Any Data, Any Time, Anywhere

Increasing data accessibility for HEP
Brian Bockelman, for the AAA team
NSF PHY-1104664
The Setup

• HEP has an incredibly large data management problem.

• To make it plausible to handle the sheer size, we must make several processing passes over the data.

• We must keep several replicas of the data to be able feed enough CPUs.

• This begets complex data management systems for tracking data provenance, location, and replication.
But Wait, There’s More!

• While we always talk about physics data due to its volume, there are other data types:

• *Conditions data*: Information (e.g., alignments) about the detector at the time the data was taken. Changes as our understanding of the detector improves.

• *Software*: The software used is complex and large, on the order of millions of lines of code. Multiple releases are available at any given time.
The Problem Statement

• I claim the HEP community has successfully demonstrated the ability to manage its data and processing challenges at its computing sites.

• We need to improve the ability to utilize outside resources - whether a physicist’s laptop or an opportunistic grid site.

The Challenge: Make any internet-connected computer useful to the CMS experiment!
Introducing AAA

- The “Any Data, Any Time, Anywhere” project is an NSF-funded initiative to increase data accessibility for the HEP community.

- We are working to build an data access infrastructure, based on existing components, that makes it possible to run CMS jobs anywhere.
Data Access
Physics Data

- The most challenging aspect is access to physics data.
- CMS manages about 50PB of disk space at its data centers.
- Remote access is mediated through a web-services protocol called “SRM” and transfers via GridFTP.
  - GridFTP typically used to transfer the complete file.
- Access to the data via CMS software is done by specialized protocols or POSIX access.
  - CMS jobs tend to read <1/3 of the file at a rate of 128KB/s to 2MB/s.
The Woe of One Event

• If I want to read a single event, how do I get the data? Options:

• **Run a grid job on that event**: best case, 15 minutes (create the job, submit it, have it run, fetch results). Worst case, hours of queue time.

**Download the file**: First you have to find it and setup the tools. If you’re lucky, only 5 minutes to download.
Direct Remote Access is Key

• We turned to the Xrootd project to provide remote, direct access to data stored at sites.
• Mature project for remote-I/O.
• Client almost always integrated into ROOT.
• Has the security mechanisms WLCG needs.
• Time to open event interactively is limited to network latency.
Key Features

- Client is robust against a multitude of networking failures and misconfigured endpoints.
- Remote sites are not run by us, so we can’t easily control their configurations or software versions.
- Protocol can batch many requests into one network round-trip (routinely hundreds of read requests are batched into one packet).
- Prefetching and request-batching ("vector reads") are essential to reducing the effect of latency; the whole system depends on this!
Introducing Federations

• Remote access gives us data for one site. We need a federation to access all sites.

• Definition of a **federated storage system**:

  • A collection of disparate storage resources managed by cooperating but independent administrative domains transparently accessible via a common namespace.

* From the Lyon workshop on Federated Data Stores: [http://indico.in2p3.fr/conferenceProgram.py?confId=5527](http://indico.in2p3.fr/conferenceProgram.py?confId=5527)
Federating Xrootd

- The simplest kind of federation is illustrated below:

Federation overlays on top of existing storage
Cross-region queries

To limit namespace query propagation, queries spill over to other regions only if nothing is found locally.
Deployment

- Currently, redirector at xrootd.unl.edu.
- Includes the FNAL T1 (dCache) and 8 T2s (5 HDFS, 1 dCache, 1 Lustre, 1 L-Store).
- During April, our monitoring recorded:
  - Over 300 unique users,
  - 900K file transfers
  - 300TB moved.
To limit the effects of latency, reads are bundled into large vectors.
| File A          | User Hash | Server Domain | Client Domain | Open Ago  | Update Ago | Read [MB] | Read [%] | Rate [

