Scalla/xrootd

Andrew Hanushevsky
SLAC National Accelerator Laboratory
Stanford University
19-August-2009
Atlas Tier 2/3 Meeting

http://xrootd.slac.stanford.edu/

Outline

- System Component Summary
- Recent Developments
- Scalability, Stability, & Performance
- ATLAS Specific Performance Issues
- Faster I/O
  - The SSD Option
- Conclusions

The Components

- xrootd
  - Provides actual data access
- cmsd
  - Glues multiple xrootd's into a cluster
- cnsd
  - Glues multiple name spaces into one name space
- BeStMan
  - Provides SRM v2+ interface and functions
- FUSE
  - Exports xrootd as a file system for BeStMan
- GridFTP
  - Grid data access either via FUSE or POSIX Preload Library

SLAC
NATIONAL ACCELERATOR LABORATORY

Recent Developments

- File Residency Manager (FRM)
  - April, 2009
- Torrent WAN transfers
  - May, 2009
- Auto-reporting summary monitoring data
  - June, 2009
- Ephemeral files
  - July, 2009
- Simple Server Inventory
  - August, 2009

File Residency Manager (FRM)

- Functional replacement for MPS scripts
  - Currently, includes…
    - Pre-staging daemon **frm_pstgd** and agent **frm_pstga**
      - Distributed copy-in prioritized queue of requests
      - Can copy from any source using any transfer agent
      - Used to interface to real and virtual MSS's
    - **frm_admin** command
      - Audit, correct, obtain space information
        - Space token names, utilization, etc.
      - Can run on a live system

Torrent WAN Transfers

- The xrootd already supports parallel TCP paths
  - Significant improvement in WAN transfer rate
    - Specified as xrdcp –S *num*
- New Xtreme copy mode option
  - Uses multiple data sources bit torrent-style
    - Specified as xrdcp –x
  - Transfers to CERN; examples:
    - 1 source (.de): 12MB/sec (1 stream)
    - 1 source (.us): 19MB/sec (15 streams)
    - 4 sources (3 x .de + .ru): 27MB/sec (1 stream each)
    - 4 sources + || streams: 42MB/Sec (15 streams each)
    - 5 sources (3 x .de + .it + .ro): 54MB/Sec (15 streams each)

Summary Monitoring

- xrootd has built-in summary monitoring
  - In addition to full detailed monitoring
- Can auto-report summary statistics
  - xrd.report configuration directive
- Data sent to up to two central locations
  - Accommodates most current monitoring tools
    - Ganglia, GRIS, Nagios, MonALISA, and perhaps more
      - Requires external xml-to-monitor data convertor
      - Can use provided stream multiplexing and xml parsing tool
        - Outputs simple key-value pairs to feed a monitor script

Ephemeral Files

- Files that persist only when successfully closed
  - Excellent safeguard against leaving partial files
    - Application, server, or network failures
      - E.g., GridFTP failures
  - Server provides grace period after failure
    - Allows application to complete creating the file
      - Normal xrootd error recovery protocol
      - Clients asking for read access are delayed
      - Clients asking for write access are usually denied
        - Obviously, original creator is allowed write access
  - Enabled via xrdcp –**P** option or **ofs.posc** CGI element

SLAC NATIONAL ACCELERATOR LABORATORY

Simple Server Inventory (SSI)

- A central file inventory of each data server
  - Does *not* replace PQ2 tools (Neng Xu, Univerity of Wisconsin)
    - Good for uncomplicated sites needing a server inventory
  - Inventory normally maintained on *each* redirector
    - But, can be centralized on a single server
    - Automatically recreated when lost
    - Updated using rolling log files
      - Effectively no performance impact
  - Flat text file format
    - LFN, Mode, Physical partition, Size, Space token
      - “cns_ssi list” command provides formatted output

SLAC
NATIONAL ACCELERATOR LABORATORY

Stability & Scalability

- xrootd has a 5+ year production history
  - Numerous high-stress environments
    - BNL, FZK, IN2P3, INFN, RAL, SLAC
  - Stability has been vetted
    - Changes are now very focused
      - Functionality improvements
      - Hardware/OS edge effect limitations
      - Esoteric bugs in low use paths
  - Scalability is already at the theoretical maximum
    - E.g., STAR/BNL runs a 400+ server production cluster

