Any Data, Anytime, Anywhere: Project Update

1. AAA Overview
2. Status of various sub-projects
3. Summary & Conclusions

Matevž Tadel, UCSD for the AAA team

AAA Project Overview

- 3-year, 3 FTE funding

  Started in Sep 2011
- Joint project of UCSD, UNL & Wisconsin
- ~3 people / institution → 10 active members

- Cover all aspects of remote data access in a VO with large-scale, distributed storage and computing infrastructure.

AAA Objective

- Maintain an infrastructure which provides access to distributed data anytime from anywhere, including interactive access.
- Integrate technologies that will allow more diverse set of resources to be used by VOs.
- Improve robustness and resiliency against failure.

AAA Modus Operandi

- AAA is a meta-project:
  - Take proven technologies, integrate them, and contribute back – Adopt, Adapt & Improve
- CMS is our main target
  - But most products can be used by any VO
  - There is, of course, some work that needs to be done on software stack of individual VO
  - Operational experience is crucial

    → we'll do most of the mistakes so you don't have to.
- Main focus on US + cooperation with CERN

AAA External Components

Projects that we are mashing up together:

- **XRootD**: serve and access data
- **CondorG**: flexible job scheduling
- **GlideinWMS**: dynamic job placement
- **CernVM FS**: software distribution

  Coupled with Parrot, IO syscall interceptor

Xrootd Infrastructure

Interactive Access

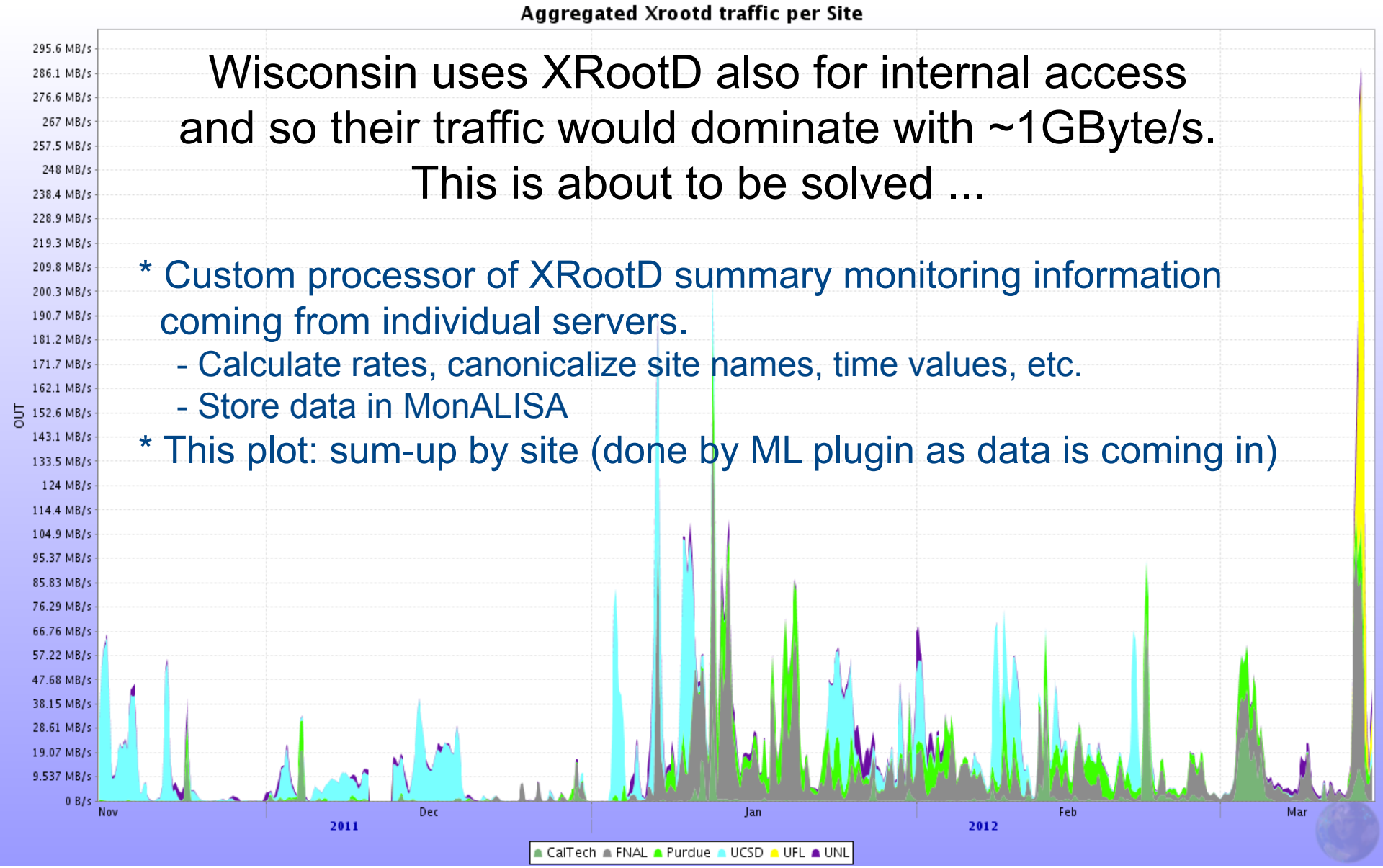
- We had XRootD servers running on some sites already before the project start-date.
  - Top-level redirector running at UNL.
  - Collectors of monitoring packets at UCSD.
- Since the beginning of this year, data at the following sites can be opened from the command line or event-display from anywhere at any time:
  - FNAL (anything that is on disk).
  - All of the US CMS T2s.
- Additional redirector in Italy with some German, Italian and UK T2s.

