Operating a production pilot factory serving several scientific domains

I Sfiligoi, F Würthwein, W Andrews, J M Dost, I MacNeill, A McCrea, E Sheripon and C W Murphy

UC San Diego, 9500 Gilman Dr., La Jolla, CA 92093, USA

E-mail: isfiligoi@ucsd.edu

Abstract. Pilot infrastructures are becoming prominent players in the Grid environment. One of the major advantages is represented by the reduced effort required by the user communities (also known as Virtual Organizations or VOs) due to the outsourcing of the Grid interfacing services, i.e. the pilot factory, to Grid experts. One such pilot factory, based on the glideinWMS pilot infrastructure, is being operated by the Open Science Grid at University of California San Diego (UCSD). This pilot factory is serving multiple VOs from several scientific domains. Currently the three major clients are the analysis operations of the HEP experiment CMS, the community VO HCC, which serves mostly math, biology and computer science users, and the structural biology VO NEBioGrid. The UCSD glidein factory allows the served VOs to use Grid resources distributed over 150 sites in North and South America, in Europe, and in Asia. This paper presents the steps taken to create a production quality pilot factory, together with the challenges encountered along the road.

1. Introduction

Leading Grid computing deployments, like the Open Science Grid (OSG) [1] and the European Grid for E-Science (EGEE) [2], are composed of hundreds of independent resource providing sites. Each site is operated by a different team, and uses potentially very different resource management software than any other site, with only a thin software layer common between them. Used directly, these Grid deployments require the scientific users to write very flexible code, severely hindering their productivity as more time is spent on coding and thus less is left for the actual scientific work. To alleviate this problem, several pilot infrastructures, creating an on-demand overlay over Grid resources, have been proposed and implemented, one of them being the glideinWMS [3].

One of the characteristics of glideinWMS is the fact that it splits the services in two categories; one set, known as a glidein factory, is in charge of the interaction with the Grid resources, while the other set, commonly referred to as the user pool, is responsible for the handling of user jobs. The two sets of services can be operated by independent teams, as long as there is a trust relationship established between them. The relationship is many-to-many, with any glidein factory potentially serving many VO frontends, and any VO frontend potentially interacting with many glidein factories.

OSG, together with the HEP experiment CMS[4], is sponsoring a glidein factory instance at the University of California San Diego (UCSD). This glidein factory is configured to interact with a large
subset of Grid resources available in North and South America, in Europe, and in Asia, and is serving several OSG scientific communities, each operating its own user pool.

Section 2 describes the setup of the UCSD glidein factory and its user community. Section 3 describes the operational experience for the period Aug 2009 to Nov 2010. Finally, section 4 provides a summary and the expected short term roadmap.

2. The UCSD glidein factory and its user community

The OSG glidein factory operated at UCSD is currently serving six user communities (also known as Virtual Organizations or VOs), submitting glideins to 177 Grid sites distributed across the North and South America, Europe, and Asia. Of the six VOs, three are responsible for most of the load. They cover a wide range of sciences, with CMS serving physics users, HCC serving math, computer science, and physics users, and NEBioGrid serving biology users. CMS is the one requesting glideins to most sites, 164 in total, while HCC and NEBioGrid use about 20 sites each. Many sites are used by multiple VOs, with NEBioGrid going to only two unique sites and HCC to six. An overview of the above can be found in Table 1.

Table 1. Users of the UCSD glidein factory

| VO       | Scientific field(s)            | Total sites | Unique sites |
|----------|---------------------------------|-------------|--------------|
| CMS      | Physics                        | 164         | 153          |
| HCC      | Math, CS, Physics, Campus Grid | 23          | 6            |
| NEBioGrid| Biology                        | 20          | 2            |

To serve them all, a single server with modest hardware resources is being used. The server used is a dual-CPU (8 total cores) Intel Xeon-based rack server, with 12 GB RAM, 128GB solid state disk (SSD) and Gigabit networking. It is a pretty low-end but reliable machine that is easily handling the load, and with ample headroom for growth. The only unconventional feature is the use of an SSD. The glidein factory handles many small files, generating a large number of pseudo-random IO transactions; traditional hard drives are not capable of handling it in a satisfactory way, while solid state drives easily can.