Performance I

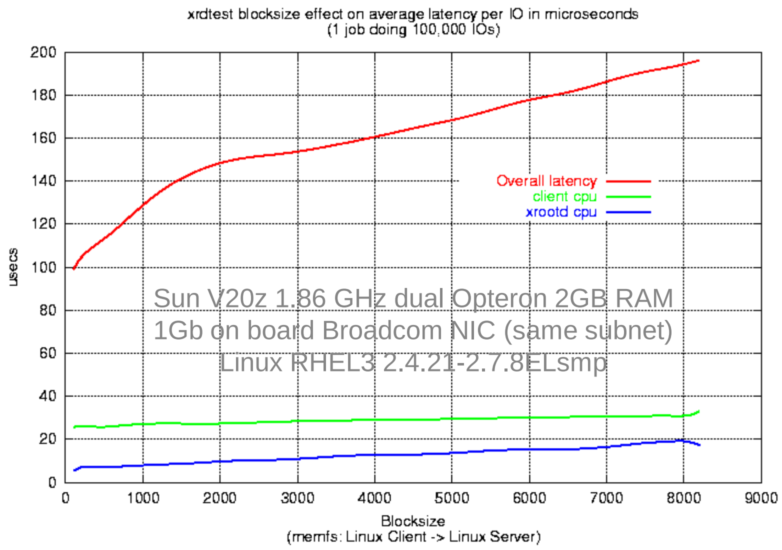
- Following figures are based on actual measurements
  - These have also been observed by many *production* sites
    - E.G., BNL, IN2P3, INFN, FZK, RAL , SLAC

CAVEAT!

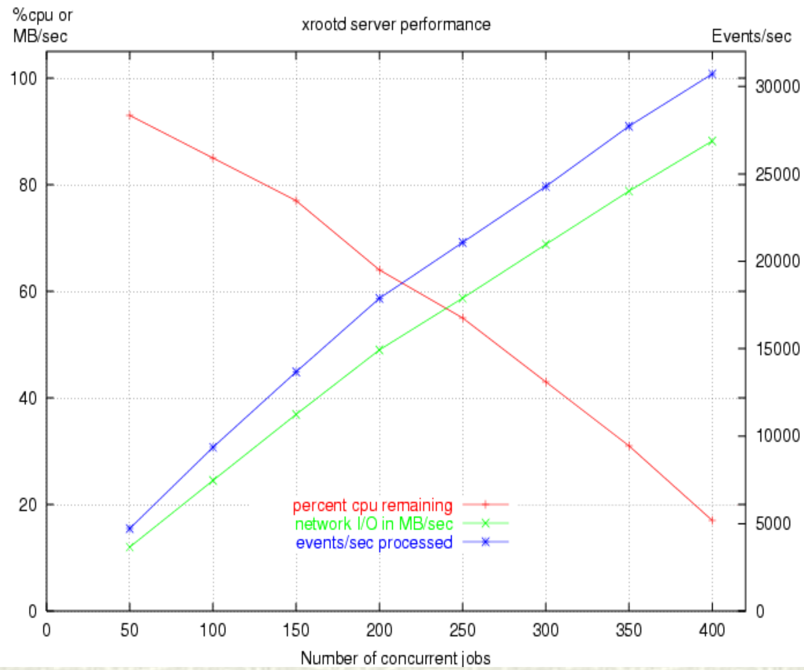
  - Figures apply only to the *reference* implementation
  - Other implementations vary significantly
    - Castor + xrootd protocol driver
    - dCache + native xrootd protocol implementation
    - DPM + xrootd protocol driver + cmsd XMI
    - HDFS + xrootd protocol driver

