Using distributed local information to improve global performance in Grids

Paula Verghelet, Diego Fernández Slezak, Pablo Turjanski

and

Esteban Mocskos

Laboratorio de Sistemas Complejos, Departamento de Computación,
Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires
Buenos Aires (C1428EGA), Argentina.
{*pverghelet, dslezak, pturjanski, emocskos*}*@dc.uba.ar*

Abstract

Grid computing refers to the federation of geographically distributed and heterogeneous computer resources. These resources may belong to different administrative domains, but are shared among users. Every grid presents a key component responsible for obtaining, distributing, indexing and archiving information about the configuration and state of services and resources. Optimizing tasks assignations and user requests to resources require the maintenance of up-to-date information about the grid.

In large scale Grids, the dynamics of the resource information cannot be captured using a static hierarchy and relying in manual configuration and administration. It is necessary to design new policies for discovery and propagation of resource information. There is a growing interest in the interaction of Grid Computing and the Peer to Peer (P2P) paradigm, pushing towards scalable solutions.

In this work, starting from the Best-Neighbor policy based on previously published ideas, the reasons behind its lack of performance are explored. A new improved Best-Neighbor policy are proposed and analyzed, comparing it with Random, Hierarchical and Super-Peer policies.

Keywords: Resource Information, Information Policies, Grid Computing, Best Neighbor

1 Introduction

Grid computing refers to the federation of geographically distributed and heterogeneous computer resources[1]. These resources may belong to different administrative domains, but are shared among users. Grid infrastructure may be confined to a small network of workstations within a corporation or large public collaborations across many countries and networks.

Every grid infrastructure needs a component responsible for obtaining, distributing, indexing and archiving information about the configuration and state of services and resources. Optimizing tasks assignations and user requests to resources require the maintenance of up-to-date information about the grid[2]. It is widely known that standard centralized organization approach has several drawbacks[3]. Static hierarchy has become the defacto implementation of grid information systems[4].

In medium-to-large scale environments, the dynamics of the resource information cannot be captured using a static hierarchy[5]. This approach has similar drawbacks to the centralized one, such as the point of failure, and poor scaling for a large number of users/providers[6, 7]. Therefore, it is necessary to design new policies for discovery and propagation of resource information.

There is a growing interest in the interaction of Grid Computing and the P2P paradigm, pushing towards scalable solutions[8, 5]. These initiatives are base in two common facts: i) very dynamic and heterogeneous environment and ii) create a virtual working environment by collecting the resources available from a series of distributed, individual entities[7].

Iamnitchi et al.[9, 10] proposed a P2P approach for organizing the information components in a flat dynamic P2P network. This decentralized approach envisages that every administrative domain maintains its information services and makes it available as part of the P2P network. Schedulers may initiate look-up queries that are forwarded in the P2P network using flooding (a similar approach to the unstructured P2P network Gnutella[11]).

A key aspect of P2P systems is how peers interact between them. Different algorithms for this interaction are available and the selection may severally impact in system performance. The most common policies are:

- **Random:** Every node chooses randomly any other node to query information from. There is no structure at all.
- **Best-Neighbor:** Some information about each answer is stored and the next neighbor to query is selected using the quality of the previous answers. At the beginning, the node has no information about its neighbors, thus it chooses randomly. As information is collected , the probability of choosing a neighbor randomly is inversely proportional to the amount of information stored.
- **Super-Peer:** 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.

Mastroiani et al.[12] evaluated the performance of these policies and analyzed the pros and cons of each solution. In despite of the majority of the evaluated aspects strongly depend on time, it is usually discarded in the analysis leading to a the missing of the inherent dynamical nature of the system. Some other structured P2P approaches have also been proposed, see for example the work of Basu et al [13].

In Mocskos et al.[14] the authors introduced a new set of metrics (LIR, GIR and GIV) that incorporate the notion of time decay of information for evaluating system performance. The best results in terms of the proposed metrics were attained by the hierarchical policy, followed by super-peer which outperformed random and best-neighbor.

Iamnitchi et al.[9, 10] introduced the Best-Neighbor policy which records the requests answered by each node and directs the following to the peer that previously answered or chooses randomly if no relevant experience exists. Following the taxonomy proposed in Ranjan et al [3], this approach can be included in the class of unstructured and non-deterministic P2P systems.

Based on these results, in Mocskos et al[14] some good initial results of this policy were shown. Best-neighbor get good performance and, mainly, the overall information known by the system increases with time. Notwithstanding, in later studies with the system growth Best-Neighbor shows a similar performance of Random policy without getting the increase of GIR with time (data not shown).

