HAL
open science

Assessing the Quality of Automatically Built Network Representations

Lionel Eyraud-Dubois, Martin Quinson

To cite this version:

Lionel Eyraud-Dubois, Martin Quinson. Assessing the Quality of Automatically Built Network Representations. Workshop on Programming Models for Grid Computing - organized at IEEE/ACM International Symposium on Cluster Computing and the Grid - CCGrid 2007, May 2007, Rio de Janeiro, Brazil. pp.795 - 800, 10.1109/CCGRID.2007.25 . inria-00188408

HAL Id: inria-00188408

https://inria.hal.science/inria-00188408v1

Submitted on 16 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assessing the Quality of Automatically Built Network Representations

Lionel Eyraud-Dubois
École Normale Supérieure de Lyon
Lyon, France
Lionel.Eyraud-Dubois@ens-lyon.fr

Martin Quinson
Nancy University / LORIA
Nancy, France
Martin.Quinson@loria.fr

***Abstract*— In order to efficiently use Grid resources, users or middlewares must use some network information, and in particular some knowledge of the platform network. As such knowledge is usually not available, one must use tools which automatically build a topological network model through some measurements. Our aim is to define a methodology to assess the quality of these network model building tools, and to apply this methodology to representatives of the main classes of model builders. Using this approach, we show that none of the main existing techniques build models that enable to accurately predict the running time of simple application kernels for actual platforms.**

keywords: Network model, topology reconstruction.

I. Introduction

Grids are parallel and distributed systems that result from the sharing and aggregation of resources distributed between several geographically distant organizations [1]. Unlike classical parallel machines, Grids present heterogeneous and sometimes even non-dedicated capacities. Gathering accurate and relevant information about them is then a challenging issue, but it is also a necessity. Indeed, the efficient use of Grid resources can only be achieved through the use of accurate network information. Qualitative information such as the network topology is crucial to achieve such tasks as running network-aware applications [2], efficiently placing servers [3], or predicting and optimizing collective communications performance [4].

However, the description of the structure and characteristics of the network interconnecting the different Grid resources is usually not available to users. This is mainly due to security (fear of Deny Of Service attacks) and privacy reasons (ISP do not want you to know where their bottlenecks are). We thus have a need for tools which automatically construct models of platform networks. There exist many such tools and projects providing information about the network. Some of them rely on simple ideas while others use very sophisticated measurement techniques. Some of these techniques, though, are sometimes ineffective in Grid environments due to security issues. Anyway, to the best of our knowledge, these different techniques have never been compared rigorously in the context of Grid computing platforms. Our aim is to define a methodology to assess the quality of these network model building tools, and to apply it to representatives of the main classes of model builders.

In this article, we make the following contributions:

- In Section II, we briefly review the main observation techniques and reconstruction algorithms that have been proposed in the literature. We first identify some observation techniques that are effective in Grid environments. Then we identify in Section II-B a few reconstruction algorithms that are representative of the existing ones.
- Assessing the quality of a reconstruction algorithm is really hard as the quality of the resulting graph is highly dependent of its future usage. We propose in Section III a few quality metrics, ranging from simplistic to sophisticated ones.
- We implement ALNeM, a lightweight distributed measurement infrastructure. ALNeM is built using GRAS [5] which enables us to run it seamlessly in real environments as well as in simulated environments.
- In Section IV, we evaluate in a real environment the quality of the different reconstruction algorithms with respect to the simplest metrics we have proposed.
- In Section V, we evaluate through simulation the quality of the different reconstruction algorithms with respect to all the metrics we have proposed. This evaluation is performed on models of real platforms and on synthetic platforms.
- These evaluations highlight the weaknesses of

simple metrics and demonstrate the need for a new generation of reconstruction algorithms.

II. Building a Network Representation

A. Measurements in a Grid Environment

Network discovery tools have received a lot of attention in the recent years. However, most of them are not suited to Grid environments. Indeed, much of the previous work (e.g., Remos [6], [7]) rely on low-level network protocols like SNMP or BGP, whose usage is generally restricted for security reasons (it is indeed possible to conduct Deny Of Service attacks by flooding the routers with requests).

