Distributed Xrootd

Derek Weitzel & Brian Bockelman

Outline

- Motivation
- Data Access and Xrootd Concepts
- WLCG Demonstrator Project
- Sample T3 sites
- Interacting with Users

Motivation

- Data scalability for the next LHC run in 2013
- Tiered architecture = many copies
- Explicit data placement implies we must be smarter than our users and know future usage patterns
- This is a common problem, internet companies have found solutions a decade ago.
- Common solution = Caching

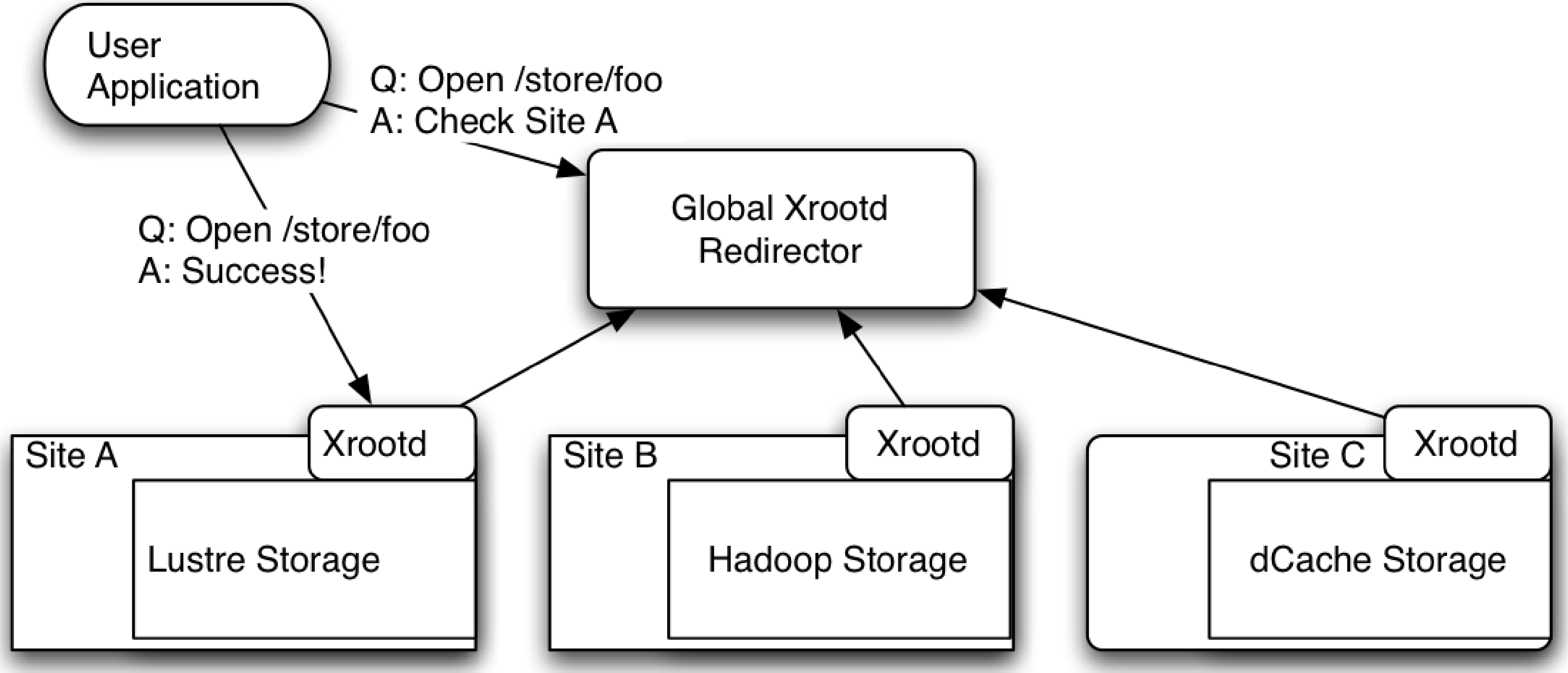
Data Access

- We want to work on improving **end-user data access**. Goals:
- **Reliability**: The end-user should never see an I/O error or failure propagated up to their application unless no USCMS site can serve the file. Failures should be caught as early as possible and I/O retried or rerouted to a different site (possibly degrading the service slightly).
- **Transparency**: All actions of the underlying system should be automatic for the user – catalog lookups, redirections, reconnections. There should not be a different workflow for accessing the data ``close by" versus halfway around the world. This implies the system serves user requests almost instantly; opening files should be a ``lightweight" operation.
- **Usability**: All CMS application frameworks (CMSSW, FWLite, bare ROOT) must natively integrate with any proposed solution. The proposed solution must not degrade the event processing rate significantly.
- **Global**: A CMS user should be able to get at any CMS file through the Xrootd service.
- *To achieve these goals, we are investigating xrootd-based technology.*

Xrootd Concepts

- Getting access to data:
  - User contacts a redirector.
  - Redirector searches through all the nodes which subscribe to it to see who has the data.
  - Selects the “Best Node” to serve the file.
  - Client is redirected to the correct data node.
- Originally, this was done within a site – the “redirector” was the headnode and the “data server” was a big disk server.