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>store/mc/Fall11</td>
<td>6DB21B7</td>
<td>hep.wisc.edu</td>
<td>unl.edu</td>
<td>25:37:11</td>
<td>21:22:11</td>
<td>2728.245</td>
<td>68.827</td>
<td>0.030</td>
</tr>
<tr>
<td>store/mc/Summer11</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>unl.edu</td>
<td>01:06:03</td>
<td>00:05:23</td>
<td>2523.107</td>
<td>62.609</td>
<td>0.637</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3D8C2B76</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>04:30:43</td>
<td>00:08:38</td>
<td>2498.118</td>
<td>54.639</td>
<td>0.154</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3D8C2B76</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>03:15:18</td>
<td>00:01:08</td>
<td>2474.463</td>
<td>57.433</td>
<td>0.211</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3D8C2B76</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>04:00:37</td>
<td>00:01:42</td>
<td>2442.054</td>
<td>57.521</td>
<td>0.169</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>00:51:58</td>
<td>00:00:58</td>
<td>2429.944</td>
<td>64.505</td>
<td>0.779</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>02:14:36</td>
<td>00:04:21</td>
<td>2420.094</td>
<td>59.497</td>
<td>0.300</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>01:26:01</td>
<td>00:04:41</td>
<td>2413.841</td>
<td>63.327</td>
<td>0.468</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3D8C2B76</td>
<td>hep.wisc.edu</td>
<td>t2.ucsd.edu</td>
<td>04:00:47</td>
<td>00:10:22</td>
<td>2411.461</td>
<td>52.835</td>
<td>0.167</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3D8C2B76</td>
<td>hep.wisc.edu</td>
<td>t2.ucsd.edu</td>
<td>01:31:55</td>
<td>00:00:50</td>
<td>2366.587</td>
<td>64.929</td>
<td>0.429</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>03:52:41</td>
<td>00:01:56</td>
<td>2365.060</td>
<td>59.156</td>
<td>0.169</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3AC086B</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>01:17:18</td>
<td>00:00:18</td>
<td>2320.691</td>
<td>62.855</td>
<td>0.500</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3D8C2B76</td>
<td>hep.wisc.edu</td>
<td>t2.ucsd.edu</td>
<td>04:41:24</td>
<td>00:08:09</td>
<td>2274.699</td>
<td>48.299</td>
<td>0.135</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3D8C2B76</td>
<td>hep.wisc.edu</td>
<td>t2.ucsd.edu</td>
<td>04:01:48</td>
<td>00:00:23</td>
<td>2212.339</td>
<td>59.802</td>
<td>0.152</td>
</tr>
<tr>
<td>store/mc/Summer11</td>
<td>3C12BE84</td>
<td>hep.wisc.edu</td>
<td>ultralight.org</td>
<td>02:02:28</td>
<td>00:16:48</td>
<td>2170.151</td>
<td>54.353</td>
<td>0.295</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3D8C2B76</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>04:27:46</td>
<td>00:05:21</td>
<td>2164.812</td>
<td>53.054</td>
<td>0.135</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>01:20:25</td>
<td>00:00:30</td>
<td>2117.791</td>
<td>53.866</td>
<td>0.439</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3C12BE84</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>03:52:41</td>
<td>00:11:16</td>
<td>2101.871</td>
<td>52.162</td>
<td>0.151</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3D8C2B76</td>
<td>fnal.gov</td>
<td>t2.ucsd.edu</td>
<td>04:20:26</td>
<td>00:00:56</td>
<td>1991.785</td>
<td>61.253</td>
<td>0.127</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3AC086B</td>
<td>ihep.ufl.edu</td>
<td>t2.ucsd.edu</td>
<td>08:41:15</td>
<td>00:00:05</td>
<td>1976.205</td>
<td>62.562</td>
<td>0.063</td>
</tr>
<tr>
<td>store/mc/Summer11</td>
<td>3C12BE84</td>
<td>unl.edu</td>
<td>ultralight.org</td>
<td>01:17:14</td>
<td>00:10:34</td>
<td>1953.870</td>
<td>53.664</td>
<td>0.422</td>
</tr>
<tr>
<td>store/mc/Fall11</td>
<td>B83B3C94</td>
<td>hep.wisc.edu</td>
<td>ultralight.org</td>
<td>11:32:01</td>
<td>02:49:41</td>
<td>1944.444</td>
<td>52.812</td>
<td>0.047</td>
</tr>
<tr>
<td>store/data/Run2012A</td>
<td>3D8C2B76</td>
<td>hep.wisc.edu</td>
<td>t2.ucsd.edu</td>
<td>03:17:33</td>
<td>00:02:03</td>
<td>1919.433</td>
<td>64.035</td>
<td>0.162</td>
</tr>
<tr>
<td>store/data/Run2011A</td>
<td>3C12BE84</td>
<td>ihep.ufl.edu</td>
<td>unl.edu</td>
<td>03:38:01</td>
<td>00:00:01</td>
<td>1908.636</td>
<td>50.158</td>
<td>0.146</td>
</tr>
</tbody>
</table>
Volume of Gigabytes Transferred By Facility
13 Days from 2012-04-09 to 2012-04-22