The glidein factory typically runs between 4k and 8k glideins, with peaks of up to 13k glideins. The glideins are being requested by all three major VOs, each one using between 2k and 5k glideins, with occasional peaks of 8k. The three VOs rarely use peak capacity at the same time, as shown in Fig. 1.

![Figure 1. UCSD glidein factory use by VO](image-url)
Each site uses its own color.

Figure 2. UCSD glidein factory use by site

Most of the time, the glideins are distributed over a large fraction of the available sites; most provide only a small fraction of the total resources, although we do have a few sites that consistently provide a large amount of resources by themselves, as shown in Fig. 2.

3. Operational experience
The OSG glidein factory at UCSD has been in stable operation for about a year, since Fall 2009. From the user point of view, it performed remarkably well, with very few problems reported by the user communities; the pilot paradigm shields the users effectively from most Grid related problems. From the glidein factory operational point of view, however, there are still many problems that need to be addressed: efficiency, Grid resource discovery and shared factory-VO error handling

3.1. Efficiency problems
The average CPU inefficiency over the past six months was about 20%, but with peaks approaching 50%, as shown in Fig. 3.

![Graph showing CPU inefficiency over six months.](image)

While the glideins hide most of the Grid problems from the users, the problems are still there. Some problems are easy to diagnose; the glideins run a set of validation scripts before pulling user jobs, and if any of those fails, the node is deemed broken, no user jobs are pulled and the error is reported to the factory operators. It is up to the factory operator then to notice the problem, apply any mitigating actions, and finally contact the Grid site(s) to get the problem fixed. A year ago, glideinWMS did not provide any tools that would aid in this task; all the work was strictly manual which resulted in problems not being addressed for extended periods of time. Realizing the problem, the UCSD team invested significant development effort, mostly by undergraduate students, to develop better monitoring and reporting tools. While still sub-optimal, these tools have been quite effective, resulting in significant reduction of aggregated waste due to validation errors, as can be seen comparing the periods before and after the summer.

The problems described above are, however, only a fraction of all Grid related problems. Many of the problems are not so trivial, and require human intervention even to be diagnosed; they range from unexpected process terminations, networking problems, to Grid middleware problems. As of now, no automated tools exist to effectively diagnose the root problems of these failure modes, so experienced factory operators are a strong requirement. On top of a large cost in human time, this also makes for a very long ramp-up time for new personnel, making the use of short term operators, such as summer students, practically impossible.
3.2. Grid resource discovery
The glidein factory is configured to submit only to Grid resources (i.e. sites) that are trusted and have been verified to meet the requirements of the user communities they are supposed to serve. The list is kept by the factory itself, and each Grid site potentially needs a slightly different configuration to account for its specific setup. While this approach minimizes Grid errors and maximizes security, it can be quite taxing on the operations team; with hundreds of sites, a rate of even a few per-cent of sites changing every week, results in extensive discovery and testing load.

The problem is compounded by the lack of automated tools for discovery and new site validation in the current glideinWMS code base. Having to do all the operations by hand, the UCSD glidein factory is currently at the limit of the number of sites that can be handled by the current team. The UCSD operators have developed a few ad-hoc solutions for making the situation bearable, but more complete tools are needed in the long term.

3.3. Shared factory-VO error handling
Glidein functionality is a joint responsibility between the factory and the VO part of glideinWMS; the factory provides the base glidein configuration while the VO provides the submission credentials and VO-specific configuration parameters. When there is a problem with a glidein, either part could be at fault, and sometimes it is a combination of the two.

Effective debugging of complex glidein problems thus often requires information from both the factory and the VO counterpart. A big problem in these situation is the dispersed nature of the error logs, part of them being located on the factory and part of them located on the VO nodes, with no automated way to collate them all together. As a result, neither side can autonomously debug such problems, resulting in time consuming interaction between factory operators and VO representatives.

In principle, the relevant error logs should be able to flow between the two sites. The problem is purely the lack of proper infrastructure and tools, so we expect them to be developed in the near future.

3.4. Glidein factory strong points
The problems described above should not induce the reader to believe the glidein factory to be a major failure. Far from it; while the operational load is indeed higher than desired, the overall experience has been very positive.
The glideinWMS software is very stable; it can be left unattended for days at end, with the only ill effect being potentially higher inefficiencies.

