Simulation of Resource Monitoring and Discovery in Grids

D. González Márquez, E. Mocskos, D. Fernández Slezak, P. Turjanski

Laboratorio de Sistemas Complejos, Departamento de Computación,
Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires
Buenos Aires (C1428EGA), Argentina

Abstract. The proliferation of huge computer centers with thousands of processors available, summed up with the new computing paradigms as cloud computing, make distributed environment a maze of new perspectives. Dynamics of resource information is increasing in complexity, and nowadays is almost impossible to be captured using a static hierarchy. Consequently, we focused on developing new tools to attack this new challenges. Grid-Matrix is a framework with an user-friendly GUI, based on the famous grid simulator SimGrid. It is focused on the analysis of information propagation policies, including automatic generation of very varied network topologies, and many different policies scenarios. As a case study, we present the generation of clique and ring networks under three different policies: hierarchical, super-peer and random. These preliminary results shows the potential of this powerful framework, plenty of features for the profound analysis of information propagation.

1 Introduction

With the advances in microprocessors, storage capacity, networking speed and telecommunications in general the world has witnessed an unprecedented progress in information technology, enabling to satisfy the increasing demands of computing power required by scientific and commercial applications.

Many High Performance Computing (HPC) centers are being built and Grid Computing arises as the new computing paradigm[15, 6, 18, 20] for interconnection and synchronized interaction between them, providing the infrastructure required for sharing diverse set of resources, including desktops, computational clusters, supercomputers, storage, data, sensors, applications, and on-line scientific instruments[34].

In this technology, Globus Toolkit 4 (GT4)[19, 1] is one of the most used tool for creating grid environments over different operating systems and computer architectures; which consists of a set of services and applications for managing the different grid features following a layered design.

One of the key features needed in Grid Computing is to maintain up-to-date information related to resources and services in an heterogeneous and highly dynamic environment[7]. The component responsible for obtaining, distributing, indexing and archiving information about the configuration and state of services

and resources is called the Grid Resource Information Service (GRIS). Standard centralized organization approach for the GRIS has several drawbacks[34]:

- Highly prone to a single point of failure.
- Lacks scalability.
- High network communication cost in the links leading to the information server (i.e. network bottleneck, congestion).
- Centralized servers may become bottlenecks or points of failure.

Monitoring and Discovery System (MDS)[4, 14], the specialized services of GT4 for resources monitoring and discovery, uses this kind of centralized approach as the default configuration.

Information technology trends suggest that scenarios will count with a enormous number of shared resources in the grid that the whole planet could be considered a global Virtual Organization (VO), where the dynamics of the resource information is almost impossible to be captured using a static hierarchy[36], which has similar problems to the centralized approach, such as the point of failure, and poor scaling for a large number of users/providers[31, 32].

Therefore, it is necessary to design new policies for discovery and propagation of resource and monitoring information is necessary. A proposal for decentralizing grid information services was made by Iamnitchi et al.[22, 23] where they proposed a Peer to Peer (P2P) based approach for organizing the MDS directories in a flat dynamic P2P network. There is a growing interest in the interaction of the Grid Computing and the P2P paradigm: both work within a very dynamic and heterogeneous environment, where the role and availability of resources may quickly change; both create a virtual working environment by collecting the resources available from a series of distributed, individual entities[32]. Extensive work has been devoted recently to obtain a new standard to replace the hierarchical policy. Naturally, a practical approach towards scalable solutions is offered by P2P models[28, 32, 36]. In this sense, many research efforts have been aimed to design new GRIS policies; their theoretical benefits are usually shown using ad-hoc simulation tools to compare to other previously published policies[29].

Many grid simulators are available for simulating different grid features, specially scheduling policies: Bricks[35], GangSim[17], BeoSim[25], GridSim[10], GSSIM[3, 26], SimGrid[12, 13], ChicSim[33] and OptorSim[9], but none of them is specifically designed to test and analyze the resource discovery and monitoring information policies.

Moreover, many useful tools are available for the automatic creation of network topologies, including from basic (unrealistic) structures to extremely complex topologies. The most famous ones are **BRITE** [30], **Inet** [24], **Tiers** [16], **GridG** [27] and **SIMULACRUM** (SIMUlated pLAtform CReation and User Modification)[5, 21].