Maximum: 17,992 GB, Minimum: 3,300 GB, Average: 8,304 GB, Current: 3,916 GB
<table>
<thead>
<tr>
<th>Source Site</th>
<th>Volume GB</th>
<th># of Transfers</th>
<th>Yesterday Diff</th>
<th>One Week Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOW</td>
<td>1,080</td>
<td>1,223</td>
<td>55%</td>
<td>1908%</td>
</tr>
<tr>
<td>GLOW_Internal</td>
<td>46,053</td>
<td>38,388</td>
<td>65%</td>
<td>32%</td>
</tr>
<tr>
<td>MIT</td>
<td>4,950</td>
<td>11,460</td>
<td>103%</td>
<td>95800%</td>
</tr>
<tr>
<td>Nebraska</td>
<td>4,195</td>
<td>9,118</td>
<td>-43%</td>
<td>65%</td>
</tr>
<tr>
<td>Purdue</td>
<td>415</td>
<td>2,168</td>
<td>279%</td>
<td>374%</td>
</tr>
<tr>
<td>T2_IT_Bari</td>
<td>0</td>
<td>6</td>
<td>Unknown</td>
<td>-96%</td>
</tr>
<tr>
<td>UCSD</td>
<td>551</td>
<td>4,245</td>
<td>-65%</td>
<td>927%</td>
</tr>
<tr>
<td>UFL</td>
<td>1,138</td>
<td>2,066</td>
<td>945%</td>
<td>1250%</td>
</tr>
<tr>
<td>USCMS-FNAL-WC1</td>
<td>1,521</td>
<td>2,222</td>
<td>-57%</td>
<td>-32%</td>
</tr>
<tr>
<td>Vanderbilt</td>
<td>14</td>
<td>746</td>
<td>400%</td>
<td>598%</td>
</tr>
</tbody>
</table>
Conditions Data

- Conditions data is, happily, mostly taken care of for us.

- All conditions data in CMS is distributed via a network of HTTP proxies.

- The actual volume is small (50GB total; small percentage of this is used per job).

- So, only inbound HTTP (preferably, a local HTTP proxy cache) is needed for this.
Software

- The base software install takes about 20GB, and 5-10GB per additional version of CMSSW. 1-2 hours to deploy it “on the fly”.

- Due to disk size and time restraints, impossible to deploy it “on the fly” with the job.

- Given an arbitrary job, only a small portion of the files are used.

How do we solve this? See the next talk!
Use Cases
Interactive Use

• Distributing the software via CVMFS, conditions via HTTP, and physics data via Xrootd, any CMS job can run on any computer - regardless of whether it is in a CMS data center.

• This covers the “interactive use case”, where a physicist is debugging their code or viewing events in the event viewer.

• A significant percentage of AAA is optimizing the I/O code to be robust in face of high-latency connections.
• In CMS, the jobs are sent to a site where the data is stored.

• Due to unpredictable transient issues, a small percentage is unavailable at any given time.

• If a job cannot open a file, access fails.

• Now, if a job cannot open a file, it uses the xrootd infrastructure instead.
Overflow

- We use the glideinWMS software to create a heterogeneous Condor pool containing worker nodes from as many sites as possible.

- Condor knows which site each slot is from and matches the jobs according to data locality.

- Due to transient non-optimal data distribution, there may be slots available with no matching slots.

- In this case, we will purposely send queued jobs to the “wrong” site if the fallback mechanism can provide the data over the WAN.

- We’re getting close to being able to send jobs to non-CMS sites in production!
Security

• The overflow job is submitted under one identity (the “pilot”), but the actual code is from the user.

• Needs to run under a different identity, as user can run arbitrary code and the pilot identity is quite powerful.

• At CMS sites, we have a setuid binary (glexec) to allow identity switching.

• A bit heavy-handed for opportunistic sites; looking to use parrot identity boxing for isolation.
Thoughts for the Future

- We use CCTools to enable opportunistic grid work.
- Xrootd deployment and client are unique in their abilities to reliably federate multiple sites.
- But the protocol is niche.
- We are starting to understand what aspects are lacking in other protocol stacks (e.g., HTTP).
- It outlines a path that others could take.
The Commoditization of HEP?

- The AAA infrastructure shows CMS can utilize non-CMS sites.
- We’ve been able to greatly decrease the “site footprint” of an HEP experiment with respect to the data management.
- With a significant investment, one could use similar techniques for HTTP. *There is not a straightforward translation.*
- The vision is “Computing as a Service”: be able to utilize a research computing site with as light a footprint as possible.
- The computing sites are only partway: the final frontier for the commoditization is showing a HEP experiment can use an off-the-shelf workflow management system.