Transparent Access to Grid-Based Compute Utilities *

Tino Vázquez, Javier Fontán, Eduardo Huedo, Rubén S. Montero, and Ignacio M. Llorente

Departamento de Arquitectura de Computadores y Automática
Facultad de Informática
Universidad Complutense de Madrid, Spain
{tinova, jfontan, ehuedo}@fdi.ucm.es, {rubensm, llorente}@dacya.ucm.es

Abstract. There is normally the need for different grid architectures depending on the constrains: from enterprise infrastructures, to partner or outsourced ones. We present the technology that enables the creation and combination of different grid architectures, offering the possibility to create federated grids that can be easily deployed as compute utilities on grid infrastructures based on the Globus Toolkit. Then, taking advantage of the uniform and standard interface of the proposed grid-based compute utilities, we present a way to enable transparent access to virtualized grids from local computing infrastructure managed by Sun Grid Engine, although similar approaches could be followed for other resource managers.

1 Introduction

Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations", as stated by Ian Foster in [1]. Therefore, due to its multi-institutional nature, a grid is built on top of potentially heterogeneous resources.

Furthermore, due to socio-political landscape, there is normally the need for different grid architectures depending on the constraints. We can differentiate three kinds of arrangements. There are, firstly, enterprise grids, contained within the boundaries of one organization. Improves internal collaboration and enables a better return from IT investment, coming from a more optimal resource utilization. Secondly, we have partner grids, created using the resources of two or more organizations and /or scientific communities. Partner grids allows the access to higher computing performance to satisfy peak demands and also provide support to face collaborative projects. As a third architecture we find outsourced grids,

* This research was supported by Consejería de Educación of Comunidad de Madrid, Fondo Europeo de Desarrollo Regional (FEDER) and Fondo Social Europeo (FSE), through BioGridNet Research Program S-0505/TIC/000101, by Ministerio de Educación y Ciencia, through research grant TIN2006-02806, and by European Union, through BEinGRID project EU contract IST-2005-034702.

although not a reality right now, it has been predicted that dedicated service providers will organize and maintain cyberinfrastructures to supply resources on demands over the Internet. This business model will be feasible when growing network capacity is enough to allow business and consumers to draw their computing resources from outsourced grids apart from enterprise and partner grids [2].

These different grid perspectives need the technology to ensure their deployment in a way that preserves the autonomy of each organization, while offering at the same time overall security (critical if organizations are planning to outsource sensitive data) and scalability (an inherent requirement of grid infrastructures).

In this paper we present the technology that enables the creation and combination of different grid architectures, i.e. we offer the possibility to create federated grids, that can be easily deployed as compute utilities on grid infrastructures based on the Globus Toolkit (GT). Furthermore, taking advantage of the uniform and standard interface of these grid-based compute utilities, we present a way to enable transparent access to virtualized grids from local computing infrastructure managed by Sun Grid Engine, although similar approaches could be followed for other resource managers.

The rest of this paper is organized with the following structure. Section 2 introduces and explains the "grid for utility" concept. Section 3 presents the GridGateWay solution for building federated grid infrastructures based on the Globus Toolkit and the GridWay Metascheduler. It also describes a testbed for a federated grid and provides some results. In Section 4, we present a complementary component to enable access to virtualized grids using the Sun Grid Engine *Transfer Queue to Globus*. To finish, Section 5 presents some conclusions and our plans for future work.

2 Grid for Utility

Originally, the concept of grid was developed from the power grid. In short, the idea behind a grid is to be able to plug a computing resource so it gets added to the total computer power with the minimum possible configuration. It also will be able to use the grid computing capability as a single computer, transparently to the user, the same way an electric device can be plugged to a power socket to draw electricity from it.

Utility computing is a service provisioning model which will provide adaptive, flexible and simple access to computing resources, enabling a pay-per-use model for computing similar to traditional utilities such as water, gas or electricity. It pursues an scenario where computing power is given or taken (consumer/producer paradigm) from the different components that conform the grid based on the demand and offer of this computing power at a given time. This posses different problems, ranging from accounting and billing to scalability and security.

One desired outcome of the "grid for utility" idea is to allow the creation of the outsourced grids identified above. It will indeed provide means to *sell* computing power to anyone interested on it. Obviously, security is essential, as sensitive data may be outsourced. Several efforts have been undergone to tackle this problem [3, 4].