In this work, we present *Grid-Matrix*[2], a tool designed to focus on the development and testing of discovery and monitoring information policies with a friendly graphical user interface and easy-to-use mechanism, based on SimGrid2 due to its current development activities and functionalities provided. Also, Grid-Matrix provide tools for the automatic creation of virtual scenarios

and propagation policies, based on SIMULACRUM because of the versatile network topologies schemes that provides for generation, including Clique, Line, Ring, Star, Uniform, Exponential, Waxman[37], Zegura[11] y Barabasi[8].

In section 2 we give some details of the SimGrid tool and present Grid-Matrix, in the section 3 we show some results, finally in the section 4 we draw some conclusions.

2 Grid-Matrix

Grid-Matrix offers an advanced Graphical User Interface (GUI) that allows the creation of complex networks topologies in an easy-to-use graphical environment and a simple scripting interface to simulate information propagation policies, including the capability to show the simulation progress during the execution (figure 1).

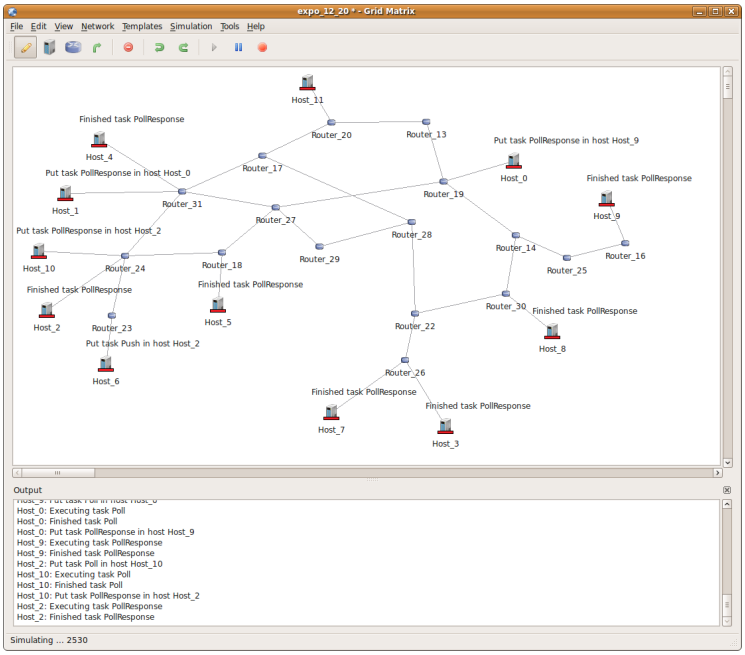


Fig. 1. Grid-Matrix GUI running a simulation of a Grid network.

Grid-Matrix is based on SimGrid2[13], a toolkit that provides the core for the simulation of distributed applications in heterogeneous distributed environments.

SimGrid2 consists of a core simulation engine, with an additional higher-level Application Programming Interface (API): **Simple application-level Simulator**

(MSG). This interface was added in SimGrid2. Formerly, it was developed to study scheduling algorithms and was upgraded to support many different features. MSG presents a large variety of functions to operate with tasks and processes.

Grid-Matrix interacts with SimGrid2, and provides a full-featured graphical interface that helps configure the network topologies and behavior. It includes an interface for scripting the node behavior in *Python*, giving access to a very powerful and complete environment to add new functionalities. Moreover, many built-in scripts for configuring the simulation to focus on information propagation, particularly MDS, are included, providing the necessary scripts for an easy setup of information propagation policies which may prevent having to synchronize manually the platform and deployment files.

In order to run a simulation, a network description file is needed (*network* file). This file may be created manually or automatically from a variety of different topologies included in the network generator incorporated in Grid-Matrix, based on SIMULACRUM. This file is later translated into the SimGrid2 format, the *platform* file. Once, the network topology is defined, a *deploy* file is needed to describe the propagation policy to use in each node. Finally, the scripts corresponding to the simulation and policies selected must be provided. The complete diagram of files interaction needed for a simulation in Grid-Matrix is shown in figure 2.

