Making Science in the Grid World: Using Glideins to Maximize Scientific Output

Igor Sfiligoi, *Fermilab, Batavia, IL, USA*

***Abstract*–The current Grid computing model is based on a set of independent computing clusters, with only a thin common software layer among them and Grid users are expected to adapt. However, scientists are supposed to do science and time spent on computing problems is subtracting time available for scientific thinking. In this paper we present the glidein approach that aims to make the Grid computing as easy as working in a local batch environment.**

I. Introduction

Many modern scientific collaborations need a lot of computing power. However, purchasing, hosting and operating huge numbers of processing units is beyond the capability of most institutes, so computers are being distributed among many locations and organized in a computer Grid[1]. While this allows sites to operate in an optimal mode, it does make the life of the average scientist much harder.

The current Grid computing model is based on a set of independent computing clusters, with only a thin common software layer shared between them. Moreover, short and medium term prospects do not seem to change this paradigm very much. Users submitting their computing workloads to the Grid must thus be flexible and learn how to deal with this heterogeneous environment, as sketched in Fig. 1.

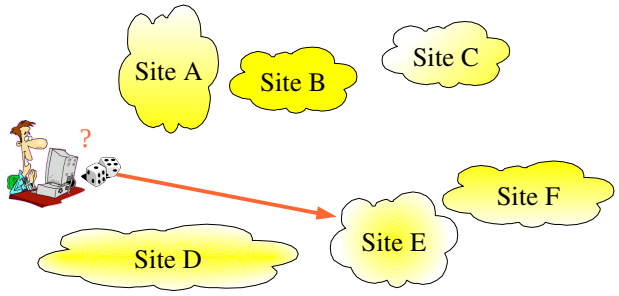

Fig. 1. The Grid environment

However, scientists are supposed to do science, and are not interested in computing problems. Moreover, many scientists use community-contributed software that has been created over several decades of incremental improvements and relies on many external tools, making it very difficult to use in a heterogeneous environment.

A tool that makes the Grid look like a uniform computing cluster is thus needed. In this paper the pilot, or pull approach, based on Condor glideins is presented.

II. The pilot paradigm

The basic principle of the pilot paradigm postulates that user jobs are never sent directly to the Grid; instead pilot jobs are the ones to be submitted directly to the Grid. Only when a pilot job has started, verified and configured a worker node, i.e. a Grid-based compute resource, will a user job be transferred there. See Fig 2.

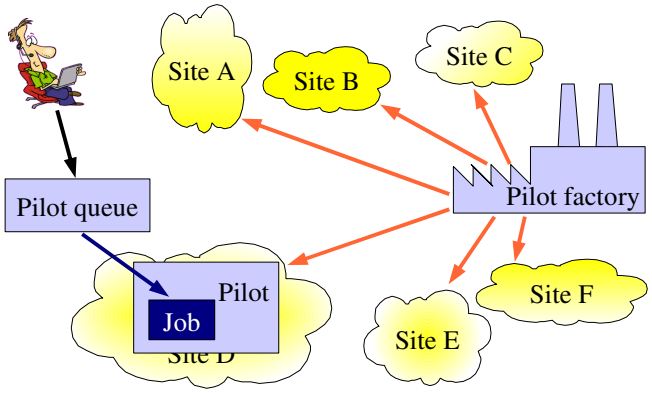

Fig. 2. The pilot paradigm

From the user point of view, the Grid now looks like a local, uniform batch system environment, as shown in Fig. 3.

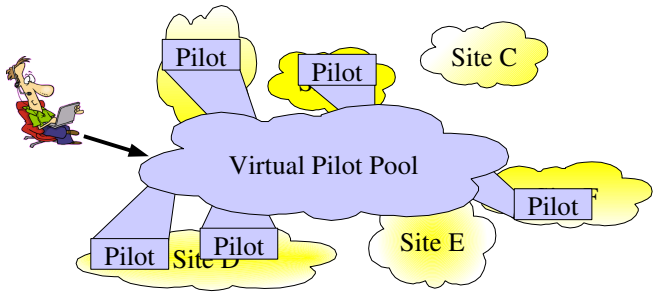

Fig. 3. A virtual private computing pool

The burden of handling heterogeneous environments has thus been shifted from the users to the pilot infrastructure. However, since the pilots are relatively simple, they are much easier to adapt to new environments. Moreover, the pilot factory can be handed over to a computer support group, isolating the scientists from the details of the Grid.

III. The glideinWMS

One general purpose pilot infrastructure currently available is glideinWMS[2,3], which is based on the Condor[4,5] glidein mechanism.

Condor is a widely deployed workload management system used primarily for managing local computing pools. However, its design is based on an intrinsically distributed architecture, making it an ideal candidate for a pilot implementation.

A Condor pool is composed of three logical units:

- a central manager (running a *collector* and a *negotiator*),
- one or more submit nodes (running a *schedd*), and
- one or more execute nodes (running a *startd*),

as shown in Fig. 4.

