Journal of Physics: Conference Series **513** (2014) 042005 doi:10.1088/1742-6596/513/4/042005

CMS Use of a Data Federation

Kenneth Bloom (for the CMS Collaboration)

Department of Physics and Astronomy, University of Nebraska-Lincoln, Lincoln, NE 68588-0299 USA

E-mail: kenbloom@unl.edu

Abstract. CMS is in the process of deploying an Xrootd based infrastructure to facilitate a global data federation. The services of the federation are available to export data from half the physical capacity and the majority of sites are configured to read data over the federation as a back-up. CMS began with a relatively modest set of use-cases for recovery of failed local file opens, debugging and visualization. CMS is finding that the data federation can be used to support small scale analysis and load balancing. Looking forward we see potential in using the federation to provide more flexibility in the location workflows are executed as the difference between local access and wide area access are diminished by optimization and improved networking. In this presentation we discuss the application development work and the facility deployment work, the use-cases currently in production, and the potential for the technology moving forward.

1. Introduction

In its implementation of the grid-computing paradigm, CMS has generally moved computing jobs to the input data, so that the job runs in the same room where the data is stored. This was a design decision made ten years ago, based on the assumption that data transfers would be slow and unreliable. This scheme has in fact proven to be successful – CMS has produced hundreds of physics results, including the observation of the Higgs boson [1] that secured the 2013 Nobel Prize for Francois Englert and Peter Higgs [2].

However, the experience of operating the distributed computing system has indicated a variety of problems. In pre-placing the data, one must guess what the most popular datasets are and how to match them to the processing capacity available at each site. When there is a mismatch, there can be long queuing times at sites holding popular data when there may be available CPU at another site. It is hard to incorporate processing resources that are not dedicated to the experiment, because they do not host the necessary data. The co-location of data and jobs requires that multiple copies of datasets be distributed across the grid sites, and it is not clear that keeping so many copies will be affordable in the coming years. Finally, users would rather run on local resources that they control rather than going out to the grid, but they may not have the money or the expertise to run their own large storage system for the data they need.

All of this can be solved if CMS could provide access to "Any Data, Anytime, Anywhere" (AAA)[1] – that is, by making it possible for any physicist to locally read any CMS data from any site, no matter their own physical location. A project to do just this was started by US CMS collaborators at the University of Nebraska-Lincoln, the University of Wisconsin-Madison

[1] This author is generally credited with coining this phrase.

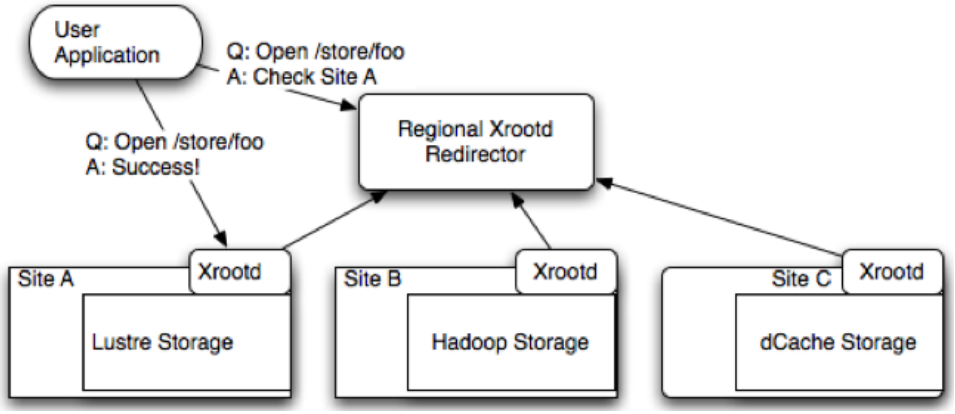

Figure 1. Schematic diagram of AAA redirection and file access.

and the University of California, San Diego in 2010; the National Science Foundation began supporting this work in Fall 2011. The goal of the project is to make data accessible over the wide-area network

- reliably: with no access failures regardless of data location
- transparently: so that a user never knows where the data actually reside
- easily: so that there are no operational burdens for physicists who want to access remote data with local resources
- universally: to fulfill the promise of opportunistic grid computing in which every available processing resource on the grid can be used for a particular application.

The technical solution for AAA is federated storage, which can be defined as a collection of disparate storage resources that are transparently accessible across a wide area via a common namespace. The CMS AAA data federation uses Xrootd [3] as its underlying technology. The structure of the system is shown in Figure 1. Xrootd provides a uniform interface in front of heterogeneous storage systems. Sites in the data federation publish their data to a redirector, which can then be queried by applications seeking to access data. If the requested data is found at a federated site, the application is directed to that site for reading over the WAN. If the data is absent in a particular region, there is a subsequent fallback to query a different redirector. All of the access is authenticated using grid credentials.

