CMS CR -2012/068

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

10 May 2012

The benefits and challenges of sharing glidein factory operations across nine time zones between OSG and CMS

Igor Sfiligoi, Jeffrey M Dost, Marian Zvada, Ignas Butenas, Burt Holzman, Frank Wuerthwein, PeterKreuzer, Scott W Teige, Robert Quick, Jose M Hernndez, Joseph Flix

Abstract

OSG has been operating for a few years at UCSD a glideinWMS factory for several scientific communities, including CMS analysis, HCC and GLOW. This setup worked fine, but it had become a single point of failure. OSG thus recently added another instance at Indiana University, serving the same user communities. Similarly, CMS has been operating a glidein factory dedicated to reprocessing activities at Fermilab, with similar results. Recently, CMS decided to host another glidein factory at CERN, to increase the availability of the system, both for analysis, MC and reprocessing jobs. Given the large overlap between this new factory and the three factories in the US, and given that CMS represents a significant fraction of glideins going through the OSG factories, CMS and OSG formed a common operations team that operates all of the above factories. The reasoning behind this arrangement is that most operational issues stem from Grid-related problems, and are very similar for all the factory instances. Solving a problem in one instance thus very often solves the problem for all of them. This paper presents the operational experience of how we address both the social and technical issues of running multiple instances of a glideinWMS factory with operations staff spanning multiple time zones on two continents.

Presented at *CHEP 2012: International Conference on Computing in High Energy and Nuclear Physics*

The benefits and challenges of sharing glidein factory operations across nine time zones between OSG and CMS

I Sfiligoi[1], J M Dost[1], M Zvada[2], I Butenas[3], B Holzman[4], F Wuerthwein[1], P Kreuzer[5], S W Teige[6], R Quick[6], J M Hernández[7], J Flix[7,8]

[1]University of California San Diego, La Jolla, CA 92093, USA
[2]Karlsruher Institut für Technologie, 76021 Karlsruhe, Germany
[3]Vilniaus universitetas, LT-01513 Vilnius, Lithuania
[4]Fermilab, Batavia, IL 60510, USA
[5]RWTH Aachen University, III. Physikalisches Institut A, 52074 Aachen, Germany
[6]Indiana University, Bloomington, IN 47405, USA
[7]CIEMAT, 28040 Madrid, Spain
[8]Port d'Informació Científica, E-08193 Bellaterra, Spain

E-mail: isfiligoi@ucsd.edu

Abstract. OSG has been operating for a few years at UCSD a glideinWMS factory for several scientific communities, including CMS analysis, HCC and GLOW. This setup worked fine, but it had become a single point of failure. OSG thus recently added another instance at Indiana University, serving the same user communities. Similarly, CMS has been operating a glidein factory dedicated to reprocessing activities at Fermilab, with similar results. Recently, CMS decided to host another glidein factory at CERN, to increase the availability of the system, both for analysis, MC and reprocessing jobs. Given the large overlap between this new factory and the three factories in the US, and given that CMS represents a significant fraction of glideins going through the OSG factories, CMS and OSG formed a common operations team that operates all of the above factories. The reasoning behind this arrangement is that most operational issues stem from Grid-related problems, and are very similar for all the factory instances. Solving a problem in one instance thus very often solves the problem for all of them. This paper presents the operational experience of how we address both the social and technical issues of running multiple instances of a glideinWMS factory with operations staff spanning multiple time zones on two continents.

1. Introduction

Many scientific communities, or Virtual Organizations (VOs), have adopted Grid computing as their base computing model to simplify the deployment and management of the tens of thousand of CPUs needed to accomplish their mission. While the Grid computing paradigm has been shown to be a boon for resource providers, allowing them to keep their administrative autonomy over the resources they manage, direct use of these resources has been shown to be difficult for standard users.

As a result, the VOs have adopted the pilot-based Workload Management System (WMS) paradigm, also known as “overlay infrastructure”. In this paradigm, resources across multiple administrative domains are aggregated into VO specific overlay pools, or “virtual clusters” (VC). Each VO has full control over its own VC, and can thus easily implement priorities between the final users. Moreover, resource provisioning is clearly separated from resource usage, with the former managed by dedicated IT personnel. Standard users are thus never exposed to the complexities of Grid infrastructure and perceive the overlay pool as just any other compute cluster.

