Journal of Physics: Conference Series **513** (2014) 032037 doi:10.1088/1742-6596/513/3/032037

Grids, virtualization, and clouds at Fermilab

S Timm[1], K Chadwick[1], G Garzoglio[1*] and S Noh[2]

[1]Scientific Computing Division, Fermi National Accelerator Laboratory

[2]Global Science experimental Data hub Center, Korea Institute of Science and Technology Information

E-mail: {timm, chadwick, garzoglio }@fnal.gov, rsyoung@kisti.re.kr

Abstract. Fermilab supports a scientific program that includes experiments and scientists located across the globe. To better serve this community, in 2004, the (then) Computing Division undertook the strategy of placing all of the High Throughput Computing (HTC) resources in a Campus Grid known as FermiGrid, supported by common shared services. In 2007, the FermiGrid Services group deployed a service infrastructure that utilized Xen virtualization, LVS network routing and MySQL circular replication to deliver highly available services that offered significant performance, reliability and serviceability improvements. This deployment was further enhanced through the deployment of a distributed redundant network core architecture and the physical distribution of the systems that host the virtual machines across multiple buildings on the Fermilab Campus. In 2010, building on the experience pioneered by FermiGrid in delivering production services in a virtual infrastructure, the Computing Sector commissioned the FermiCloud, General Physics Computing Facility and Virtual Services projects to serve as platforms for support of scientific computing (FermiCloud & GPCF) and core computing (Virtual Services). This work will present the evolution of the Fermilab Campus Grid, Virtualization and Cloud Computing infrastructure together with plans for the future.

1. Introduction

The Fermilab Computing Division participated in several early Grid computing research and development projects. In 2004, the Computing Division management made the strategic decision to unify all Fermilab high-throughput computing (HTC) resources into a meta-facility (now known as a Campus Grid) called FermiGrid [1,2]. This strategy was designed to allow the optimization of resources at Fermilab, to make a coherent way of integrating Fermilab into the Open Science Grid (OSG) [3,4], to save effort through the implementation of shared services, and to fully support the OSG and the Large Hadron Collider (LHC) Computing Grid. Experiments (such as CDF, D0 and CMS) that had previously provisioned large dedicated clusters would have first priority for these resources. Opportunistic access via common Grid interfaces would be enabled when the resources were not being fully utilized. FermiGrid still follows this strategy today, as a meta-facility of several clusters rather than a monolithic cluster with set allocations for major virtual organizations.

In the early days of the project to deliver FermiGrid, several key policy and strategic decisions were made. These included:

* To whom any correspondence should be addressed.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

- The development of security policies for the Open Science Enclave (later renamed to the Open Science Environment) that addressed required system management configurations;
- The development of a site wide gateway that automatically routed Grid jobs from various communities, such as the OSG, to clusters within the Campus Grid;
- A program of work that established interoperation for our major stakeholders and Campus Grid clusters that ensured that it was possible for all major stakeholders to run jobs on all of the Campus Grid clusters;
- The incorporation of a unified Grid credential to Unix UID mapping service across all the Grid clusters together with a central credential banning service;
- The decision to ensure that support for a minimum of two batch systems was maintained (currently HTCondor and Torque/PBS).

In addition, Fermilab has deployed a Short Lived Credential Service (SLCS) Kerberos Certificate Authority (KCA) that supports automatic generation of short-lived X.509 credentials derived from Kerberos 5 credentials. Each of these decisions will be discussed in detail below.

2. The Open Science Environment

The strategy that led to the development of the Open Science Environment was to separate the function of computational resources that run arbitrary jobs from external grid sources (Open Science Environment) from the functions of interactive login (General Computing Environment). This is done by severely limiting interactive logins on the nodes, making sure that home directories are not shared, and that those application and data areas that are shared are not executable on any interactive node. Grid jobs execute in different accounts that do not have shells and thus can't be logged into. This limits the ability of grid jobs to put arbitrary executables or modify user startup scripts in areas where users can log in. We require professional system management and prompt application of all security patches. The authorization servers for the Open Science Enclave are in locked racks in a card-access-restricted computer room.