Status of Fallback

- At file-open failure, CMSSW can fallback to remote data-access – decision based on site configuration.
- All US T2s (+ some US T3s) have deployed XRootD based fallback to redirector at UNL.
- Monitoring in place, including daily email summaries.
  - We track fallback-induced failure rates daily.
  - Try to look at the end-to-end and feed back fixes to either XRootD or CMSSW. Or improve monitoring.

Had some problems operationally on occasion but overall, the system is providing value to CMS.

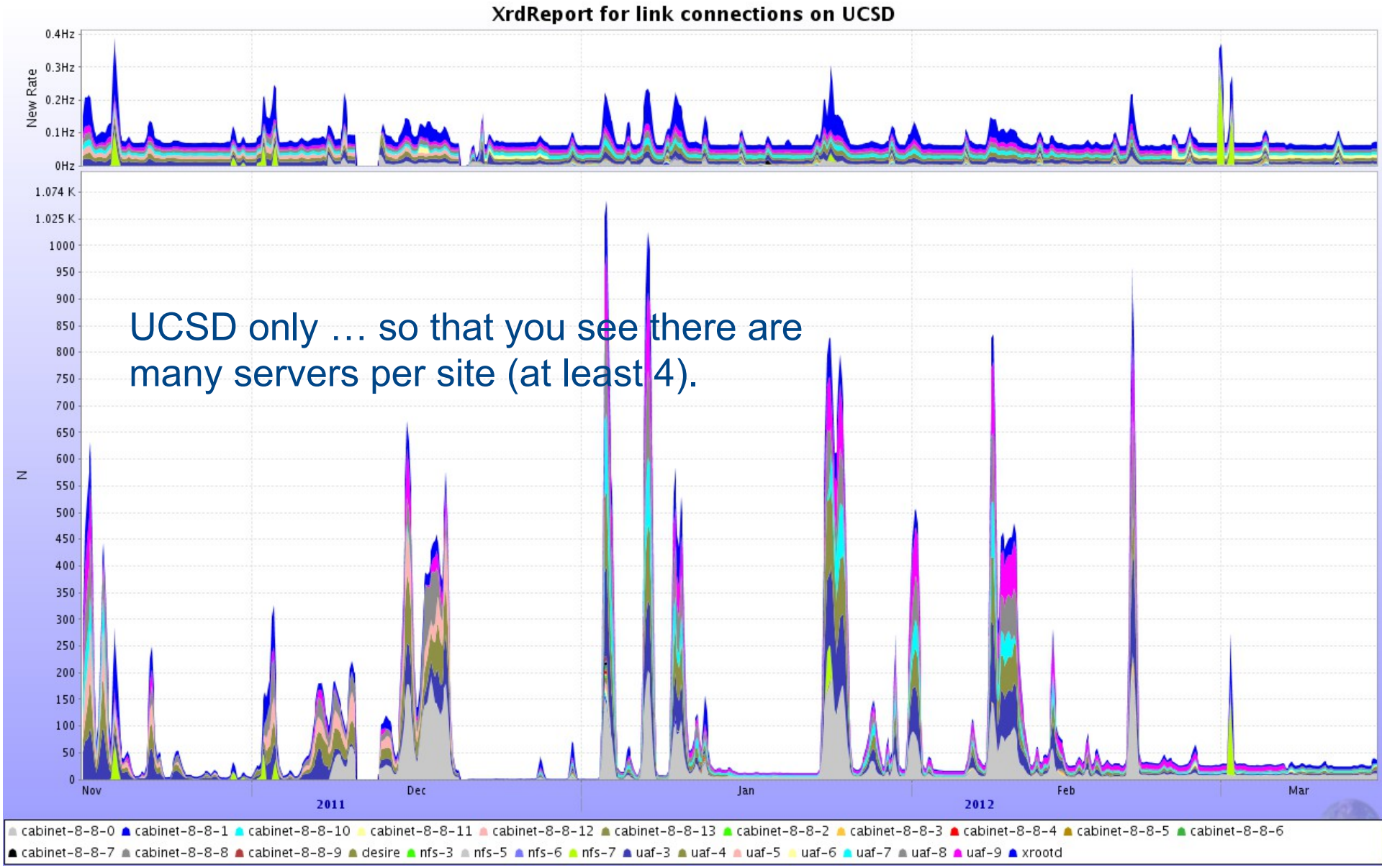
Outgoing Traffic per Site



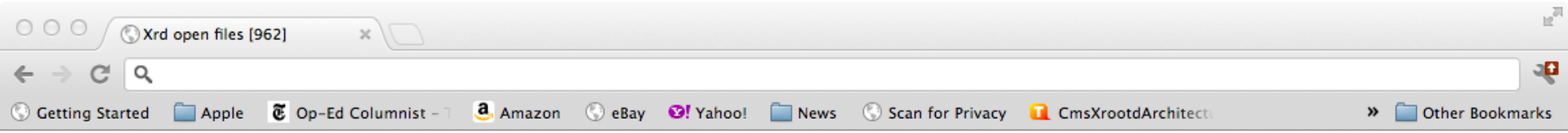
Wisconsin uses XRootD also for internal access and so their traffic would dominate with ~1GByte/s. This is about to be solved ...

* Custom processor of XRootD summary monitoring information coming from individual servers.
  - Calculate rates, canonicalize site names, time values, etc.
  - Store data in MonALISA
* This plot: sum-up by site (done by ML plugin as data is coming in)

Number of connections on UCSD servers



Real-time file access monitor



File	OpenAgo	ServerDomain	ClientDomain	User	Read [MB]	UpdateAgo
/store/test/xrootd/T1_US_FNAL/store/data/Run2011B/ForwardTriggers/AOD/PromptReco-v1/000/179/307/78D47D5F-1EFD-E011-A0F2-BCAEC53296F5.root	27:21:59	fnal.gov	unl.edu	Brian Bockelman	0.000	27:21:59
