Scalla

Back Through The Future

Andrew Hanushevsky

SLAC National Accelerator Laboratory
Stanford University
8-April-10
http://xrootd.slac.stanford.edu

Outline

- Introduction
  - History
  - Design points
  - Architecture
- Scalla Usage
- Future projects involving Scalla
- Conclusion

What is Scalla?

- **S**tructured **C**luster **A**rchitecture for **L**ow **L**atency **A**ccess
  - Low Latency Access to data via xrootd servers
    - POSIX-style byte-level random access
      - Arbitrary data organized as files
      - Hierarchical directory-like name space
    - Protocol includes high performance features
  - Structured Clustering provided by cmsd servers
    - Exponentially scalable and self organizing

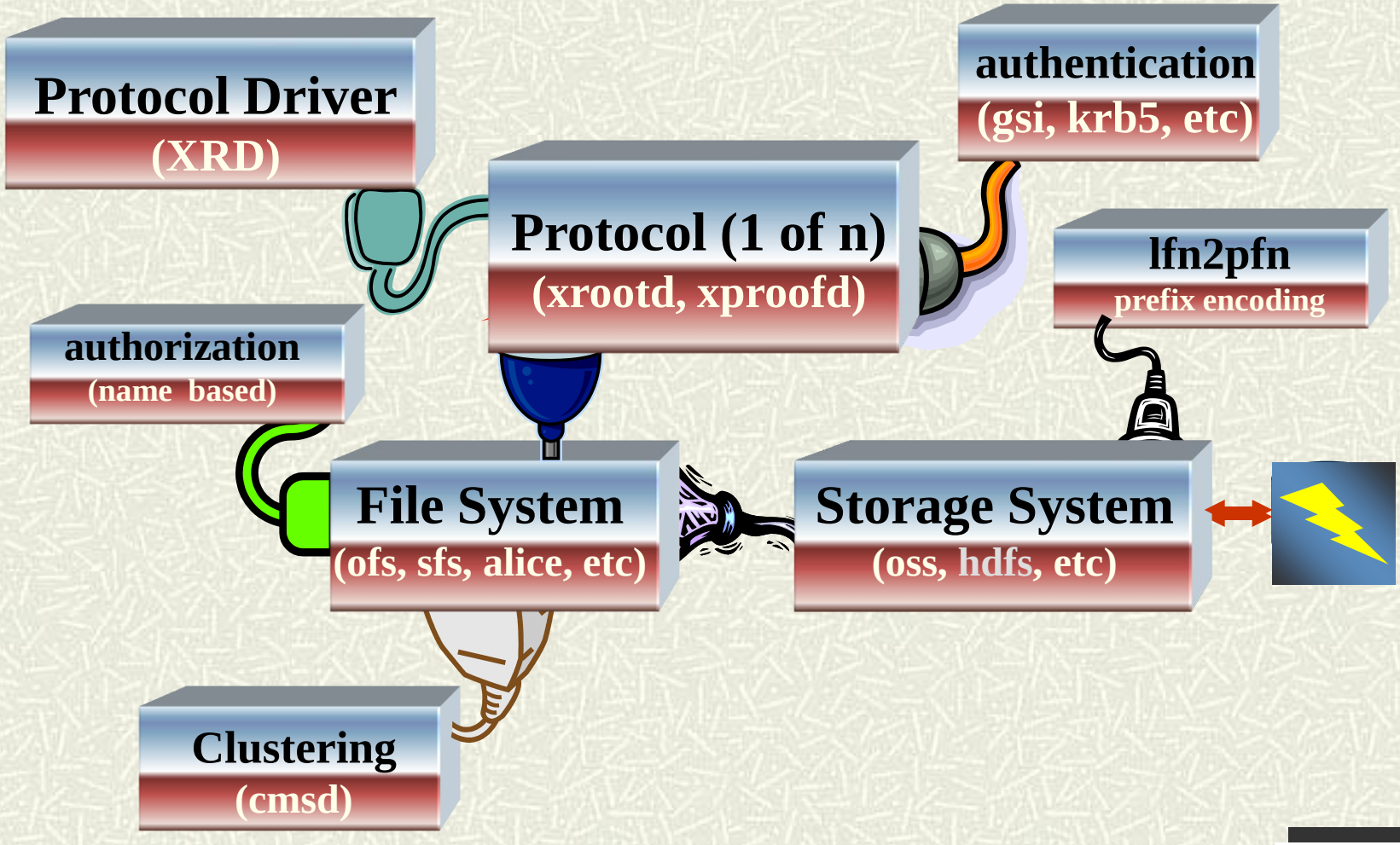
Brief History

- 1997 – Objectivity, Inc. collaboration
  - Design & Development to scale Objectivity/DB
    - First attempt to use commercial DB for Physics data
    - Very successful but problematical
- 2001 – BaBar decides to use root framework vs Objectivity
  - Collaboration with INFN, Padova & SLAC created
    - Design & develop high performance data access system
    - Work based on what we learned with Objectivity
- 2003 – First deployment of xrootd system at SLAC
- 2005 – Collaboration extended
  - Root collaboration & Alice LHC experiment, CERN
  - Over 100 deployment sites across the world