3 The GridGateWay Solution for Grid Federation

In order to leverage the practical implementation of the concepts of the previous section we provide means to build grid federations, although admittedly only in a somewhat still crude form.

A GridGateWay provides the standard functionality required to implement a gateway to a federated grid. The main innovation of our model is the use of Globus Toolkit services to recursively interface to the services available in a federated Globus-based grid. Such a combination allows the required virtualization technology to be created in order to provide a powerful abstraction of the underlying grid resource management services. The GridGateWay acts as a utility computing service, providing a uniform standard interface based on Globus protocols and services for the secure and reliable submission and control of jobs, including file staging, on grid resources.

A GridGateWay offers the possibility of encapsulating a virtualized grid inside a WS (Web Service) GRAM (Globus Resource Allocation and Management), using GridWay as the underlying LRMS (Local Resource Management System). To interface GridWay through GRAM, a new scheduler adapter has been developed along with a scheduler event generator. Also, a scheduler information provider has been developed in order to feed MDS (Monitoring and Discovery Service) with scheduling information.

The grid hierarchy in our federation model is clear. An enterprise grid, managed by the IT Department, includes a GridGateWay to an outsourced grid, managed by the service provider. The outsourced grid provides on-demand or pay-per-use computational power when local resources are overloaded. This hierarchical grid organization may be extended recursively to federate a higher number of partner or outsourced grid infrastructures with consumer/provider relationships. Figure 1 shows one of the many grid infrastructure hierarchies that can be deployed with GridGateWay components. This hierarchical solution, which resembles the architecture of the Internet (characterized by the end-to-end argument [5] and the IP hourglass model [6]), involves some performance overheads, mainly higher latencies, which have been quantified elsewhere [7]. The access to resources, including user authentication, across grid boundaries is under control of the GridGateWay service and is transparent to end users. In fact, different policies for job transfer and load balancing can be defined in the GridGateWay. The user and resource accounting and management could be performed at different aggregation levels in each infrastructure.

There are several characteristics of this architecture that we identify as advantages. First one, it is completely based on standard protocols and uses a *de facto* standard grid middleware as Globus Toolkit is. Secondly, it doesn't require for extra deployment for clients as it uses a GRAM interface. And, as a third

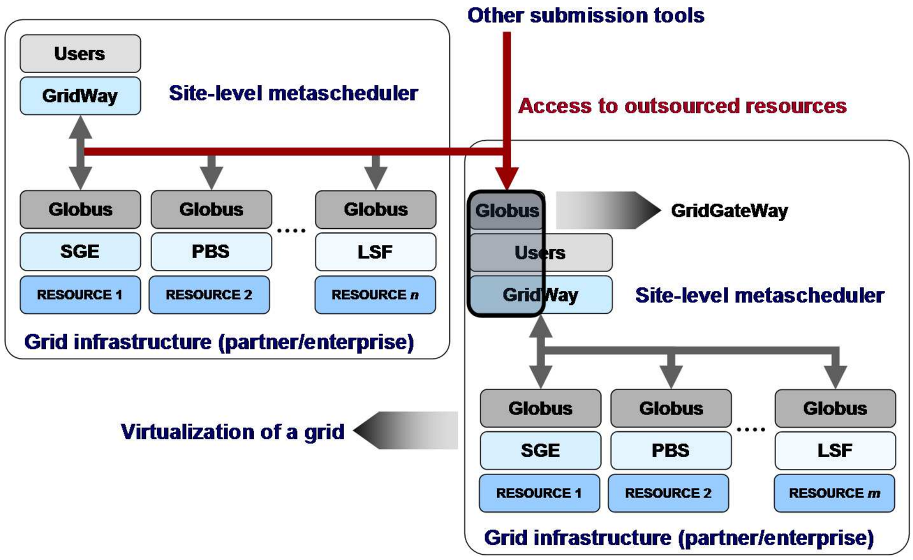

Fig. 1. GridWay encapsulated under a Globus WS GRAM service.

advantage, also being its main objective, it allows for a complex grid federation, based on recursive architecture.

