XRootD Release 4.5 And Beyond

XRootD Workshop Tokyo

Stanford University/SLAC

November 10, 2016

Andrew Hanushevsky, SLAC

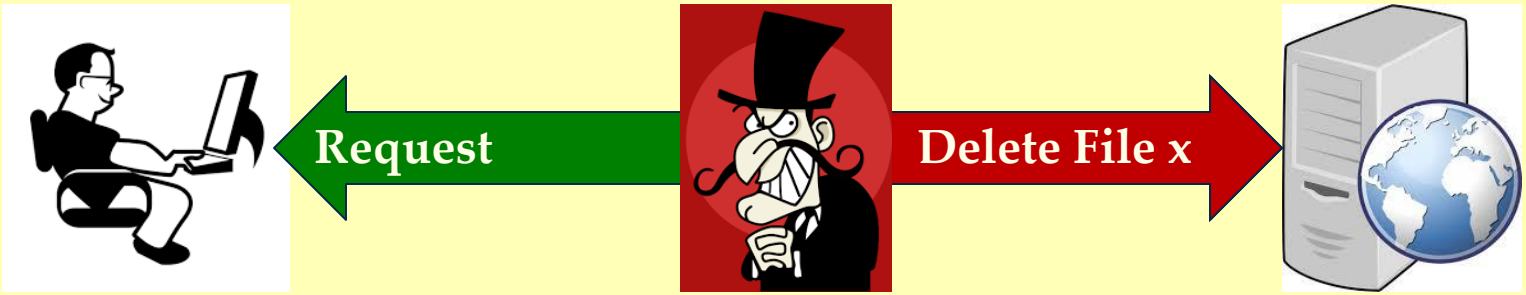
http://xrootd.org

November Release 4.5 Highlights

- Request signing
- Dual stack networking option
- Client reporting of release at login time
- Separate negative cache timeout in cmsd
- Allow file names with spaces
- Allow host names starting with digit
- Automatic URL prefixing
- Zip archive support

Request Signing

- Protects **XRootD** servers from bad actors



- Client can cryptographically sign requests
  - Sever verifies request came from the same authenticated client
    - Bad actor problem avoided

Enabling Request Signing

- Request signing gives you peace of mind
  - When allowing R/W access to the server
    - Especially on the WAN
- Server configuration option
  - **sec.level** {**all**|**local**|**remote**] [**relaxed**] *level*
    - **all** applies level to local and remote client
      - **local** and **remote** provide split options
    - **relaxed** provides a migration path
      - It requires signing only for 4.5 and up clients

Request Signing Levels

- **none**
  - The default
- **compatible**
  - Only destructive operations
    - Is compatible with R/O access for old clients
- **standard | intense | pedantic**
  - Each requires more operations to be signed

Request Verification by Level

Operation	Compatible	Standard	Intense	Pedantic
admin	verified	verified	verified	verified
auth	---	---	---	---
bind	---	---	verified	verified
chmod	verified	verified	verified	verified
close	---	---	verified	verified
decrypt	---	---	---	---
dirlist	---	---	---	verified
endsess	---	---	verified	verified
getfile	verified	verified	verified	verified
locate	---	---	---	verified
login	---	---	---	---
mkdir	---	verified	verified	verified
mv	verified	verified	verified	verified
open read	---	verified	verified	verified
open Write	verified	verified	verified	verified
ping	---	---	---	---
prepare	---	---	---	verified
protocol	---	---	---	---
putfile	verified	verified	verified	verified
query	---	---	---	verified
query special	---	---	verified	verified
read	---	---	---	verified
readv	---	---	---	verified
rm	verified	verified	verified	verified
rmdir	verified	verified	verified	verified
set	---	---	verified	verified
set special	verified	verified	verified	verified
sigver	---	---	---	---
stat	---	---	---	verified
statx	---	---	---	verified
sync	---	---	---	verified
truncate	verified	verified	verified	verified
verifyw	---	---	verified	verified
write	---	---	verified	verified

Dual Stack Networking Option

- Pre-4.3 clients may report as IPv6 only
  - This is a big headache for IPv4-only servers
- New server-side option
  - **xrd.network assumev4**
  - Server will assume client has IPv4
  - Only applied to 4.4 or older clients
    - Server can't detect a client's release level until…

