A Performance Model for Federated Grid Infrastructures *

Constantino Vázquez Eduardo Huedo Rubén S. Montero
Ignacio M. Llorente
Departamento de Arquitectura de Computadores y Automática
Facultad de Informática, Universidad Complutense de Madrid, Spain

E-mail: {tinova, ehuedo}@fdi.ucm.es, {rubensm, llorente}@dacya.ucm.es

Abstract

A performance model, previously proposed to characterize the performance of grid infrastructures, is extended to evaluate federations of grids by aggregating their performance parameters. These parameters can then be used to develop scheduling policies based on them. The new model can be used to take scheduling decisions based on them and hence to aid in the development of scheduling policies. The model has been validated using the performance results obtained in the execution of a high throughput computing application on an enterprise grid composed of Globus Toolkit Web Service resources and a GridGateWay giving access to gLite resources from the EGEE infrastructure.

1. Introduction

The deployment of *enterprise grids* enables diverse resource sharing to improve internal collaboration and achieve a better return from IT investment. On the other hand, *partner grids* of several scales are being mainly deployed within the context of different research projects, whose final goal is to provide large-scale, secure and reliable sharing of resources among partner organizations and supply-chain participants. Such partner grids allow access to a higher computing performance to satisfy peak demands and also provide support to face collaborative projects.

Different studies suggest that growing network capacity will allow businesses and consumers to draw their computing resources from outsourced grids apart from enterprise and partner grids. Therefore, future grid infrastructures will be composed of several enterprise, partner and outsourced grids, showing a hierarchical architecture, which preserves the autonomy of each organization and improves both security and scalability.

In a previous work [1], we have proposed a solution for federating grids that can be deployed on a grid infrastructure based on the Globus Toolkit (GT). A similar approach for the federation of grid infrastructures has been previously applied to meet LCG and GridX1 infrastructures [2], hosting a GridX1 user interface in a LCG computing element. However, this solution imposes software, middleware and network requirements on worker nodes. The Globus project is also interested in this kind of *recursive* architectures, and is working on Bouncer [3], which is a Globus job forwarder initially conceived for federating TeraGrid and Open Science Grid infrastructures. There are other approaches to achieve middleware interoperability, for example between UNICORE and Globus [4] and between gLite and UNICORE.

A lot of work has been done to benchmark, model, predict or even control the performance of grid infrastructures. Previous works consider the characteristics of grid infrastructures and applications, like dynamism, heterogeneity or adaptation, but few of them deal with the aggregation of performance models (or their parameters) to model the performance of federated grid infrastructures [5, 6].

In this work we extend a performance model, previously proposed, to characterize federated grid infrastructures. Then we apply it to a federation of grid based on GridGateWays technology.

The rest of this paper is as follows. Section 2 describes the performance model ans its extensions to deal with federated grids. Section 3 presents a solution for building federated grid infrastructures based on the Globus Toolkit and the GridWay Metascheduler by means of GridGateWays. Section 4 presents some experimental results to validate the performance model and, finally, Section 5 provides some conclusions and plans for future work.

*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.

2. Performance Model

In the execution of High-Throughput Computing applications, a grid can be considered, from the computational point of view, as an array of *heterogeneous* processors [7]. Therefore, the number of tasks completed as function of time is given by the following equation:

$$n(t) = \sum_{i \in G} N_i \left\lfloor \frac{t}{T_i} \right\rfloor \quad (1)$$

where N_i is the number of processors of resource *i* in a grid *G* that can compute a task in T_i seconds.

The best characterization of a grid can be obtained if we take the average behavior of the system. The next formula represents *n(t)* using the r_∞ and $n_{1/2}$ parameters defined by Hockney and Jesshope [8]:

$$n(t) = r_\infty t - n_{1/2} \quad (2)$$

These parameters are called:

- Asymptotic performance (r_∞): the maximum rate of performance in tasks executed per second. In the case of an homogeneous array of N processors with an execution time per task T, we have $r_\infty = N/T$.
- Half-performance length ($n_{1/2}$): the number of tasks required to obtain the half of asymptotic performance. This parameter is also a measure of the amount of parallelism in the system as seen by the application. In the homogeneous case we obtain $n_{1/2} = N/2$, so $2 \cdot n_{1/2}$ represents the apparent number of –homogeneous– processors (thus, equal to N in the homogeneous case).

These parameters can be obtained through intrusive or non-intrusive benchmarking, as proposed in [7].