/store/test/xrootd/T1_US_FNAL/store/mc/JobRobot/RelValProdTTbar/GEN-SIM-DIGI-RECO/MC_3XY_V24_JobRobot-v1/0001/56E18353-982C-DF11-B217-00304879FA4A.root	07:26:17	fnal.gov	unl.edu	Brian Bockelman	0.000	07:26:17
/store/data/Run2011B/MuOnia/AOD/19Nov2011-v1/0000/4ACEB592-041C-E111-AF33-003048FF9AA6.root	05:14:25	fnal.gov	rcac.purdue.edu	Caterina Mascolo	292.456	01:37:10
/store/data/Run2011B/MuOnia/AOD/19Nov2011-v1/0001/5CE6311F-091C-E111-BC20-00248C0BE018.root	05:14:00	fnal.gov	rcac.purdue.edu	Caterina Mascolo	274.046	00:01:28
/store/data/Run2011B/MuOnia/AOD/19Nov2011-v1/0001/9E50C964-071C-E111-A04F-003048679008.root	05:13:50	fnal.gov	rcac.purdue.edu	Caterina Mascolo	187.190	00:04:32
/store/data/Run2011B/MuOnia/AOD/19Nov2011-v1/0001/1E179270-0A1C-E111-BE46-00248C0BE016.root	04:56:17	fnal.gov	rcac.purdue.edu	Caterina Mascolo	221.642	00:01:45
/store/mc/Fall11/W1Jet_TuneZ2_7TeV-madgraph-tauola/AODSIM/PU_S6_START42_V14B-v1/0002/3A079775-51F6-E011-B2C4-003048678C9A.root	04:29:08	fnal.gov	hep.wisc.edu	Jeremy Andrea	2008.832	00:15:36
/store/data/Run2011B/MuOnia/AOD/19Nov2011-v1/0000/10C4473B-041C-E111-A948-003048678BE8.root	03:44:45	fnal.gov	rcac.purdue.edu	Caterina Mascolo	210.768	00:00:01
/store/mc/Fall11/DYJetsToLL_TuneZ2_M-50_7TeV-madgraph-				Philip		