Performance II

Latency



Capacity vs. Load



xrootd latency < 10µs → network or disk latency dominates

Practically, at least ≈10,000 Ops/Second with linear scaling

xrootd+cmsd latency *(not shown)* 350µs →»1000 opens/second

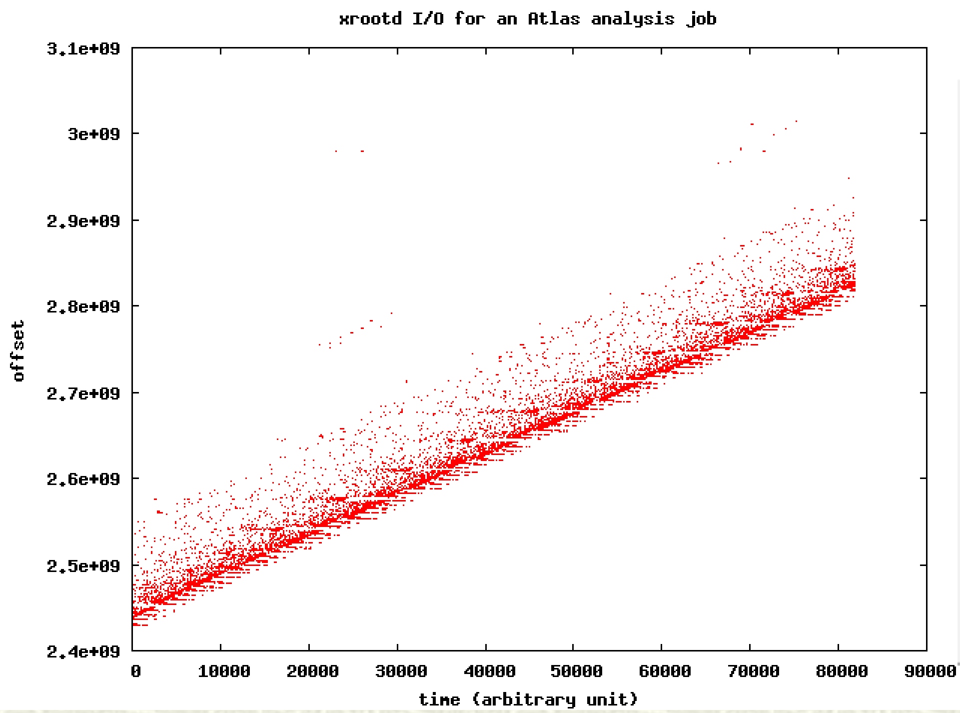
Performance & Bottlenecks

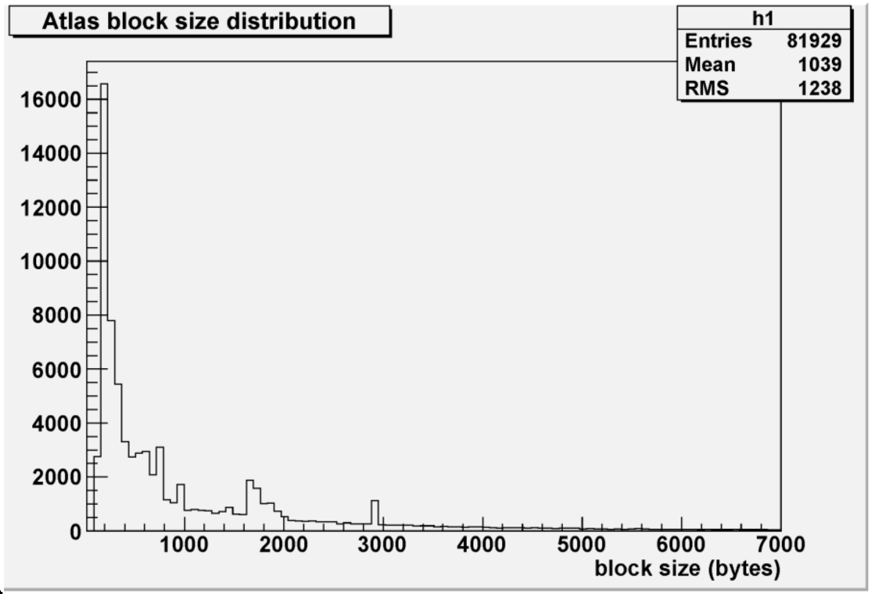
- High performance + linear scaling
  - Makes client/server software virtually transparent
    - A 50% faster xrootd yields 3% overall improvement
    - Disk subsystem and network become determinants
      - This is actually excellent for planning and funding

HOWEVER

  - Transparency makes other bottlenecks apparent
    - Hardware, Network, Filesystem, or Application
      - Requires deft trade-off between CPU & Storage resources
    - But, bottlenecks usually due to unruly applications
      - Such as ATLAS analysis