Client Release Reporting

- Client will report it's release level
  - Happens at login time
- This allows future server-side bypasses
  - When a particular release has a bug
  - Upgrading server can bypass old client bugs
  - Make client migration much easier

Separate Negative Cache Timeout

- The cmsd caches file location
  - Implicitly caches missing files as well
- Default is 8 hours
  - Incorrectly missing files will be missing 8 hours
    - Unless data server updates the cahe
- New server configuration option
  - **cms.fxhold noloc** *ntime*[**h**|**m**|**s**]
    - *ntime* expiration for cached missing files only

Allow File Names With Spaces

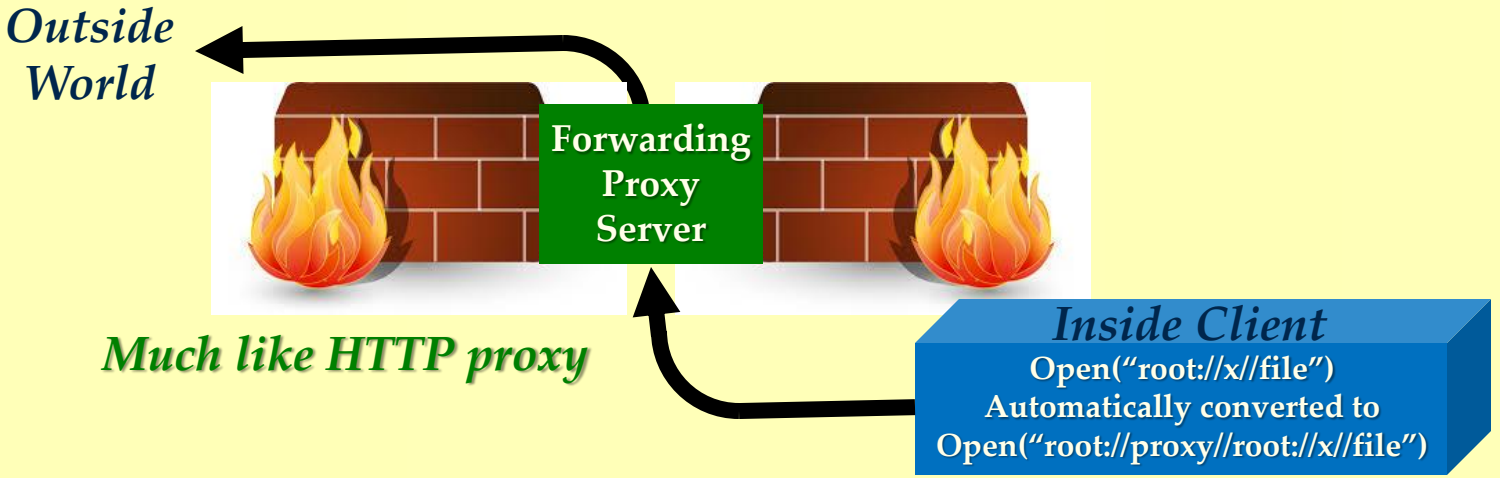
- XRootD allows spaces in file names
  - Every operation *except* **rename**
- Protocol extension covers **rename** now
  - All 4.5 plus clients use the protocol extension

Host Names Starting With Digits

- XRootD originally adhered to RFC 952
  - Hostnames may contain letters, digits, dashes
    - But may not start with a digit
- Now it adheres to RFC 1123
  - Supplants RFC 952 allows 1st char as a digit
- Required for auto-generated hostnames
  - Typically a problem for VM's and containers

Automatic URL Prefixing

- Required by fully firewalled sites



- Done as a configurable client plug-in
  - Usable by 4.0 and above clients
    - Will address multi-tenant sites later

Zip Archive Support

- Fully implemented as a client feature
  - Allows extraction of file from archive
    - No need to transmit the whole archive
  - Covered by Elvin's talk