* Based on detailed XRootD monitoring stream.
* We did quite some work to improve & extend XRootD reporting and to develop a *collector / analyser* of monitoring packets.

Missing Pieces

- We have monitoring for both jobs and data access, we are lacking easy way to correlate the two.
- At this point things get VO specific. What we did / are doing for CMSSW:
    - Inject JobID into XRootD monitoring stream;
    - Use custom XrdFileAdapter (not TFile)
        - Gather statistics, improve vector read usage
    - Include relevant information in end-of-job-report:
        - Redirection history with failures
        - Branch access counters and statistics

Towards making more diverse resources useful to CMS

In a nutshell…

- Operate data / software / job distribution as now.
- If a job waits for a slot at its target site(s) for more than N hours, then try to find resources at another T2 → *remote read access*.
- If all T2s are full then find resources elsewhere on grid → *opportunistic use*.
- If all grid resources are occupied, and "we need more" → *buy them at commercial vendor*.
  - Where "we need more" is a policy decision made by someone else.

Technical Issues

- Procuring resources outside target site(s) when backlog exceeds threshold.
- Transparently switching to WAN read access.
- Dynamic access to VO software releases without the need for fixed installations.

AAA is delivering these capabilities (and more) within the next 3 years

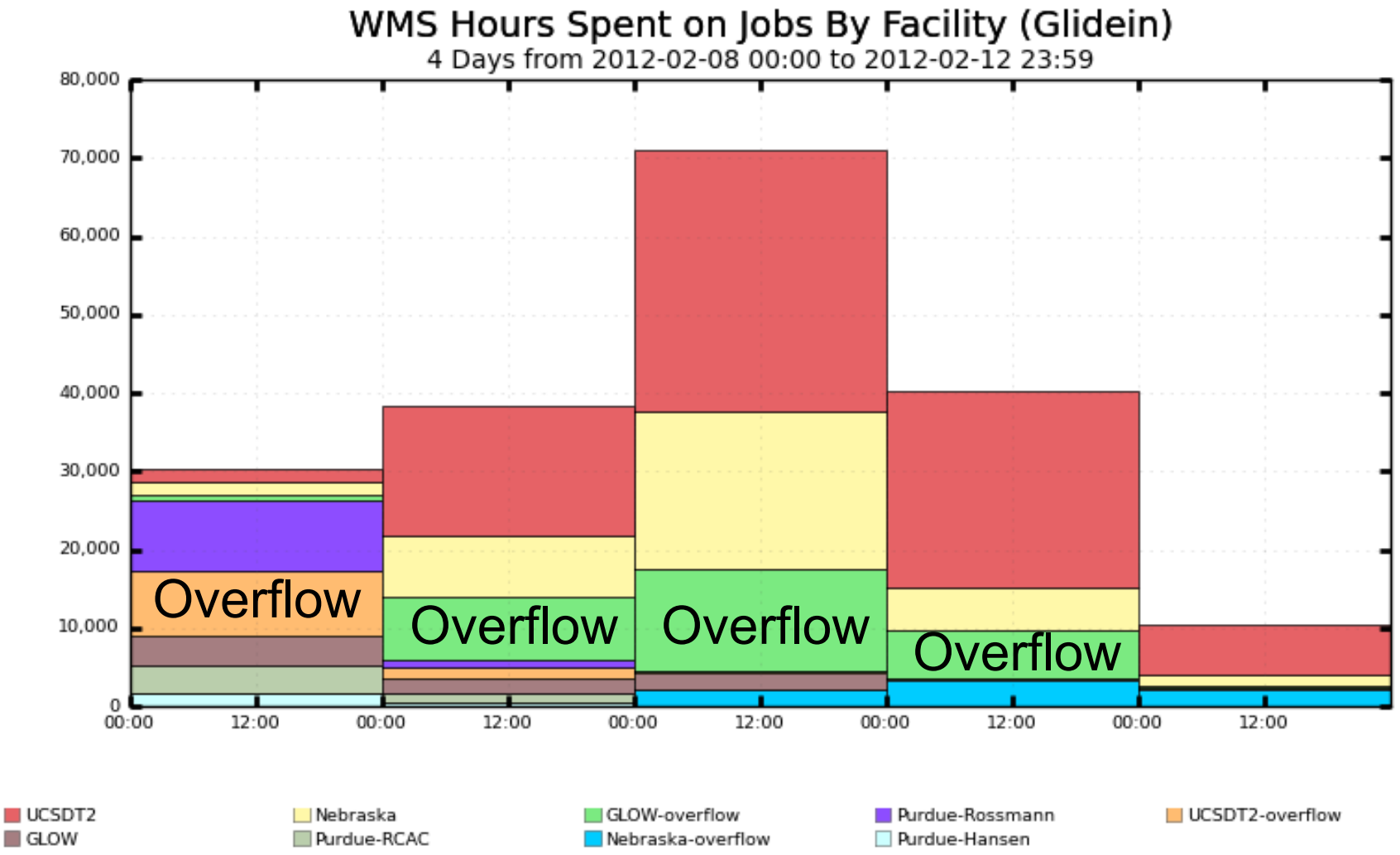
Status of Overflow

- If a job queues up in *glideinWMS* for more than 6 hours then glideins are sent to one of UW, UNL, UCSD or Purdue for spare resources.
- Job now runs at a site that does not have the data → fallback kicks in and data gets read via XRootD.
- All XRootD-enabled production sites are used in a round robin fashion.

  Optimizing data-source selection is also on our to-do list.
- Efficiency penalty at most 10%.

  Should be much less with latest vector-read optimizations.

Example: Feb. 8-12 2012



Dynamic Access to VO software

- This is still at the prototyping stage
- Concept:
  - Use CVMFS via Parrot at grid sites.
  - All the niceties of CVMFS, but done *only* in user space. Much easier to run at non-WLCG sites.
- Have run a few CMSSW tasks using this; scaling it up toward “production-ready”.
  - Currently, the “penalty” is below 5%.