In this work, we start from the Best-Neighbor policy based on the ideas of Iamnitchi et al.[9, 10] and explore the reasons behind its lack of performance. We propose and analyze some improvements to the policy. Finally, we compare the obtained policy with Random, Hierarchical and Super-Peer.

2 Materials and Methods

To evaluate the different scenarios and policies, we used GridMatrix[1], an open source tool focused on the analysis of discovery and monitoring information policies, based on SimGrid2[16]. This simulator includes three different metrics[14] for the study of information propagation, described below:

- **Local Information Rate (LIR)**: captures the amount of information that a particular host has from all the entire grid in a single moment. For the host k, LIR_k is:

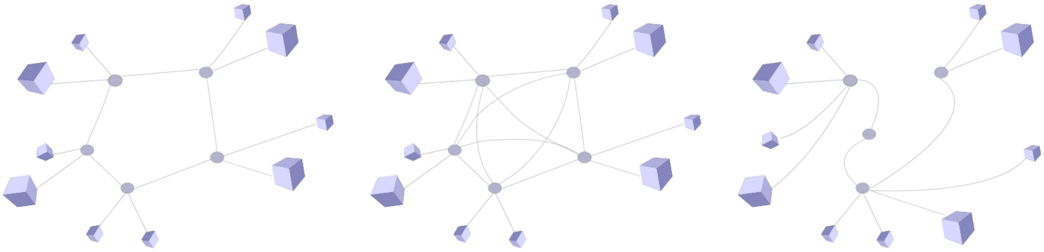
$$LIR_k = \frac{\sum_{h=1}^{N} f(age_h, expiration_h) \cdot resourceCount_h}{totalResourceCount} \tag{1}$$

  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.

- **Global Information Rate (GIR)**: captures the amount of information that the whole grid knows of itself, calculated as the mean value of every node's LIR.

- **Global Information Variability (GIV)**: measures the variability of GIR in the system (less is better), calculated as the standard deviation of GIR.

Three topologies were used to study the information dynamics: Ring, Clique and Exponential (see figure 1). In a Ring topology, every node is connected exactly to two other nodes, forming a cycle (figure 1a). Clique topology proposes a scenario where every node is connected to every other node (figure 1b). To represent a more realistic network, the exponential distribution model is used for the connections, where the amount of connections of each node follows an exponential distribution law (figure 1c), commonly seen in the Internet or collaborative networks[17, 18]. All theses topologies and scenarios were generated by the included features in the GridMatrix simulator.



(a) Ring topology **(b)** Clique topology **(c)** Exponential topology

Figure 1: Schemes of the network topologies analysed in this work. In the Ring topology(a) each node connects to exactly two other nodes, while the Clique(b) is an all-to-all connected network. Exponential topology(c) is formed following an exponential distribution law.

The standard best-neighbor implementation (**BN**) ranks the nodes with the following scoring function[14]:

$$f_{scoring} = a * \text{RES\_COUNT} - b * \text{RTT} - c * \text{RESPONSE\_FAILED}$$

where RES_COUNT is the amount of available resources in the node, RTT corresponds to the Round Trip Time and RESPONSE_FAILED counts the number of messages looses. a, b, and c are parameters to change the weight of each variable.

[1] For a complete description of this tool, see [14] ([15])

We present **fBN**, an implementation of Best-Neighbor policy that incorporates a new term which captures information about the amount of local resources of the node:

$$f_{scoring} = a * \text{RES\_COUNT} - b * \text{RTT} - c * \text{RESPONSE\_FAILED} + d * \text{OWN\_RES\_COUNT}$$

where the new term OWN_RES_COUNT is the amount of local resources in the node and d is the weighting coefficient of this variable.

3 Results and Discussion

The standard best-neighbor implementation (**BN**) strongly depends on knowing as much nodes as possible in the network. When the policy starts, the nodes are randomly selected until sufficient nodes are known (some threshold value is selected) creating a local database with the information about the known neighbors. In medium-to-large scale networks, knowing the whole network can be very demanding, and so starting the best-neighbor strategy may be delayed leading to extremely large "random" stage (also known as *learning stage*).

To achieve this, many methods have been proposed, from which we choose *merging lists* for our implementation. This technique consists of sharing the lists of neighbors that a particular node has to any other node that communicates with it. With such simple implementation, significant improvements are reported and all nodes know about almost all the network greatly shortening the learning stage. In figure 2, we show the learning curve of the nodes in two exponential networks (30 and 400 nodes) using the merging list algorithm versus just randomly exploring the network. Using this improvement, all the nodes of the network are quickly known and best neighbor method can start choosing the most appropriate nodes (figure 2, blue lines). On the other hand, as network sizes scale, knowing every node in the infrastructure is increasingly demanding (figure 2, green lines), leading eventually to a situation where the learning stage become the strongly dominant phase.