3. Single site gateway and interoperability

At the present time, FermiGrid has twelve Compute Elements (CEs) that serve seven different Grid clusters (CMS, CDF, D0 x 4, and General Purpose). The majority of these CEs are not directly accessible outside of the Fermilab site. A single site gateway serves as an intermediary, accepting jobs from external OSG VOs as well as many on-site submitters who have chosen to run opportunistically across FermiGrid. From each of the individual Grid clusters, the site gateway receives information such as memory, disk, operating system version, number of jobs in execution or waiting and how many free resources are available for each Virtual Organization (VO). This information is transferred via the CEMon utility in the form of HTCondor classads and collected by the Resource Selection Service (ReSS) [5] Information Gatherer. The site gateway determines the potential candidate cluster to forward jobs to using HTCondor-G matchmaking, finding the cluster with free nodes that match the published job requirements.

If a matched job does not start on the selected cluster within two hours, it is rescheduled on a different cluster where free slots are likely to be available. Most of the major users have configured their jobs to be able to run on any FermiGrid cluster. When VO members are scheduled on their "home" cluster, they are not subject to preemption. Opportunistic jobs can be preempted if the primary user needs the resources. The opportunistic jobs, however, are given 24 hours to complete their processing when the preemption signal is given to the job.

4. Overlay batch system

Prior to widespread grid deployment, users were accustomed to submitting jobs to a local cluster. The idea of glideins using Condor through the system known first as glideCAF and later as glideinWMS allowed us to aggregate resources from across the complicated site into a single pool. All of the certificate authorization was generated automatically on the users' behalf, using automatically

generated (Robot) X.509 credentials from the Fermilab KCA. This was a key technology in transitioning from expert-based grid production use to transparent use of the grid by all users.

5. Unified user mappings and central banning server

The Grid User Mapping Service (GUMS) is designed to assign privileges to the user from her Virtual Organization, Group and Role. This information is encapsulated in the extended user credentials in the Distinguished Name and Fully Qualified Attribute Name (FQAN). GUMS is called by all resource gateways for compute elements, worker nodes, and dCache-based storage elements. The calling machine sends a message in XACML format, based on an agreed interoperability profile [6,9,10]. Some virtual organizations request to have pool accounts, in which each user's Distinguished Name is mapped to an individual username. Others are mapped to group accounts in which all members of a virtual organization run as the same username.

The Site AuthoriZation (SAZ) service returns a simple Permitted/Denied response. It allows our site to veto any grid identity based on Distinguished Name, Certificate Authority, VO, Role, or certificate serial number. It uses the same XACML-based messaging format to receive its information and is also called both from compute elements and from worker nodes. The ability to ban single users immediately during incident response investigation, rather than waiting for certificate revocations to propagate through the grid, is key to fast and effective incident response and is also used to quickly terminate jobs which are putting undue load on the batch and storage resources.

6. Virtualization and high availability

The FermiGrid central authentication services serve a number of different stakeholders, all of which have different and potentially incompatible scheduled maintenance windows. If central services such as these are unavailable, no jobs can start or stop. Our service level agreement specified a 99.9% uptime and we designed for 99.999%. We also had to design for the potential of 5000 or more simultaneous clients all trying to contact the server at once. Yet most of the time these central services required fairly small resources in terms of CPU, memory, and disk space under normal conditions, and could not co-exist within the same operating system image because they all wanted to run on some of the same ports. We turned to virtual machines and high availability to address these problems.

By using Xen for virtualization, we were able to transition the VOMS, GUMS, and SAZ services, which previously had run on three separate machines, to virtual machines on a single server. All three of these services hold all state in a MySQL database so we made a fourth virtual machine to be a common database backend for all three services. We then made a second server also with copies of these four virtual machines. In the initial days of Xen we needed to use a custom-built Xen kernel, but support was later added into Scientific Linux 5.