As a matter of fact, in a Grid environment, even traceroute or ping-based tools (e.g., TopoMon [8], Lumeta [9], IDmaps [10], Global Network Positioning [11]) are getting less and less effective. Indeed, these tools rely on ICMP which is more and more often disabled by administrators, once again to avoid Deny Of Service attacks based on flooding. For example, the **Skitter** project [12], which keeps track of the evolution of the macroscopic connectivity and performance of the Internet, reports that in 5 years of measurements the number of hosts replying to ICMP requests decreases by 2 to 3% per month.

Even if recent works have proposed similar or even better functionalities without relying on ICMP, some of them (e.g., pathchar [13]) require specific privilege on the machines, which make them unusable in our context. It is thus mandatory to rely on tools that only use *application-level measurements*, i.e., a measurement that can be done by any application running on a computing Grid without any specific privilege.

That kind of measurement comprises the common end-to-end measurements, like bandwidth and latency, but also interference measurements (i.e., whether a communication between two machines A et B has non negligible impact on the communications between two machines C et D). Many projects rely on "application-level" measurements.

An example is the NWS (Network Weather Service) [14] software, which constitutes a *de facto* standard in the Grid community as it is used by major *Grid middlewares* like Globus [15] or *Problem Solving Environments* (PSEs) like DIET [16], NetSolve [17], or NINF [18] to gather information about the current state of the platform as well as about its future evolutions. NWS is able to report the end-to-end bandwidth, latency and connection time, which are typical application-level measurements. However, the NWS project focuses on quantitative information and does not provide any kind of topological information. It is however natural to address this issue by aggregating all NWS information in a single clique graph and use this labeled graph as a network model.

In another example, interference measurements have been successfully used in ENV [19] and enabled to detect, to some extent, whether some machines are connected by a switch or a hub.

A last example is ECO [20], an efficient collective communication library, that uses plain bandwidth and latency measurements to propose optimized collective communications (e.g., broadcast, reduce, etc.). These approaches have proved to be very effective in practice, but they are generally very specific to each problem.

B. Studied Reconstruction Algorithms

Application-level measurements are thus the measurements of choice in a Grid environment, and can lead to meaningful results. However, all previous projects are ad-hoc and a more general framework would enable any network-aware application to benefit from such information. In most of the previous works, the underlying network topology is either a clique [14], [20] or a tree [21], [19]. That is why we propose in the following to evaluate these three simple and widely-used reconstruction algorithms: clique, minimal spanning tree on latencies, maximal spanning tree on bandwidths.

III. Assessing the Quality of Reconstructions

We want to thoroughly assess the quality of the reconstruction algorithms. To compare fairly various topology mapping algorithms, we have developed ALNeM (Application Level Network Mapper). ALNeM is developed with GRAS [5] that provides a complete API to implement distributed application on top of heterogeneous platforms. Thanks to two different implementations of GRAS, ALNeM can work seamlessly on real platforms as well as on simulated platforms with SimGrid [22]. ALNeM is made of three main parts:

1) a measurement repository (MySQL database);
2) a distributed collection of sensors performing bandwidth, latency and interference measurements;
3) a topology builder with some reconstruction algorithms that use the repository.

The evaluation of the quality of model builders is not an easy task. To perform such an evaluation, we use three different and complementary approaches. For each approach, we will consider a series of original platforms; and for each of these platforms we will

compare the original platform and the models built from it.

The three approaches can be seen as different point of views on the models: a structural one, a communication-level one, and an application-level one.

A. Visual Evaluation

This evaluation is the most subjective one. We simply display side-by-side the graph of the original platform and the model graph. Then we visually check whether the two graphs match.

