xrootd Update

Alice Tier 1/2 Workshop

Karlsruhe Institute of Technology (KIT)

January 24-26, 2012

Andrew Hanushevsky, SLAC

http://xrootd.org

Outline

- Recent additions
- Recent changes
- On the horizon
- Near future
- Things that you might not know
- The xrootd collaboration
- Conclusion
- Acknowledgements

Recent Additions

- Extended Attribute Framework
- Integrated checksums
- Shared-Everything File System Support
- Static ENOENT redirection
- Caching proxy server
- Federated site shares
- Monitoring Extended
- Read-fork-read client

Agnostic Extended Attributes

- FS-Independent Extended Attribute Framework
  - Used to save file-specific information
  - File system must support extended attributes
    - Normally not a problem except for EXT3
      - May need to manually enable support in /etc/fstab
        - /dev/hda3 /home ext3 defaults,**user_xattr** 1 2
    - If disabled and needed, you will get a log error message
      - Functions needing xattr information then turned off

Current Xattr Information

- XrdCks.*xxx* — *xxx* (e.g. md5) checksum
- XrdFrm.*yyy* — **F**ile **R**esidency **M**anager info
  - XrdFrm.Cpy — Time file was copied out
  - XrdFrm.Mem — Memory mapped access attributes
  - XrdFrm.Pin — Disk pinning attributes
  - XrdFrm.Pfn — File's physical name

    Only for *symlinked* spaces

Future Xattr Information

- Original file creator
  - The inexorable start of quotas, sigh
- Extended Access Information
  - Keep last three accesses
    - Number of parallel readers, % read, access time
  - Can be used for better migration/purging

Viewing Xattr Information

- getfattr -d [-e *en*] [-m *pattern*] *pathname*
  - See Linux man page for details
  - Most information is binary so must use proper *en*
  - Use *pattern* to restrict what variable is displayed
    - In Linux, variables are always prefixed by “user.”
- frm_admin {chksum | find} . . .
  - Displays info in human understandable terms
    - See http://xrootd.org/doc/prod/frm_config.htm

Will I Need To Migrate?

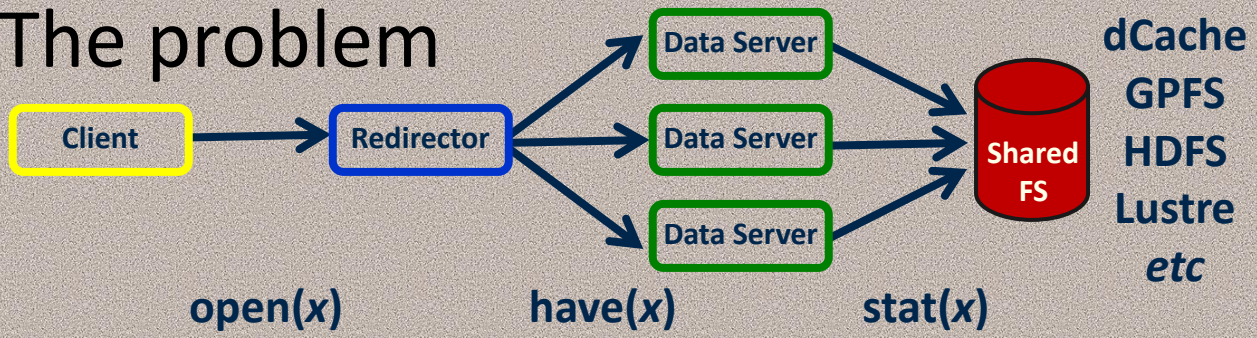
- Yes, if you use
  - MPS (**M**igration, **P**urge, and **S**taging) *or*
  - oss.cache or oss.space directives
- No, otherwise or run in compatibility mode
  - oss.runmodeold directive
    - Will be supported until 2014
- Use “frm_admin convert” for conversion
  - See http://xrootd.org/doc/prod/frm_migr.htm

Integrated Checksums

- Can configure xrootd to handle checksums
  - **xrootd.chksum** [**max** *num*] {adler32|crc32|md5}
    - Add new ones via **ofs.ckslib** directive and plug-in
    - Optimize vis **ofs.cksrdsz** directive
  - Checksum saved in extended attributes
    - Displayed on query
    - Automatically recomputed when file changes
      - Can be disabled
  - External program no longer needed
    - But this is still supported