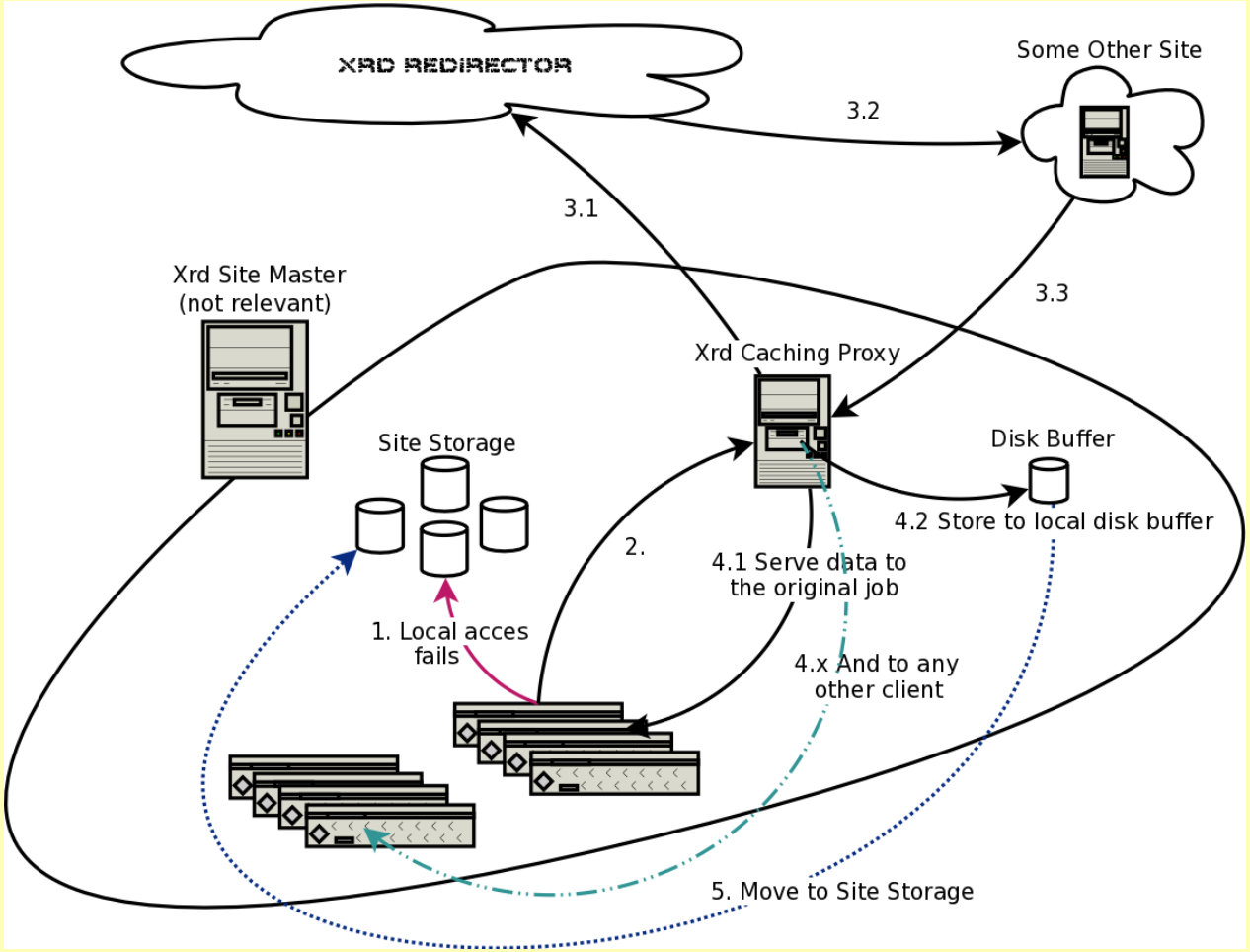
January Release 4.6

- Async I/O Proxy Handling
- Disk Caching Proxy
- Possible other addition
  - Extreme (multi-source) copy
- Perhaps others to be determined

