **space**
    - Negotiate and assign space to users and manage *lifetime* of spaces
  - Manage **files** on behalf of user
    - Pin files in storage till they are released
    - Manage *lifetime* of files
  - Manage file sharing
    - Policies on what should reside on a storage or what to evict
  - Bring the files from remote locations
  - Manage multi-file requests
    - a brokering function: queue file requests, pre-stage

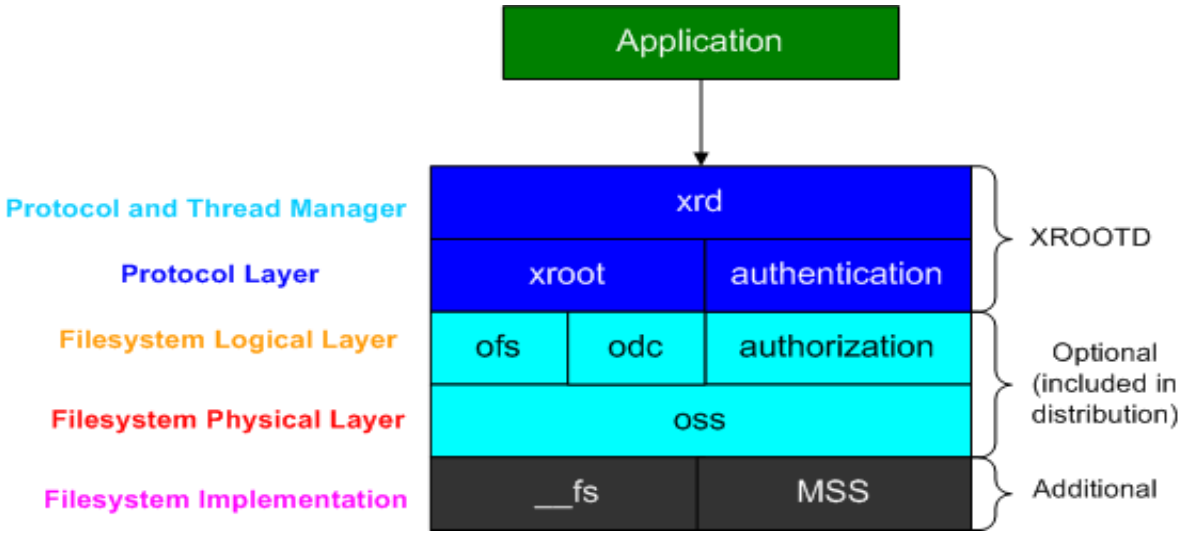
Types of SRMs

- **Types of storage resource managers**
  - Disk Resource Manager (DRM)
    - Manages one or more disk resources
  - Tape Resource Manager (TRM)
    - Manages the tertiary storage system (e.g. HPSS)
  - Hierarchical Resource Manager (HRM=TRM+HRM)
    - An SRM that stages files from tertiary storage into its disk cache
- **SRMs and File transfers**
  - SRMs DO NOT perform file transfers
  - SRMs DO invoke transfer service(GridFTP, FTP, HTTP, . . .)
  - SRMs DO monitor transfers and recover from failures