We use Linux Virtual Server (as distributed in the Piranha package of Scientific Linux) to serve the public facing service IP's and then directly route the traffic to one or both of the two virtual machines based on a weighted-least-connections algorithm. This allowed us to have an active/active configuration, using both copies of the service when both were available and failing all the traffic to one when the other was down. The Linux Virtual Server itself has a backup that is supported by heartbeat. The FermiGrid-HA configuration went live in late 2007 and has delivered better than 99.9% service availability from that time until now.

We later added the Squid web proxy service and MyProxy to the high-availability services we support. MyProxy stores its state in a file system. We replicate this using the DRBD file system and control the service using Heartbeat. The Gratia Accounting services that we operate for the Fermilab site and for the Open Science Grid also are in a partial high availability configuration, in which there is always a web server and database available for reporting resource usage, using one-way MySQL replication.

In 2011, after a series of electrical and network failures, we divided the two halves of the services between two different buildings at Fermilab in the FermiGrid-HA2 project. The network topology has also been changed such that even if we lose one of the two buildings we can still get network

switching and routing. This configuration successfully worked hours after deployment due to a building failure and has worked several times since that. The flexibility in doing upgrades and routine maintenance that this service structure affords is key to high reliability over the long term.

7. Virtualization and cloud at Fermilab

After several years of successful operation of high-availability static virtualization in FermiGrid and elsewhere on site, the Computing Division launched three new virtualization projects in 2010. An enterprise-class VMWare virtualization infrastructure was set up with its focus on the core IT services. The General Physics Computing Facility was set up to provide static virtualization for long-lived scientific stakeholder application such as interactive login, local batch submission, and other auxiliary services. The FermiCloud project was launched to investigate cloud technologies and establish an on-demand Infrastructure as a Service facility for scientific stakeholders. This includes the developers, integrators, and testers who build and operate grid middleware systems for scientific stakeholders, who have a large need for a test facility. We anticipated that the various projects would eventually share technology solutions and interoperate.

The FermiCloud project has consisted of four phases to date. In the first phase we collected the requirements, bought the hardware, and selected the OpenNebula open-source cloud management software stack. In the second phase we deployed a variety of virtual machines for scientific stakeholders and did systematic testing on real applications in virtualized hardware, including various open-source distributed storage applications [7,8], MPI over virtualized Infiniband, and networking performance. We also added X.509 authentication and secure contextualization. The third phase included expanding the cloud to two buildings linked by a replicated SAN-based file system. We also added accounting, automated configuration via puppet, and monitoring mechanisms. In the fourth phase during the summer of 2013, we completed a joint Cooperative Research and Development Agreement with the Korean Institute of Supercomputing and Technology Information (KISTI) in which we leveraged all these technologies to run workflows of real scientific users on a distributed cloud.

8. Grid technologies on the cloud

The key contributions of the FermiCloud project to cloud computing stem from our successful experience with grids and virtualization. In broad terms, these include security authentication, security policy, accounting, virtualization and high availability. We were able to leverage existing grid authentication and authorization services in the cloud. We use automatically generated certificates from our Kerberos Certificate Authority to use X.509 authentication, code that we contributed back to the core OpenNebula software. We used existing software modules of the Authorization Interoperability profile to contact our existing GUMS and SAZ services for cloud authorization.

FermiCloud differs from most other private clouds at national laboratories in that it is on the main site network with access to all resources both on and off site. Given the extensive experience in defining the security policies and controls for the Open Science Environment, we were able to predict some key security policy issues. We developed a secure contextualization process such that no persistent secrets, such as X.509 personal and host certificates or Kerberos 5 credentials, are stored in the image repository. They are loaded at launch time of the virtual machine and stored in a RAM disk so that they don't persist after the machine shutdown/reboot. We are working on a special security scanner for inbound virtual machine images that will scan them before providing access to the public network. Again permissions on shared network file systems are important, particularly since the infrastructure-as-a-service users have root access on their virtual machines.