Shared FS Support

- The problem

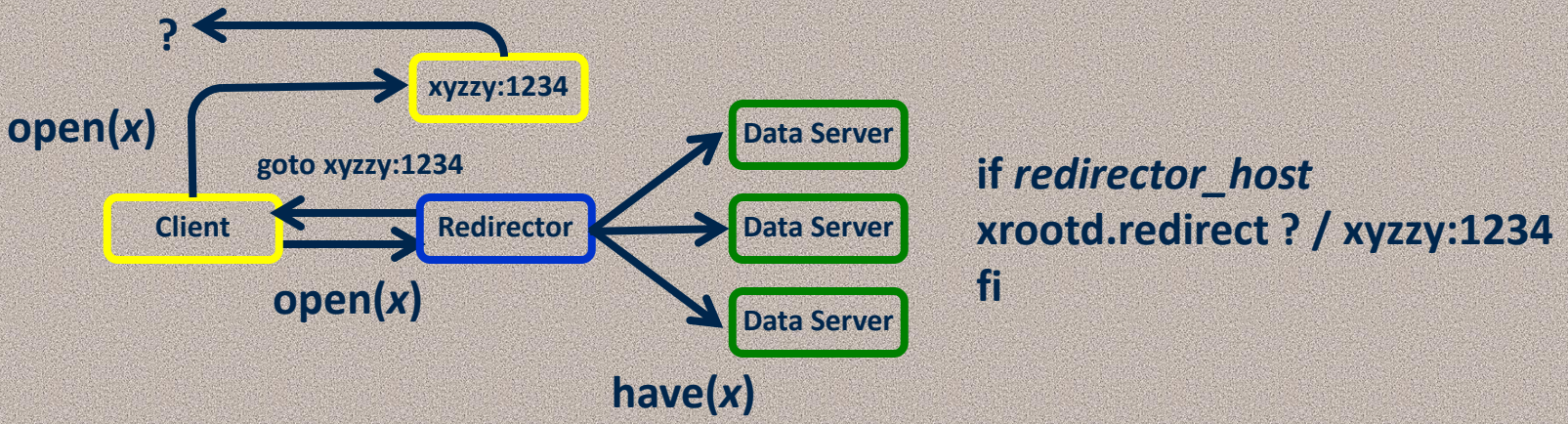


- The solution
  - Use **cms.dfs** directive to eliminate duplicate hits
    - Many tuning options available
      - Lookup at redirector, limits, caching, etc
      - See http://xrootd.org/doc/prod/cms_config.htm

Static ENOENT Redirection

- Allows you to redirect client on ENOENT
  - Works for redirectors and data servers
    - Any operation involving a file
- **xrootd.redirect ?** *path host*:*port*
  - Up to four different "?" redirects allowed
    - Provides for selective redirects based on path
- Allows more specialized missing file handling
  - Alternative to the FRM staging add-on

ENOENT Redirects



- What we envision. . .
  - xyzzy is a caching proxy server or proxy cluster
    - Provides high performance WAN access
      - Client accesses data via WAN in this case

But You Can Do Much More!

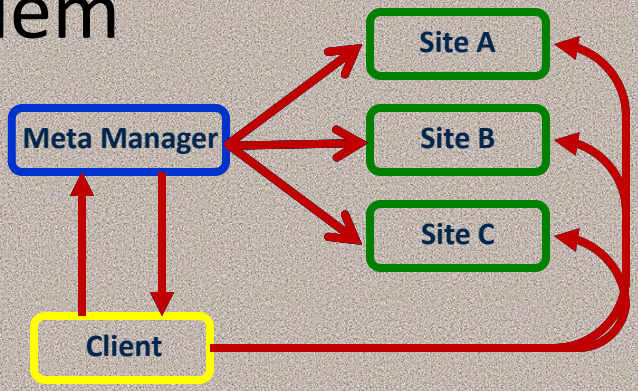
- The xyzzy host can do anything really
  - Have a specialized reselection
  - Then redirect client elsewhere
    - Requires a plug-in implementation
- ENOENT redirection is a powerful tool
  - Can implement any site-specific recovery actions
    - The logic is centralized and can be uniform

Caching Proxy Server

- The proxy server plug-in has in-memory cache
  - Must be enabled via **pss.memcache** directive
- Many tuning options; some of which are . . .
  - Cache and page size
  - Read ahead size
  - Maximum block size to cache
  - Root file access optimization
- See http://xrootd.org/doc/prod/ofs_config.htm

Federated Site Shares

- The problem



Each site gets an *equal* number of requests all things being equal

- The solution
  - Use **cms.sched** directive to establish site share
    - Use the **gshr** and **gsdflt** options
      - See http://xrootd.org/doc/prod/cms_config.htm