The data federation was easily implemented at CMS because the experiment had a globally consistent data file namespace to begin with. At a particular CMS computing site, each physical filename is constructed from a local prefix with a logical filename appended; the logical filename uniquely identifies a file in the CMS data management system. But the successful operation of the federation relies on earlier work to optimize the CMS software I/O stack to reduce read latencies. Data formats were carefully designed to maximize the potential for partial reads when data is filtered in analysis [4]. In addition, wide-area networking has proven to be robust, in contrast to the expectations of ten years ago, allowing reliable access to remote filesystems. It is important to remember that CPU efficiency is still better for jobs that read data locally. Analysis jobs at CMS Tier-2 centers run at 92% efficiency for local access and 86% efficiency for remote access (with 0.48 s/event and 0.65 s/event, respectively). However, AAA technology will allow CMS to make better use of expanding WAN bandwidth because of the efforts to reduce latencies.

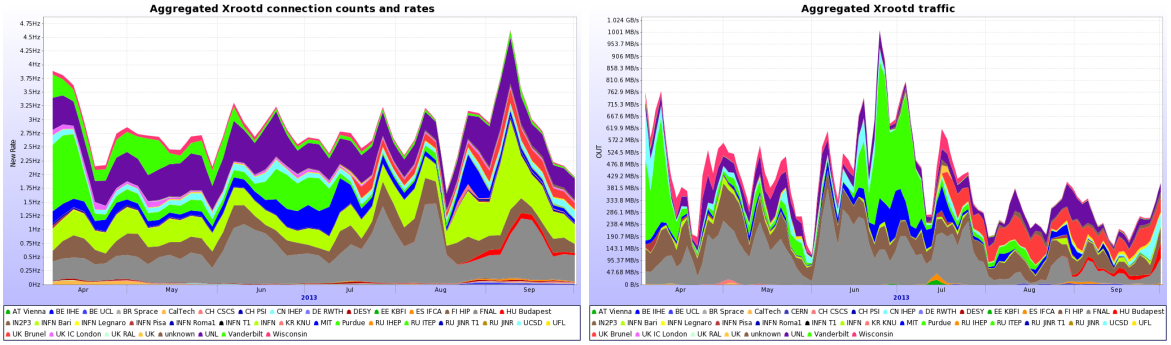

Figure 2. Rate of new file connections (left) and total outbound traffic (right) in the CMS data federation over a six-month period. Each color represents a different source site.

2. Deployment

CMS has set a goal for all of its Tier-1 sites and as many Tier-2 sites as possible to join the data federation by the start of Run 2 of the LHC in 2015. The deployment is now well underway. Three of the seven Tier-1 sites – those in Italy, the UK and the US – have placed all of their disk-resident data in the federation. This number will grow as sites implement disk-tape separation. As for the Tier-2 sites, 39 of 51 sites are now federated. The missing sites are typically smaller and/or less performant, so that more than 95% of the unique datasets that are resident at Tier-2 sites – more than 11 PB – are available in the federation. The Xrootd technology works with the heterogeneous storage technologies at the sites, such as dCache, Hadoop, DPM, StoRM and CASTOR. The status of the infrastructure is monitored through site availability monitoring (SAM) and Nagios tests.

An extensive system has been deployed to monitor the usage of the data federation [5]. Figure 2 (left) shows the rate of new file connections over the past six months. The average rate is 3 Hz, with spikes at various times. Figure 2 (right) shows the total traffic through the system. The output rate averages to 450 MB/s, which corresponds to 39 TB/day. In contrast, the daily average amount of data transferred by CMS through the PhEDEx subscription service was 81 TB/day during the same period. Thus a significant fraction of the experiment's data movement is occurring through the AAA federation. CMS also gathers more detailed information that monitors activity at individual sites, as shown in Figure 3, and even at the level of individual users and files [6]. Only 22 sites are publishing the detailed monitoring information as of this writing, but this should grow quickly with the recent WLCG release of a plugin that will work with dCache systems.

On any given day, AAA is typically used by 20 to 30 unique users with five to fifteen different destination networks. 5-10% of user analysis jobs at Tier-2 sites are using AAA in some fashion. Since more growth is anticipated before the start of Run 2 in 2015, CMS is now conducting scale tests of the Xrootd redirectors and servers. So far, it has been shown that the redirector can handle a 10 Hz rate of file-open requests, which should be sufficient for the steady-state load, but perhaps not a burst of requests from many jobs starting simultaneously.

