Scalla Advancements

xrootd /cmsd (f.k.a. olbd)

Fabrizio Furano
CERN – IT/PSS

Andrew Hanushevsky
Stanford Linear Accelerator Center

US Atlas Tier 2/3 Workshop
Stanford University/SLAC
28-November-07

http://xrootd.slac.stanford.edu

Stanford Linear Accelerator Center

Outline

- Current Elaborations
  - Composite Cluster Name Space
  - POSIX file system access via FUSE+xrootd
- New Developments
  - Cluster Management Service (cmsd)
    - Cluster globalization
- Conclusion

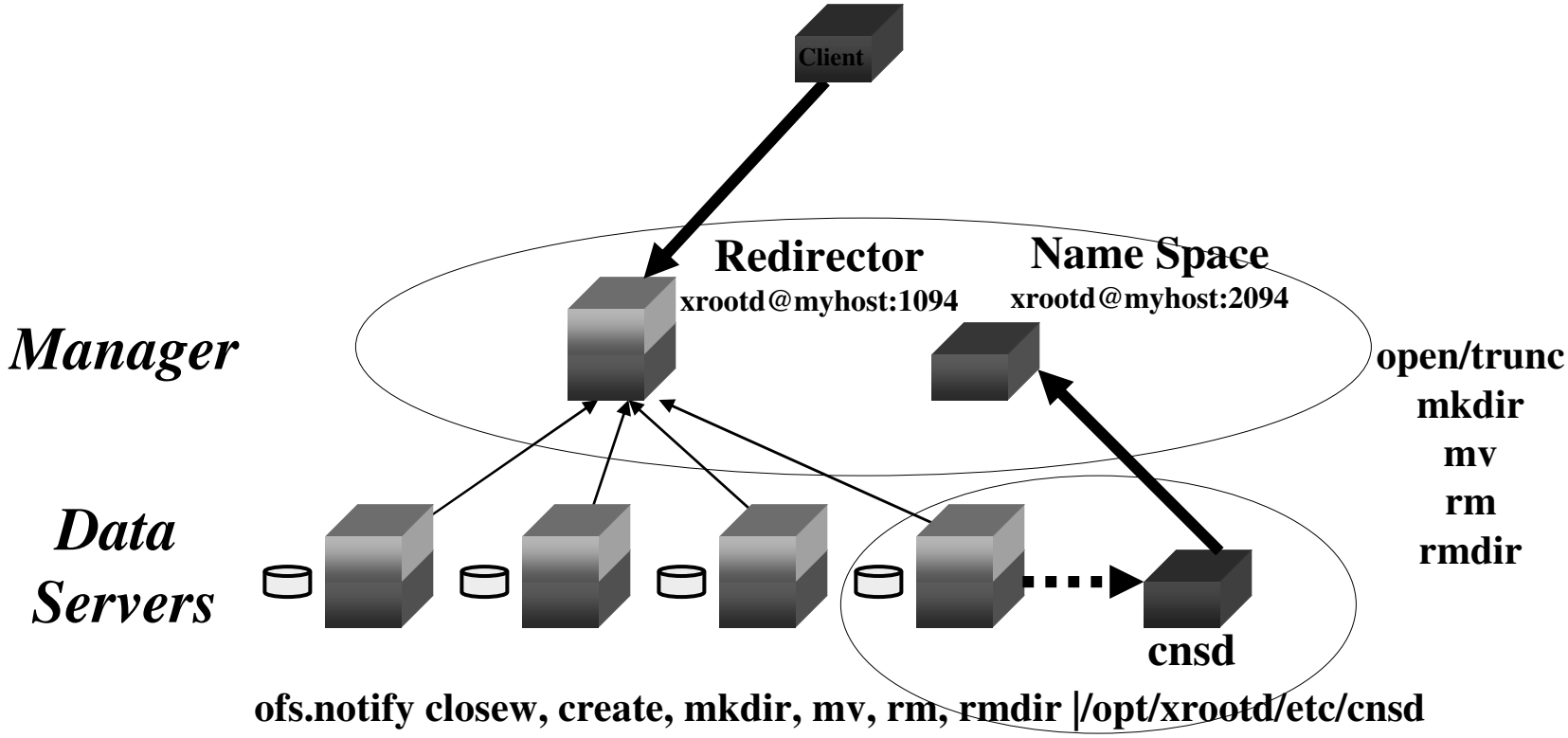
Stanford Linear Accelerator Center

The Distributed Name Space

- **Scalla** implements a distributed name space
  - Very scalable and efficient
  - Sufficient for data analysis
- Some users and applications (e.g., SRM) rely on a centralized name space
  - Spurred the development of a Composite Name Space (cnsd) add-on
    - Simplest solution with the least entanglement