The implementation most broadly adopted across different scientific domains today is **glideinWMS**[1]. One characteristic of glideinWMS is the clear separation between the VO-facing layer implementing the provisioning logic and the Grid-interfacing layer responsible for the actual provisioning; the former is called a **VO Frontend**, while the latter is a **Glidein Factory**. This clear division allows the VOs to keep the control of their provisioning policies, while outsourcing the operations of the resource provisioning service to a dedicated team of Grid experts. This results in a clear division of labor, with the former supporting the domain scientists and their applications, and the latter working closely with IT professionals that physically manage and control the resources in each administrative domain. This separation has the additional advantage of making the Glidein Factory operations mostly independent of the served VO, allowing for instances serving multiple VOs and thus reducing the total cost of ownership (TCO) through economies of scale[2].

CMS[3] and OSG[4] jointly operate four such Glidein Factory instances, one at CERN, one at the OSG Operations facility at Indiana University (GOC), one at Fermilab and one at the University of California San Diego (UCSD). The UCSD and GOC factories serve VOs across several domains[5], spanning biology, chemistry, climate science, computer science, economics, engineering, mathematics, medicine, and physics. The rational for having a joint operations team is threefold. First, it provides service redundancy across several locations. Second, it provides extended hours of human intervention as the operations team includes people that span nine time zones from Europe to the Western United States. And third, most operational issues stem from Grid-related problems and are very similar for all the Glidein Factory instances; solving a problem in one instance thus very often solves the problem for all instances. In addition to these three anticipated advantages, we discovered that the operations principles for operating a service at the four different institutions is sufficiently different that this provided opportunities in its own right.

This paper presents the benefits and challenges of the above setup. Technical and social implications are described in separate sections, with both benefits and challenges presented.

2. Technical implications of operating multiple Glidein Factory instances

On the technical side, the most obvious benefit of having multiple Glidein Factories is the elimination of the single point of failure in a glideinWMS deployment. There are other advantages, too, but there are problems as well. This section provides an overview of the glideinWMS architecture followed by a discussion on the benefits and drawbacks of operating multiple Glidein Factories.

2.1. The glideinWMS architecture

As mentioned in the introduction, the glideinWMS architecture is based on two different services, the VO Frontend and the Glidein Factory. A VO Frontend instance implements the resource provisioning logic, while a Glidein Factory instance performs the actual resource provisioning. There is an n-to-m relationship between them.

Logically, a Glidein Factory instance is just a slave. It receives orders from one or more VO Frontend instances and acts accordingly; i.e. by itself, it does not request any resources and it does not try to second-guess the needs of the VO Frontends. Each VO Frontend is also treated in an independent manner; i.e. the Glidein Factory does not do any prioritization between the VO Frontends. The main added value of a Glidein Factory is to insulate the served VO Frontends from the details of resource provisioning. The benefits are threefold. First, the Glidein Factory provides an abstract description of the available resource providers, adding derived information if needed. Second, the Glidein Factory provides the site-specific configuration of the pilot, abstracting away most of the heterogeneity of the provisioned resources. The Glidein Factory operators also discover any new resource providers, extract and interpret the site specific information, validating them before making the resource provider available to the VO Frontends. Finally, the Glidein Factory operators monitor, debug and fix Grid-specific resource provisioning problems, often resulting in discussions with the various resource operators.

The logical role of a VO Frontend instance is instead being the manager of a virtual cluster; i.e. each VC is managed by its own VO Frontend. The VO Frontend instance monitors the local user job queues, decides if more resources are needed and, if appropriate, instructs one or more Glidein Factories to provision the needed resources. The VO Frontend owns the credential(s) needed to obtain access to the remote resources, delegating them to the Glidein Factories. It also provides any VO-specific pilot configuration needed to accommodate the needs of its user community.

The protocol between a VO Frontend and a Glidein Factory is based on the principle of constant pressure. When a VO Frontend needs a large number of additional resources, it does not ask for all of them in well defined chunks; instead, it asks a Glidein Factory for a stream of resource provisioning pilots, with the understanding that the VO Frontend will tell the Glidein Factory when to stop provisioning more. A nice property of this approach is the possibility to request multiple streams from the same resource provider, e.g. through the use of multiple Glidein Factories, without having to guess how effective these requests will be. As long as at least one stream is active, the desired resources will be provisioned in the optimal manner.

The above description is necessarily over-simplified. A reader interested in the architectural details should consult [1] and [6].

2.2. The benefits of operating multiple Glidein Factory instances

The most obvious benefit of having multiple Glidein Factories is providing redundancy to the glideinWMS ecosystem, thus eliminating the single point of failure. In case one Glidein Factory instance stops working, others pick up the load and the VOs hardly ever notice it. This benefit extends both to unscheduled and scheduled downtimes, adding the nice side effect of making maintenance of the services a relatively transparent process for users.

