

GIVE ME MY DATA

Working with users so they can work
with each other

The Truth about Scalability

- You can't scale up if you can't scale down.
 - By “scale down”, I mean have the system reasonably usable by a novice user.
 - Put another way, if enough people can't figure out the system, you won't get enough buy-in from users.
- How well are we doing with this in grid storage?
 - I caught a very experienced grid user using “scp” to move files to FNAL and CERN because he couldn't figure out a way to collaborate with colleagues.

We do big well

- We can copy files at multiple Gbps.
- We can hold many terabytes of data.
- CMS applications can pound the SE as much as they want without causing failures.
 - During the recent October exercise, we only experienced a single transient failure in a read() call out of (probably) billions of accesses. We probably see about 1 internal error a day per node, but these are never encountered by the users due to error recovery in the client.
 - Saw more failures due to jobs being sent to the site to access a dataset that was deleted while they were in the queue.

Doing small well

- Here's a draft list of things that are required to make a system “easy” on users.
 1. Very reliable. Never encounter failures unless the site is in downtime. Never loses files.
 2. Can easily move files in and out.
 3. Can easily share files with others.
 4. Can move lots of files in / out (generally, this means a recursive copy).

Thinking Small

- Could always move files using scp if you have a login to the T2 cluster.
- Michael Thomas really started this off when he wanted to make files accessible to his end users and T3.
 - Mounted HDFS on a webserver and does a SSL-secured export out to local users.
- I started thinking about this when I found Ken Bloom trying to share a few ROOT files with colleagues at CERN and FNAL. The best advice I could give was to use “scp”.
 - Yuck!

