XRootD Roadmap

ATLAS Software & Computing Week

CERN

October 15-19, 2012

Andrew Hanushevsky, SLAC

http://xrootd.org

Outline

- Things definitely for 3.3.0 (minor release)
  - Already in git head
- Things possible for 3.3.0 (minor release)
  - Not yet in git head
- Things likely for 4.0.0 (major release)
- Conclusion
- Acknowledgements

The actual presentation is an edit

Things Definitely For 3.3.0

- Already in git head
  - New “f” stream monitoring
  - Integrated checksums part 2
  - Redirection & opaque information forwarding
  - Third party copy
  - New client
  - Plug-in version checking

Covered here

The Real-Time “f” Stream

Details

- Binary stream for only file-based information
  - Intermediate detail between summary and detail
    - Provides accurate Real-Time per-file transfers
    - Computed Sigma’s for I/O block size
  - Configured via the xrootd.monitor directive
    - Option: fstats *interval* [fn] [io] [ops] [sigma]

fn	include filename in open record
io	provide per-file I/O statistics each *interval*
ops	include operation counts in close record
Sigma	calculate sigma values

Integrated Checksums

- Can configure xrootd to handle checksums
  - **xrootd.chksum** [**max** *num*] {adler32|crc32|md5}
    - Checksums have been part of xrootd for a long time
- Now can be configured for a manager node!
  - Checksum for a data server or manager equivalent
    - From client’s perspective endpoints are the same
      - Manager will redirect client to appropriate data server
  - This also eases implementing checksum plug-ins
    - E.g. DPM, EOS, HDFS, etc

[Static] Redirection

- Allows you to redirect client
  - Can also redirect only when file not found
- The problem has to do with the “old” client
  - Opaque information only passed for open()
  - This *may* make EOS N2N service problematic
    - N2N handled via opaque information
      - Problematic for admin functions only
- Client now always passes through opaque

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
  - In git head as xrdcpy
- Does not require certificate delegation
  - We plan to provide this as an option

New Client I

- 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
  - Completely thread and fork-safe
  - Uses new detailed monitoring plug-in
- Will be the platform for future features
  - E.G. plug-in caches, local redirects, etc

New Client II

- We realize this is a disruptive change
  - The new client will be phased in
  - Phase 1
    - xrdcp will use the new client
      - We now have xrdcp (old), xrdcpy (old+new), xrdcopy (new)
      - Some functionality is still missing but will be added
  - Phase 2
    - The POSIX interface will use the new client
      - This affects a host of systems (e.g. XRootDFS, proxies)
  - Phase 3
    - Complete switch (likely a major release)

This may be sped up!
CMSSW already converted but some issues need to be solved.

More

Things Possible For 3.3.0

- Not in git head
  - Dynamic Node Disablement
  - Standalone cmsd
  - Monitoring signposts
  - EPEL Conformance
  - The cms space directive enhancement

Covered here

Dynamic Node Disablement

- Sites expressed interest in RT disablement
  - Temporarily disable badly behaving sites
    - At the redirector level
  - Still exploring the best way to do it
    - Active: inform redirector about site status
      - I.E. via admin interface {enable | disable} *nodename*
    - Passive: mark site in some well-known directory
      - E.G. touch /*adminpath*/disabled/*nodename*

Standalone cmsd

- Currently, always pair cmsd with an xrootd
    - Some sites think this is odd for certain systems
        - dCache when using the dCache xrootd door
    - The cmsd always supported stand-alone mode
        - But didn’t allow a virtual data port (i.e. non-xrootd)
        - Adding this feature allows full standalone mode
            - I.E. client would be redirected to the dCache xrootd door
            - No need to run a separate xrootd with a static redirect
    - We are still not sure this is a good idea
        - http://savannah.cern.ch/bugs/?98119

Monitoring Signposts

- Monitoring allows application signposts
  - I.E. insertion of an application defined marker
- Currently, requires application-level call
  - We can automate this via special envar’s
    - No application code changes needed
    - Allows tracking of actual application
    - We may always do this for common applications
      - E.G. xrdcp

More

Things For 4.0.0

- Not in git head
  - Readv passthrough
  - Allow home directory creation
  - IPV6

Covered here

Readv Passthrough

- Currently, xrootd un-roles readv requests
  - The passes them singly to the file system plug-in
- This is OK for most systems but not all
  - HDFS can do better given the read vector
  - Proxy servers suffer most
    - Due to increased LAN/WAN requests/responses
- Plan to allow end-to-end readv requests
  - Requires ofs and oss interface extension
    - http://savannah.cern.ch/bugs/?98149
- Clearly, a major release!

Allow Home Directory Creation

- Access control allows a fungible write rule
    - **u = /*basepath*/@=/ a**
        - Where @= is the authenticate username
        - Hence, user’s have r/w access to their home directory
    - However, this requires directory pre-creation
- We plan to allow users to create their own
    - Should a fungible rule exists
    - https://savannah.cern.ch/bugs/index.php?93902

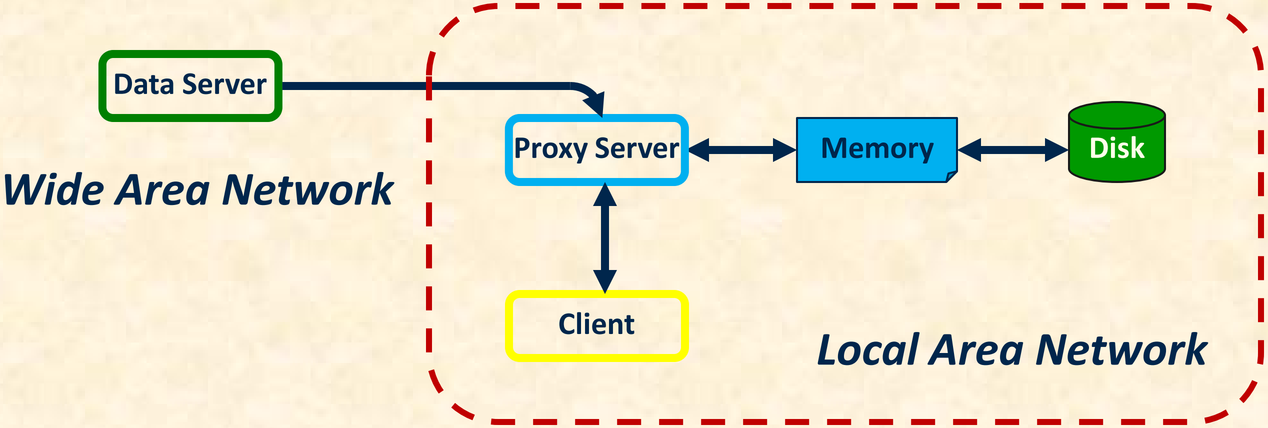
More

Things in the near future

- Disk caching proxy server *Covered here*
- Extended POSC
- Additional extended attributes
- Specialized meta-manager
- Integrated alerts
- New async I/O model

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
  - This being actively developed by CMS experiment



More

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