Composite Cluster Name Space

opendir() refers to the directory structure maintained by xrootd:2094



ofs.notify closew, create, mkdir, mv, rm, rmdir |/opt/xrootd/etc/cnsd

Stanford Linear Accelerator Center

cnsd Specifics

- Servers direct name space actions to common xrootd(s)
  - Common xrootd maintains composite name space
    - Typically, these run on the redirector nodes
  - Name space replicated in the file system
    - No external database needed
    - Small disk footprint
  - Deployed at SLAC for Atlas
    - Needs synchronization utilities, more documentation, and packaging
      - See Wei Yang for details
    - Similar mySQL based system being developed by CERN/Atlas
      - Annabelle Leung <annabelle.leung@cern.ch>

Stanford Linear Accelerator Center

Data System vs File System

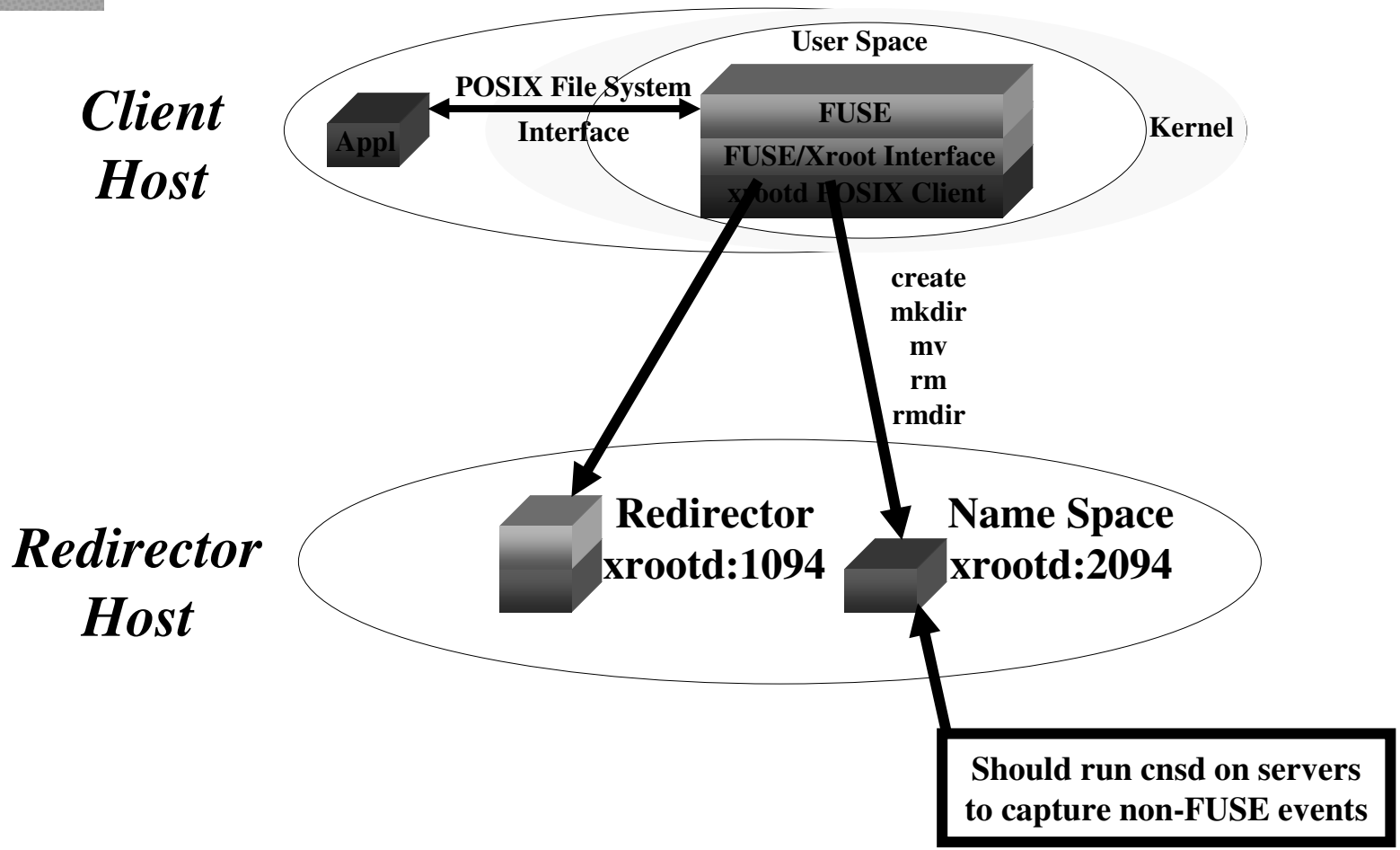
- **Scalla** is a data access system
  - Some users/applications want file system semantics
    - More transparent but many times less scalable
- For years users have asked ….
  - Can **Scalla** create a file system experience?