Uniformity of interface -> Compatibilty of SRMs

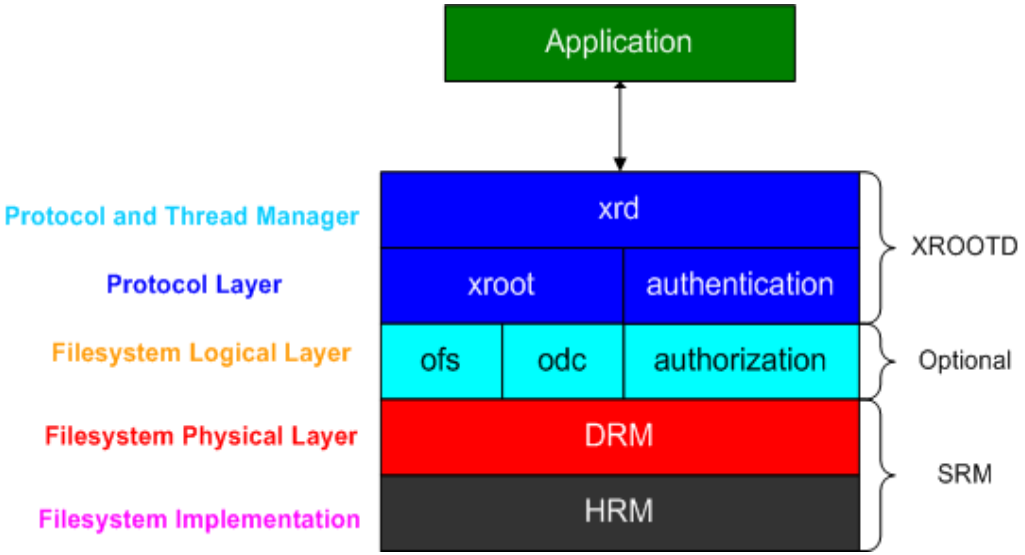


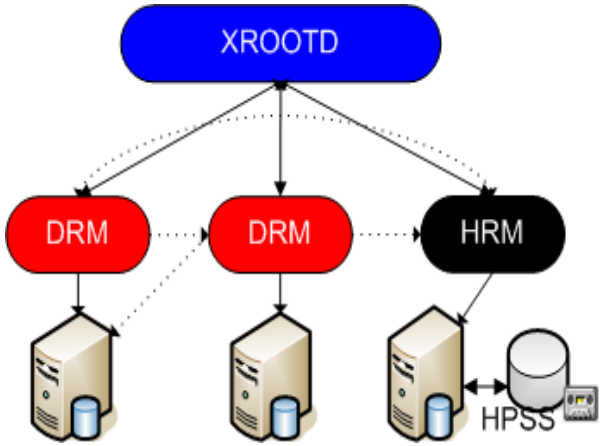
XROOTD components architecture



- xrd - provides networking support, thread management and protocol scheduling
- xroot - implements xrootd protocol
- ofs - provides enhanced first level access to file data (responsible for coordinating activities of oss, odc, auth)
- oss - provides access to underlying storage system (controlled by ofs and invokes meta-data operations)

XROOTD+SRM components architecture





- xrootd is responsible for managing the disk cluster
- DRM is responsible for managing the disk cache
- HRM is responsible for staging files from MSS

Status of XROOTD+SRM integration

There are **2 parallel non-overlapping** projects:

1. Integration of the SRM with xrootd where xrootd becomes a client of the DRM
   - DRM is responsible for managing the disk pool and xrootd for coordinating SRM
2. Integration of the FNAL SRM with xrootd where the FNAL SRM becomes an xrootd client
   - xrootd is responsible for the server selection and directing SRM requests to the correct node

Ongoing/future work

- need to coordinate requests to MSS
  - will use DataCarousel which is in use in STAR for this purpose
  - conceptual approach demonstrated in other context (old rootd based model)

Ongoing/future work

- need to coordinate requests to MSS
  - will use DataCarousel which is in use in STAR for this purpose
  - conceptual approach demonstrated in other context (old rootd based model)
- implement the interface of generic plugin for LFN/PFN conversion for a more flexible LFN to PFN conversion

Ongoing/future work

- need to coordinate requests to MSS
  - will use DataCarousel which is in use in STAR for this purpose
  - conceptual approach demonstrated in other context (old rootd based model)
- implement the interface of generic plugin for LFN/PFN conversion for a more flexible LFN to PFN conversion
- complete XROOTD+SRM integration

Ongoing/future work

- need to coordinate requests to MSS
  - will use DataCarousel which is in use in STAR for this purpose
  - conceptual approach demonstrated in other context (old rootd based model)
- implement the interface of generic plugin for LFN/PFN conversion for a more flexible LFN to PFN conversion
- complete XROOTD+SRM integration
- we propose to move toward "**object on demand**"
  - we want to move from file based access to object based access (in HENP, objects = events for example)
  - will take the advantage of *GridCollector* (event grid catalog) which is used in STAR

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)
- modulo few fixes in year 2005 the system looks stable and easily configurable

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)
- modulo few fixes in year 2005 the system looks stable and easily configurable
- load balancing and handshake with MSS make the system resilient to failures

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)
- modulo few fixes in year 2005 the system looks stable and easily configurable
- 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

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)
- modulo few fixes in year 2005 the system looks stable and easily configurable
- 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
- simultaneous PFN/LFN support allowed for smooth transition from ROOTD to XROOTD

Summary

- Xrootd is deployed on 320 nodes (the biggest production deployment of xrootd)
- modulo few fixes in year 2005 the system looks stable and easily configurable
- 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
- simultaneous PFN/LFN support allowed for smooth transition from ROOTD to XROOTD
- remaining concern an un-coordinated requests to MSS could be resolved with SRM back-end interface (as interim use DataCarousel)