Monitoring

- Readv requests fully monitored
  - By default, only a readv request summary provided
    - Can request a full unwind of a readv request
- Per client I/O monitoring now flushable
  - Before, I/O statistics flushed when buffer full
  - Can specify a flush window
    - Based on code provided by Matevz Tadel, CMS
- Authentication info can now be monitored

Read-Fork-Read Client

- The current client allows forking
  - This allows sharing data between processes
    - Read conditions data
    - Fork *n* times for parallel processing
    - Read event data using pre-read conditions data
- Extensively used by CMS
  - Substantially reduces memory load
    - Critical for large multi-core worker nodes

Recent Changes

- Switched to git and cmake for build
- Using EPEL guidelines

Switched To cmake

- We have dropped old build procedures
  - CVS, autotools, and configure.classic
- Using git repository for source code
  - git clone http://xrootd.org//repo/xrootd.git
  - git clone http://xrootd.cern.ch/repos/xrootd.git
- Standardized on cmake
  - Source builds, RPM builds, and installs
    - Version 2.8 recommended but 2.6 should work
  - See README file in the top level source directory

Switch Had Side Effects

- Xrootd will not be part of the root package
  - Will be able to get it separately via RPM process
    - Use yum to install
    - Repositories: xrootd.org or xrootd.cern.ch
      - See http://xrootd.org/dload.html
      - Must include repository in /etc/yum.repos.d
  - Process normally determined by experiment

Conform To EPEL Guidelines

- New guidelines prohibit installing '.a' files
  - All '.a' files are replaced by '.so' files
    - But we consolidated libraries
      - This limits the number of installed shared libraries
      - Still, there are more than there used to be
  - Impact is minimal except for plug-in writers
    - Will likely need to change your link step
    - But now you can easily re-use supplied classes

On the horizon

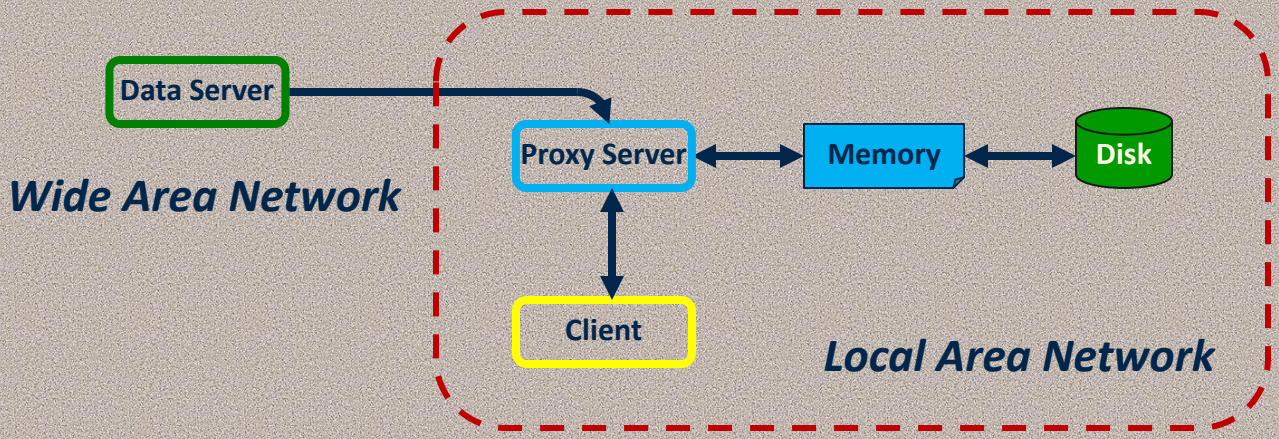
- Extended monitoring
- Disk caching proxy server
- Integrated 3rd party copy
- New client
- Dropping RH4 support
  - Planned for 3.2 release

Extended Monitoring

- Redirect information will be provided
  - Selectable via configuration option
  - Will provide extensive information
    - Who went where for what operations
      - Currently, only available via debug log output (yech)
- Authentication details
  - Number of attempts, failures, bypasses, etc

Disk Caching Proxy Server

- The current proxy server will be extended
  - Will allow for memory as well as disk caching
    - Data can stick around on the proxy for re-use
  - We will be fleshing out the details in February