Stanford Linear Accelerator Center

What is FUSE

- Filesystem in Userspace
  - Used to implement a file system in a user space program
    - Linux 2.4 and 2.6 only
    - Refer to http://fuse.sourceforge.net/
  - Can use **FUSE** to provide xrootd access
    - Looks like a mounted file system
  - SLAC and FZK have xrootd-based versions of this
    - Wei Yang at SLAC
      - Tested and practically fully functional
    - Andreas Petzold at FZK
      - In alpha test, not fully functional yet

XrootdFS (Linux/FUSE/Xrootd)

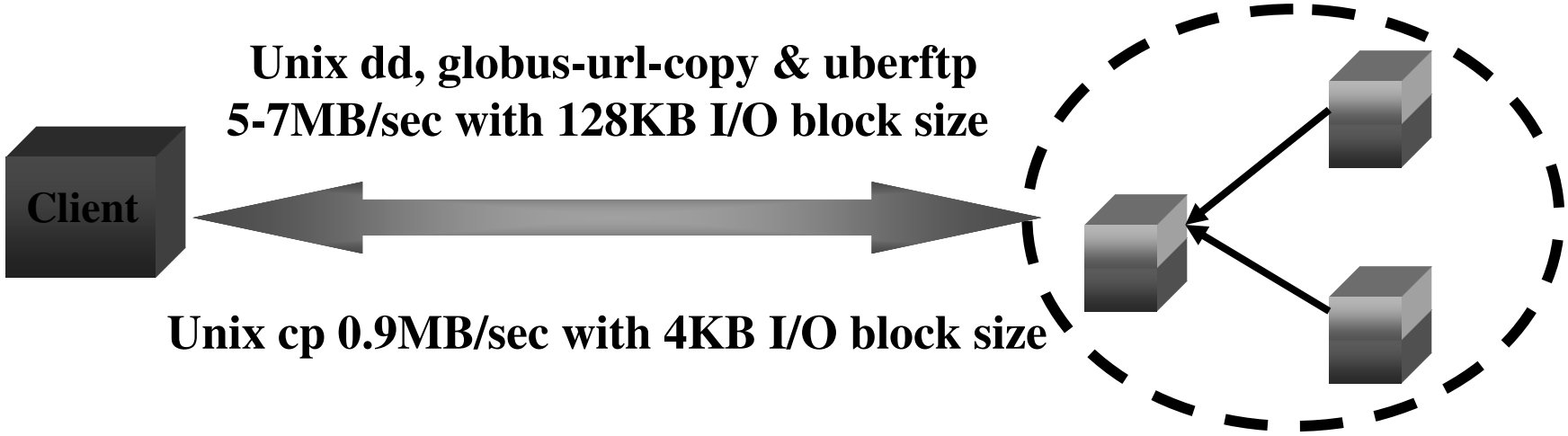


XrootdFS Performance

Sun V20z
RHEL4
2x 2.2Ghz AMD Opteron
4GB RAM
1Gbit/sec Ethernet

VA Linux 1220
RHEL3
2x 866Mhz Pentium 3
1GB RAM
100Mbit/sec Ethernet

Unix dd, globus-url-copy & uberftp
5-7MB/sec with 128KB I/O block size



Unix cp 0.9MB/sec with 4KB I/O block size

Conclusion: Better for some things than others.