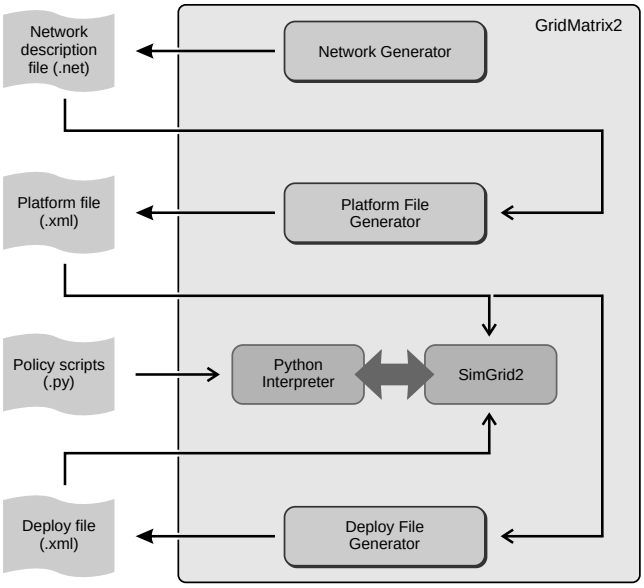


Fig. 2. Files needed for a Grid-Matrix simulation.

3 Results

In this section, we analyze the behavior of different distribution policies in two network topologies: clique and ring. In all the cases, the underlying network was composed of 50 routers with an increasing amount of nodes (100, 200 and 300 nodes).

In order to compare the different policies, we define the following information rates that capture the change in the total information, the amount of grid information available in each node and the age of this information, and the variability of this information among the nodes:

- Local Information Rate (LIR): this coefficient is bounded between 0 and 1 and captures the amount of information that a particular host has from all the entire grid in a single moment, weighting the expiration time. A value of 1 would mean the host knows every single piece of information about the whole grid and is as new as it can be. The definition of LIR for the host k is:

$$LIR_k = \frac{\sum_{h=1}^{N} \frac{(expiration_h - age_h)}{expiration_h} \cdot resourceCount_h}{totalResourceCount}$$

  where N is number of hosts in the system, $expiration_h$ is the expiration time of the resources of host h in host k, age_h is the time passed since the information was obtained from that host, $resourceCount_h$ is the amount of resources in host h and $totalResourceCount$ is the total amount of resources in the whole grid. The expression $expiration_h - age_h$ is similar to the well known parameter *time-to-live* from the `IP` protocol and, due to its definition, can only has no negative values.
- Global Information Rate (GIR): this coefficient is also a value between 0 and 1, and captures the amount of information that the whole grid knows of itself. It is calculated taking the mean value of every node's LIR.
- Global Information Variability (GIV): this coefficient is the standard deviation of GIR. It measures the variability of GIR in the system (less is better).

In the results presented in this section, the GIV values obtained were very low and similar in all cases, so they are not shown.

3.1 Evaluated policies

Based on the work of Mastroianni et al.[29] we selected three paradigmatic policies to be evaluated with Grid-Matrix.

- **Hierarchical:** in this kind of policy, a hierarchy is established beforehand and the resource information is sent using this fixed structure. In this way, the nodes in the top of the hierarchy exchange information with the ones below them. MDS actually uses this type of information propagation policy[4].

– **Random:** there is no structure at all, every node chooses randomly a neighbor to get the information from. This policy does not save information about quality of each query and the distribution of queries is uniform. All the nodes can query every other connected node[22].
– **Super-Peer:** the unstructured P2P policy spends part of the network bandwidth in work that might be done with a local cache, thus saving useful resources. For this reason, the super-peer networks were developed as an hybrid system which lies between totally distributed systems and cache-based services[32]. A super-peer network operates like a unstructured P2P, but some nodes are defined as *super-peers* working like servers for a subset of nodes and as peers in the network of super-peers. In this way, a two level structure is defined such that the *normal* nodes are allowed to *talk* only with a single super-peer and the cluster defined by it. An unstructured P2P is a degenerated super-peer network, where the cluster size is exactly one[38]. In this work, hosts send information to a local super-peer and this information is distributed among the others using a random policy.

To simplify the analysis and implementation of the simulations, the messages obtained by each node contain no details about the *real* resource information, but only the expire time (*time-to-live*) and the amount of carried information. The message between each host has two parts: one with fixed size (similar to the header of a real network message) and one in which the size is proportional to the resource information being transmitted (constant size for each resource). In this way, we have no need to introduce the resource details, focusing only on the very rich dynamics of the system.

