Any Data, Any Time, Anywhere

Improving Data Access for the LHC
using Xrootd

Introducing…

- "Any Data, Any Time, Anywhere" is a NSF-funded research project on enhancing data accessibility in HEP.
  - Award #s 1104549, 1104664, 1104447; a collaboration between UNL, UCSD, and UW.
  - We'd like to thank the NSF for their support of this project.

Improving Data Accessibility

- The WLCG has built a marvelous grid in support of the LHC experiment's computing programs.
  - In 2011 especially, it has demonstrated the its worthiness in terms of throughput.
  - Unfortunately, latency to physics data is fairly high.
    - I have seen folks submit a grid job in order to read out a single event. Not pretty.
    - I have seen students try to figure out how to use SRM directly. Also not pretty.

Ways to improve Data Access

- Over the next three years, AAA is going to focus on getting data to physicists. We'll be:
  - Making sure the physics applications are amenable to high-latency operation.
  - Building and operating reliable data federations.
  - Educating physicists on using the federation.
  - Expand the number of sites that can do work for CMS by reaching T3 and opportunistic sites.
- All while trying to keep an eye on the vertical stack – from the application software to the underlying storage.

Long-Lasting Contributions

- The AAA project isn't about building a new piece of software.
    - Instead, we'll be synthesizing several existing pieces of software,
    - Writing new modules and functionality,
    - And writing small, peripheral pieces of software where we identify holes.
        - But try to find a permanent home for them!
- Message: To be long-lasting on a temporary grant, we must focus on collaboration!

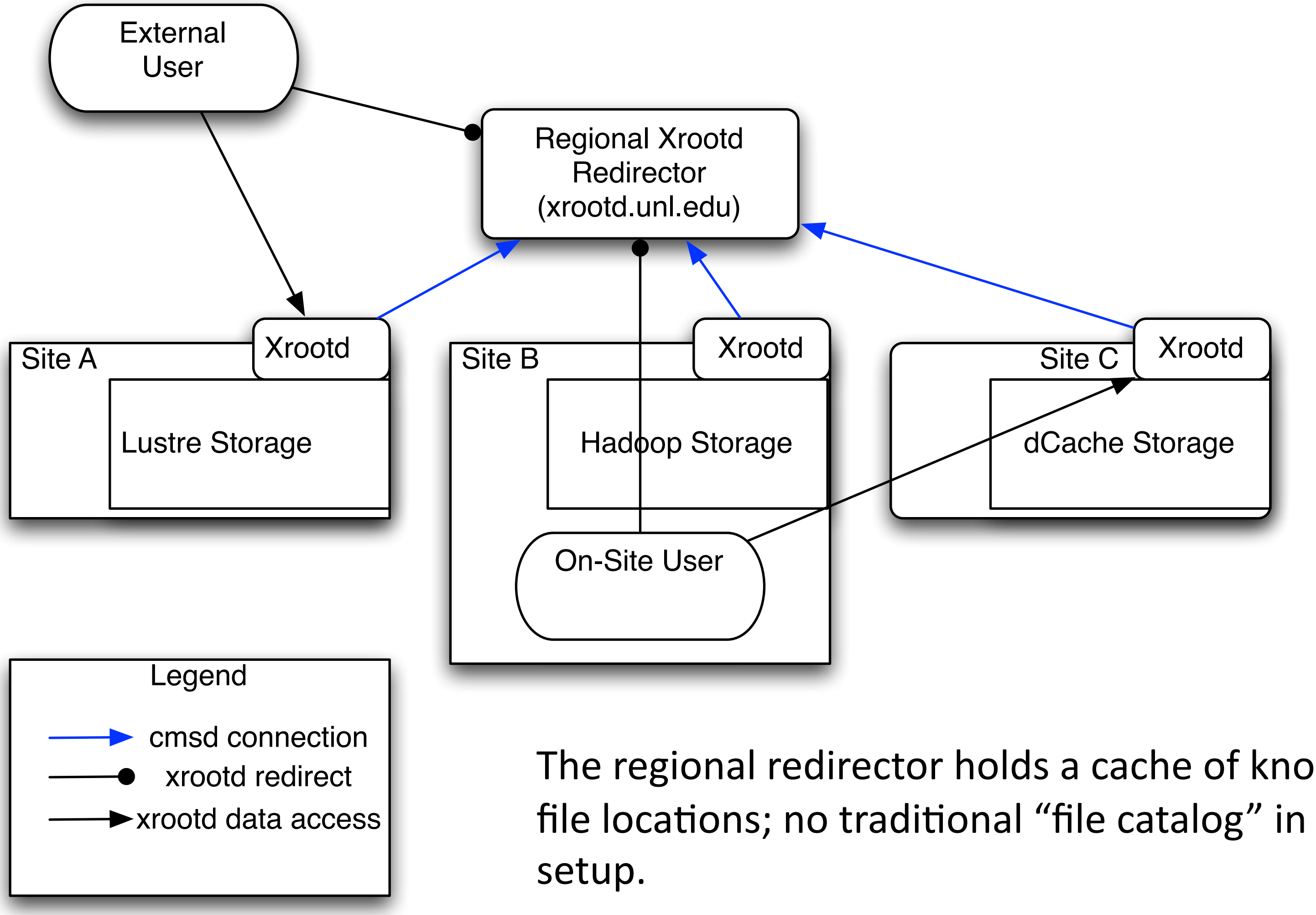
Aside: What is a Data Federation?