Stanford Linear Accelerator Center

Why XrootdFS?

- Makes some things much simpler
  - Most SRM implementations run transparently
  - Avoid pre-load library worries
- But impacts other things
  - Performance is limited
    - Kernel-**FUSE** interactions are not cheap
    - Rapid file creation (e.g., tar) is limited
  - **FUSE** must be administratively installed to be used
    - Difficult if involves many machines (e.g., batch workers)
    - Easier if it involves an SE node (i.e., SRM gateway)

Stanford Linear Accelerator Center

Next Generation Clustering

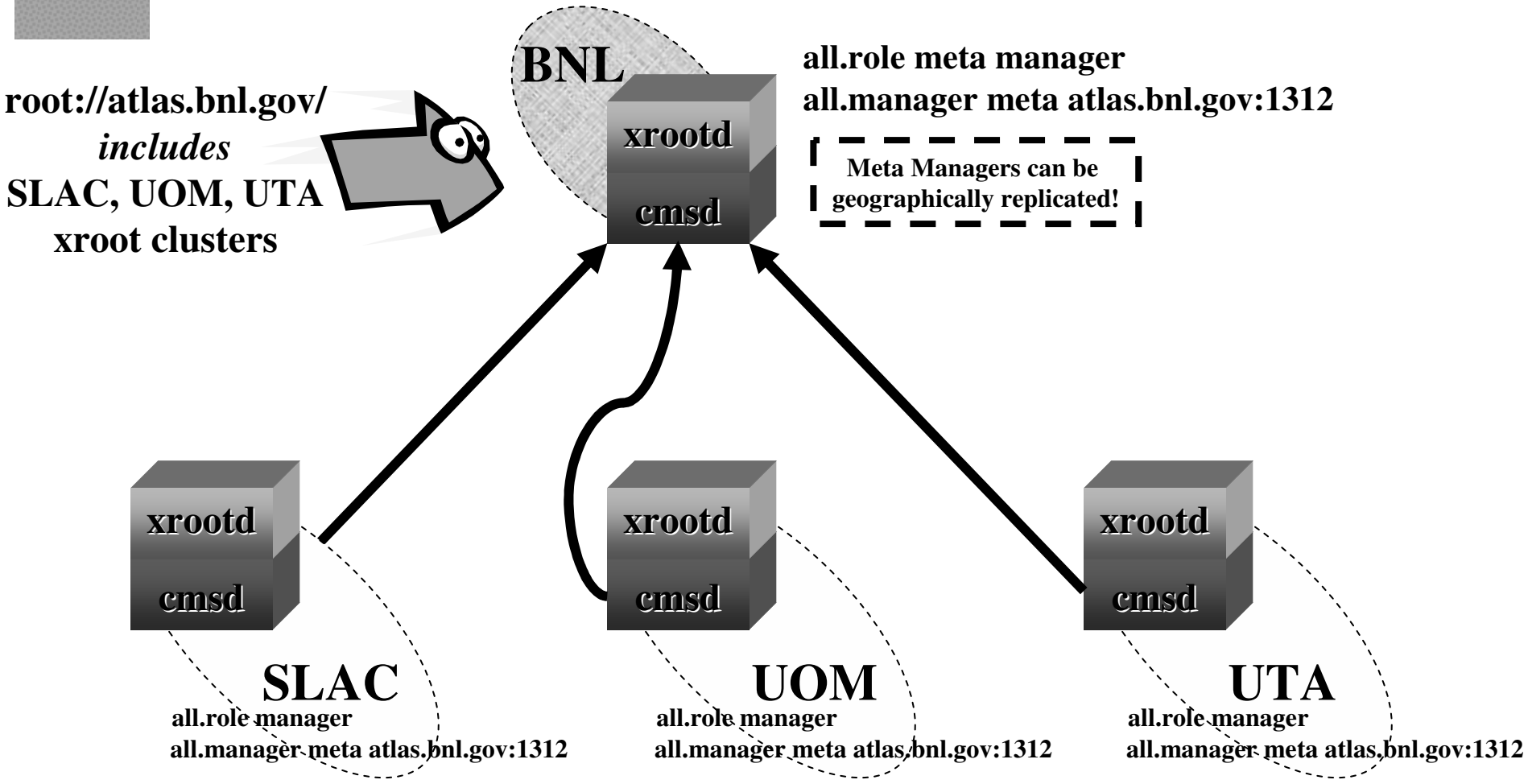
- Cluster Management Service (cmsd)
  - Functionally replaces olbd
    - Compatible with olbd config file
      - Unless you are using deprecated directives
    - Straight forward migration
      - Either run olbd or cmsd everywhere
  - Currently in alpha test phase
    - Available in CVS head
    - Documentation on web site