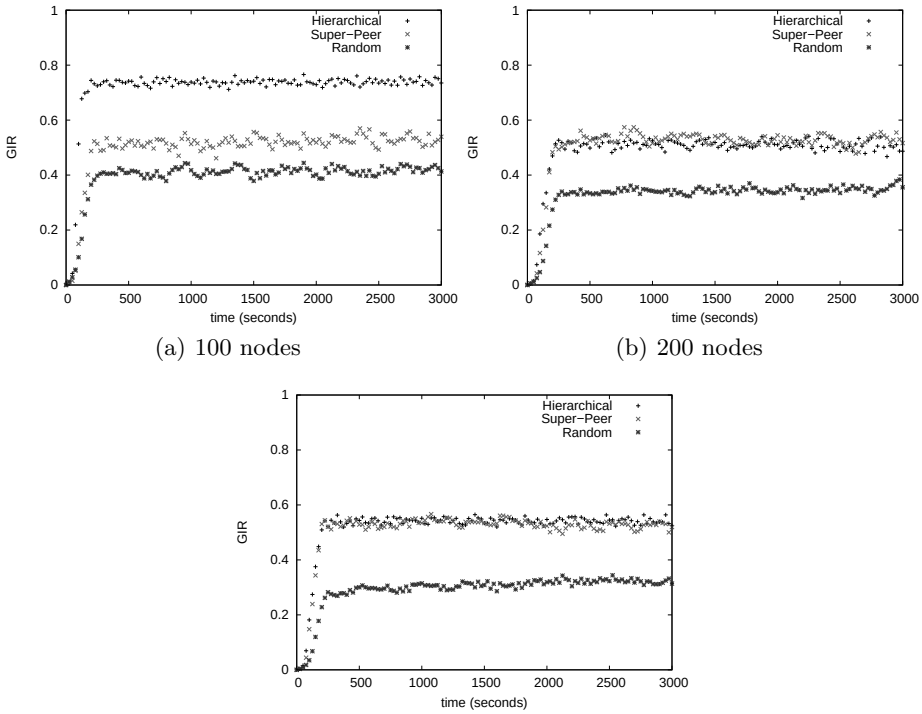
3.2 Clique Topology

In this topology, all the routers are fully connected and the nodes are inserted keeping the balance of the structure, i.e. all the routers have approximately the same amount of nodes. Due to this organization, the longest path between every two nodes has two hops and the shortest only one, expecting a good behavior of any policy in this scenario.

The figure 3 shows the global behavior of the policies in this topology in three different system sizes, gradually incrementing the nodes.

The Random policy shows a progressive decline in the performance directly related with the system size. With the addition of nodes and resources, this policy is less efficient to gather the information due to the lack of use of the information included in every message (i.e. the amount of resource information received, the remaining time of the information and the delay to receive the message). Nevertheless, with an increment of 300% in the system size, the Random policy shows only a fall of approximately 50% of GIR.

In the considered scenarios, Super-peer policy behaves very stable with a good performance despite the growth of the system. The Hierarchical policy presents some remarkable results, for 100 nodes overmatches Random and Super-peer as can be seen in the figure 3(a). For larger systems it has a fall in the performance matching Super-peer policy. The defined hierarchy has three levels:



(a) 100 nodes

(b) 200 nodes

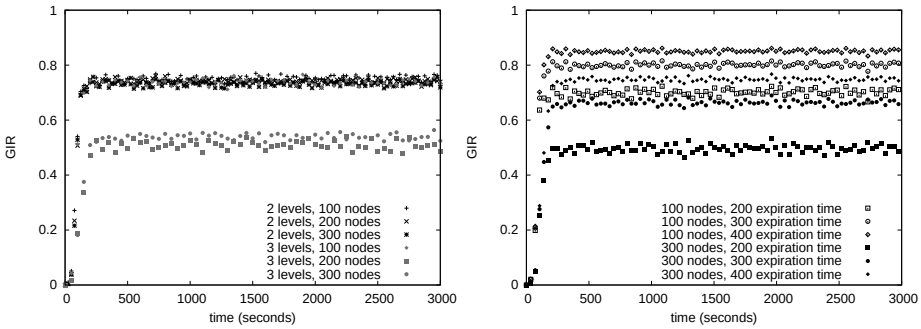
(c) 300 nodes

Fig. 3. Clique topology: GIR for the evaluated policies. In all the cases the underlying network has 50 routers and the mount of nodes are increased in each case. The information expiration time is uniformly distributed in the range from 200 to 300 seconds. Super-Peer presents a very stable behavior in all the cases, while the Hierarchical policy only overmatches it in the 100-nodes system. Random rapidly losses performance with the growth of the system, but it can be an interesting choice because its performance is only 50% worst with a system 300% larger.