A related benefit is scalability. By partitioning the glidein requests over multiple Glidein Factory instances the total number of supported provisioned resources scales approximately linearly with the number of instances. If we were to hit scalability limits in our deployments, we could easily overcome them by instantiating yet another instance, either at one of the four existing data centers, or anywhere else.

2.3. The problems of operating multiple Glidein Factory instances

Having multiple Glidein Factory instances brings with it the problem of synchronization. In order to achieve full benefits from shared operations, each resource used by a VO must be configured at each and every instance it uses. Ideally, one would want full synchronization to keep operations as simple as possible. However, this is not possible operationally, due to the differences at the various instances both in terms of available hardware and local deployment policies.

One obvious problem is the difference in network setup. Each instance has a different node name, and this node name must be entered into the Glidein Factory configuration file in several places. Another quite obvious issue is the directory structure. Different institutions provide nodes with different mount points, based on the local cluster conventions. The location of the available directory trees must also be put in the instance's configuration file. Simply synchronizing the various configurations byte-by-byte is thus not possible.

Moreover, different instances may support a different set of Grid resources, and may even run a slightly different version of the glideinWMS software. This is typically due to the difference in update policies of the different institutions. For example, OSG Operations at Indiana University maintains a strict change management policy along with dedicated change management windows for production services; all services at this location must be tested on the associated test Glidein Factory instance and shown to be stable for at least a week before moving to the production system. On the opposite side of the spectrum, the UCSD instance will deploy bleeding-edge software changes at any time, if the benefits are non-negligible, with only a short period of testing on the associated Glidein Factory instance. This has led to most changes being applied initially on the UCSD instance and moved to Indiana University instance after stable operation is confirmed.

To keep the operation load manageable, we have developed a tool that does selective cloning of attributes from one factory to another. This allows the operations team to perform semi-automated synchronizations with periodic corrections to the parameters used to run the tool and only occasional manual adjustments in the actual configuration files of the target instance.

For the most part this tool has served us well. On a typical week, we can synchronize the various instances with no post-clone human intervention at all. There have however been short periods of time when automatic cloning was not possible; this happened when the different instances were running substantially different glideinWMS versions. Major version changes often bring with them a change in the configuration file semantics, which tend to be backwards but not forward compatible; trying to clone an instance running a newer software version into an instance running an older version typically would confuse the tool, requiring substantial manual corrections.

Manual changes are of course error prone; if the needed customization is not done, or is done wrong, the target factory will likely misbehave. However, that's not the only human related problem we may face. For example, there is also the possibility of an operator performing the synchronization in the wrong direction, resulting in lost information, or worse.

3. Social implications of distributed operations

The joint operations team includes people that span nine time zones from Europe to the Western United States, and who are employed by five different organizations. This arrangement has of course advantages, like extended support hours, but it requires careful execution in order to work. This section provides an overview of the nature of Glidein Factory operations, as well as the benefits and drawbacks of executing it with a highly distributed team.

3.1. The nature of Glidein Factory operations

Glidein Factory operations are composed of mainly two sets of tasks; keeping up with the changes in the Grid landscape, and to monitor and debug site-specific problems. We do occasionally provide generic glideinWMS-related know-how to the VO Frontend operators, too.

The Grid landscape is composed of hundreds of independent resource providers, so at least one new service is added and one old deprecated every single week. Since the Glidein Factory insulates the VOs from these changes, the Glidein Factory operators must keep up with them. This includes both noticing the change, as well as making sure that it is both legitimate and, if a new service is added, properly configured. We have developed tools that help us reduce the human effort needed in doing this, but it is far from being fully automated.

Sending provisioning requests to sites is however not sufficient. The Glidein Factory operators are also responsible for monitoring the success rates of such requests, and act if too many are failing. The main challenge is the sheer number of provisioning requests flowing through the system; the UCSD instance alone processes about 50k glideins per day. We have developed tools to filter out the logs of well behaving glideins, and some that flag the logs of the obviously broken ones. However, this still leaves a substantial number of logs that require some human parsing.

3.2. The benefits of distributed operations

The nature of Glidein Factory operations lends itself naturally to be distributed among many independent operators, which brings with it several advantages.

Having the operations team separated by up to nine hours allows for a natural arrangement of the operations in shifts, thus providing up to 17 hours of support a day with each operator working only his regular business hours; the difference in Holiday dates between nations further extends the coverage period. The actual coverage is of course not complete; e.g. the hardware problems are dealt with by the local people only. However, the redundant nature of glideinWMS mitigates this problem significantly, i.e. as long as at least one Glidein Factory is fully functional the served VOs do not suffer, so the perceived coverage is indeed complete.