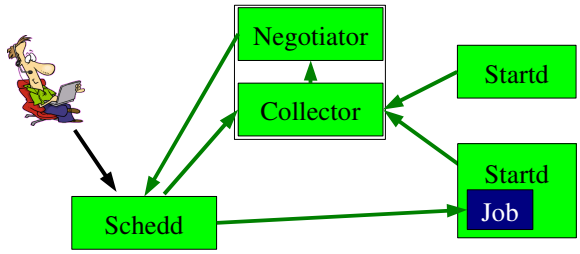

Fig. 4. Condor overview

A *Condor glidein* is a properly configured startd, started as a Grid job. By using glideins, it is possible to create a virtual-private computing pool, as shown in Fig. 5. Users may not even be aware that their jobs are running on the Grid.

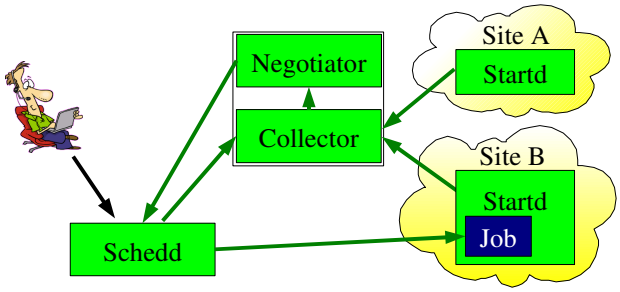

Fig. 5. Condor glideins

GlideinWMS is a software package that provides the tools to create pilot jobs that will start a Condor startd, and a pilot factory that will submit these pilot jobs as needed.

A. GlideinWMS pilot jobs

GlideinWMS pilot jobs are simple scripts that load and launch other executables. Like configuration and software installation scripts, and the Condor daemons. The glideinWMS provides a set of basic configuration scripts, which, for example, set up the security infrastructure for the Condor daemons. Pilot admins can and should add scripts that will configure the environment and possibly install the needed software for their target user community.

B. GlideinWMS pilot factory

The glideinWMS pilot factory is composed of two types of services:

- VO frontends, which monitor user queues and regulate glidein submission rates.
- Glidein factories, which handle glidein configuration and submit the glideins.

The services do not talk directly to each other, but use a Condor collector as a message board, as shown in Fig. 6.

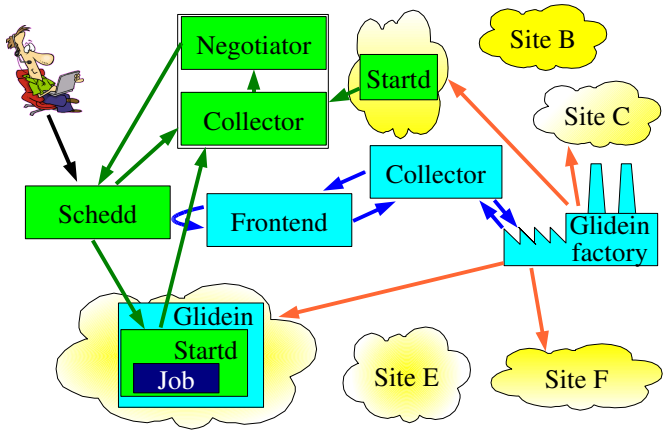

Fig. 6. The glideinWMS

Glidein factories use Condor-G for submission to various Grid sites. The list and characteristics of the Grid sites are determined by the pilot administrators, using their favorite Grid information system, and fed to the factories by means of configuration files. More technical details can be found in [2].

IV. Glidein deployments in High Energy Physics

Glideins, and more generally pilots, have been used in several High Energy Physics (HEP) collaborations.

The first production deployment of a glidein factory was used by CDF at the so-called GlideCNAF[6] in 2005. Since then, CDF has been increasingly relying on glidein-based computing pools[7] for both data reconstruction, organized Monte Carlo production, and user analysis, serving over 400 users.

Based on CDF experience, both CMS and ATLAS have recently started using glideins, although primarily for organized data reconstruction and Monte Carlo analysis.

V. Conclusions

Doing science in the Grid world can be hard, differences between Grid sites do exist. Pilot jobs are an easy way to abstract the Grid computing by making a virtual private computing pool. With careful configuration by pilot

administrators, the obtained pool can actually be completely uniform.

The glideinWMS is an implementation of the pilot infrastructure based on the Condor glidein infrastructure. Glideins have been successfully used by several HEP collaborations; other scientific groups could benefit as well.

Acknowledgment

The author gratefully acknowledges the support of the CMS experiment.

References

[1] I. Foster and C. Kesselman, “The Grid: Blueprint for a New Computing Infrastructure”, San Francisco, CA: Morgan Kaufmann Publishers, 1998.

[2] I. Sfiligoi, “glideinWMS - A generic pilot-based Workload Management System”, *Journal of Physics: Conference Series*, submitted for publication.

[3] http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/

[4] D. Thain, T. Tannenbaum, and M. Livny, "Distributed Computing in Practice: The Condor Experience" *Concurrency and Computation: Practice and Experience*, vol. 17, no. 2-4, pp. 323-356, Feb.-Apr. 2005.

[5] http://www.cs.wisc.edu/condor/

[6] S. Sarkar and I. Sfiligoi, “GlideCNAF : A Purely Condor Glide-in Based CDF Analysis Farm”, *CDF/DOC/COMP UPG/PUBLIC/7630P*, May 2005.

[7] I. Sfiligoi, “CDF Computing”, *Computer Physics Communications*, vol. 177, no. 1-2, pp. 235-238, July 2007.