- the first level only includes the root node.
- in the second level the nodes directly connected to the root through the same router are found, i.e. the path between these nodes and the root has only one hop.
- the rest of the nodes are included in the third (and last) level.

This organization does not make use of the underlying network infrastructure. Having three levels means that the information exchange between two nodes in the third level, involves traveling a much longer path than going directly between them.

The figure 4(a) presents a scenario comparing the three-level hierarchy with other with only two levels, i.e. a root node and the rest of the nodes connected to it representing a centralized scheme. For the 100-node system, both hierarchies behave very similar, but with larger systems, the two-level hierarchy overmatches



(a) Two and three level hierarchies (b) Variation of the expiration time

Fig. 4. Focusing on the impact of the amount of levels and the expiration time. In the figure (a) a two-level hierarchy behaves better than a three-level one in systems with 200 and 300 nodes, while in a 100-node system both present a similar behavior. In the figure (b) the expiration time is gradually increased using a 100 and 300 node hierarchy with three levels. The Hierarchical policy performs better when a longer expiration time is used.

the other one. In scenarios strongly connected, the centralized scheme is very appropriate, however it does not takes advantages of the network infrastructure.

On the other hand, the figure 4(b) presents a scenario focusing on the impact of the expiration time in the behavior of the Hierarchical policy. As was previously mentioned, a three-level hierarchy implies that the information would travel five levels to go from one end of the network to the other. An increment in the expiration time will cause that the information will remain valid for more time, allowing a more extensive distribution.

In the figure 3, the expiration time used was uniformly distributed in the range from 200 to 300 seconds using a three-level hierarchy. While in the figure 4(b) the expiration time was fixed in 200, 300 and 400 seconds using the same defined hierarchy.

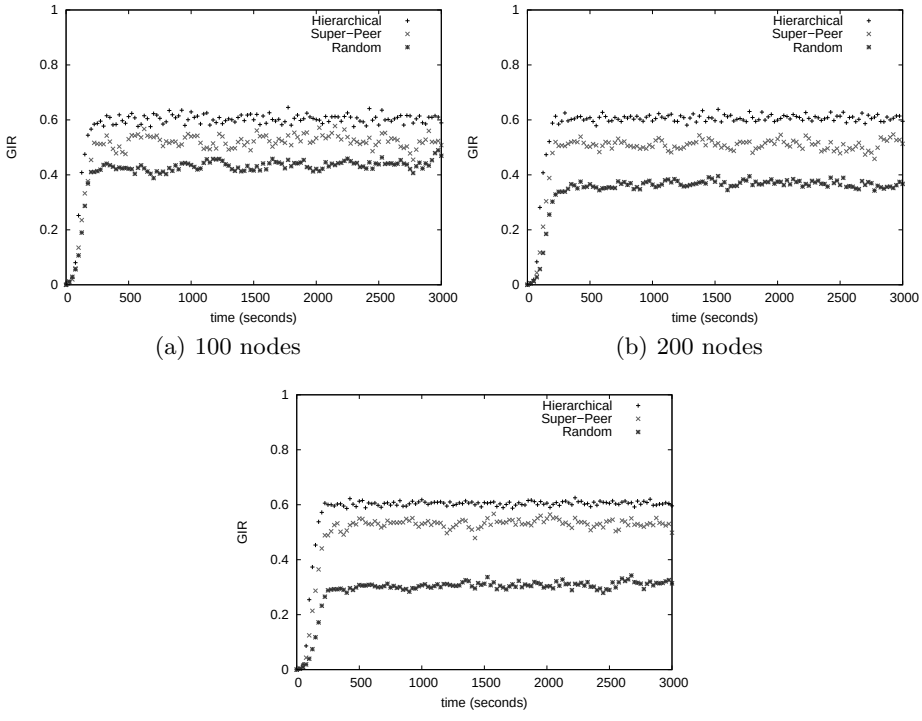
The impact of increasing the expiration time is remarkably well captured in the figure 4(b), specially evident in the 300-node system: the longer the selected expiration time, the better is the global behavior of the system. The situation is similar in the case of 100-node system. A further increment in the expiration time does not improve the behavior, concluding that for these cases the studied range is enough to allow the information to reach all the hierarchy without expiring.