The following equation defines the performance of the system (tasks completed per time) on actual applications with a finite number of tasks based on the linear relation of Eq. 2:

$$r(n) = n(t)/t = \frac{r_\infty}{1 + n_{1/2}/n} \quad (3)$$

The simplicity of this model allows the characterization of grid infrastructures by using a very reduced set of metrics (just the r_∞ and $n_{1/2}$ parameters), while fully capturing their behavior. Moreover, it is straightforward to obtain these parameters in the case of federated grid infrastructures by just adding the parameters of the grid infrastructures building up the federation. Let assume that FG is the set of federated grids, each one characterized by its linear performance model (i.e. r^i_∞ and $n^i_{1/2}$, $\forall i \in FG$):

$$r_\infty = \sum_{i \in FG} r^i_\infty$$
$$n_{1/2} = \sum_{i \in FG} n^i_{1/2} \quad (4)$$

The suitability of this approach will be evaluated in section 4 with the GridGateWay architecture. It is worth mentioning that this model can be applied to the federation architecture.

As another application of this model, it is possible to obtain the optimum number of jobs that should be submitted to each infrastructure in order to achieve the minimum computational time. We will formulate the general problem as follows. There are J jobs to be processed, each job can be processed in any grid of the federation FG. We assume that each grid can process j_i jobs simultaneously. We are interested in optimize the maximum completion time criterion (*makespan*), C_{max}. The time needed for the grid i to process the jobs assigned to it, can be easily derived from Eq. 2, therefore the makespan yields:

$$C_{max} = \max_{i \in FG} \frac{j_i + n^i_{1/2}}{r^i_\infty} \quad (5)$$

This problem can be formulated as an integer linear programming problem as follows:

$$\begin{aligned} &\textbf{minimize} \\ &\quad C_{max} \\ &\textbf{subject to} \\ &\quad \sum_{i \in FG} j_i - J = 0; \\ &\quad j_i \geq 0 \;\; \forall i \in FG \end{aligned} \quad (6)$$

We will apply this optimization problem in the experiments section.

3. Federation of Grid Infrastructures through GridGateWays

The Globus Toolkit [9] provides a uniform, secure and reliable interface to many different DRM (Distributed Resource Manager) systems, allowing the development of grid workload managers that optimize the use of the underlying computing platforms.

Our solution uses Globus Toolkit services to recursively interface to the services available in a federated Globus based grid. A set of Globus Toolkit services hosting a GridWay Metascheduler[1] [10], what we call a *GridGateWay*, provides the standard functionality required to implement a gateway to a federated 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 computing service, providing a uniform standard interface based on Globus interfaces, protocols and services for the

[1] http://www.gridway.org

secure and reliable submission and control of jobs, including file staging, on grid resources.

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. This solution involves some performance overheads, mainly higher latencies, which have been quantified before [1].

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.

4. Experiments

We set up a federated grid infrastructure where a client runs an instance of the GridWay Metascheduler interfacing local resources in an enterprise grid (UCM, in this case), based on GT4 Web Services interfaces, and a GridGateWay that gives access to resources from a partner grid (*fusion* VO of EGEE, in this case), based on GT pre-Web Services interfaces found on gLite 3.0. The simultaneous use of different adapters to access multiple partner grid infrastructures has been demonstrated before [11]. In fact, that could be an alternative for the coexistence of different grid infrastructures, although based on distinct middleware (GT2, GT4, LCG, gLite...).

In this configuration, draco is the client machine, providing access to the enterprise grid, and cepheus is the GridGateWay, providing access to the partner grid (see Figure 1 (top)). Notice that, in this case, the GridGateWay is hosted in the enterprise. However, in a typical business situation it would be hosted in the partner infrastructure, enforcing its security and usage policies and providing services for accounting and billing.

In the case of EGEE resources, and in order to avoid saturation of the testbed (which is supposed to be at production level) with our tests, we limited the number of running jobs in the same resource to 10, and the number of running jobs belonging to the same user to 30. This kind of usage policies can be defined by the service provider in the instance of GridWay running in the GridGateWay.

The application used was the distributed calculation of the π number as $\int_0^1 \frac{4}{1+x^2}dx$. Each task computes the integral in a separate section of the function and all results are finally added to obtain a good approximation of π. The required computational time is about 10 seconds per task on a 3.20GHz Pentium 4; and file transfer cost, including the executable and the standard input/output/error streams, is about 10KB per task. Is worth noting here that the fact that the π number calculation is a toy program doesn't affect it's execution and transfer times, so it doesn't affect the validity of the model.

Table 1. Parameters of the performance model for each experiment.