ATLAS Data Access Pattern





SLAC NATIONAL ACCELERATOR LABORATORY

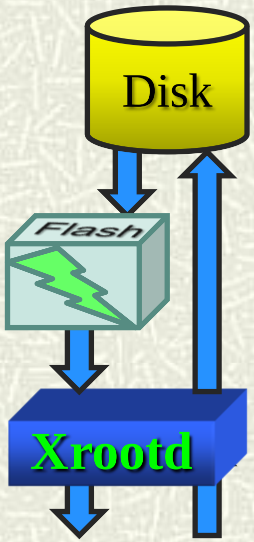
ATLAS Data Access Problem

- Atlas analysis is fundamentally indulgent
  - While xrootd can sustain the request load the H/W cannot
- Replication?
  - Except for some files this is not a universal solution
    - The experiment is already disk space insufficient
- Copy files to local node for analysis?
  - Inefficient, high impact, and may overload the LAN
  - Job will still run slowly and no better than local disk
- Faster hardware (e.g., SSD)?
  - This appears to be generally cost-prohibitive
    - That said, we are experimenting with smart SSD handling

Faster Scalla I/O (The SSD Option)

Latency only as good as the hardware (xrootd adds < 10μs latency)

- Scalla component architecture fosters experimentation
  - Research on intelligently using SSD devices



ZFS Specific

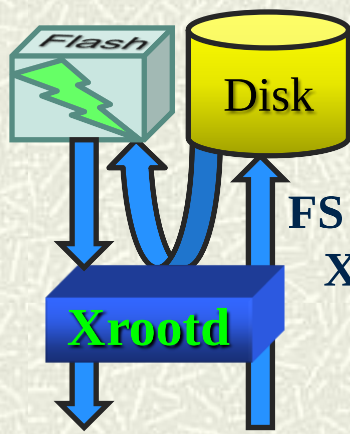
ZFS caches disk blocks

Xrootd I/O:

Data sent from RAM/Flash

Data received sent to Disk

R/O Disk Block Cache



R/O Disk File Cache

FS Agnostic

Xrootd caches files

Xrootd I/O:

Data sent from RAM/Flash

Data received sent to Disk

The ZFS SSD Option

- Decided against this option (for now)
  - Too narrow
    - OpenSolaris now or Solaris 10 Update 8 (likely 12/09)
    - Linux support requires ZFS adoption
      - Licensing issues stand in the way
  - Current caching algorithm is a bad fit for HEP
    - Optimized for small SSD’s
    - Assumes large hot/cold differential
      - Not the HEP analysis data access profile

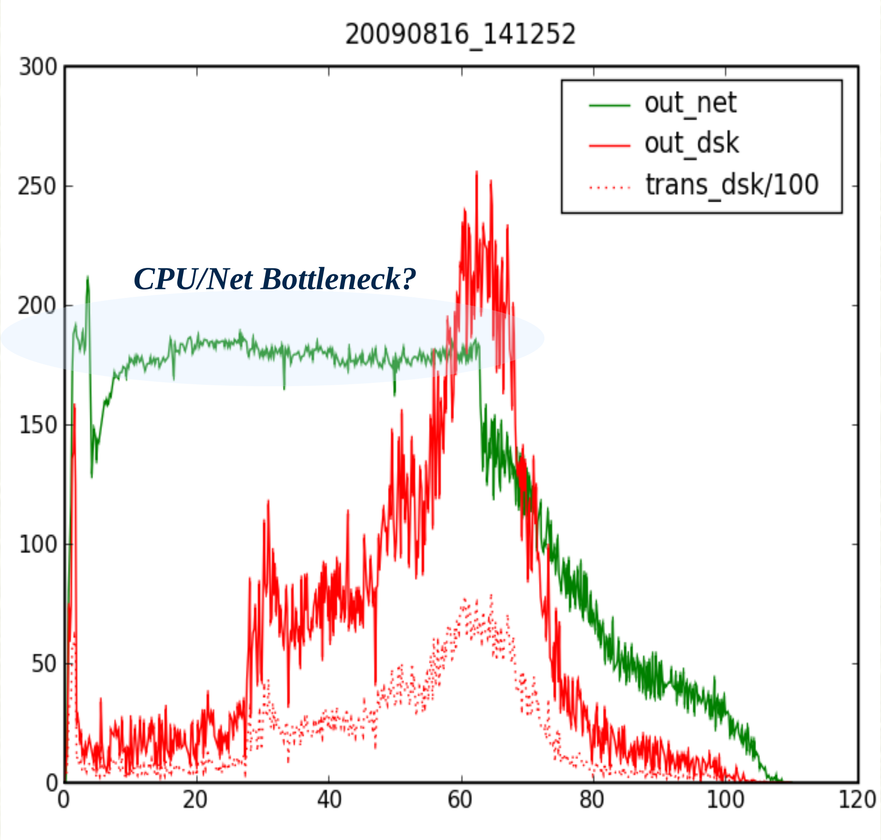
The xrootd SSD Option

- Currently architecting appropriate solution
  - Fast track is to use staging infrastructure
    - Whole files are cached
    - Hierarchy: SSD, Disk, Real MSS, Virtual MSS
  - Slower track is more elegant
    - Parts of files are cached
      - Can provide parallel mixed mode (SSD/Disk) access
    - Basic code already present
      - But needs to be expanded
- Will it be effective?