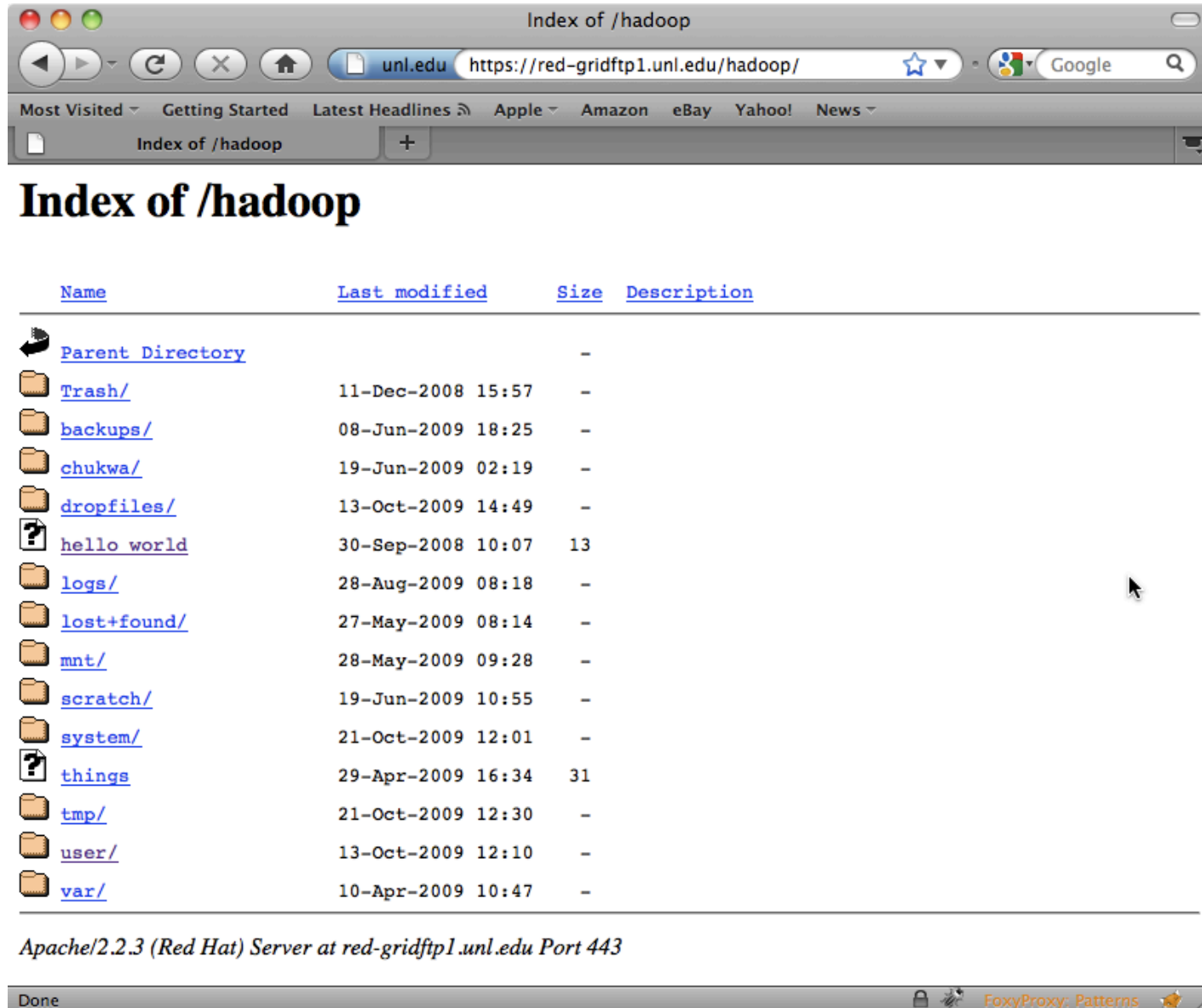
Thinking Small

- HTTP and scp access allows secure GSI/KRB5 access, but the latency for collaboration is too high.
 - What if I just wanted to look at one event? I have to copy over the whole 11GB file.
- Xrootd offers the ability to stream events in a reasonable and secure manner.
 - Decided we want to try this; it does limit our “solution” to HEP users.













HTTP

- Very simple solution – Apache server reading from a mounted HDFS instance.
- Controlled via grid-mapfile basically.
- Documentation:
 - <https://twiki.grid.iu.edu/bin/view/Storage/HadoopApache>
 - We'll probably turn this into an RPM in the future – it takes 1 script and 2 config files to set up; these will be mostly the same between sites.
- Simple, unscalable, but incredibly usable. Everyone knows how to use a browser.

Screenshot



The screenshot shows a web browser window titled "Index of /hadoop". The address bar displays "unl.edu https://red-gridftp1.unl.edu/hadoop/". The browser's navigation bar includes "Most Visited", "Getting Started", "Latest Headlines", "Apple", "Amazon", "eBay", "Yahoo!", and "News". The main content area shows the "Index of /hadoop" directory listing. The listing is a table with columns for "Name", "Last modified", "Size", and "Description". The entries include "Parent Directory", "Trash/", "backups/", "chukwa/", "dropfiles/", "hello world", "logs/", "lost+found/", "mnt/", "scratch/", "system/", "things", "tmp/", "user/", and "var/". The "hello world" and "things" entries have a size of 13 and 31 respectively. At the bottom of the page, it says "Apache/2.2.3 (Red Hat) Server at red-gridftp1.unl.edu Port 443". The browser's status bar at the bottom shows "Done" and "FoxyProxy: Patterns".

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 Trash/	11-Dec-2008 15:57	-	
 backups/	08-Jun-2009 18:25	-	
 chukwa/	19-Jun-2009 02:19	-	
 dropfiles/	13-Oct-2009 14:49	-	
 hello world	30-Sep-2008 10:07	13	
 logs/	28-Aug-2009 08:18	-	
 lost+found/	27-May-2009 08:14	-	
 mnt/	28-May-2009 09:28	-	
 scratch/	19-Jun-2009 10:55	-	
 system/	21-Oct-2009 12:01	-	
 things	29-Apr-2009 16:34	31	
 tmp/	21-Oct-2009 12:30	-	
 user/	13-Oct-2009 12:10	-	
 var/	10-Apr-2009 10:47	-	

Apache/2.2.3 (Red Hat) Server at red-gridftp1.unl.edu Port 443

Done FoxyProxy: Patterns

Xrootd/HDFS

- I implemented an OSS plugin for Xrootd which accesses HDFS through the C API.
 - As it is an OSS component, it means that the xrootd clustering “stuff” still works.
- This allows the xrootd server to export our HDFS instance.
- We only allow GSI-authenticated access; user mapping done with grid-mapfile.