3.3 Ring Topology

In this topology, the routers are connected forming a line with the same amount of nodes each one. Finally, a connection is made between the two extreme routers closing the *ring*. In this way, the topology becomes symmetric and the longest path is directly related with the half of the existing routers.

Intuitively, this topology seems to be poor suited for the information interchange. If the communication between the nodes surpass the available bandwidth in any part of the ring, the congestion will affect all the system and, as a consequence, a decrease in the system performance. This issue has a particular relevance in the Hierarchical policy due to the selection of the *root* node and the built hierarchy that can be seriously affected by congestion. In the others policies, congestion can impact the system's efficiency, but the random part of each one can help (at least in part) to overcome it.

The focus of the experiments of this section is on the behavior of the policies and not in the congestion issue, so enough available bandwidth was configured in the underlying network. Like the previous case, each policy was tested in a scenario with a fixed amount of routers and increasing the number of nodes.



(a) 100 nodes

(b) 200 nodes

(c) 300 nodes

Fig. 5. GIR in a ring topology. The network consists in 50 routers and 100, 200 and 300 nodes. The expiration time is uniformely distributed in the range from 200 to 300 seconds. The Hierarchical policy shows the best performance, stably followed by Super-peer. Random policy, like in the clique topology, presents a decrease in the performance with the system size.

In the figure 5 the behavior of the three policies is shown. It is worthwhile noting that in spite of the increment in the path length respect the clique topology, the Random policy behaves in a similar way: the GIR obtained is lower with the increment in the system size. This indicates that the behavior of this policy is directly related with the system size and independent of the underlying network topology.

On the other hand, the Hierarchical and Super-peer policies show very stable GIR values, but the first presents higher in the three considered scenarios.

4 Conclusions

Grid Computing proposes unlimited access to different computational resources. A key feature needed in Grid Computing is to maintain up-to-date information related to distributed resources and services in an heterogeneous and highly dynamic environment.

The dynamics of the resource information cannot be captured using a static hierarchy, so we must be prepared to work in a scenario where adaptive information policies such as P2P strategies should be implemented. Many grid simulators are available for simulating different grid features, especially scheduling policies, but none of them are specifically designed to test and analyze the resource discovery and monitoring information policies.

In this work, we presented *GridMatrix2*, a very powerful simulation tool designed to focus on the development and testing of discovery and monitoring information policies with a friendly graphical user interface and easy-to-use mechanism, based on SimGrid2. Moreover, GridMatrix provides tools for the automatic creation of virtual scenarios and propagation policies.

Using GridMatrix, we created scripts that implement three different information propagation policies: Hierarchical, Super-peer and Random. This policies were studied in two different network environments, Clique and Ring topologies, that were automatically built using GridMatrix. In all the cases, the tool allowed us to get very useful information and insight into the details of the working system.

We introduced three metrics that capture and summarize the information propagation behavior: LIR, GIR and GIV, showing that Super-Peer presented a very stable behavior in all the cases, while Random rapidly losses performance with growth of the system indepently of the underlying topoly. The Hierarchical policy shows good performance in the case of a centralized scheme, but using a three-level hierarchy produces a fall in the performance.

On the other hand, the impact of the expiration time in the behavior of the Hierarchical policy was analized. We found that longer expiration times presented a better global behavior.

The modern and modular design of the scripting features included in GridMatrix, in close conjunction with the user friendly and always handy GUI happened to be a very powerful tool for the evaluation of new propagation policies of resource information.