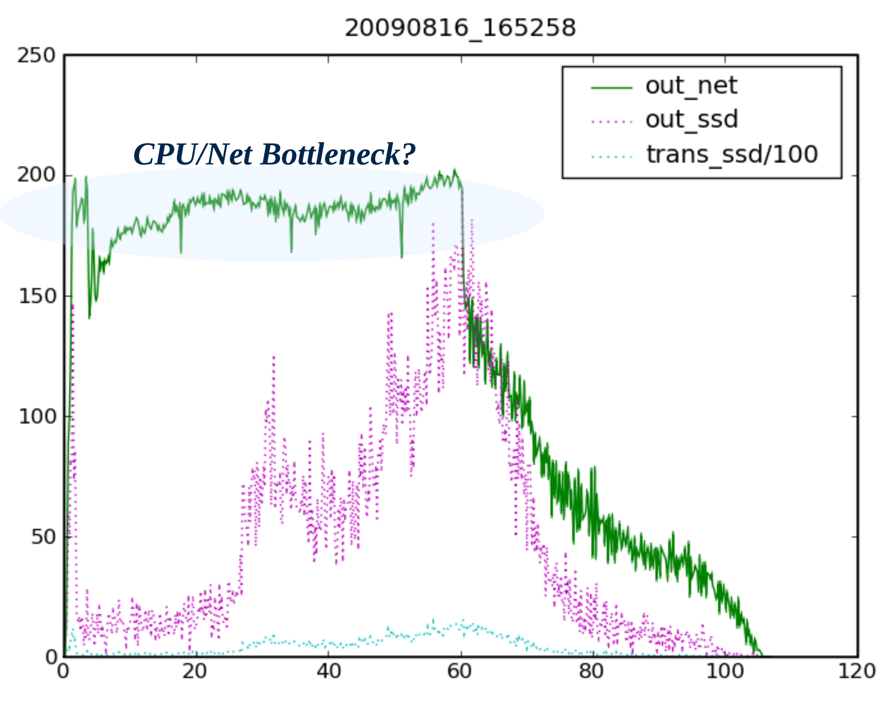
Disk vs SSD With 323 Clients

MB/s

Min



Disk I/O



SSD I/O

What Does This Mean?

- Well tuned disk can equal SSD Performance
  - True when number of well-behaved clients < *small* **n**
    - Either 343 Fermi/GLAST clients not enough or
    - Hitting some undiscovered bottleneck
- Huh? What about ATLAS clients?
  - Difficult if not impossible to get
    - Current grid scheme *prevents* local tuning & analysis
      - Desperately need a “send *n* test jobs” button
  - We used what we could easily get
    - Fermi read size about 1K and somewhat CPU intensive

Conclusion

- Xrootd is a lightweight data access system
  - Suitable for resource constrained environments
    - Human as well as hardware
  - Rugged enough to scale to large installations
    - CERN analysis & reconstruction farms
  - Flexible enough to make good use of new H/W
    - Smart SSD
  - Available in OSG VDT & CERN root package
- Visit the web site for more information
  - http://xrootd.slac.stanford.edu/

Acknowledgements

- Software Contributors
  - Alice: Derek Feichtinger
  - CERN: Fabrizio Furano , Andreas Peters
  - Fermi/GLAST: Tony Johnson (Java)
  - Root: Gerri Ganis, Beterand Bellenet, Fons Rademakers
  - SLAC: Tofigh Azemoon, Jacek Becla, Andrew Hanushevsky, Wilko Kroeger
  - LBNL: Alex Sim, Junmin Gu, Vijaya Natarajan (BeStMan team)
- Operational Collaborators
  - BNL, CERN, FZK, IN2P3, RAL, SLAC, UVIC, UTA
- Partial Funding
  - US Department of Energy
    - Contract DE-AC02-76SF00515 with Stanford University