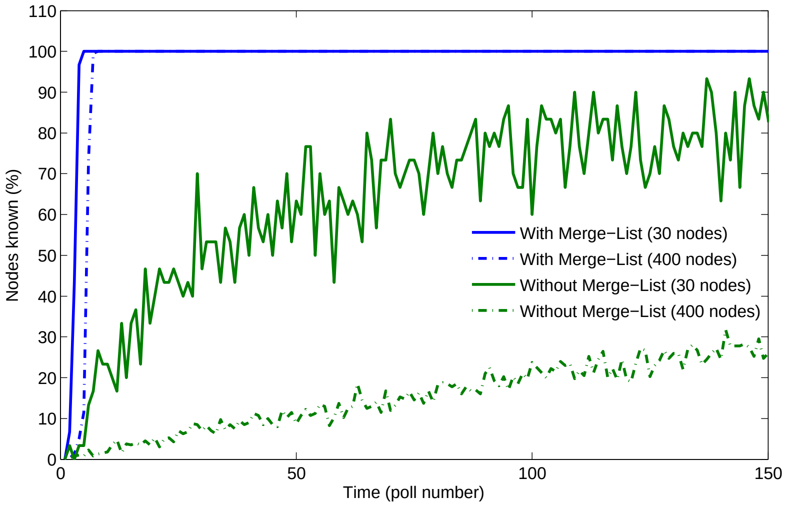
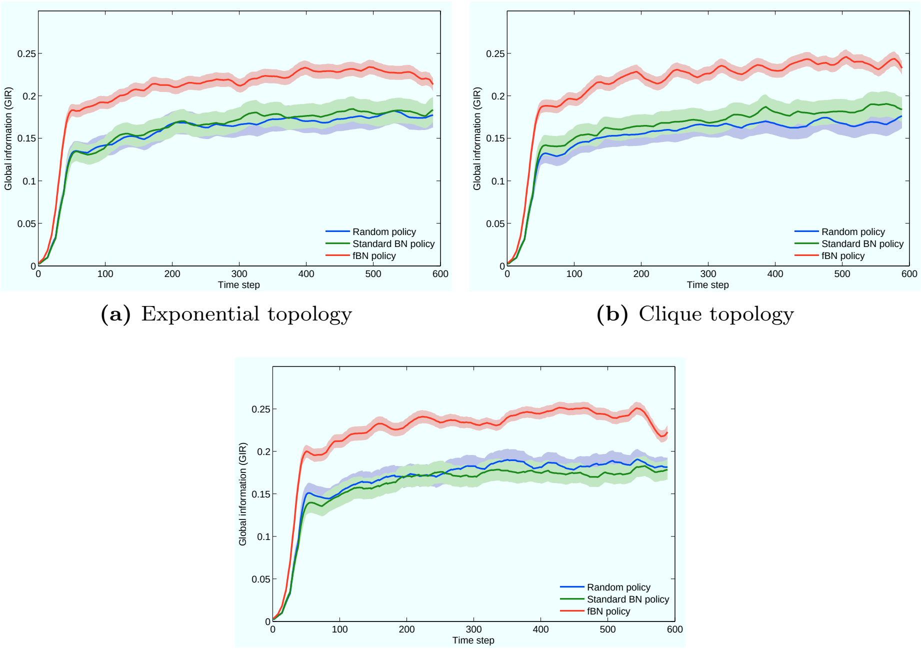


Figure 2: Learning curve of the nodes in two exponential networks (30 and 400 nodes) using the merging list algorithm versus just using the random nodes selected. In very little time steps, with the merge-list method, all nodes of the network are known using less messages.

Once the network is sufficiently well known, Best-Neighbor method can rank the nodes to connect with, and select the most informative one following the scoring function. This function involves an implicit relation between their weighting coefficients (see Materials and Methods for details). The selection of each weight in this relation leads to focusing on some of the aspect of the system, in this work a standard set of parameters were used following previous works[19, 20, 14]. Using the standard scoring function, the scoring function may select a node that do not have much proper information and instead has lots of data about its neighbors. This fact would penalize the amount of information collected due to the time delay of propagation of information.

In figure 3, green line shows the evolution of GIR for this situation, performing just over random policy (blue line). To overcome this problem, we introduce the fBN, a modification to the original Best-Neighbor policy that takes into account the amount of proper information available. In all the topologies and networks sizes studied (only shown 400 nodes networks in figure 3), fBN outperforms Random and BN policies.