It is also extremely easy to maintain. The system has very little state, and very little of it is critical. The system can be restarted at any time, e.g. to apply system and/or application patches, with no ill effect for the served user communities. In case of a catastrophic loss, all the critical state can be swiftly and easily recreated from scratch using only a few configuration files, removing the need for expensive hot-spare solutions. Regular backups of the critical configuration files is all that is needed.

3.5. Crisis moments
In the year of production operation, the UCSD glidein factory experienced only two crisis moments; once reaching the scalability limits of the available hardware, and once discovering a major security flaw. Neither impacted significantly the served user community, but they each did require a major change in the factory setup.

As more users started to use the glidein factory, the load on the factory increased accordingly. For several months, the increase was incremental and it seemed to be perfectly under control. However, once the IOWait exceeded 10%, the situation started to get out of hand; the load started to rise much faster, and within weeks the system was heavily overloaded, to the point that it was difficult to log in. After some investigation, the problem was pinpointed to be the disk access pattern, inherent to the glideinWMS factory implementation. Luckily SSDs had just become affordable enough, so we added one to the server and moved all the factory data to it; the load dropped to a comfortable 3 and stayed in that area since then.

In February 2010, a security assessment of the glideinWMS architecture discovered a major security flaw in the factory installation. Until then, the factory was running all processes under a single UNIX account, but the system was still supposed to be safe due to the handling of access controls internally in the glideinWMS factory code. However, the submission of Grid jobs through GRAM has a side effect of allowing the remote system to access any file owned by the submitting user; while this is not a problem for personal Grid submissions, it is a major security risk for a multi-user portal[5]. The glideinWMS development team fixed the problem by adding effective privilege separation through the use of multiple UNIX accounts, while still running the factory as a non-privileged user. The UCSD installation was upgraded as soon as the code was available; the system has since been operated in a secure way, although debugging has become somewhat harder due to the addition of privilege separation.

4. Summary
The OSG glidein factory at UCSD has been in production for over a year, since Fall 2009. In this period it has been successfully used by several user communities, lowering their operational load and drastically improving their science user experience.

The factory operations experience has also been quite positive. The system is reliable and scalable. The major problem is the excessive operational human load, which is resulting in a reduced overall efficiency of the system, due to the limited human resources available to debug and fix Grid errors.

The OSG glidein factory is expected to continue to operate for the foreseeable future, serving several additional user communities. Reducing the human operation load will however be imperative, and we expect to achieve this with better monitoring and debugging tools.
References

[1] The Open Science Grid Executive Board, “A Science Driven Production Cyberinfrastructure - the Open Science Grid”, OSG Document 976, 2010. http://osg-docdb.opensciencegrid.org/cgi-bin/ShowDocument?docid=976

[2] Laure E and Jones B, “Enabling Grids for e-Science: The EGEE Project”, in “Grid computing: infrastructure, service, and applications”, CRC Press, pp 55-74, April 16, 2009, ISBN: 978-1420067668

[3] Sfiligoi I, Bradley D, Holtzman B, Mhashilkar P, Padhi S and Wuerthwein F, “The Pilot Way to Grid Resources Using glideinWMS”, Proceedings of Computer Science and Information Engineering, 2009 WRI World Congress on, pp. 428-432, March 2009, ISBN: 978-0-7695-3507-4, http://dx.doi.org/10.1109/CSIE.2009.950

[4] The CMS Collaboration, “The CMS experiment at the CERN LHC”, Journal of Instrumentation Volume 3, August 2008. http://dx.doi.org/10.1088/1748-0221/3/08/S08004

[5] Sfiligoi I, “Condor-G: A few lessons learned”, Presentation at Condor Week 2010. http://www.cs.wisc.edu/condor/CondorWeek2010/thursday_condor.html

Acknowledgments

The OSG glidein factory at UCSD would not be possible without major help from the glideinWMS and Condor teams. Many thanks also to the CMS and HCC user pool operators for all the constructive feedback.

This work is partially sponsored by the US Department of Energy under Grant No. DE-FC02-06ER41436 subcontract No. 647F290 (OSG), and the US National Science Foundation under Grants No. PHY-0533280 (DISUN), PHY-0612805 (CMS Maintenance & Operations), and OCI-0943725 (STCI).