Xrootd access

- ROOT is pretty much universally available to our CMS users.
- In every CMSSW release, the GSI plugins are available.
 - So, once the user gets his/her certificate, they probably have a client already installed.
- To open a file in ROOT:
 - `TFile::Open("root://xrootd.unl.edu//foo/bar");`
- Alternately, one can use `xrdcp`:
 - `xrdcp root://xrootd.unl.edu//foo/bar /tmp/test`
 - `Xrdcp` allows for multiple sources, multiple streams, etc.

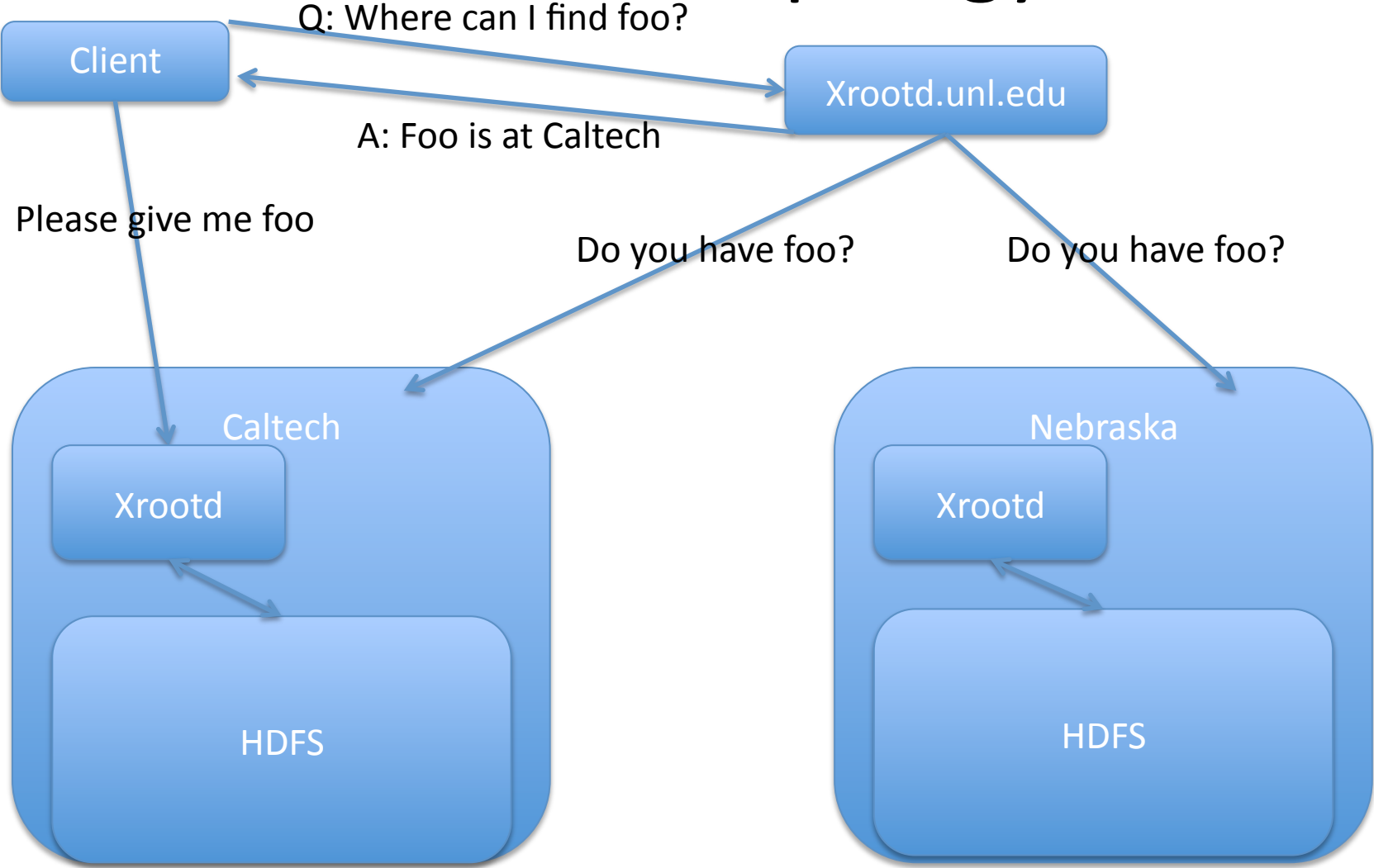
Xrdcp

- Xrdcp allows recursive copies.
 - Wonderful!
 - This is a top-request from end-users.
 - We can watch for abuses via Gratia.
 - This is much easier at a T2 as opposed to a T1.
The cost to stream indefinitely at 1Gbps is negligible at T2. At a T1, recursive operations can potentially touch many files, causing many stage operations. This is *very* expensive.

Xrootd deployment

- We have a redirector installed at `xrootd.unl.edu`
- So, you only need to remember a single address and you get redirected to the correct data server.
 - I.e., you never know your file is actually coming from `red-gridftp1.unl.edu:14235`
- This is not new; ALICE uses this on a worldwide basis.

Xrootd Topology



Xrootd on your cluster

- We have documentation here:
 - <https://twiki.grid.iu.edu/bin/view/Storage/HadoopXrootd>
- This provides a “yum install xrootd” experience and init scripts to start the server on boot.
 - The default configuration files have you utilize a manager at UNL.
 - So, even Caltech users access the Caltech cluster via:
 - `root://xrootd.unl.edu//some/Caltech/path`

Upcoming improvements

- You no longer need to remember the hostname to access the file. Why should you have to remember the PFN?
 - Working on a plugin that converts the LFN to PFN for CMS files. Any VO could do the same thing.
- Gratia Xrootd probe has been written, tested, and committed. Needs documentation.
- Packaging, packaging, packaging.

Xrootd bugs and drawbacks

- Xrootd is highly optimized for reading ROOT files and has optimizations for WAN access too.