3. Applications

CMS has found many applications for the AAA technology that can make the entire distributed computing system more robust and efficient, and make it easier to incorporate resources beyond those dedicated to the experiment. Almost all of these rely on the so-called "fallback mechanism." Usually, if a job fails to open a file, the job crashes, with any existing output typically lost. AAA cures this problem via the fallback mechanism. On a local file-open failure,

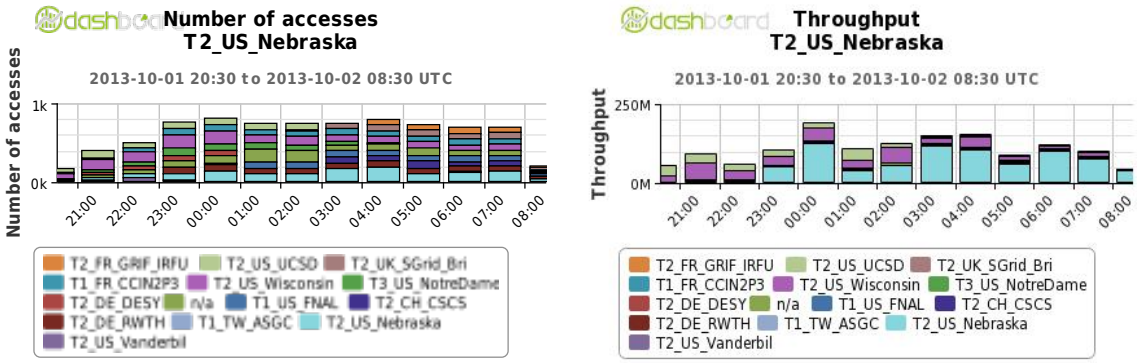

Figure 3. Detailed monitoring at one source site, showing the number of accesses (left) and throughput (right) over an hour, broken down by client site.

the CMS software stack asks the redirector to find the file elsewhere. The job then reads the remote file and continues on, without the user noticing. This makes users much less sensitive to storage problems at any given site, and gives them more throughput with less CPU time wasted on failed jobs.

There are many ways to take advantage of this ability to straightforwardly run at a site that does not host the desired data. For instance, sites that host popular datasets can have very long batch queues, and users can spend much time just waiting for their jobs to start. But since the job need not run at that particular site, it is possible to re-direct ("overflow") queued jobs to another site that does not host the desired data but has free job slots. The jobs will then read the data through AAA. The resulting CPU efficiency will be lower, but at least the jobs can start up sooner. This overflow scheme was implemented by changing scheduling policies in the glideinWMS layer. So far, overflow has only been deployed amongst four sites in the US, with O(1K) jobs targeted, but expansion of the scheme is planned.

A site that doesn't host any data locally at all can obviously make use of the fallback mechanism. Some Tier-3 sites are now completing entire data analyses through AAA. A single site has run ∼800 simultaneous jobs ingesting 2-3 Gb/s over the WAN sustained for the week with a 99% rate of successful execution. A physicist at the site expressed great satisfaction with being able to do his analysis only with resources under his own control, saying, "At this point, I basically don't pay attention to where the data is and just assumes that jobs will find the data and run." Going beyond this, CMS can consider accepting diskless Tier-2 sites at well-networked centers, and large sites that temporarily lose their data due to a storage downtime (planned or unplanned) can continue to operate as normal through the fallback mechanism. This allows the continuity of system-wide processing capacity.

CMS had a demonstration of large-scale remote data-reading during Spring 2013, when a "legacy" reprocessing of the 2012 LHC dataset and associated simulation samples was performed. All of the input data was resident at Tier-1 sites. While those sites ran on all of the LHC data, Tier-2 sites processed the simulations by reading the input through AAA. Figure 4 shows the numbers of reprocessing jobs running at Tier-1 and Tier-2 centers during this period. By moving some of the processing load from Tier-1 to Tier-2 the entire task was completed more quickly and the data was handed over to physicists sooner.

Any data, anywhere also means any data on any computer, not just those that are owned by CMS. AAA provides a simple way of incorporating opportunistic resources that are not directly connected to CMS into the experiment's computing system. The CMS software must still be provided to the opportunistic machine, but this can be done through Parrot and CVMFS for