- One personal “goal” for the workshop is to figure out the precise definition of a data federation.
- Here’s my working one:
  - An infrastructure which provides the user the ability to access data uniformly across multiple resources, without knowledge of data location, using a global namespace.
  - Often done by having the user accessing a single, well-known endpoint.

Data Federations

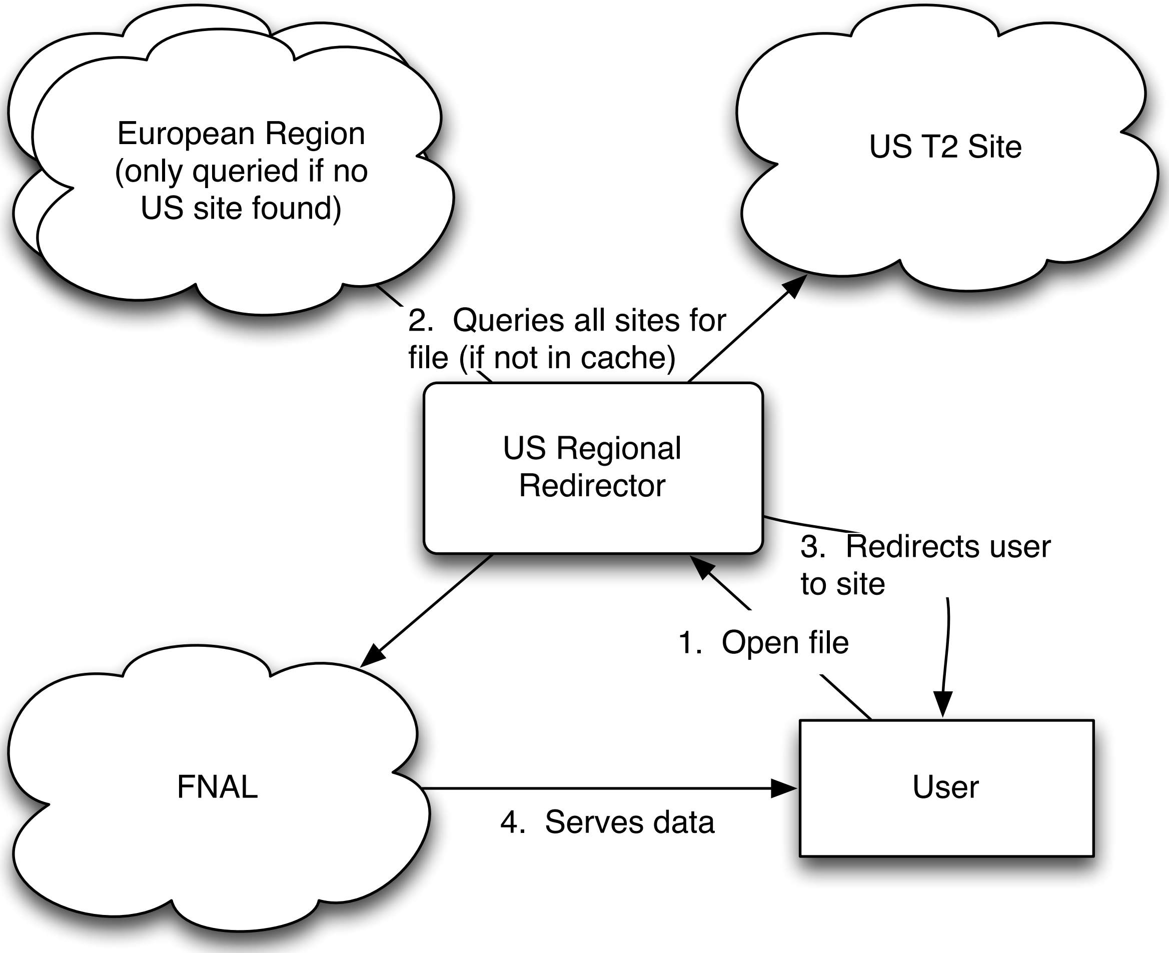
- We form a data federation by:
  - Adding an **Xrootd interface** to each site (using either native implementation or modular plugin). Sometimes the interface goes directly to data servers, sometimes it is a proxy.
  - Using cmsd, we cluster nearby sites together, so they are accessible via a **regional redirector**.
  - Using **peering**, we connect multiple regions together (work in progress).