The Scalla Design Point

- Write once read many times processing mode
  - Capitalize on simplified semantics
- Large scale small block sparse random access
  - Provide very low latency per request
- Secure large compute investment
  - Provide high degree of fault-tolerance
- Accommodate more data than disk space
  - Integrate offline storage (Mass Storage System)
- Adapt to disparate deployment environments
  - Robust security framework
  - Component based replaceable subsystems
  - Simple setup with no 3rd party software requirements
    - In typical cases

Scalla Plug-in Architecture



Clustering

- xrootd servers can be clustered
  - Increase access points and available data
  - Allows for automatic failover
- Structured point-to-point connections
  - Cluster overhead (human & non-human) scales linearly
    - Cluster size is not limited (easily accommodates 262,144 servers)
    - I/O performance is not affected
- Symmetric cookie-cutter management
  - Always pairs xrootd & cmsd server

How Scalla Is Accessed

- The root framework
  - Used by most HEP experiments (MacOS, Unix and Windows)
- A mounted FUSE Scalla file system (Linux and MacOS only)
- SRM and gridFTP
  - General grid access (Unix only)
- POSIX preload library
  - Any POSIX compliant application (Unix only, no recompilation needed)
- xrdcp
  - The copy command (MacOS, Unix and Windows)
- xprep
  - The redirector seeder command (MacOS, Unix and Windows)
- xrd
  - The admin interface for meta-data operations (MacOS, Unix and Windows)

Who Uses Scalla?

- US
  - ATLAS (SLAC, ANL, UWisc$_{\text{160-node cluster}}$, UTA, UVIC$^{\text{Canada}}$, more to follow)
  - BaBar (SLAC, IN2P3$^{\text{France}}$, INFN$^{\text{Italy}}$)
  - Fermi/GLAST (SLAC & IN2P3$^{\text{France}}$)
  - STAR (BNL $_{\text{600-node cluster}}$)
- CERN (Switzerland)
  - To support most LHC local site data analysis
- LHC ALICE (World-Wide)
  - Global cluster of over 90 sites
- IN2P3 (France)
  - Over 12 unrelated physics, astronomy, & biomed experiments
- There are many many more
  - E.G., all sites running the Parallel Root Facility (PROOF)

Scalla Flexibility

- Engineered to play well in different contexts
  - Flexible plug-in architecture
  - Robust multi-protocol security
    - KRB4/5, GSI, SSL, Shared Secret, password, unix
  - Highly scalable and fault-tolerant
  - Multi-platform
    - HP/US, Linux, MacOS, Solaris, Windows (client only)
- A good framework for the future

Future Scalla-Based Projects I

- SSD Based Multi-Tiered Storage
  - Use Scalla's automatic data migration facility to keep active data on expensive SSD.
  - Determine usability of SSD's in this mode to improve performance and/or reduce overall storage costs
  - LDRD submitted

Future Scalla-Based Projects II

- Data Direct Networks Port
  - Port Scalla/xrootd to run natively on the DDN head-node for decreased latency, perhaps increased throughput.
  - Waiting for DDN to supply hardware
    - Requires lots of accounting/support co-ordination

Future Scalla-Based Projects III

- PetaCache
  - Pair SLAC-developed SOFI (Mike Huffer's Sea Of Flash) system with Scalla to determine it's usability for high-performance data access in typical analysis environments
    - In progress, waiting for software interface specifications and access to hardware

Future Scalla-Based Projects IV

- ExaScale mySQL
  - Use Scalla framework to cluster large numbers of mySQL servers to provide a fault-tolerant map/reduce functionality for relational databases
  - ASCR proposal submitted to DOE
    - Selection will be announced in the fall
  - This is in support of LSST

Future Scalla-Based Projects V

- Secure export of Lustre Filesystem
  - Use Scalla to securely and efficiently export Lustre to “laptops” for remote researchers
    - Can be done via FUSE
      - Have a working version for Linux and MacOS
      - Need to pair Windows xrootd client with Dokan
        - Will provide equivalent functionaility
  - In support of LCLS researchers
    - Idea is currently being floated around

Future Scalla-Based Projects VI

- Multi-Tiered Storage
  - Provide mechanism to mix and match hardware
    - Expensive disk, Cheap Disk, Tape
      - Yet maintain high I/O performance
  - Already supported by the software
    - A matter of configuration
  - In support of ATLAS SLAC Tier2
    - Increase effective disk capacity at highly reduced cost
    - In progress

External Future Projects

- Scalla + HDFS (Brian Bockelman, University of Nebraska)
  - Provide the best features of Hadoop and Scalla
- Scalla + DPM (David Smith, CERN)
  - Enables Disk Pool Manager global participation
- Scalla + Castor (Andreas Peters, CERN)
  - Overcomes the high latency of CERN's MSS
    - Near-term plan is to move toward pure Scalla
- Scalla + SSD (Andreas Peters, CERN)
  - Provide block level SSD caching
    - In alpha test at BNL

Conclusion

- Scalla is a highly successful HEP s/w system
  - It works as advertised
  - The feature set is tuned for large scale data access
  - Collaboratively designed and developed
  - The only system of its kind freely available
- It is widely used in the HEP community
  - Increasing use in the Astro community
- Being actively developed and adapted

Acknowledgements

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