In order to test the GridGateWay we used the following testbed. On one server (cepheus) we have an instance of GridGateWay that virtualizes a partner grid, in this case is going to be the *fusion* Virtual Organization (VO) of EGEE. Also, we have a local cluster managed by GridWay, that also has access to the EGEE partner grid. As it can be seen in Figure 2, cepheus appears as having a GW (GridWay) LRMS and it just displays the number of total and free nodes, and no more configuration details, it is in fact virtualizing the partner grid. The users doesn't know the details of the grid behind cepheus Then again, he really doesn't need to know, but if needed a host information provider could be developed.

```
tinova@draco:~$ gwhost
HID OS              ARCH   MHZ %CPU MEM(F/T)      DISK(F/T)   N(U/F/T) LRMS HOSTNAME
0   Linux2.6.16-2-6 x86    3216    0 831/2027 114644/118812      0/0/1 Fork cygnus.dacya.ucm.e
1   Linux2.6.16-2-a x86_6 2211  100 671/1003   76882/77844      0/2/2 SGE  aquila.dacya.ucm.e
2   Linux2.6.16-2-6 x86    3215    0 153/2027 114541/118812      0/0/1 Fork draco.dacya.ucm.es
3   Linux2.6.16-2-a x86_6 2211  100 674/1003   76877/77844      0/2/2 PBS  hydrus.dacya.ucm.e
4   NULLNULL        NULL      0    0      0/0           0/0 6/665/1355 GW   cepheus.dacya.ucm.
```

Fig. 2. Resources in UCM (enterprise grid) as provided by GridWay **gwhost** command. cepheus hosts a GridGateWay and appears as another resource with GW (GridWay) as LRMS (Local Resource Management System), only providing information about its status (used, free and total nodes).

Figure 3 shows the throughput achieved with this configuration. We used a small parameter sweep application composed of 100 tasks, each one needing about 10 seconds to execute on a 3.2 GHz Pentium 4 and about 10 KB of input/output data. UCM resources executed 73 jobs, thus EGEE resources executed only 27 jobs, because their response time was much higher than that of UCM resources (the first job executed on EGEE completes 7.45 minutes after the experiment starts). This was in part due to network latencies and the use of the GridGateWay, but mainly because EGEE resources were heavily loaded with production jobs and, therefore, the submitted jobs suffered from high queue wait times. UCM and EGEE respectively contributed with 297.96 and 121.65 jobs/hour to the aggregated throughput of 408.16 jobs/hour. Results with this configuration has been proved to give a close to optimal performance, according to predictions made using a performance model for federated grid infrastructures [8].

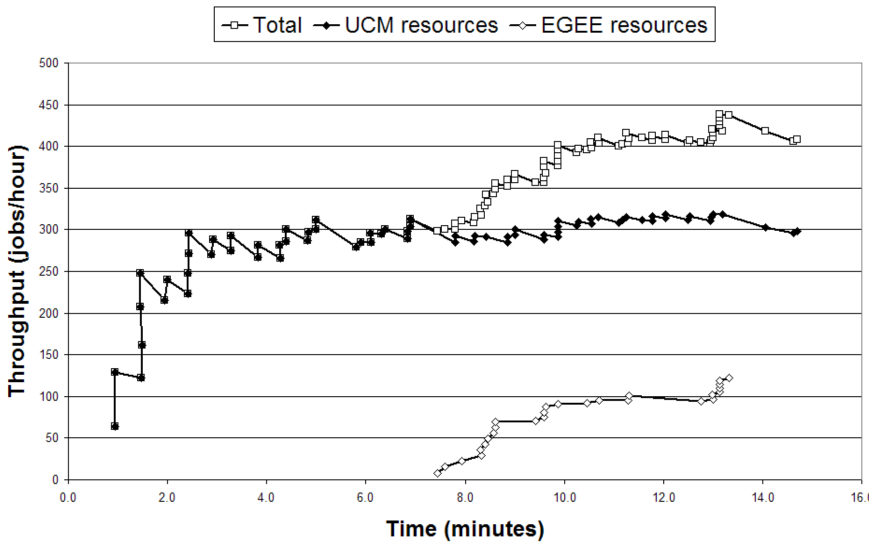

Fig. 3. Throughput achieved in UCM when provisioning resources from EGEE (*fusion* VO) through a GridGateWay.