In a crudely drawn picture…



The regional redirector holds a cache of known file locations; no traditional “file catalog” in this setup.

Peering Between Regions


European Region (only queried if no US site found)
US T2 Site
2. Queries all sites for file (if not in cache)
US Regional Redirector
3. Redirects user to site
1. Open file
FNAL
User
4. Serves data

CMS-isms

- Since we're working within CMS, it's probably worth noting a few items:
  - **Global namespace**: Xrootd exports the CMS global namespace; the mapping from global-to-local is a well-defined function at each CMS site.
    - Mapping is defined by a set of rules in an XML file.
  - **Authorization**: Authorization is based upon the same GSI/VOMS model as GridFTP or SRM.
    - Maps GSI/VOMS to a Unix user, then use underlying Xrootd user authorization scheme.

Uses of Data Federations

- **Fallback**: If a file is inaccessible or truncated locally, the user application can transparently fallback to the regional redirector.
- **Interactive Access**: End-user can access files directly for the event viewer or interactive session, regardless of where they are.
- **“Diskless” sites**: A small site could operate using the data federation as their exclusive source for experiment data.

Uses of Data Federations

- **Overflow**: If a job has been in the queue for over a threshold, allow it to go to a site which does not have the required input data.
  - Can be done for jobs in a global queue.
  - Or jobs in the local batch system.
- We have recently enabled the use of overflow for CMS jobs, and have gained lots of operational experience (both positive and negative) in dealing with the federation.