Resiliency and Robustness

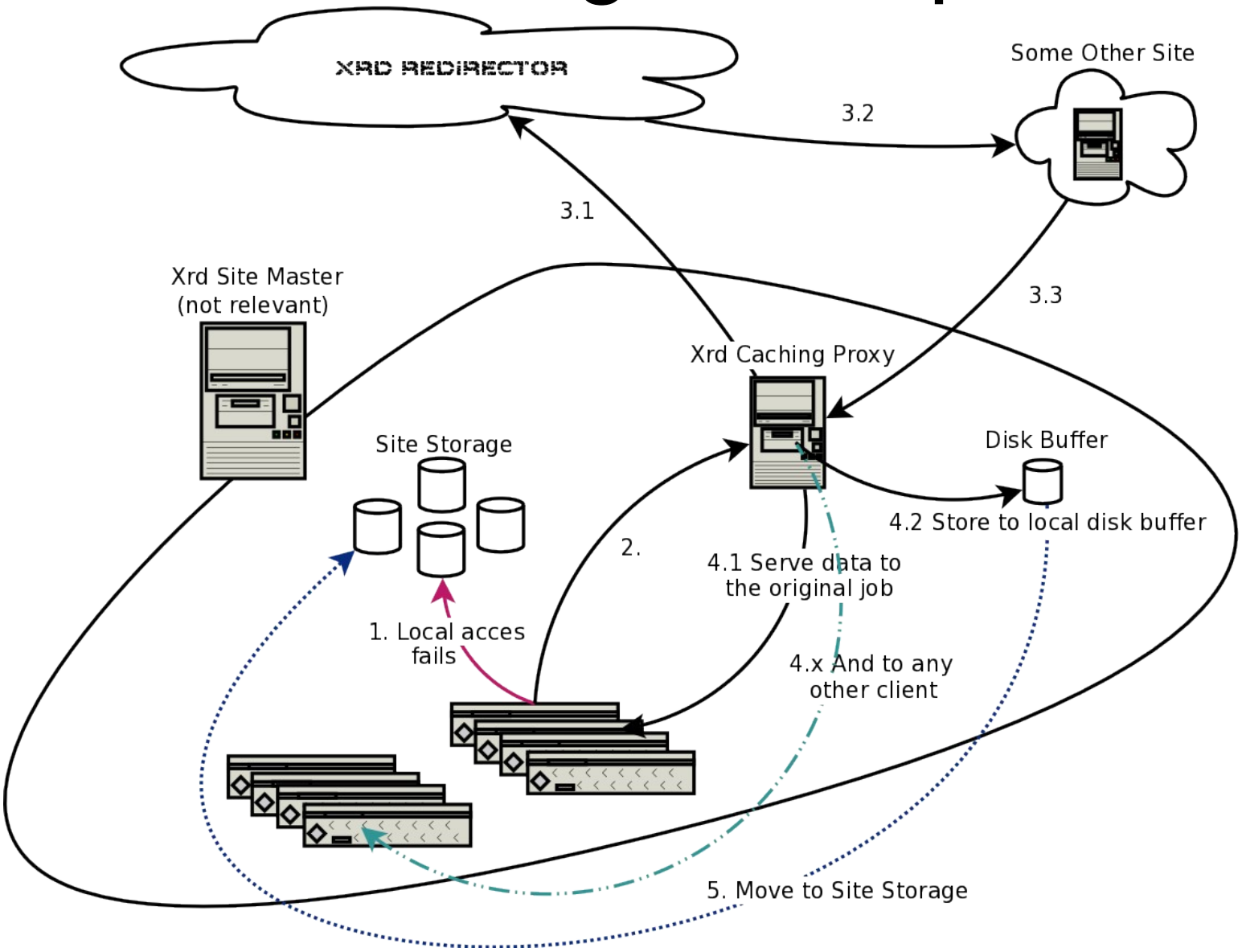
Robustness and Resiliency

- First layer of resiliency is the use of fallback mechanism.
  - Already deployed at most US sites.
- Next, we want to explore caching.
  - Initial use case: automatically restore files lost due to disk failures – for the subset of data files where this is possible.

Robustness Costs

- HDFS-based sites use cheap disks in worker nodes, but have two replicas of every file.
- This saves a huge amount of costs – no file servers and saves a lot of admin time.
- It's a life-saver for unique files, but rather costly for files that are replicated elsewhere in the US.
  - We could add an additional 3PB (500TB x 6 sites) of T2 disk space if we didn't replicate!
  - Or we could buy more CPUs and less disk in the next purchasing round.

Caching Concept



Caching Usage

1. Automatic Storage Healing

   Files are pulled from elsewhere without error and replaced in local storage ... without anybody knowing.

   Avoid local file replication for data that also exists off-site.

2. “Targeted overflow”

   Large demand for a specific data-set. e.g. SUSY in Dec 2011

   Use local cache as short-term (~weeks) storage.

   Spy on waiting jobs to make reasonable decisions; react faster than we currently can for pre-placement.

3. T3s, Opportunistic resources, clouds

   WAN traffic slow or expensive.

Caching Implementation

- In touch with XRootD team (in fact, were coerced to join)
  We know how to do it.
- Now studying access patterns to optimize cache management and read-ahead
  - Have access patterns from detailed monitoring.
  - We also contribute to CMSSW IO layer
    - many optimizations already in place

But one needs to keep standing on one's toes as both file structure and our access improve

Summary & Conclusions

- AAA project is providing technical solutions to fully integrate diverse resource usage into data intensive analysis workflows.
- Initial target is to better use CMS owned resources.
- Longer term target is opportunistic and commercial resources.
- Regain disk space by reducing replication.