4 Sun Grid Engine *Transfer Queue to Globus*

An interesting application of the GridGateWay solution is the outsourcing of computing power to sustain peak demands. This can be done using a pure GridWay based solution, where a local GridWay instance manages the local resources and uses a GridGateWay to satisfy computing power demand when the local cluster/grid is overloaded.

We are aware that in the real, non academic, world (and even there) there is a huge inertial component that makes difficult to assume changes. With that in mind, we are committed to provide solutions for a wide range of the current possible configurations, so as to allow our concept of grid federation without any major change in the available infrastructure (i.e. both the grid middleware or the underlying LRMS). GridWay already offers the technology to construct a grid out of heterogeneous sub-grids, based on different flavors of Globus (pre-WS and WS) [9, 10]. On top of that, we now offer the possibility of accessing a federated grid not only from GridWay, but also from a LRMS like Sun Grid Engine (SGE).

Sun Grid Engine[1] is an open source batch-queuing system, supported by Sun Microsystems[2]. This LRMS offers the possibility to define custom queues, through the development of scripts to submit, kill, pause and resume a job. It is already available from the vendor a *Transfer Queue to Globus* [11], but just for GT2, with pre-WS interfaces. As stated above, GridGateWay uses WS GRAM interface, so a Transfer Queue to GT4 has been developed[3]. What we obtain from this component is the ability to submit jobs to a virtualized grid (utility computing service) from a local, well known, LRMS in a transparent way for the user.

Figure 4 shows SGE with one transfer queue configured. In the SGE domain we find the normal SGE local queues and a special grid queue, that enables SGE to submit jobs to a WS GRAM service or to GridWay. This WS GRAM service can, in turns, virtualize a cluster (with several known LRMS) or it can encapsulate a GridWay (thus conforming a GridGateWay) that gives access to a Globus-based outsourced infrastructure. The special grid queue can be configured to be available only under certain special conditions, like peak demands or resource outage.

5 Conclusions and Future Work

In this paper we have shown a utility computing solution that can be deployed on any grid infrastructure based on existing Globus Toolkit components. This solution enables the creation of enterprise, partner and outsourced grids using automated methods on top of *de facto* grid standards in a simple, flexible and adaptive way. Moreover, we have developed components that intend to make the deployment as simple as possible, ideally requiring the minimum changes to the existing local infrastructure. As of now, we just support one LRMS, but is in the scope of the Grid4Utility project[4] to provide a wider range of components to handle the broader set of pre-existing configurations as possible. Other aims of

[1] http://gridengine.sunsource.net
[2] http://www.sun.com
[3] Available at http://www.grid4utility.org/#SGE
[4] http://www.grid4utility.org