Acronyms

API Application Programming Interface . . . 3
GIR Global Information Rate . . . 5
GIV Global Information Variability . . . 5
GRIS Grid Resource Information Service . . . 2
GT4 Globus Toolkit 4 . . . 1
GUI Graphical User Interface . . . 3
HPC High Performance Computing . . . 1
LIR Local Information Rate . . . 5
MDS Monitoring and Discovery System . . . 2
MSG Simple application-level Simulator . . . 3
P2P Peer to Peer . . . 2
VO Virtual Organization . . . 2

Acknowledgments

E.E.M. has a scholarship from CONICET. This work was partially supported by grants from CONICET (PIP 1087/09) and Universidad de Buenos Aires (UBACyT X132/08).

References

[1] Globus Alliance web page, `http://www.globus.org/`, last visited on 03/20/2010
[2] Grid matrix home page, `http://lsc.dc.uba.ar/hpc-grid/grid/grid-matrix`, last visited on 07/12/2009
[3] The grid scheduling simulations portal, `http://www.gssim.org`, last visited on 07/19/2009
[4] MDS in Globus Alliance, `http://www.globus.org/toolkit/mds/`, last visited on 03/20/2010
[5] PDA, the Platform Description Archive, `http://pda.gforge.inria.fr/`, last visited on 02/24/2010
[6] Abbas, A.: Grid Computing: A Practical Guide to Technology and Applications. Charles River Media, Hingham, MA (2003)
[7] Aloisio, G., Cafaro, M., Epicoco, I., Fiore, S., Lezzi, D., Mirto, M., Mocavero, S.: Resource and service discovery in the igrid information service. In: Computational Science and Its Applications - ICCSA. pp. 1–9 (2005)
[8] Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (October 1999)
[9] Bell, W.H., Cameron, D.G., Millar, A.P., Capozza, L., Stockinger, K., Zini, F.: Optorsim: A grid simulator for studying dynamic data replication strategies. International Journal of High Performance Computing Applications 17(4), 403–416 (2003), `http://hpc.sagepub.com/cgi/content/abstract/17/4/403`
[10] Buyya, R., Murshed, M.: Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrency and Computation: Practice and Experience 14, 1175–1220 (December 2002)
[11] Calvert, K.L., Doar, M.B., Zegura, E.W.: Modeling Internet Topology. IEEE Communications Magazine 35(6), 160–163 (June 1997), `citeseer.ist.psu.edu/calvert97modeling.html`

[12] Casanova, H.: Simgrid: A toolkit for the simulation of application scheduling. In: CCGRID '01: Proceedings of the 1st International Symposium on Cluster Computing and the Grid. p. 430. IEEE Computer Society, Los Alamitos, CA, USA (2001)

[13] Casanova, H., Legrand, A., Quinson, M.: Simgrid: A generic framework for large-scale distributed experiments. In: UKSIM '08: Proceedings of the Tenth International Conference on Computer Modeling and Simulation. pp. 126–131. IEEE Computer Society, Los Alamitos, CA, USA (2008)

[14] Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: HPDC '01: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing. pp. 181–194. IEEE Computer Society, Washington, DC, USA (2001)

[15] De Roure, D., Baker, M., Jennings, N., Shadbolt, N.: Grid computing - making the global infrastructure a reality, chap. The evolution of the Grid, pp. 65–100. John Wiley & Sons Ltd (2003), http://eprints.ecs.soton.ac.uk/6871/

[16] Doar, M.: A better model for generating test networks. In: Proceedings of GLOBECOM'96. Global Telecommunications Conference. pp. 86–93. IEEE (Nov 1996), citeseer.ist.psu.edu/doar96better.html

[17] Dumitrescu, C., Foster, I.: Gangsim: a simulator for grid scheduling studies. In: CCGRID '05: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05). vol. 2, pp. 1151–1158. IEEE Computer Society, Washington, DC, USA (2005)

[18] Foster, I., Kesselman, C., Nick, J., S., T.: The physiology of the grid: An open grid services architecture for distributed systems integration. Tech. rep., Open Grid Service Infrastructure WG, Global Grid Forum (2002), http://www.globus.org/research/papers/ogsa.pdf

[19] Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, H., Reed, D., Jiang, W. (eds.) IFIP International Conference on Network and Parallel Computing 2005. Lecture Notes in Computer Science, vol. 3779, pp. 2–13. Springer Berlin / Heidelberg (2005)

[20] Foster, I., Kesselman, C.: The Grid 2: Blueprint for a New Computing Infrastructure. The Morgan Kaufmann Series in Computer Architecture and Design, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (Nov 2003)