Uses of Data Federations

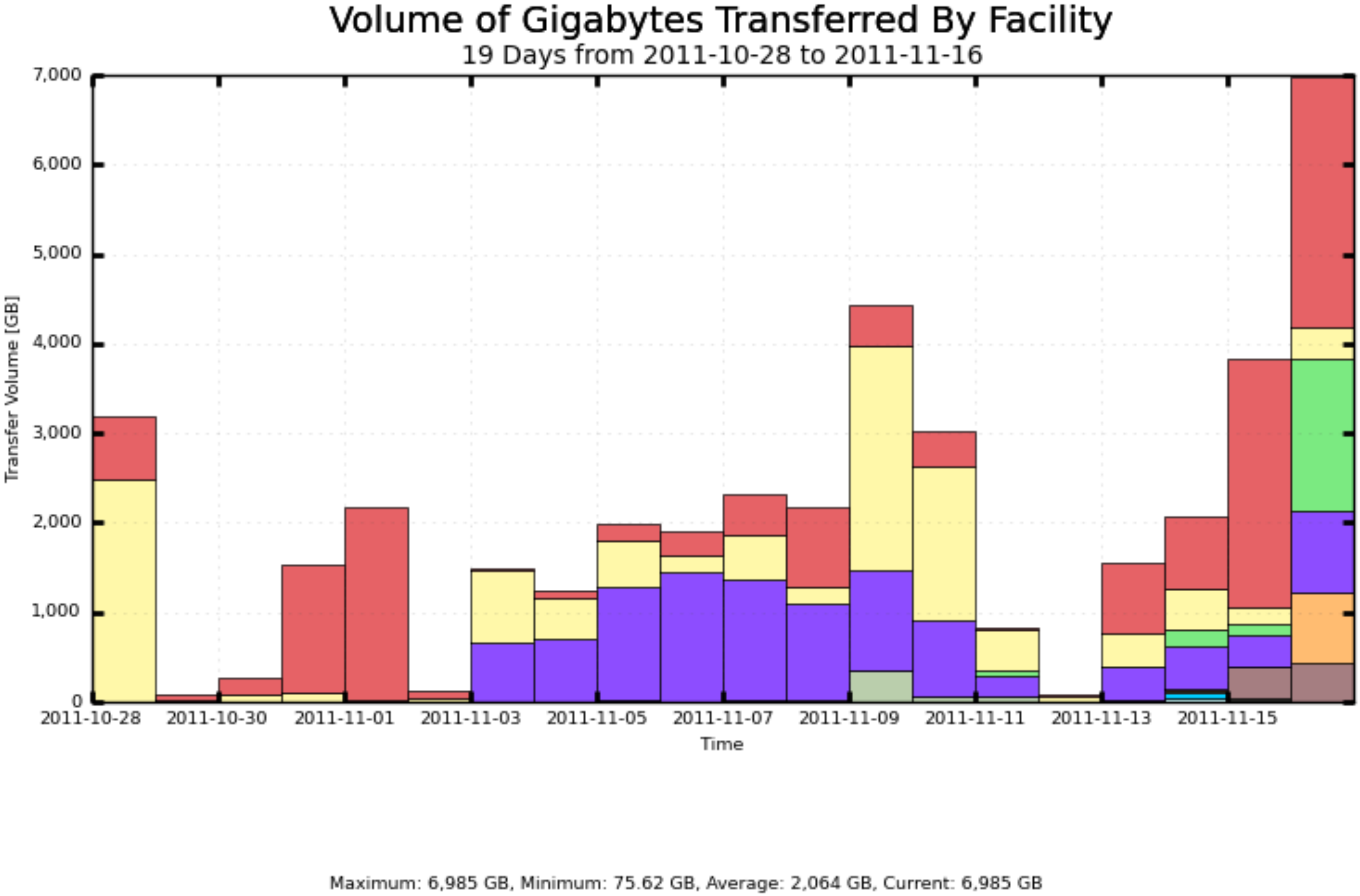
- We spend a lot of money on reliability:
  - High manual cost to recover data or lost time if user hits a problem file, so we have 2 copies of everything.
  - If there is reliable WAN fallback, local reliability is less important.
  - Hence, we could replicate a subset of our data only once, save money on disk, and deploy more CPUs.