Integrated 3rd Party Copy

- Currently, xrd3cp provides 3rd party copy
  - We plan to include this functionality in the base
    - Actual protocol will differ since pull is a simpler model
      - This does not change xrootd protocol just the ofs plugin
  - Part of the xrdcp rewrite
    - Better handling of streams
    - More understandable options

New Client

- Current client uses a dedicated thread model
  - Limits scaling and is resource intensive
- New client will use a thread pool model
  - Scalable and fully asynchronous
- Will be the platform for future features
  - E.G. plug-in caches, local redirects, etc

Things in the near future

- Extended POSC
- Specialized meta-manager
- Integrated alerts
- New async I/O model

Extended POSC

- Currently, adding “?ofs.posc=1” enables POSC
  - Persist On Successful Close
- This can be extended to support checksums
  - E.G. “?ofs.posc=adler32:*csval*”
- File persists on successful close AND Supplied checksum matches
  - Privilege & error ending states not yet defined

Specialized Meta-Manager

- Current MM is a regular manager with mm role
  - This limits what the meta-manager can do
    - Extending it unduly impacts the manager's code
- The specialized MM is a separate daemon
  - Will allow many more subscribers
    - Limit now is 64 will likely increase it to 256.
  - Can better optimize handling federated managers

Integrated Alerts

- Currently, alerts based on using monitoring
  - Monitoring provides broad usage information
    - Alerts are therefore macro-scale
- We want to send a separate alert stream
  - Based on unusual internal events
    - E.G. unexpected latency, server recovery actions, etc
- Part of message and logging restructuring

New Async I/O Model

- Currently xrootd uses OS supplied async I/O
  - This is not particularly useful
    - In Linux it is simulated as it was never kernel level
    - In other OS's it uses a lot of CPU resources
      - In fact, xrootd normally bypasses it for most requests
- The next version will use a thread model
  - Based on looking ahead on the request stream
  - This should be more applicable to most requests

IPV6

- Currently, all new code supports IPV6
  - But existing code needs to change
  - We are not sure how critical this really is
  - And it has side-effects
    - E.G. all IP addresses in messages would change
- No implementation plan yet
  - How critical is this in practice?

Did You Know?

- You can do load based server selection
  - Avoid overloaded servers
- Step 1: Get a load collector program
  - We supply one called cms_MonPerf
- Step 2: Configure its use
  - cms.perf [**int** *time*] [**pgm** *prog*]
- Step 3: Configure how to calculate load
  - cms.sched *parameters*

The cms.sched Directive I

- cms.sched *parameters*
  - cpu *pct*
  - io *pct*
  - mem *pct*
  - pag *pct*
  - runq *pct*
  - space *pct*
    - Percentage on indicated item used to calculate load
      - Total percentages should equal 100!

The cms.sched Directive II

- cms.sched *parameters*
  - fuzz *pct*
    - Percentage difference two load value must have in order to be considered different
      - Default is 0
  - maxload *value*
    - Maximum load allowed to select server
      - If it's the only server that has the file and it can't be selected the client gets an ENOENT condition
      - This can be used to replicate the file or select an external source of the file

Xrootd Collaboration

- Mutually interested institutions contributing effort for development and maintenance
  - SLAC (founder)
  - CERN (2010)
  - JINR (fall 2011)
  - UCSD (winter 2011)
    - Newest member!

Conclusion

- xrootd is under active development
  - Always looking for new ideas
    - Feel free to suggest them
  - Be a contributor
    - You too can contribute to the code base
  - Consider joining the xrootd collaboration
    - It costs no money to join
- See more at http://xrootd.org/

Acknowledgements

- Current Software Contributors
  - ATLAS: Doug Benjamin, Patrick McGuigan, Danila Oleynik, Artem Petrosyan
  - CERN: Fabrizio Furano, Lukasz Janyst, Andreas Peters, David Smith
  - CMS: Brian Bockelman (unl), Matevz Tadel (ucsd)
  - Fermi/GLAST: Tony Johnson
  - FZK: Artem Trunov
  - LBNL: Alex Sim, Junmin Gu, Vijaya Natarajan (**BeStMan** team)
  - Root: Gerri Ganis, Beterand Bellenet, Fons Rademakers
  - OSG: Tim Cartwright, Tanya Levshina
  - SLAC: Andrew Hanushevsky, Wilko Kroeger, Daniel Wang, Wei Yang
- Operational Collaborators
  - ANL, BNL, CERN, FZK, IN2P3, SLAC, UCSD, UTA, UoC, UNL, UVIC, UWisc
- US Department of Energy
  - Contract DE-AC02-76SF00515 with Stanford University