Async I/O Proxy Handling

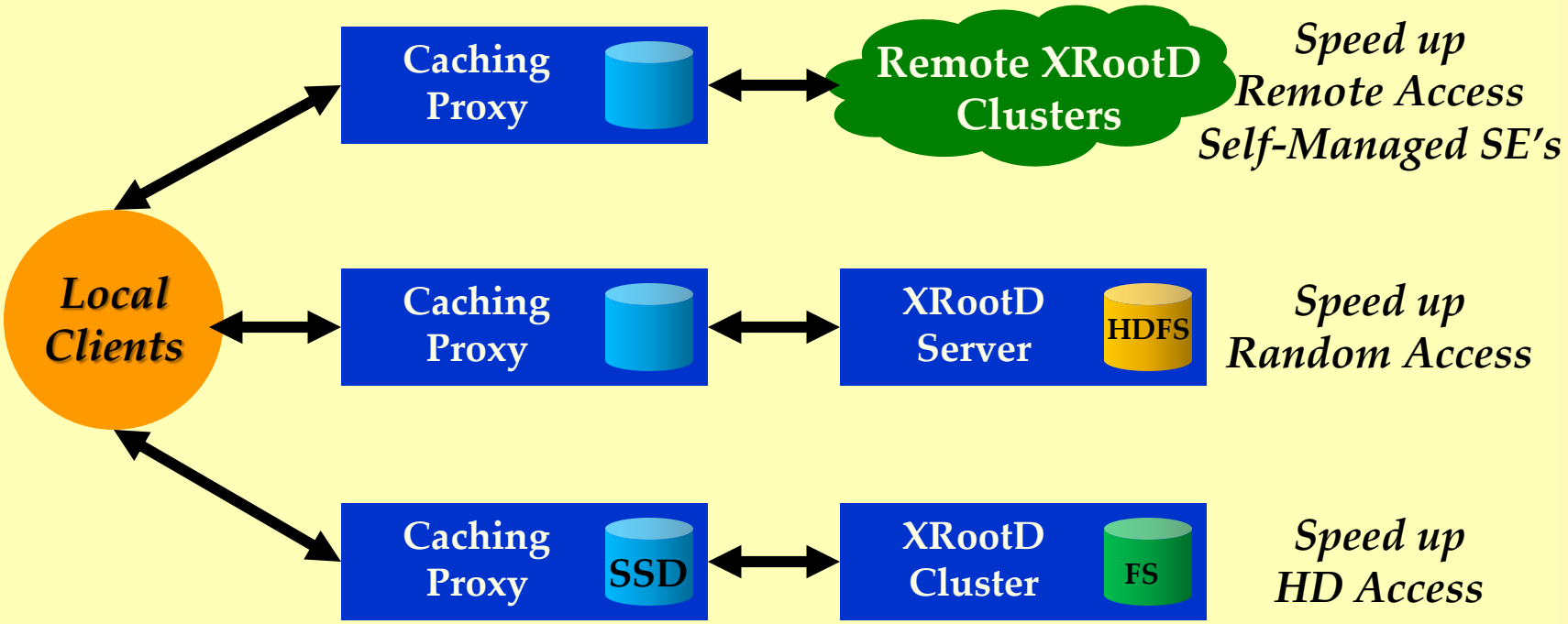
- Proxy now handles async I/O requests
  - Previously converted async to sync I/O
- Motivation
  - Improve streaming performance of xrdcp
  - Improve Disk Caching Proxy performance
    - Certain background operations (e.g. pre-fetch)
- May require tuning to get best performance
  - See **xrootd.async** directive

Disk Caching Proxy



File or block level caching

Typical Disk Caching Proxy Uses



More On Disk Caching Proxy

- High potential to solve vexing problems
  - Reduced remote latency for user analysis
  - Just in time data
    - Avoids pre-placement delays
  - Optimized disk space utilization
- Cached data access via xroot and http
- Currently being tested at scale
  - MWT2, SLAC, University of Notre Dame

Disk Caching Proxy Caveats

- Needs installation of tcmalloc or jemalloc
  - Avoids memory growth problems in glibc
- It's very easy to overload the proxy
  - Lager sites should consider caching clusters
    - Two or more proxies clustered together
    - Fully supported upon release

Disk Caching Proxy Deployment

- Target sites without ATLAS managed disk
  - Opportunistic sites
  - OSG diskless sites
  - Pacific Research Platform sites (NSF funded program)
    - NASA, NREN, U Washington, UC School System
      - Includes University California LHC sites (3 ATLAS T3's)
    - Interconnected via CalREN, ESNET, & Pacific Wave
      - Sites and networking may expand

In The Pipeline for 4.7 or 4.8

- Space quotas
- Scalable Service Interface

Space Quotas

- Experiments want write access to **XRootD**
- Current development
  - Non-hierarchical logical path based quotas
  - Quota is soft
    - Roughly within a specified resolution
    - Periodic recalibration
  - Will be a plug-in so can be replaced
- This is much harder than imagined!