(a) Exponential topology

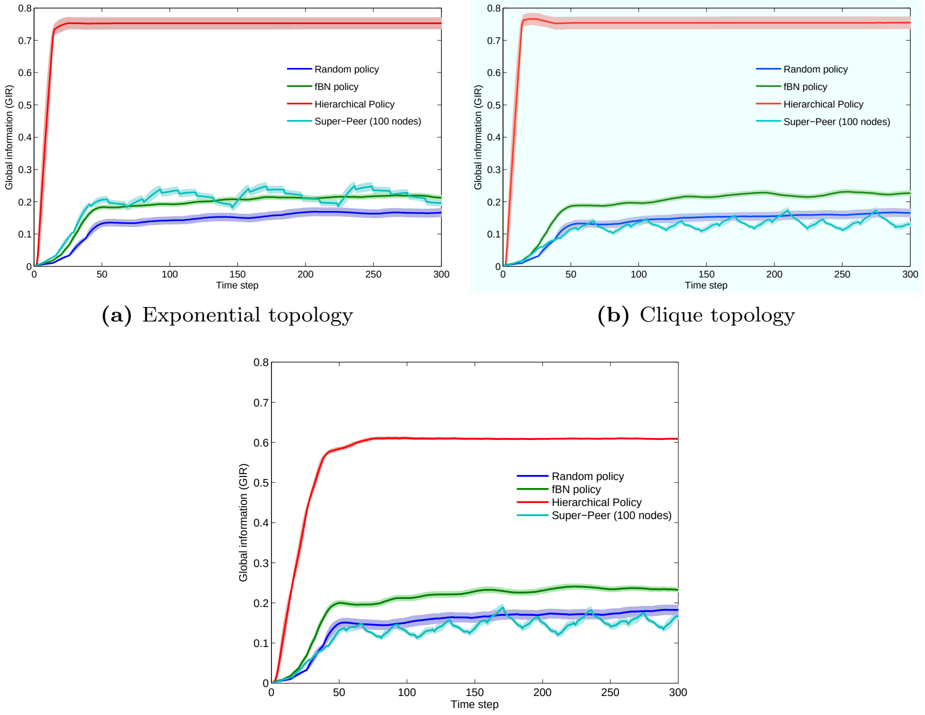
(b) Clique topology

(c) Ring topology

Figure 3: Evolution of GIR for Random, BN and fBN policies in three topologies with 400 nodes. BN shows similar behavior to Random, while fBN outperforms both other policies.

Next, we compare this new implementation with the different policies: Random, Hierarchical and Super-peer. These policies have different needs of administration. Hierarchical consists of a human supervised construction of a logical hierarchy using the nodes. Evidently, this policy has a very high cost of configuration and maintenance, but would result in very high GIR values. We compare this policy with other policy which needs very little supervision: Super-peer. In the used setup, 100 nodes are selected to act as super-peers. Finally, we present the comparison to the proposed implementation of Best-Neighbor, a completely unsupervised policy.

In figure 4, we show the evolution of GIR for the exposed policies. Data is smoothed by taking the moving average 5 points to each side of each point. Hierarchical (red line) shows the higher GIR values, far from the other implementations. This policy shows a lower GIR in the case of Ring topology due to the underlying network infrastructure and the longer paths needed to send messages between the nodes. On the other hand, random (blue line) shows the worst values. In the middle, closer to Random policy, Super-peer (cyan line) and best-neighbor (green line) shows a similar average GIR in the case of Exponential topology. For the other two policies, Super-peer overlaps with Random policy. Super-peer results in a more variable GIR over time, while best-neighbor shows a very stable behavior.



(a) Exponential topology

(b) Clique topology

(c) Ring topology

Figure 4: Evolution of GIR for Random, fBN, Super-Peer (100 nodes) and hierarchical policies. fBN shows similar behavior to Super-Peer, indicating that unsupervised method may obtain comparable result to supervised ones. Both results perform better Random, but very far from the hierarchical policy.

4 Conclusions

Grid computing refers to the federation of geographically distributed and heterogeneous computer resources. Every grid infrastructure needs a component responsible for obtaining, distributing, indexing and archiving information about the configuration and state of services and resources. The dynamics of the resource information cannot be captured using a static hierarchy due to similar drawbacks as the centralized one. Therefore, it is necessary to design new policies for discovery and propagation of resource information.

Four policies are usually considered: Random, Best-Neighbor, Super-Peer and Hierarchical, all of them have different needs of administration and supervision.