Stanford Linear Accelerator Center

cmsd Advantages

- Much lower latency
- New very extensible protocol
- Better fault detection and recovery
- Added functionality
  - Global clusters
  - Authentication
  - Server selection can include space utilization metric
  - Uniform handling of opaque information
  - Cross protocol messages to better scale xproof clusters
- Better implementation for reduced maintenance cost

Cluster Globalization



Stanford Linear Accelerator Center

Why Globalize?

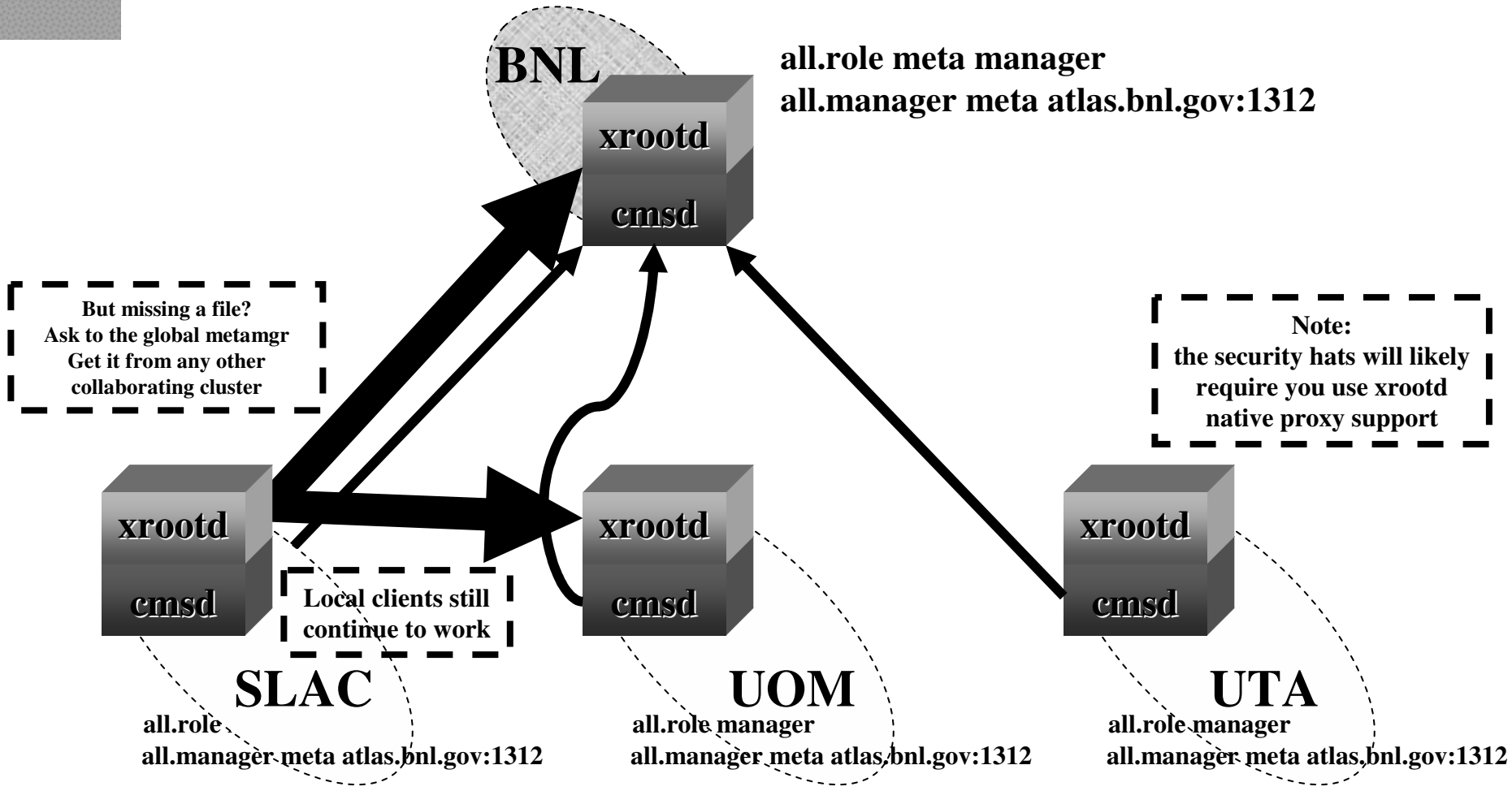
- Uniform view of participating clusters
  - Can easily deploy a virtual MSS
    - Included as part of the existing MPS framework
  - Try out real time WAN access
    - You really don't need data everywhere!
- Alice is slowly moving in this direction
  - The non-uniform name space is an obstacle
    - Slowly changing the old approach
    - Some workarounds could be possible, though