Scalable Service Interface

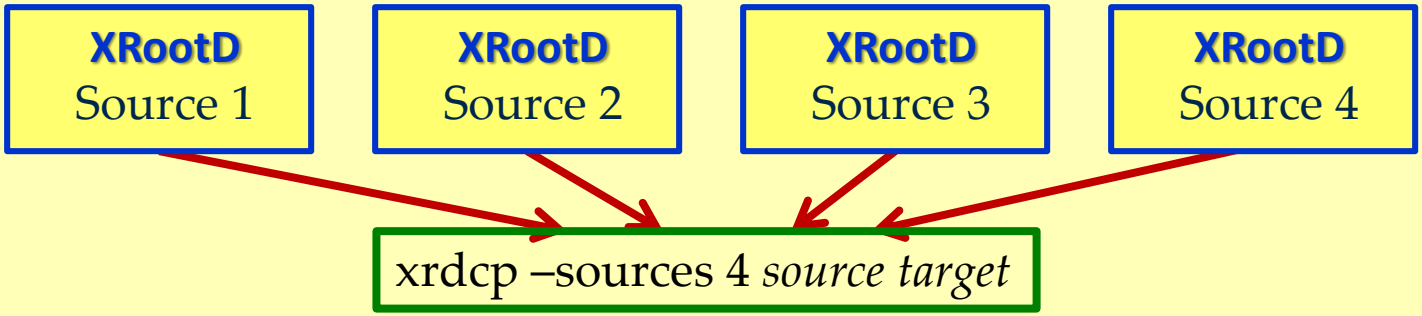
- Framework for distributed services
  - Builds on robust & recoverable **XRootD**
    - Uses a remote object execution model
- Current deployed for LSST qserv
  - Distributed unshared mySQL servers
    - Successfully being used with 100's of nodes
- API is still being refined
  - Will be released when finalized

Future Enhancements (not yet set)

- Multi-source copy
- Multi-Source load balancing client
- Eliminating **cmsd** write lookup delays
- Tracking file ownership
- Eliminating 64-node limit (exploratory)
- HTTP 2 plug-in

Multi-Source Copy

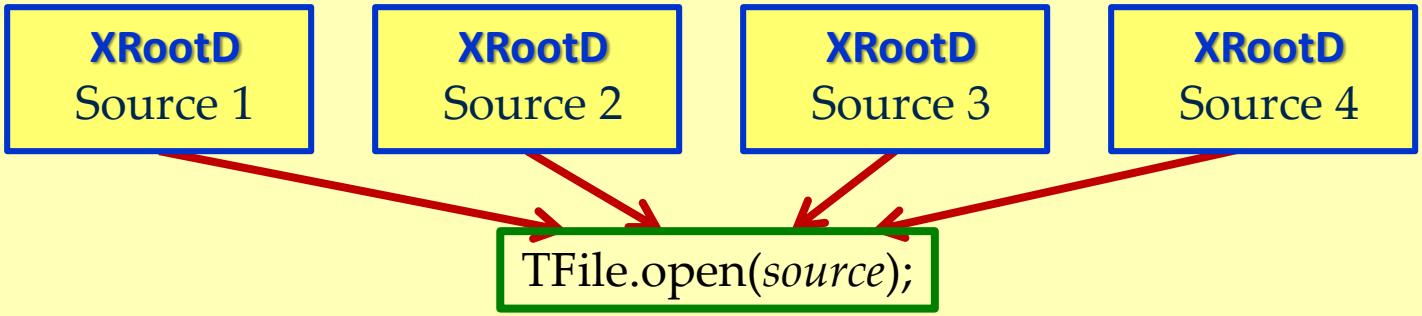
- Implement the –sources **xrdcp** option



- The *source* can be a redirector or metalink
  - I/O automatically balanced across sources
  - Advanced algorithm to avoid ending tail