Two modifications are introduced to improve the performance of Best-Neighbor policy obtaining fBN: i) merge the lists of neighbors during the learning stage to decrease the length of this phase, ii) a new term which considers the amount of local resources provided by the node is added to the scoring function.

fBN presents a short learning phase which maintains almost constant with the considered system sizes. On the other hand, fBN outperforms Random policy and shows similar behavior as Super-peer. Hierarchical shows the best performance, but on the other hand, is the policy needing more setup and administration.

fBN results in a good trade-off between fully automated policy and obtained performance.

Acknowledgments

The authors specially thank Intel Corporation and Centro de Cómputo de Alto Rendimiento (`cecar.fcen.uba.ar`) for allowing the usage of essential computing equipment needed for obtaining the presented results. This work was partially supported by the grant from Universidad de

Buenos Aires (UBACyT 20020100100889).

References

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

[2] G. Aloisio, M. Cafaro, I. Epicoco, S. Fiore, D. Lezzi, M. Mirto, and S. Mocavero, "Resource and service discovery in the igrid information service," in *Computational Science and Its Applications - ICCSA*, 2005, pp. 1–9.

[3] R. Ranjan, A. Harwood, and R. Buyya, "Peer-to-peer-based resource discovery in global grids: a tutorial," *IEEE Commun Surv Tut*, vol. 10, no. 2, pp. 6–33, 2008.

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

[5] P. Trunfio, D. Talia, C. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi, "Peer-to-Peer resource discovery in Grids: Models and systems," *Future Gener Comp Sy*, vol. 23, no. 7, pp. 864–878, Aug. 2007.

[6] B. Plale, C. Jacobs, S. Jensen, Y. Liu, C. Moad, R. Parab, and P. Vaidya, "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.* Washington, DC, USA: IEEE Computer Society, Apr. 2004, pp. 277–284.

[7] D. Puppin, S. Moncelli, R. Baraglia, N. Tonellotto, and F. Silvestri, "A grid information service based on peer-to-peer," in *Euro-Par 2005 Parallel Processing*, ser. Lecture Notes in Computer Science, J. C. Cunha and P. D. Medeiros, Eds., vol. 3648. Springer, 2005, pp. 454–464.

[8] C. Mastroianni, D. Talia, and O. Verta, "A super-peer model for resource discovery services in large-scale Grids," *Future Gener Comp Sy*, vol. 21, no. 8, pp. 1235–1248, October 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X05000701

[9] A. Iamnitchi, I. Foster, and D. Nurmi, "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).* Edinbourgh, UK: IEEE, Jul 2002, p. 419.

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

[11] "Gnutella protocol development," last visited on 30/09/2012. [Online]. Available: http://rfc-gnutella.sourceforge.net/index.html

[12] C. Mastroianni, D. Talia, and O. Verta, "Designing an information system for grids: Comparing hierarchical, decentralized P2P and super-peer models," *Parallel Comput*, vol. 34, no. 10, pp. 593–611, 2008.

[13] S. Basu, L. Costa, F. Brasileiro, S. Banerjee, P. Sharma, and S.-J. Lee, "Nodewiz: Fault-tolerant grid information service," *Peer Peer Netw Appl*, vol. 2, pp. 348–366, 2009.

[14] E. E. Mocskos, P. Yabo, P. G. Turjanski, and D. Fernandez Slezak, "Grid matrix: a grid simulation tool to focus on the propagation of resource and monitoring information," *Simul-T Soc Mod Sim*, May 2012.

[15] "Grid matrix home page," last visited on 30/09/2012. [Online]. Available: http://lsc.dc.uba.ar/hpc-grid/grid/grid-matrix

[16] H. Casanova, A. Legrand, and M. Quinson, "Simgrid: A generic framework for large-scale distributed experiments," in *10th IEEE International Conference on Computer Modeling and Simulation.* Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2008, pp. 126–131.

[17] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, October 1999.

[18] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the World-Wide Web," *Nature*, vol. 401, pp. 130–131, sep 1999. [Online]. Available: http://adsabs.harvard.edu/abs/1999Natur.401..130A

[19] D. G. Márquez, E. E. Mocskos, D. F. Slezak, and P. G. Turjanski, "Simulation of resource monitoring and discovery in grids." in *Proceedings of HPC 2010 High-Performance Computing Symposium*, 2010, pp. 3258–3270. [Online]. Available: http://www.39jaiio.org.ar/node/121

[20] D. G. Márquez, D. F. Slezak, P. G. Turjanski, and E. E. Mocskos, "Gaining insight in the analysis of performance for resource monitoring and discovery in grids," in *Proceedings of HPC 2011 High-Performance Computing Symposium*, 2011. [Online]. Available: http://www.40jaiio.org.ar/sites/default/files/T2011/HPC/1227.pdf