We were able to reuse most of our Gratia accounting system for the grid to do user accounting for the cloud. After we wrote a new probe to interpret OpenNebula virtual machine records, the system naturally provided collection, reporting, and display. We also leverage the Nagios and RSV systems, typically used to monitor grid services, for monitoring essential services on the cloud, using available Nagios plugins for the cloud software.

The auxiliary services that help us run FermiCloud, such as puppet, cobbler, mysql, LVS, webservers, NIS servers, Nagios, and the secure secrets repository, are all themselves on virtual machines. We have plans to make the OpenNebula head node itself a virtual machine. We have used a variety of high availability tactics to keep the OpenNebula service up. We started with Heartbeat and DRBD/GFS2 in an active/active mode to control the OpenNebula service and the file repository. We have now moved the image repository to our replicated SAN and are using Clustered LVM (CLVM) and GFS2 for the file system that is shared between all nodes. The shared SAN-based file system allows for live migration of virtual machines and faster launching. The rgmanager function of the Red Hat clustering system is used to make sure that the OpenNebula service is running on one of the two identically configured head nodes.

9. Highlights of recent work

In the joint Cooperative Research and Development Agreement with KISTI recently completed this summer, we worked on three major activities, all geared toward enabling scientific workflows to run on federated clouds.

The Virtual Infrastructure Automation and Provisioning program was organized in three thrusts. First, we tested that the GlideinWMS system could directly submit pilot jobs as virtual machines to both FermiCloud and Amazon EC2. Second, we presented our cloud resources through a local grid gatekeeper; regular jobs received are locally queued and, through the vcluster system, virtual machines are provisioned at FermiCloud, KISTI's Gcloud, and Amazon EC2. Third, we commissioned a system to periodically check whether virtual machines on FermiCloud are idle and, in case, suspend them to reclaim the computing slot. All of these technology tests were successful and are on their way to being made part of our production cloud and job submission infrastructure. We used the GlideinWMS setup to submit a significant fraction of the NOvA experimental cosmic ray simulation to FermiCloud at the scale of 50-75 simultaneous virtual machines. The virtual machine submitted via glideinWMS is a bare-bones virtual machine, with the user applications delivered via the CernVM-FS (CVMFS) system.

Interoperability and Federation of Cloud Resources task investigated and documented differences between various clouds in the format of their virtual machine images and in the way in which they emulate the Amazon EC2 web services API. This produced a large document complete with examples and instructions for users [11].

High-Throughput Fabric Virtualization was a program of work to continue and repeat the earlier work we had done with virtualized high-speed Ethernet and Infiniband. We successfully repeated our earlier results with MPI applications over virtualized Infiniband. We have now added options to our production OpenNebula cluster so that a user can request machines with a virtual Infiniband interface.

10. Future plans

Fermilab is and has been a strong catalyst for scientific computing for multi-domain science and, in particular, for HEP at all the frontiers. As we move into the future, the infrastructure is shaped by the changing and increasing needs of our physics communities, from the collider experiments of the Energy Frontier to the more diverse world of Intensity and Cosmic Frontier. We are moving from the need of federating our campus for a few large collaborations, to supporting many smaller collaborations in addition to a few big ones (CMS, LSST 2018, etc.). This implies a push from statically allocated virtualization to a dynamic one (cloud) i.e. from statically allocated services to on-demand ones; from four large clusters to possibly two larger ones.

We are moving from being a large provider of opportunistic cycles to becoming a large user of those on the OSG; from relying on resources federated across academic institutions to including commercial ones to address coincident peak needs of our grown community. This means that we need to aid many user communities, which have run only on-site, in getting their applications to work on external grids and clouds. We will also re-examine our campus grid federation mechanism based on the new technologies, such as glideinWMS, that have become available since we originally deployed it

in 2005. Various new methods of authentication and authorization are also proliferating in the scientific computing world and we will have to be flexible to support those as well as the traditional X.509 based methods of authentication.