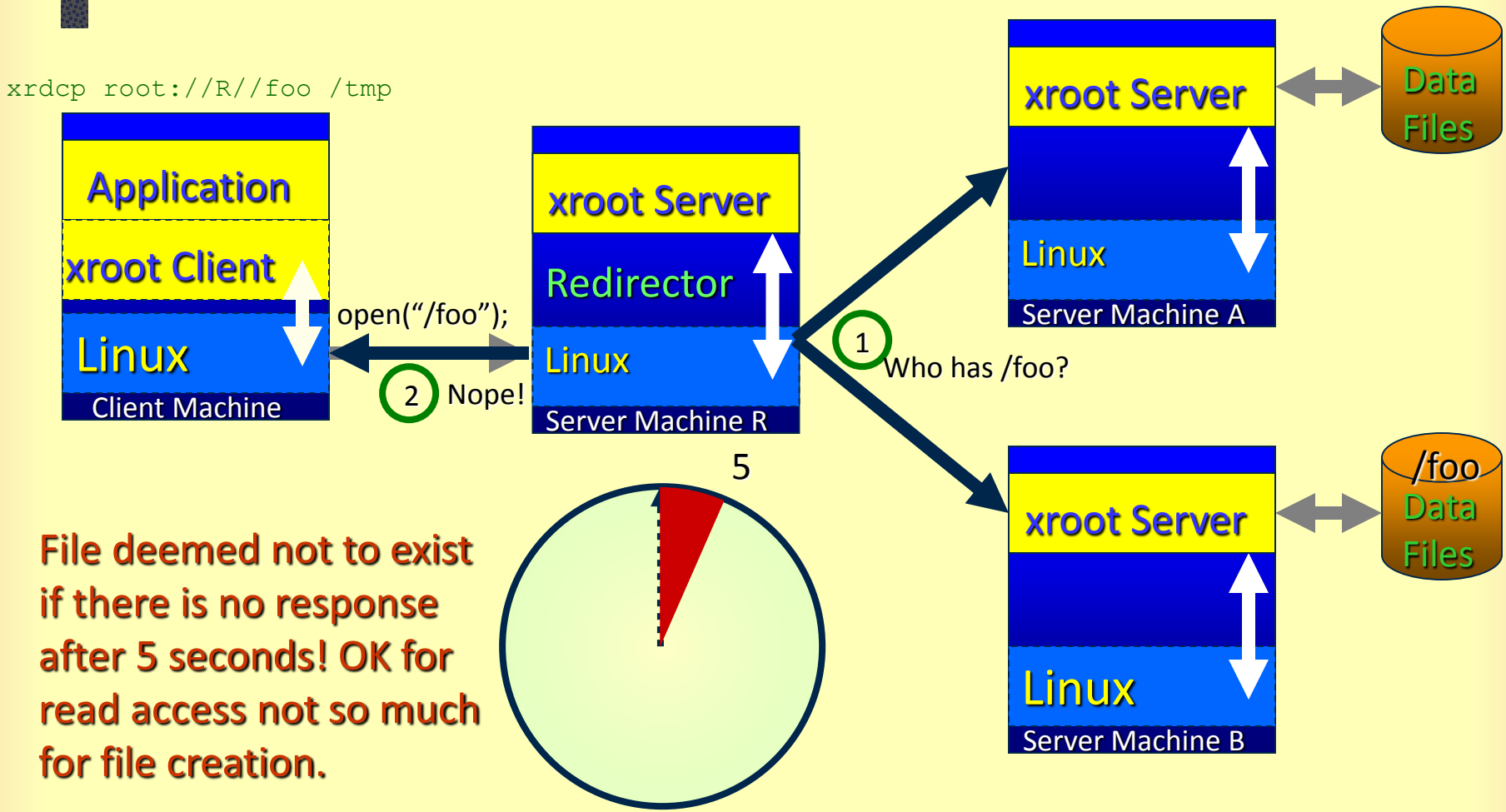
Multi-Source Load Balancing Client

- Similar to **xrdcp** but with a big twist



- The *source* can be a redirector or metalink
  - A new source is added only if current one slow
    - Can bounce around multiple sources
      - Determines by real-time performance metrics

The Missing File Problem



Eliminating cmsd Write Lookup Delay

- The cmsd uses a no response model
  - No response -> file does not exist
- Extremely scalable for analysis use case
  - Usually always looking for existing files
- Not so good for creating files
  - A small change in protocol can fix this
- Required for efficient handling of experiments desire for writable clusters