Xrootd Concepts



Xrootd Concepts

- In fact, we can make the redirector a global node and the "data server" an entire site, as demonstrated in the previous slide.
- This is a one-layer tree: you can actually have an arbitrary number of layers.
  - Xrootd will first query "down the tree" to look for close-by replicas; if none are found, it goes "up the tree" to find others in the cluster.
- In order to provide high reliability, the concepts of multiple parent nodes was added – meaning Xrootd is really providing a resilient network to find data. Can be an arbitrary graph, not necessarily a tree.

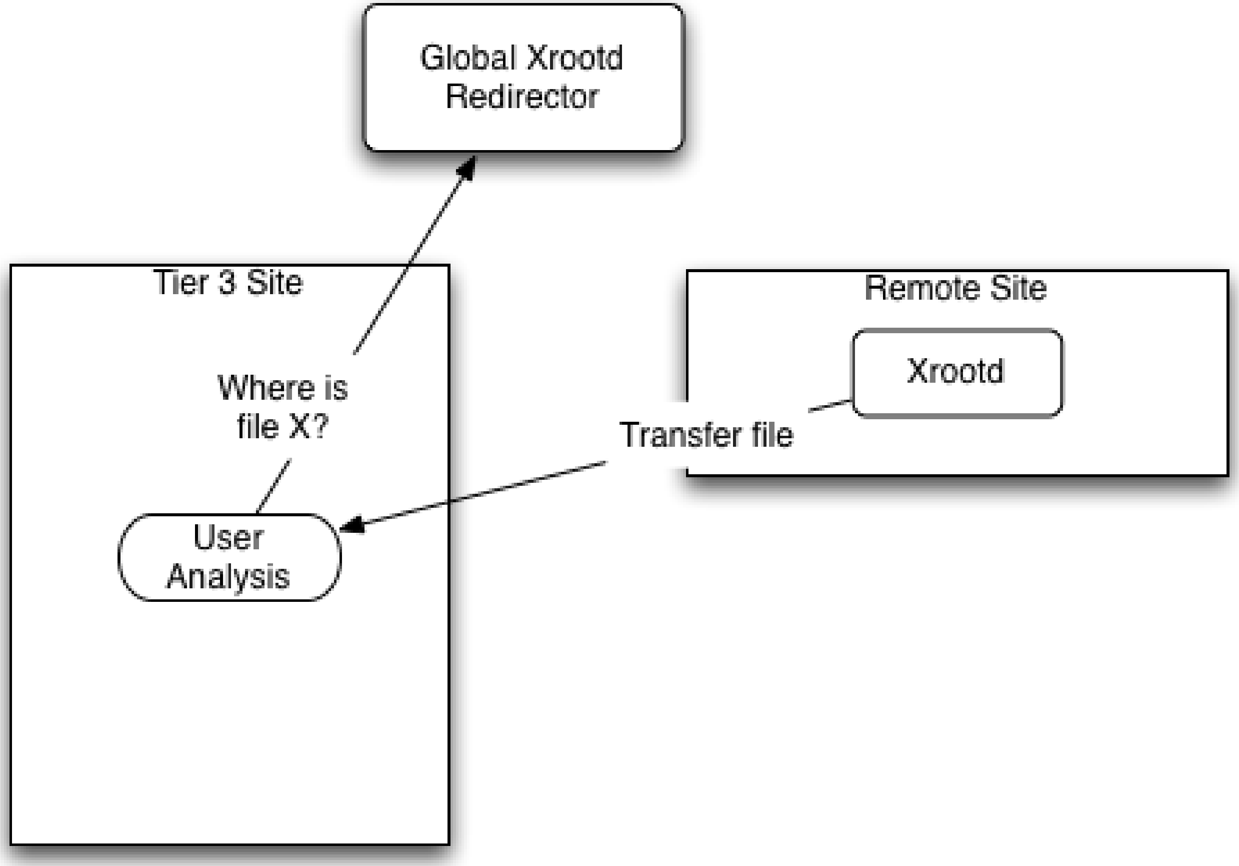
WLCG Demonstrator Project Deliverables

- Create a global-scale data access architecture architecture targeted for user analysis jobs.
  - Based upon xrootd, using both remote streaming and local caches.
  - Discover and understand the weak points of the xrootd software which would prevent the system from moving into production for CMS.
- Improve CMSSW I/O to decrease sensitivity to latency (reduce # of reads by a factor of 10).
- Target T3s and physicist workstations as initial consumers of this architecture.
- Understand effects of the new architecture on the network and storage system.