	Experiment 1		Experiment 2	
	r_∞	$n_{1/2}$	r_∞	$n_{1/2}$
UCM	171.75	5.01	322.16	2.93
fusionVO	236.83	17.24	224.61	24.65
Experimental	395.63	20.27	458.77	12.32
Estimated	408.58	22.25	546.77	27.57

Figure 2 (bottom) shows the dynamic throughput achieved in UCM when provisioning partner resources from EGEE (*fusion* VO) through a GridGateWay. The number of tasks submitted was 100. Besides network connection and the use of a GridGateWay, there are differences in latency between enterprise and partner resources due to the production status of partner resources, as they are under heavy usage. The aggregated throughput achieved in the first experiment was 347.5 jobs/hour. Enterprise and partner resources executed almost the same number of jobs (49 and 51 jobs, respectively) and contributed almost equally (170.3 and 181.4 jobs/hour, respectively) to the aggregated throughput. In the second experiment, the aggregated throughput achieved was higher, 408.16 jobs/hour, due to a better response of enterprise resources, which executed 73 jobs contributing with 297.96 jobs/hour to the total throughput. Therefore, a lower usage of partner resources was needed, executing only 27 jobs and contributing with 121.65 jobs/hour to the final throughput.

Figure 2 shows the result of applying the performance model to the above experiments, while Table 1 summarize the r_∞ and $n_{1/2}$ parameters obtained.

In the first experiment, the aggregated model is almost equal to the one obtained with the experimental data. In the second experiment, they are not so equal, because the model doesn't capture well the behavior of the partner infrastructure. Due to the low number of jobs submitted to the partner grid, the steady state of testbed saturation is not reached and, thus, the testbed seems to have a bigger number of more heterogeneous resources (notice the high value of $n_{1/2}$, almost equal to the number of submitted jobs). This makes the aggregated model to have a higher r_∞, but also a higher $n_{1/2}$. Nevertheless, the problem is in the input data,

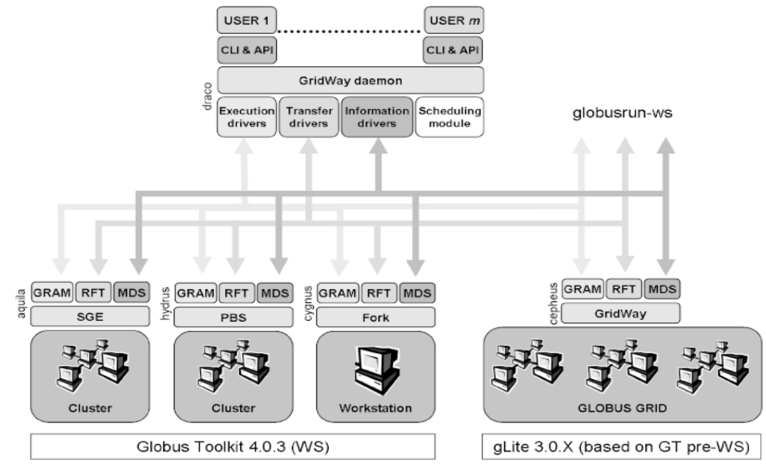

Figure 1. Experimental environment.

Table 2. Optimum and actual number of jobs submitted to each infrastructure.

	Optimum		Actual	
	UCM	fusionVO	UCM	fusionVO
Exp. 1	54	46	51	49
Exp. 2	72	28	73	27

not in the performance model.

A comparison between the optimum number of jobs needed to minimize the optimum makespan predicted by Eq. 6 (with $FG = \{UCM, fusionVO\}$) and the actual number of jobs submitted by GridWay to each infrastructure is shown on Table 2. It can be seen that GridWay numbers in both experiments are very close to the optimum ones.

5. Conclusions and Future Work

We have shown the suitability of the proposed performance model for obtaining a straightforward characterization of federating grid infrastructures by applying it to our solution based on GridGateWays. However, we have identified a limitation of the model when there are not enough samples.

We have applied the performance model to data obtained before-hand in order to validate the model, but the final aim is to automatically compute the parameters of the model (e.g. by periodically executing a given benchmark) in order to take scheduling decisions based on them. Therefore, future work includes the development of scheduling policies considering these parameters to reduce the total execution time of a whole workload, while also taking into account resource ownership, to maximize the use of local resources and so reduce costs.

Finally, we want to extend the experimental scenario with more enterprise, partner and outsourced grids. For the latter, economic models should be proposed and, due to the complexity of such an infrastructure, the use of simulation tools will be of great help. These new ideas, as well as new components for scheduling, negotiation, service level agreement, credential management, and billing, are currently being developed in the context of the Grid4Utility project[2].

6. 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[3] of the EGEE project for giving us access to their resources.