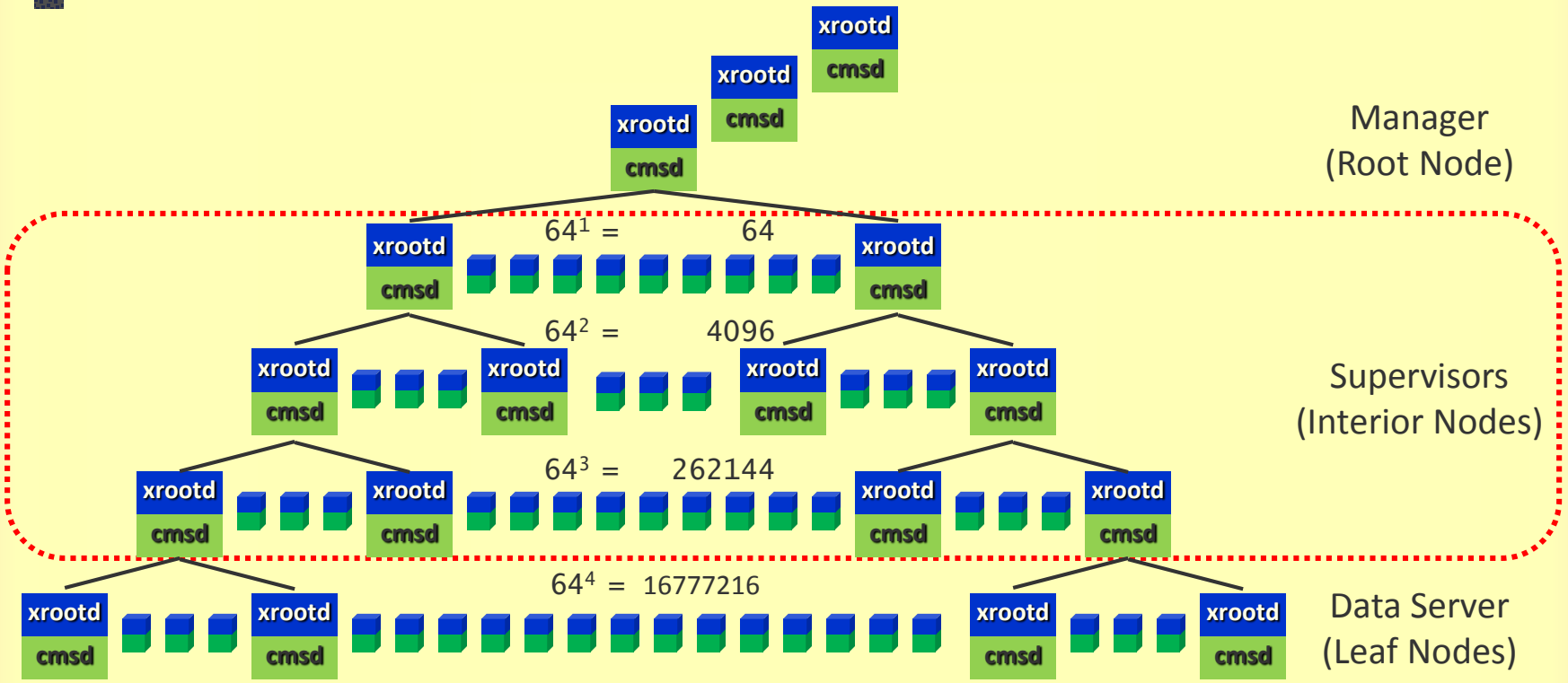
Tracking File Ownership

- UID/GID tracking of ownership
  - Available for certain authentication methods
    - GSI with a gridmap file
    - Kerberos (in domain only)
    - Simple Shared Secret
    - Unix
  - Must start **XRootD** as root
    - Security considerations abound
  - May allow of uid/gid based quotas

New Third Party Transfer

- New 3rd Party Transfer
  - Plan to use forwardable credentials
    - X.509 (i.e. Grid Certificates)
  - Allows almost universal 3rd party access
    - Only one of three parties needs to support it
      - The File Residency Manager already does this
  - Will coexist with current mechanism

Eliminating 64-Node Limit I



Eliminating 64-Node Limit II

- A B^{64} tree architecture is generally ideal
  - Fast scaling and highly parallel file search
- But it's cumbersome for large clusters
  - Need to deploy sufficient supervisor nodes
- Exploring different type of trees
  - B^{128} B^{256} B^{512} etc
- Parallelism is the biggest stumbling block
  - However, it would simplify configuration

That's All!

What's Your Wish List?