Application Support

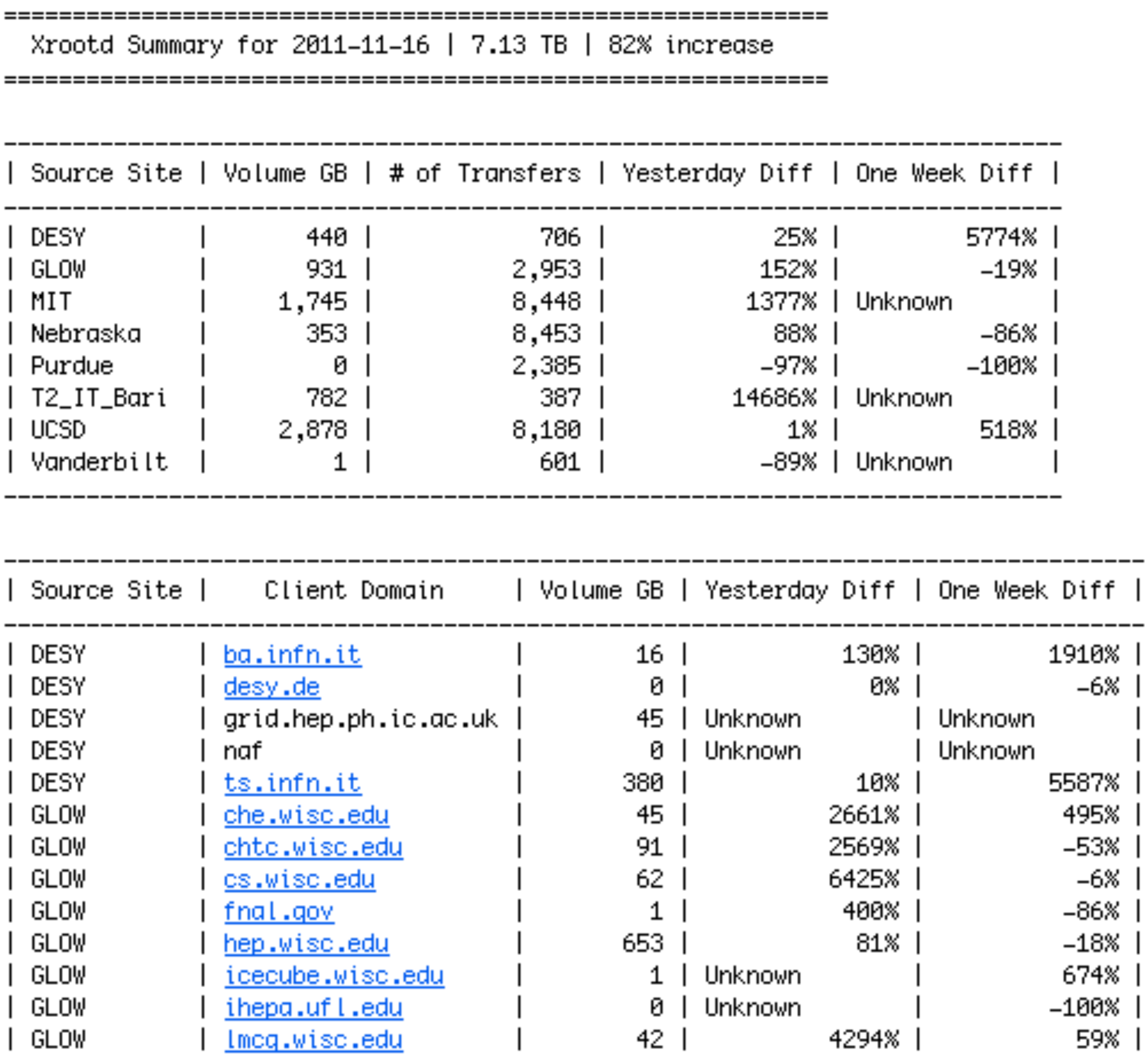
- Without proper application support, the whole exercise is fairly useless.
- It is a long, hard year's worth of work to decrease your application's sensitivity to latency. Prerequisite to remote I/O.
- We are working on building a standard candle to detect regressions in CMSSW I/O.
- Also working to improve exit codes and error messages for the Xrootd fallback case.

Monitoring

- There will be a dedicated presentation on Tuesday.
- Highlights:
  - We can monitor application CPU efficiency.
  - We can correlate a user’s activity across sites.
  - We have basic site usage statistics.
- As we progress, we still find “holes” in what we watch and what Xrootd provides. This system will be growing for awhile.
  - Particularly, I’m interested in the “end-to-end”: correlate application monitoring with Xrootd monitoring.
  - Also, in identifying problematic servers/sites ASAP.



Example daily usage graph



```
============================================================
  Xrootd Summary for 2011-11-16 | 7.13 TB | 82% increase
============================================================

-----------------------------------------------------------------------------
| Source Site | Volume GB | # of Transfers | Yesterday Diff | One Week Diff |
-----------------------------------------------------------------------------
| DESY        |       440 |            706 |            25% |         5774% |
| GLOW        |       931 |          2,953 |           152% |          -19% |
| MIT         |     1,745 |          8,448 |          1377% | Unknown       |
| Nebraska    |       353 |          8,453 |            88% |          -86% |
| Purdue      |         0 |          2,385 |           -97% |         -100% |
| T2_IT_Bari  |       782 |            387 |         14686% | Unknown       |
| UCSD        |     2,878 |          8,180 |             1% |          518% |
| Vanderbilt  |         1 |            601 |           -89% | Unknown       |
-----------------------------------------------------------------------------

-----------------------------------------------------------------------------------
| Source Site |     Client Domain      | Volume GB | Yesterday Diff | One Week Diff |
-----------------------------------------------------------------------------------
| DESY        | ba.infn.it           |        16 |           130% |         1910% |
| DESY        | desy.de              |         0 |             0% |           -6% |
| DESY        | grid.hep.ph.ic.ac.uk |        45 | Unknown        | Unknown       |
| DESY        | naf                  |         0 | Unknown        | Unknown       |
| DESY        | ts.infn.it           |       380 |            10% |         5587% |
| GLOW        | che.wisc.edu         |        45 |          2661% |          495% |
| GLOW        | chtc.wisc.edu        |        91 |          2569% |          -53% |
| GLOW        | cs.wisc.edu          |        62 |          6425% |           -6% |
| GLOW        | fnal.gov             |         1 |           400% |          -86% |
| GLOW        | hep.wisc.edu         |       653 |            81% |          -18% |
| GLOW        | icecube.wisc.edu     |         1 | Unknown        |          674% |
| GLOW        | ihepa.ufl.edu        |         0 | Unknown        |         -100% |
| GLOW        | lmcg.wisc.edu        |        42 |          4294% |           59% |
```