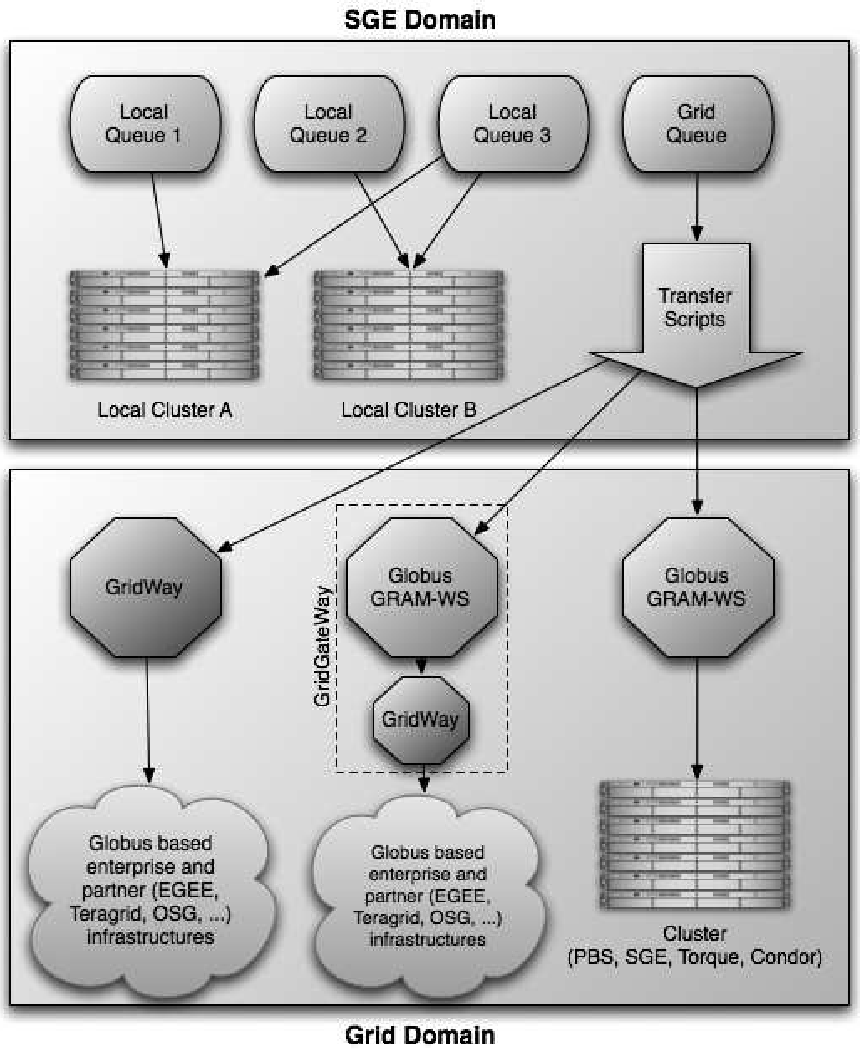

Fig. 4. Grid federation achieved through GridGateWay and accessible through a local LRMS, in this case, SGE.

the project are the development of new components for scheduling, negotiation, service level agreement, credential management and billing.

Acknowledgments

This work makes use of results produced by the Enabling Grids for E-sciencE project, a project co-funded by the European Commission (under contract number INFSO-RI-031688) through the Sixth Framework Programme. EGEE brings together 91 partners in 32 countries to provide a seamless grid infrastructure available to the European research community 24 hours a day. Full information is available at http://www.eu-egee.org.

We want to acknowledge all institutions belonging to the *fusion* VO[5] of the EGEE project for giving us access to their resources.

References

1. Foster, I.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. High Performance Computing Applications **15**(3) (2001) 200–222
2. Llorente, I.M., Montero, R.S., Huedo, E., Leal, K.: A Grid Infrastructure for Utility Computing. In: Proc. 15th IEEE Intl. Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), IEEE CS Press (2006)
3. Humphrey, M., Thompson, M.: Security Implications of Typical Grid Computing Usage Scenarios. Cluster Computing **5**(3) (July 2002) 257–264
4. Azzedin, F., Maheswaran, M.: Towards Trust-Aware Resource Management in Grid Computing Systems. In: 2nd IEEE/ACM International Symp. Cluster Computing and the Grid (CCGrid 2002). (2002) 452–457
5. B. Carpenter, E.: Architectural Principles of the Internet. RFC 1958 (June 1996)
6. Deering, S.: Watching the Waist of the Protocol Hourglass. In: 6th IEEE International Conference on Network Protocols. (1998)
7. Vázquez, T., Huedo, E., Montero, R.S., Llorente, I.M.: Evaluation of a Utility Computing Model Based on the Federation of Grid Infrastructures. In: Proc. 13th Intl. Conf. Parallel and Distributed Computing (Euro-Par 2007). (2007) in press.
8. Vázquez, T., Huedo, E., Montero, R.S., Llorente, I.M.: A Performance Model for Federated Grid Infrastructures. Submitted to the 8th IEEE/ACM Intl. Conf. Grid Computing (Grid 2007) (2007)
9. Vázquez-Poletti, J.L., Huedo, E., Montero, R.S., Llorente, I.M.: Coordinated Harnessing of the IRISGrid and EGEE Testbeds with GridWay. J. Parallel and Distributed Computing **66**(5) (May 2006) 763–771
10. Huedo, E., Montero, R.S., Llorente, I.M.: A Modular Meta-Scheduling Architecture for Interfacing with Pre-WS and WS Grid Resource Management Services. Future Generation Computer Systems **23**(2) (February 2007) 252–261
11. Seed, T.: Transfer-queue Over Globus (TOG): How To. Technical Report EPCC-SUNGRID-TOG-HOW-TO 1.0, EPCC, The University of Edinburgh (July 2003) Available at http://gridengine.sunsource.net/download/TOG/tog-howto.pdf.

[5] http://grid.bifi.unizar.es/egee/fusion-vo