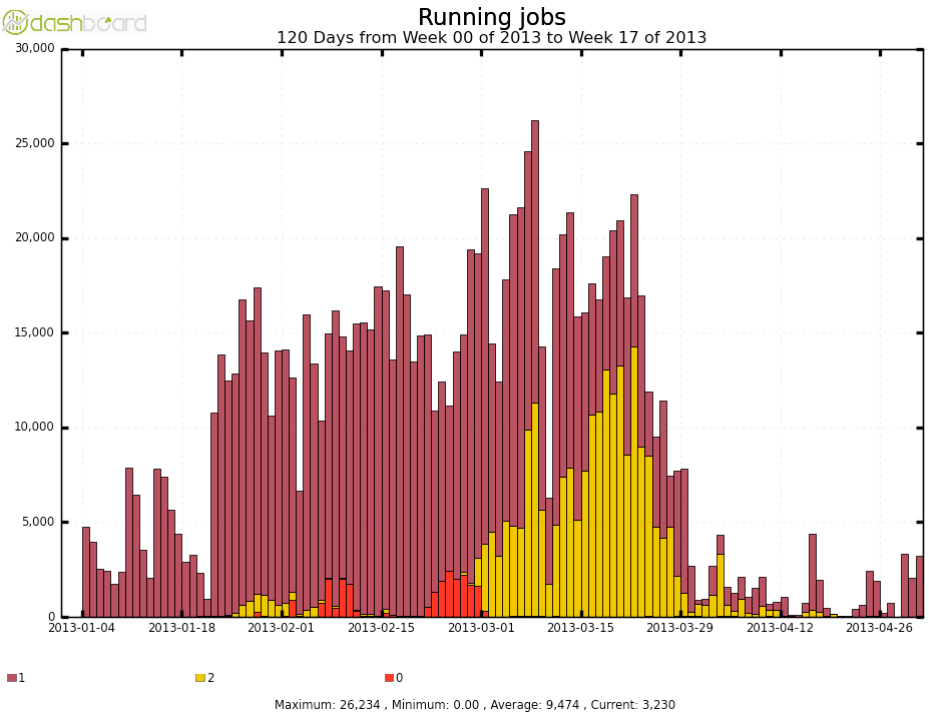

Figure 4. Count of average number of running jobs each day during the re-processing of the 2012 LHC data and corresponding simulations. Maroon indicates jobs running at Tier-1 sites and yellow indicates jobs running at Tier-2 sites.

download on demand in such a way that only 500 MB of files are required rather than the full 17 GB CMS software install. But data need not be resident; it can be read through the AAA fallback mechanism. Typical jobs run only 2% slower on opportunistic resources compared to CMS sites. This setup opens the door to the use of any opportunistic resource, including clouds. There is much CMS development work underway in this area [7]. CMS has done a successful demonstration on the Amazon cloud, and on the Open Science Grid, where CMS has successfully run ∼2000 simultaneous jobs across fifteen sites, starting up at a rate of 3 Hz. The total usage on the OSG so far has been 1.2M CPU hours, some of which were obtained from ATLAS sites.

4. Upcoming developments

The AAA development team, along with CMS collaborators, still has several projects in progress. At the moment, the CMS software can recover from an unopenable file through the fallback mechanism, but not a bad block in the middle of a file. A file-healing system is being developed that will read a bad block from a remote site, and also make a copy to the local site to repair the file. A prototype for this system already exists for HDFS. Totally dynamic caching is being explored. In such a scheme, when a site requests a file that is not available locally, the remote file is not just ready but copied into a dynamic cache. This cache can then be managed on the basis of how files in the cache are used. This opens the possibility of cache-only storage at sites, rather than subscribed files. There is interest in network-aware technologies for both the initial redirection, and then for dynamic file streaming [4]. An important future application will be the use of AAA monitoring information to determine dataset popularity, which can then drive automated data placement and deletion for the most efficient use of the available disk space.

5. Outlook

By deploying a data federation, CMS has made virtually all of its data available to all of its users, anytime and anywhere. The AAA system has now been deployed throughout most of CMS, and is moving towards widespread use and acceptance by physicists, who are beginning to understand the power of this technology. Applications such as fallback, overflow and diskless centers have already allowed more efficient use of existing resources, and also open the door to greater use of opportunistic resources. Coming developments will lead to more stable operations and even greater efficiency of resource use.

But perhaps the most important application of a data federation is the empowering individual users to access the data when, where and how they want, so that they can do their analyses more easily, with less knowledge of computing details required. This supports the ultimate goal of CMS computing, which is to get the physics out fast – and first.

Acknolwedgements

I thank my collaborators on the AAA project for their many contributions, including their feedback on this presentation. This work is supported in part by the National Science Foundation through awards PHY-1104664, PHY-1104549 and PHY-1104664.

References

[1] CMS Collaboration, "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," Phys. Lett. **B716** 30 (2012), http://dx.doi.org/10.1016/j.physletb.2012.08.021; ATLAS Collaboration, "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," Phys Lett. **B716** 1 (2012), http://dx.doi.org/10.1016/j.physletb.2012.08.020

[2] "The Nobel Prize in Physics 2013," http://www.nobelprize.org/nobel_prizes/physics/laureates/2013/ for more details.

[3] See http://xrootd.slac.stanford.edu for more details.

[4] B. Bockelman, "Optimizing High-Latency I/O in CMSSW," these proceedings.

[5] Plots for the global monitoring described here can be found at http://xrootd.t2.ucsd.edu.

[6] Plots for the detailed monitoring described here can be found at http://dashb-cms-xrootd-transfers.cern.ch.

[7] D. Hufnagel and P. Kreuzer, "Opportunistic Resource Usage in CMS," these proceedings.