In addition to the number and size of the stakeholders changing, also the nature of the workload is changing, with bigger data sets that are not easily split in smaller sets. Some jobs require large scratch space; others require more memory than usual or multiple CPU's. The work tends to come in bursts rather than a steady rate. We also support legacy experiments with old versions of operating systems and applications that may require software-defined networking to contact mass storage services and databases. All of this means that we will need to use a variety of facilities in both grid and cloud and make their configuration and provisioning more flexible.

The next phase of the FermiCloud project will focus on increasing the scale of the federated cloud science workflows to 1000 or more virtual machines. We will look at provisioning algorithms, focusing on when to extend to commercial clouds, including allocations for spot pricing, and how to optimally shift workflow execution locations based on resource availability. We will also focus on data movement services to and from the cloud and how to scale their capacity based on demand.

11. Conclusion

The careful service and policy design at the beginning of FermiGrid has served us well through the grid era and well into the cloud era. Our efforts to date have resulted in a much better utilization of our on-site resources and high availability and reliability of our central services. We now have demonstrated a clear path to enable smaller stakeholders to make use of resources on the Open Science Grid, FermiCloud, and commercial clouds. Our focus going forward is to continue to optimize that experience for our scientific stakeholders.

Acknowledgements

Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359, and by CRADA FRA 2013-0001 / KISTI-C13013.

References

[1] K. Chadwick, E. Berman, P. Canal, T. Hesselroth, G. Garzoglio, T. Levshina, V. Sergeev et al. "FermiGrid — Experience and future plans." In Journal of Physics: Conference Series, vol. 119, no. 5, p. 052010. IOP Publishing, 2008.

[2] V. Sergeev, I. Sfiligoi, N. Sharma, S. Timm., DR Yocum, E. Berman, P. Canal, K. Chadwick, T. Hesselroth, G. Garzoglio, and T. Levshina, "FermiGrid" In proceedings of TeraGrid 2007, Madison, WI (2007).

[3] Pordes, R. et al. (2007). "The Open Science Grid", J. Phys. Conf. Ser. 78, 012057.doi:10.1088/1742-6596/78/1/012057.

[4] Sfiligoi, I., Bradley, D. C., Holzman, B., Mhashilkar, P., Padhi, S. and Wurthwein, F. (2009). "The Pilot Way to Grid Resources Using glideinWMS", 2009 WRI World Congress on Computer Science and Information Engineering, Vol. 2, pp. 428–432. doi:10.1109/CSIE.2009.950.

[5] P. Mhashilkar, G. Garzoglio, T. Levshina, and S. Timm. "ReSS: Resource Selection Service for National and Campus Grid Infrastructure." In Journal of Physics: Conference Series, vol. 219, no. 6, p. 062059. IOP Publishing, 2010.

[6] G. Garzoglio, J. Bester, K. Chadwick, D. Dykstra, D. Groep, J. Gu, T. Hesselroth et al. "Adoption of a SAML-XACML Profile for Authorization Interoperability across Grid Middleware in OSG and EGEE." In Journal of Physics: Conference Series, vol. 331, no. 6, p. 062011. IOP Publishing, 2011.

[7] G. Garzoglio, K. Chadwick, T. Hesselroth, A. Norman, D. Perevalov, D. Strain, and S. Timm. "Investigation of storage options for scientific computing on Grid and Cloud facilities." In International Symposium on Grids and Clouds and the Open Grid Forum, vol. 1, p. 47. 2011.

[8] G. Garzoglio. "Investigation of Storage Options for Scientific Computing on Grid and Cloud Facilities." In Journal of Physics: Conference Series, vol. 396, no. 4, p. 042021. IOP Publishing, 2012.
[9] R. Ananthakrishnan, G. Garzoglio, O. Koeroo, "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids", Open Grid Forum GFD.205 (2012)
[10] G. Garzoglio et al., "Definition and Implementation of a SAML-XACML Profile for Authorization Interoperability across Grid Middleware in OSG and EGEE", Published in the Journal of Grid Computing, Vol. 7, Issue 3 (2009), Page 297, DOI 10.1007/s10723-009-9117-4
[11] Virtual machine interoperability documentation: http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=5208