xrootd Roadmap

Atlas Tier 3 Meeting

University of Chicago

September 12-13, 2011

Andrew Hanushevsky, SLAC

http://xrootd.org

Outline

- Recent additions
  - FS-Independent Extended Attribute Framework
  - Shared-Everything File System Support
  - Meta-Manager throttling
- On the horizon
- Near future
- Way out there
- Conclusion
- Acknowledgements

Recent Additions

- FS-Independent Extended Attribute Framework
  - Used to save file-specific information
    - Migration time, residency requirements, checksums
- Shared-Everything File System Support
  - Optimize file discovery in distributed file systems
    - dCache, DPM, GPFS, HDFS, Lustre, proxy xrootd
- Meta-Manager throttling
  - Configurable per-site query limits

On the horizon

- Security
- New source build procedures
- EPEL Guidelines
- Integrated checksums
- Extended monitoring
- Alternate Name2Name Plug-in
- Dropping RH4 support
  - Planned for 3.2 release

Security Part 1

- Enabling x509 authentication & Authorization
  - Motivated by server-server transfers via FRM
    - Required by ATLAS security
- Requires additional site configuration
  - Obtaining site x509 certificates
    - These are certificate and private key pem files
  - Placing these in /etc/grid-security/xrd
    - By default, xrdcert.pem and xrdkey.pem

Security Part 2

- Periodically creating voms proxy certificates
  - This is to get a valid voms ATLAS extension
  - Probably done via cron job
    - These are used by the FRM to authenticate site
- Installing the ATLAS x509 mapping plug-in
  - Likely distributed via the OSG rpm
    - We haven't fleshed out the details

Security Part 3

- Periodically creating voms proxy certificates
  - This is to get a valid voms ATLAS extension
  - Probably done via cron job
    - Though are considering long-lived proxy certificate
  - These are used by the FRM to authenticate site
- Installing the ATLAS x509 mapping plug-in
  - Likely distributed via the OSG rpm
    - We still have to discuss this with OSG

Security Part 4

- Configuring xrootd to force x509 authentication
  - xrootd.seclib *libpath*/XrdSec.so
  - sec.protocol libpath gsi \
    -authzfun:libXrdAuthzAtlas.so \
    -gmapopt:10 -gmapto:0
  - ofs.authorize
  - acc.authdb *path*/*dbfname*

    The *dbfname* contains the line: **g atlas / rl**

Security Part 5

- What all this does
  - Requires client to provide ATLAS certificate
    - The xrootd.seclib and sec.protocol directives
  - libXrdAuthzAtlas.so maps valid cert to group atlas
    - The authzfun parameter in sec.protocol
  - The *authdb* file says group atlas has r/o access
    - ‘g’ for group
    - ‘/’ for everything you export
    - ‘rl’ for read and search access

Security Part 6

- Simplifying the side-effects
  - Normally, this requires everyone to have a cert!
    - This is very intrusive for most T3 sites
  - We can restrict this to *only* the proxy server
    - This means you need to run a proxy server
      - Many if not most sites will need to run one due to firewalls
    - Only outside clients will need to have a valid cert
  - It is possible to do this without a proxy
    - The configuration becomes a bit more complicated

Certificate Issues!

- Voms certs only issued to individuals
  - This makes site and host certs problematic
    - We really have no real solution to this now
- Virtual solution
  - Get user cert that corresponds to xrootd user
  - Have that cert validated by voms
  - Use it as site service certificate
    - For host identify & individual host access
      - We don't really know if this will work, SIGH

Security Status

- We have a working autz mapping function
  - Based on s/w from Matevz Tadel & Brian Bockelman
    - Needs some clean-up and better packaging
- Distribution needs to be decided
  - Likely via OSG just like gridftp add-ons
- Certificate plan needs to be put into place
  - How to obtain one & creating voms proxy certificates
  - Where to place all of these
    - Will likely need additional sec.protocol options

Build using cmake

- Currently, we support two source build methods
  - autotools & configure.classic
- The 3.1 release will use cmake
  - This will displace autotools
  - Support for configure.classic yet unclear
    - The root team wants it maintained for MacOS
      - We are in negotiations
- This means you will need to install cmake
  - Only if you want to build from the source

EPEL Guidelines

- New guidelines prohibit installing '.a' files
  - Most '.a' files will be replaced by '.so' files
    - We are trying to consolidate libraries
    - This will limit the number of installed shared libraries
  - Impact is minimal except for plug-in writers
    - Will likely need to change your link step
- This will occur when we switch to cmake
  - Planned for 3.1 release

Integrated Checksums

- Currently, checksum calculated outboard
  - Program is specified via configuration file
  - The xrdadler32 command also checksums
    - Records checksum in extended attributes for future
- New xrootd will do inboard checksumming
  - Will record checksum in extended attributes
    - Many configuration options available
    - Should speed up SRM queries
- Planned for 3.1

Extended Monitoring

- Redirect information will be provided
  - Selectable via configuration option
  - Will provide information on who went where
    - Currently, only available via debug log output (yech)
- Per client I/O monitoring will be flushable
  - Currently, I/O statistics flushed when buffer full
  - Will be able to specify a flush window
    - Based on code provided by Matevz Tadel, CMS
- Planned for 3.1 (redirect info likely in 3.1.x)

Alternate N2N Plug-in

- Currently, name-to-name plug-in limited
  - It is not passed contextual information
- Will be able to specify alternate plug-in
  - Alternate will get contextual information
    - I.E. url cgi information
  - This makes it binary compatible with past plug-ins
- Planned for 3.1

Things in the near future

- Extended POSC
- New xrootd client
- 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
- Planned for 1Q12

New Client

- Current client uses basic thread model
  - Limits scaling and is resource intensive
- New client will use robust thread model
  - Scalable and fully asynchronous
- Will be the platform for future features
  - E.G. plug-in caches, local redirects, etc
- Planned for 1Q12
  - This will likely be an alpha release

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
  - Can better optimize handling federated managers
- Planned 2Q12

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
- Planned 3Q12
  - 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
- Planned 4Q12

Way Out There

- Xinetd proxy support
- IPV6

Xinetd Based Proxy

- Currently, proxy support provided by xrootd
- With some tinkering it's possible to use xinetd
- We don't know yet if this is useful
  - Internal security will apply to external clients
  - Also, seems limited to very small static clusters
    - Requires re-config every time the cluster changes
  - But it may appeal to some sites
- No implementation plan yet
  - Does anyone really want this?

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 address in messages would change
- No implementation plan yet
  - How critical is this in practice?

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
    - Currently CERN, SLAC, and Duke are members
- 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