References

[1] Vázquez, T., Huedo, E., Montero, R.S., Llorente, I.M.: Evaluation of a Utility Computing Model Based on the Federation of Grid Infrastructures. In: 13th Intl Conference Parallel and Distributed Computing (August 2007), LNCS (2007)

[2] http://www.grid4utility.org
[3] http://grid.bifi.unizar.es/egee/fusion-vo

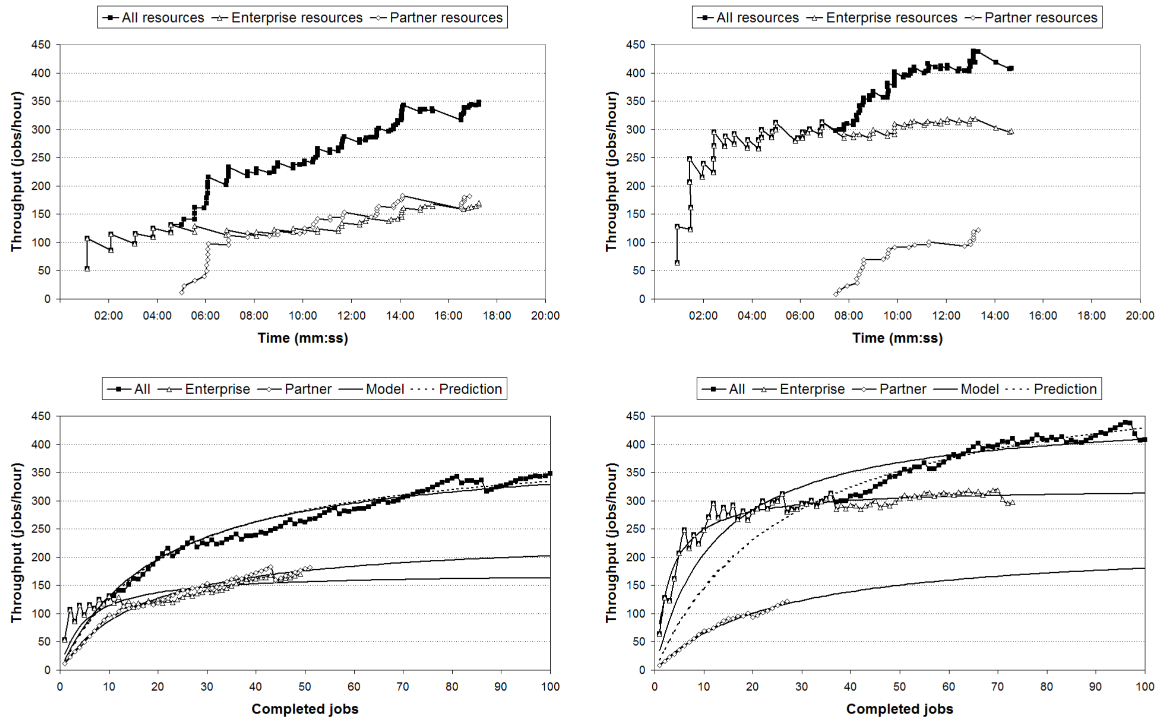

Figure 2. Dynamic throughput achieved in the first (left) and second (right) experiment (top). Performance model and estimation applied to both experiments (bottom).

[2] Walker, R., Vetterli, M., Impey, R., Mateescu, G., Caron, B., Agarwal, A., Dimopoulos, A., Klektau, L., Lindsay, C., Sobie, R., Vanderster, D.: Federating Grids: LCG Meets Canadian HEPGrid. In: Proc. Computing in High Energy and Nuclear Physics (CHEP 2004). (2004)

[3] Baumbauer, C., Goasguen, S., Martin, S.: Bouncer: A Globus Job Forwarder. In: Proc. 1st TeraGrid Conf. (2006)

[4] Breuer, D., Wieder, P., van den Berghe, S., von Laszewski, G., MacLaren, J., Nicole, D., Hoppe, H.C.: A UNICORE-Globus Interoperability Layer. Computing and Informatics **21** (2002) 399–411

[5] Nasri, W., Steffenel, L.A., Trystram, D.: Adaptive Performance Modeling on Hierarchical Grid Computing Environments. In: 7th IEEE Intl Symp on Cluster Computing and the Grid, CCGRID. (2007) 505–512

[6] Cao, J., Kerbyson, D., Nudd, G.: Performance evaluation of an agent-based resource management infrastructure for grid computing. In: 1st IEEE/ACM Intl Symp on Cluster Computing and the Grid, CCGRID. (2001) 311–318

[7] Montero, R.S., Huedo, E., Llorente, I.M.: Benchmarking of High Throughput Computing Applications on Grids. Parallel Computing **32**(4) (April 2006) 267–279

[8] Hockney, R., Jesshope, C.: Parallel Computers 2:Architecture Programming and Algorithms. Adam Hilgee Ltd. (1998)

[9] Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Proc. IFIP Intl. Conf. Network and Parallel Computing (NPC 2005)

[10] Huedo, E., Montero, R.S., Llorente, I.M.: A Framework for Adaptive Execution on Grids. Software - Practice and Experience **34**(7) (2004) 631–651

[11] 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