[21] Marc-Eduard Frincu, Martin Quinson, Frédéric Suter: Handling Very Large Platforms with the New SimGrid Platform Description Formalism. Technical Report RT-0348, INRIA (2008), http://hal.inria.fr/inria-00256883/en/

[22] Iamnitchi, A., Foster, I., Nurmi, D.: A peer-to-peer approach to resource discovery in grid environments. In: Proceedings of the 11 th IEEE International Symposium on High Performance Distributed Computing HPDC-11 (HPDC' 02). p. 419. IEEE, Edinbourgh, UK (Jul 2002)

[23] Iamnitchi, A., Foster, I.: Grid resource management: state of the art and future trends, chap. A peer-to-peer approach to resource location in Grid environments, pp. 413–429. Kluwer Academic Publishers, Norwell, MA, USA (2004)

[24] Jin, C., Chen, Q., Jamin, S.: Inet: Internet Topology Generator. Tech. Rep. CSE-TR443 -00, Department of EECS, University of Michigan (2000), citeseer.ist.psu.edu/jin00inet.html

[25] Jones, W., Pang, L., Stanzione, D., Ligon, W.I.: Job communication characterization and its impact on meta-scheduling co-allocated jobs in a mini-grid. In: Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS'04. p. 253 (Apr 2004)

[26] Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J.: Grid scheduling simulations with GSSIM. In: Proceedings of 13th International Conference on Parallel and Distributed Systems (ICPADS'07). vol. 2, pp. 1–8. IEEE Computer Society, Los Alamitos, CA, USA (2007)
[27] Lu, D., Dinda, P.A.: Gridg: Generating realistic computational grids. ACM Sigmetrics Performance Evaluation Review 30, 33–40 (2003)
[28] Mastroianni, C., Talia, D., Verta, O.: A super-peer model for resource discovery services in large-scale Grids. Future Generation Computer Systems 21(8), 1235–1248 (October 2005)
[29] Mastroianni, C., Talia, D., Verta, O.: Designing an information system for grids: Comparing hierarchical, decentralized P2P and super-peer models. Parallel Computing 34(10), 593–611 (2008)
[30] Medina, A., Lakhina, A., Matta, I., Byers, J.: Brite: Universal topology generation from a user's perspective. Tech. rep., Boston, MA, USA (2001)
[31] Plale, B., Jacobs, C., Jensen, S., Liu, Y., Moad, C., Parab, R., Vaidya, P.: Understanding grid resource information management through a synthetic database benchmark/workload. In: CCGRID '04: Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid. pp. 277–284. IEEE Computer Society, Washington, DC, USA (Apr 2004)
[32] Puppin, D., Moncelli, S., Baraglia, R., Tonellotto, N., Silvestri, F.: A grid information service based on peer-to-peer. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005 Parallel Processing. Lecture Notes in Computer Science, vol. 3648, pp. 454–464. Springer (2005)
[33] Ranganathan, K., Foster, I.: Decoupling computation and data scheduling in distributed data-intensive applications. In: HPDC '02: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing. p. 352. IEEE Computer Society, Washington, DC, USA (2002)
[34] Ranjan, R., Harwood, A., Buyya, R.: Peer-to-peer-based resource discovery in global grids: a tutorial. Communications Surveys & Tutorials, IEEE 10(2), 6–33 (2008)
[35] Takefusa, A., Matsuoka, S., Nakada, H., Aida, K., Nagashima, U.: Overview of a performance evaluation system for global computing scheduling algorithms. In: HPDC '99: Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing. pp. 97–104. IEEE Computer Society, Washington, DC, USA (1999)
[36] Trunfio, P., Talia, D., Papadakis, C., Fragopoulou, P., Mordacchini, M., Pennanen, M., Popov, K., Vlassov, V., Haridi, S.: Peer-to-Peer resource discovery in Grids: Models and systems. Future Generation Computer Systems 23(7), 864–878 (Aug 2007)
[37] Waxman, B.M.: Routing of multipoint connections. IEEE Journal on Selected Areas in Communications 6(9), 1617–1622 (December 1988)
[38] Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network. In: Proceedings of 19th International Conference on Data Engineering (ICDE'03). p. 49. IEEE Computer Society, Los Alamitos, CA, USA (2003)