 - However, these all fail *miserably* for CMS files. Total disaster; reading a single event from CERN takes longer than copying the file. The suggested workaround causes a segfault in ROOT. Will be at FNAL to brainstorm early November.
 - In fact, accessing events via HDFS directly in ROOT or Xrootd/HDFS is 30-50% slower than through FUSE.
 - Until then, simply turn caching off works fine, but you don't get any optimizations.

Xrootd Drawbacks

- There is additionally a bug in the cmsd that requires FUSE to be mounted on the Xrootd server.
 - Fix should be in hand in about a week.
- Xrootd GSI security does not integrate with VOMS/GUMS; a grid-mapfile must be used, which is a PITA.
 - After spending 30 minutes looking at code, this integration is not going to be done easily. There are overrides for callout authz functions, but those only get the DN, not the whole certificate.
 - PRIMA does not have a clean API, decent documentation, or a straightforward build process.
 - Ted, could we fix this easily? With some TLC, this could be converted to a nice component.

But isn't this unscalable?

- How do we plan on controlling rates?
 - We use the xrootd/http servers themselves as bottlenecks. The underlying system is far more scalable than the load one client can produce.
- HTTP and Xrootd have redirecting mechanisms that let us deploy new servers as necessary.
- The streaming rate with xrootd is enough to share a few events, but too slow to run an analysis against.
 - We DO NOT want to replace grid-based analysis; instead, we want to enable new features.
- Ultimately, each server that exports data is approximately equivalent to a worker node. Can only do so much damage.

Summary

- We have the following “new” access methods that generally don’t “scale up” well, but “scale down” to be usable for a single user or small collaborations:
 - Direct scp (yuck!)
 - HTTP with client cert / web browser
 - lcg-cp (yuck!)
 - ROOT / xrootd / xrdcp.
- I hope, with these options, sharing files can become routine and simple.