Example daily usage email

Status

- Two regions, EU and US.
- EU region has 1 T1 and 4 T2s participating (small percentage of possible sites).
- US region has T1 and 8 T2s participating (all sites).
  - Sees 1-5TB, 10-20 users / day.
  - Overflow in place for all users, typically up to 10% of total glidein usage. Drop in CPU efficiency varies, but average is between 5 and 10%.
- Ready and willing to include more sites and more users.

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

Things Cooking

- The next few slides are directions we plan on investigating in the next year.
  - By staying awake for the next five minutes, you are making an irrevocable agreement to not hold us to any forward-looking statements!

Data Caching

- Data caching is a tricky topic.
  - Advantages of caching are well known.
  - But don't forget the disadvantages of caching (cache poisoning, thrashing) are also well known.
- Probable strategy:
  - Develop capabilities for a known, controlled use case (i.e., restoring lost data at a T2). Develop corresponding monitoring.
  - Extend caching to under-served use cases (T3s).
  - Evaluate performance, and decide whether to extend.

Data Caching

- I feel we haven't really scratched the surface until we get a thorough cache thrashing.
  - Similar to how the first "real experience" for cross-site access is when you have to debug it!
  - Recall the cache is really defined by the eviction policy!
- A likely approach will be to:
  - Implement a "caching proxy" that plugs into the OSS.
  - Implement a modular eviction policy (and policy for adding files to the cache) so we can do experiments.

Smarter Redirection

- A primary driver for region-based federations is the cost of mis-redirection.
  - A US client really doesn’t want to go to the EU unless it is necessary.
  - Regions bound the network distance between sites; too-high latency can harm CPU efficiency.
  - But even within regions, not all redirections have the same cost (consider a client on Caltech campus; they would prefer to go to the on-campus cluster than Wisconsin).
- We hope to study this over the next year to come up with some smarter approaches; metrics is “can’t be worse than current round-robin”.

Data-Aware Scheduling

- Our current systems (Condor-based) use crude data scheduling: prior to submit, the client looks up the sites with the necessary data, and hardcodes this as the list of possible sites.
  - No regard to changing conditions after submit (new copies, deleted copies).
  - No regard to site storage health.
- This is another topic for research – feeding dynamic location and site data into Condor.

Opportunistic and Marginal Resources

- Here, “marginal” refers to the amount of time the resource is willing to deal with HEP, not the size!
- Goal is to “fly-in” a complete CMS environment:
  - Use Xrootd to provide data.
  - HTTP caching for conditions.
  - CVMFS for software.
  - Ideally, only pre-requisite is the ability to launch a user process.
- As CVMFS requires site support, we are looking into making it userland-only using Parrot.
  - http://nd.edu/~ccl/software/parrot/

Concerns

- Federations make problems harder to debug!
  - Now, all sites are coupled together.
  - Must make sure the system reliability isn't the minimum of all involved sites!
- Must still learn to quantify the system in terms of capacity and limits.
- Must make sure the system is future-proof, and not a function of quirks in today's data analysis styles.

Take-Home Message

- The AAA project is about data access. We're:
  - Using Xrootd to build a data federation for low-latency access.
  - Combining with Condor to open up access to more CPUs.
  - Working to make everything "production quality".
  - And have 3 years to do it!