The increased head count also leads to the establishment of a collective memory. This allows for an easier handling of both personal needs of the various operators, such as vacation and sick days, as well as personnel turnover. This is especially important for CMS, since due to various reasons the CMS operators at CERN can be hired for at most two years; we indeed already had a change of operator recently.

The high turnover expectation lead also to better documentation of both the glideinWMS architecture and the actual operational procedures. UCSD has recently hosted two glideinWMS related workshops, the material of which was used to train both the then active operators and the new hires. As an added bonus, the same training material has also been used as educational material for undergraduate students who occasionally collaborate with us.

3.3. Social problems

Organizing a team of people as a coherent group is always a challenge, but trying to do this when the people are physically located in multiple countries on several continents and employed by different institutions tends to multiply the issues. Nevertheless, the joint, distributed operations is working reasonably well thanks to the arrangements described below.

One major problem of having the team physically distributed across multiple continents is communication. Given there are up to nine hours of time zone difference between various operators, there is effectively no overlap in business hours between them. This means that most communication happens through a bulletin board-like medium, where the operators of one shift leave notes to the next one. We are still experimenting with various tools, so no details about which products are used will be given in this paper.

Text-based, e-communication is of course not ideal, so occasional off-hour communication is still required. One regularly scheduled occurrence is a weekly meeting in which the operators in Europe work an hour later and the operators in California start an hour earlier. This is useful mostly for the establishment of social ties, although sometimes it is also an invaluable forum where operational questions can be addressed. For rare major problems, when either waiting several days is not an option or more time and effort is needed, it is up to the group coordinator to work off-hours and indirectly bridge the gap between the various team members.

Apart from the challenges of making the team work as a cohesive unit, we also noticed effort related problems stemming from the distributed nature of the team. Most operators work only part time on the Glidein Factory operations and spend the rest of the time dealing with other responsibilities. Inevitably, there are situations when there is more work to be done than there is available effort. During these times, operators get pressured on spending more time on one activity at the expense of the others. Local activities often get prioritized over the global Glidein Factory operations due to the closer physical proximity of local leaders and the perceived redundancy of operators due to a relatively large operator pool. Indeed, the work unable to be performed by one operator can occasionally be compensated for by the others in the group. However, it is not sustainable for extended periods of time. So we are making an effort to closely monitor the situation and, if needed, apply any necessary mean to correct it.

4. Conclusions

CMS and OSG are jointly operating four Glidein Factory instances across two continents to serve a large number of scientific communities. The two organizations have chosen the joint operations path to minimize downtime, provide extended hours of human intervention and lower the operation cost for each party. The distributed nature of the joint operations of course introduced also some challenges, some technical and some social.

The experience so far has been quite positive, with no show-stoppers, and with the benefits well outweighing the drawbacks. We have lived through unscheduled local downtimes, three-fold increase in number of served resources, the addition of VOs on different timezones, and personnel turnover without significantly affecting the quality of service. We of course do not claim that everything is going completely smoothly, and we will have to improve on the remaining issues. Nevertheless, the overall experience has been very positive, and we look forward on continuing on this path.

References

[1] Sfiligoi I, Bradley D C, Holzman B, Mhashilkar P, Padhi S and Würthwein F 2009 The Pilot Way to Grid Resources Using glideinWMS *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol.2, pp.428-432. doi: 10.1109/CSIE.2009.950

[2] Sfiligoi I, Würthwein F, Dost J M, MacNeill I, Holzman B and Mhashilkar P 2011 Reducing the Human Cost of Grid Computing With glideinWMS *Proc. of Cloud Computing* 2011, pp. 217-221, ISBN: 978-1-61208-153-3.

[3] The CMS collaboration 2005 The CMS Computing Project Technical Design Report *In proc. of CMS Technical design reports*, pp.1-169. ISBN: 92-9083-252-5.

[4] Pordes I et al. 2007 The open science grid *J. Phys. Conf. Ser.* 78, 012057. doi:10.1088/1742-6596/78/1/012057.

[5] Sfiligoi I et al. 2011 Operating a production pilot factory serving several scientific domains *J. Phys.: Conf. Ser.* 331, 072031, doi:10.1088/1742-6596/331/7/072031

[6] Sfiligoi I, Hass B, Würthwein F and Holzman B 2011 Adapting to the Unknown With a few Simple Rules: The glideinWMS Experience *Proc. Of Adaptive 2011*, pp. 25-28, ISBN: 978-1-61208-156-4.