CMSSW Deliverables

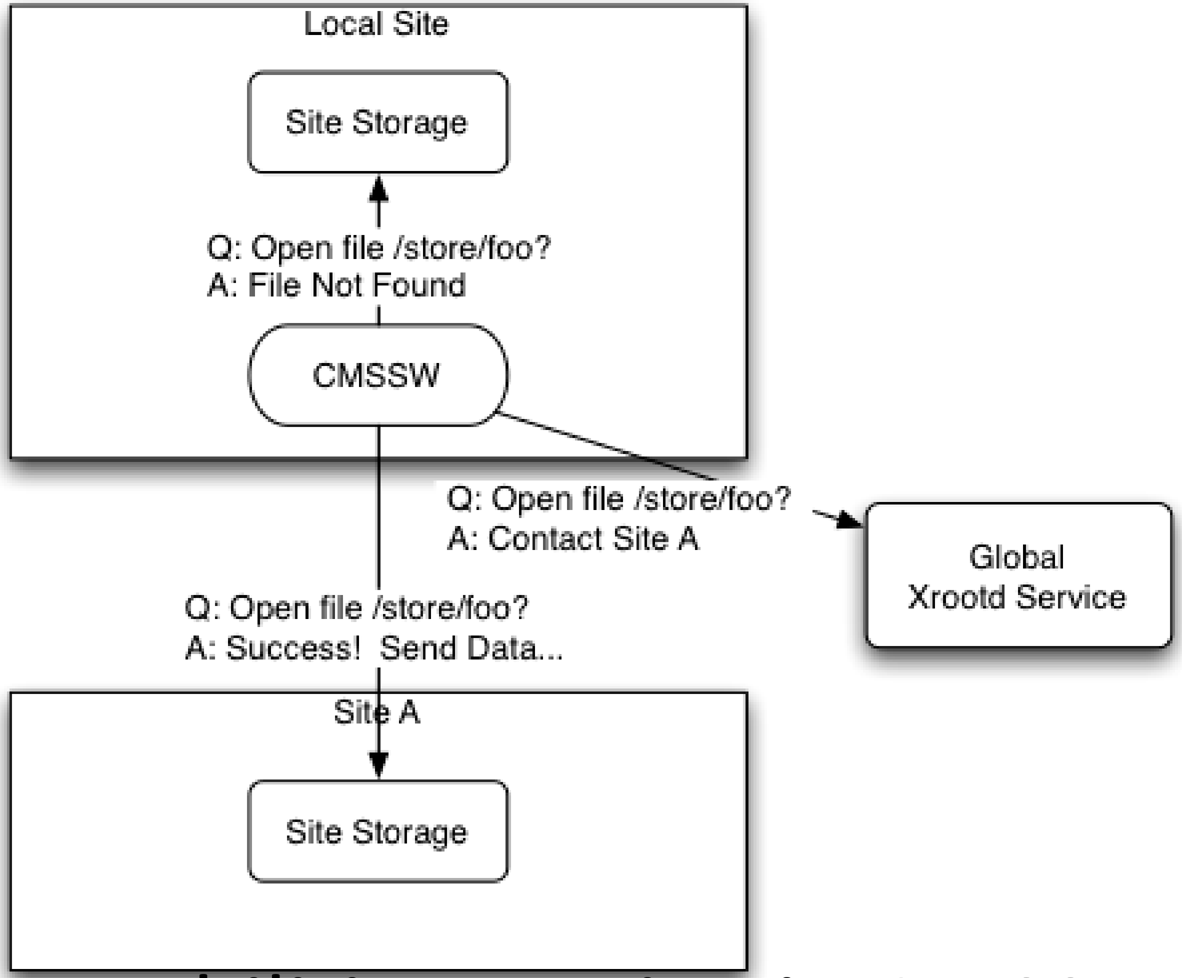
- Fault tolerant file opens
  - If a file is not found locally, try another method (such as reading from global xrootd).
- Resistant to latency
  - Reduce number of reads (good for the soul)
  - Grouping reads into 1 transaction with storage (“vector reads”)
  - Issuing reads before needing data (prefetching)