Stanford Linear Accelerator Center

Virtual MSS

- Powerful mechanism to increase reliability
  - Data replication load is widely distributed
  - Multiple sites are available for recovery
- Allows virtually unattended operation
  - Based on BaBar experience with real MSS
  - Automatic restore due to server failure
    - Missing files in one cluster fetched from another
      - Typically the fastest one which has the file really online
  - File (pre)fetching on demand
    - Can be transformed into a 3rd-party copy
      - When cmsd is deployed
  - Practically no need to track file location
    - But does not preclude the need for metadata repositories

The Virtual MSS Realized



Stanford Linear Accelerator Center

Dumb WAN Access*

- Setup: client at CERN, data at SLAC
  - 164ms RTT time, available bandwidth < 100Mb/s
- Test 1: Read a large ROOT Tree
  - (~300MB, 200k interactions)
    - Expected time: 38000s (latency)+750s (data)+CPU➔10 hrs!
- Test 2: Draw a histogram from that tree data
  - (6k interactions)
    - Measured time 20min
      - Using xrootd with WAN optimizations disabled

*Federico Carminati, *The* ALICE *Computing Status and Readiness*, LHCC, November 2007

Stanford Linear Accelerator Center

Smart WAN Access*

- Exploit xrootd WAN Optimizations
  - TCP multi-streaming: for up to 15x improvement data WAN throughput
  - The ROOT TTreeCache provides the hints on ”future” data accesses
  - TXNetFile/XrdClient ”slide through” keeping the network pipeline full
- Data transfer goes in parallel with computation
  - Throughput improvement comparable to “batch” file-copy tools
    - 70-80%, improvement and we are doing a live analysis, not a file copy!
- Test 1 actual time: 60-70 seconds
  - Compared to 30 seconds using a Gb LAN
    - Very favorable for sparsely used files
- Test 2 actual time: 7-8 seconds
  - Comparable to LAN performance
    - 100x improvement over dumb WAN access (i.e., 20 minutes)

*Federico Carminati, *The* ALICE *Computing Status and Readiness*, LHCC, November 2007

Stanford Linear Accelerator Center

Conclusion

- Scalla is a robust framework
  - Elaborative
    - Composite Name Space
    - XrootdFS
  - Extensible
    - Cluster globalization
- Many opportunities to enhance data analysis
  - Speed and efficiency

Stanford Linear Accelerator Center

Acknowledgements

- Software Collaborators
  - INFN/Padova: Alvise Dorigo
  - Root: Fons Rademakers, Gerri Ganis (security), Beterand Bellenet (windows)
  - Alice: Fabrizio Furano (client) , Derek Feichtinger, Guenter Kickinger
  - CERN: Andreas Peters (Castor)
  - STAR/BNL: Pavel Jackl
  - Cornell: Gregory Sharp
  - SLAC: Jacek Becla, Tofigh Azemoon, Wilko Kroeger, Bill Weeks
  - BaBar: Pete Elmer (packaging)
- Operational collaborators
  - BNL, CNAF, FZK, INFN, IN2P3, RAL, SLAC
- Funding
  - US Department of Energy
    - Contract DE-AC02-76SF00515 with Stanford University