Sample T3 Configs - Diskless



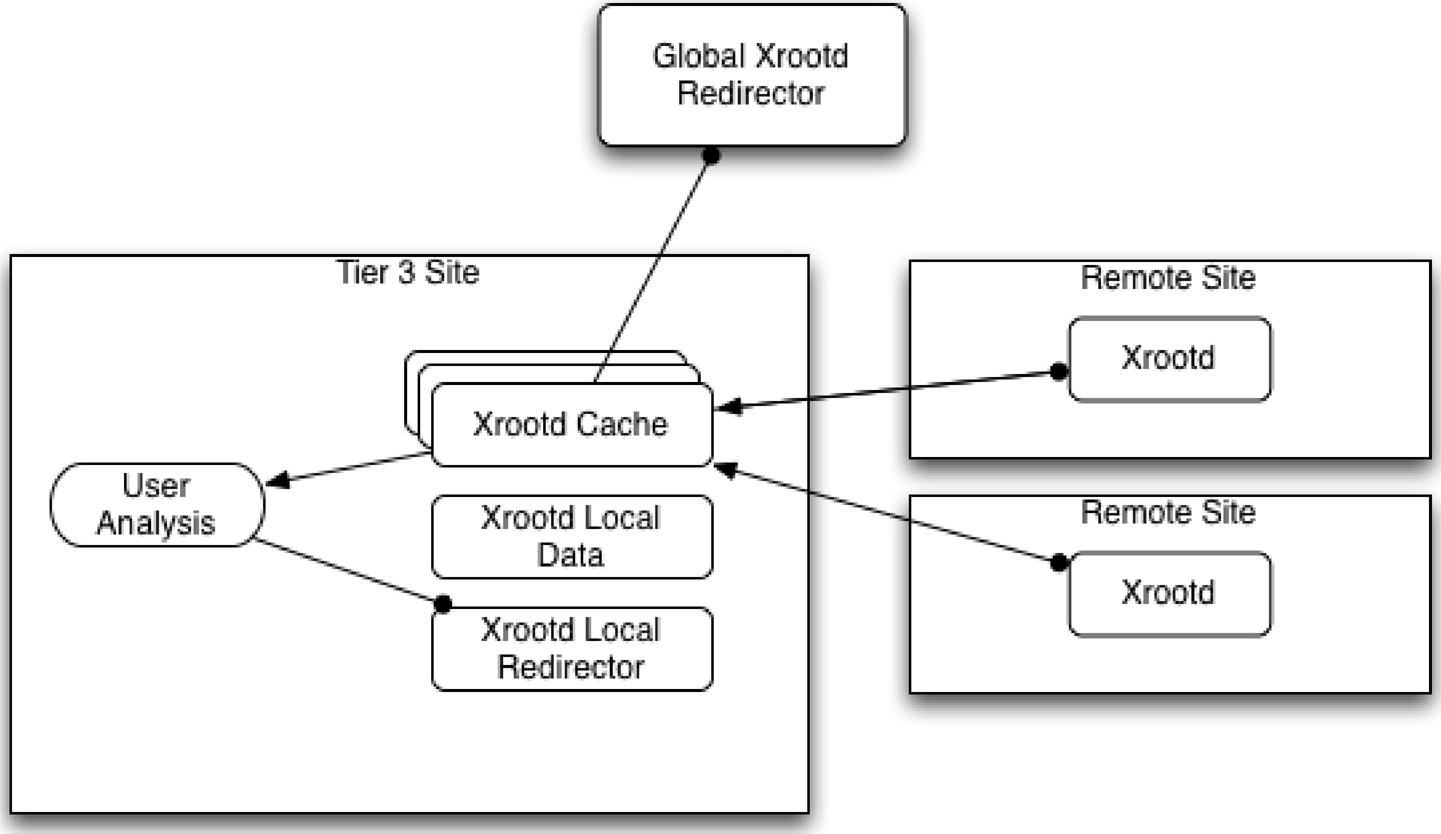
Deployed at T3_US_Omaha!

Sample T3 Config – Remote Fallback



Requires capabilities coming in CMSSW_3_9_0

Sample T3 Configs –Full Xrootd Deploy



Deploying at T3_US_UCR!

Sample T3 Configs

- Note that *you don’t need to switch your site to xrootd* except in the last picture.
  - However, the last case is the most tightly integrated *and* has the best caching.
  - So, you can benefit and participate with hardly any work on your side

Interacting with Users

- Any CMSSW user tool already works with this.
- Doesn’t require grid software (Mac friendly!)
  - But does require grid certificate!
- (Demo with Fireworks, network permitting)

Documentation

- https://twiki.cern.ch/twiki/bin/view/Main/CmsXrootdArchitecture