B. End-to-End Metric

A platform model is "good" if it allows to accurately predict the running time of applications. The accuracy of the prediction depends on the model capacity to render different aspects and characteristics of the network. Most of the time, researchers only focus on bandwidth predictions. However, latencies and interferences can also greatly impact an application performance. Therefore, we consider the three following characteristics:

1) Bandwidth: This is the most obvious characteristic. We need to know the bandwidth available between processors as soon as the different tasks of an application, or the different applications run concurrently, send messages of different lengths.

2) Latencies: Obviously, latencies are very important for small messages. They are, however, often overlooked in the context of Grid computing, because of the usual assumption that in this framework processes only exchange large messages. Casanova presented an example [23] on the TeraGrid platform where one third of the time needed to transfer a 1 GByte of data would be due to latencies. Therefore, latencies cannot always be neglected even for large messages, and models must be able to predict them accurately. In practice, latencies can range from 0.1 ms for intra-cluster communications, to more than 300 ms for intercontinental satellite communications. Applications must be aware of the magnitude of the latencies to be able to organize their communications efficiently.

3) Interferences: Many distributed applications use collective communications (e.g., broadcasts or all-to-all) or, more generally, independent communications between disjoint pairs of processors. The only knowledge of the available latencies and bandwidths between any two pairs of processors does not allow to predict the time needed to realize two communications between two disjoint pairs of processors. Indeed, this depends on whether the two communications use a same physical link[1]. Legrand, Renard, Robert, and Vivien have shown [2] that knowing the network topology, and thus being able to predict communication interferences, enable to derive algorithms far more efficient in practice.

Methodology: Our evaluation methodology is based on simulations. Given one original platform, we measure the end-to-end latencies and bandwidths between any two pairs of processors. We also measure the end-to-end bandwidths obtained when any two pairs of processors simultaneously communicate. We then perform the same measurement on the reconstructed models, and we compare the results. This approach enables us to build a quality index for each reconstruction algorithm, for each graph, and for each studied network characteristic. The index for latencies and bandwidths is built as follow. We compute the ratios between the metric measured on the reconstructed platform and on the original one for each pair of nodes. Then we keep as a summary the minimum, maximum and geometric mean of these ratios. The index for interferences is the number of correct interferences predictions, false interferences predictions and false independence predictions versus the actual number of interferences.

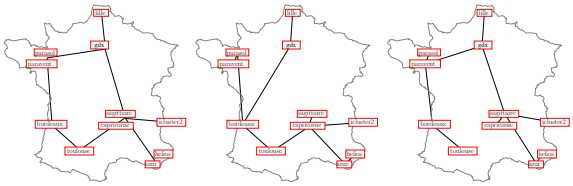
C. Application-Level Measurements

To simultaneously analyze a combination of the characteristics studied with end-to-end measurements, we also compare, through simulations, the performance of several classical distributed routines when run on the original graph and on each of the reconstructed graphs. This allows us to evaluate the predictive power of the reconstruction algorithms with applications with more complex but realistic communication patterns. This approach gives us an evaluation of the quality of reconstructions at the application level, rather than at a single communication level like end-to-end measurements.

We study the following simple distributed algorithms:

- **Token ring:** a token circulates three times along a randomly built ring (the ring structure has a priori no correlation with the interconnection network structure).
- **Broadcast:** a randomly picked node sequentially sends the same message to all the other nodes.

[1] In some cases, two communications sharing the same physical communication link do not interfere with each other. This may happen, for example, when the only shared communication links are backbones, as exemplified by Casanova [23].



(a) Real platform (b) TreeLat (c) TreeBW

1: Topologies reconstructed by the spanning tree algorithms on the G5K platform, from real measurements.

- **All-to-all:** all the nodes simultaneously perform a broadcast.
- **Parallel matrix multiplication:** a matrix multiplication is realized using ScaLAPACK outer product algorithm [24].

The evaluation can only be done through simulations. Indeed, the measurements on the reconstructed models can obviously not be done experimentally. Furthermore, the comparison of experimental (original platform) and simulated (reconstructed models) measurements would introduce a serious bias in the evaluation framework, the bias due to the differences between the actual world and the simulator.

IV. Experiments on a Real Platform

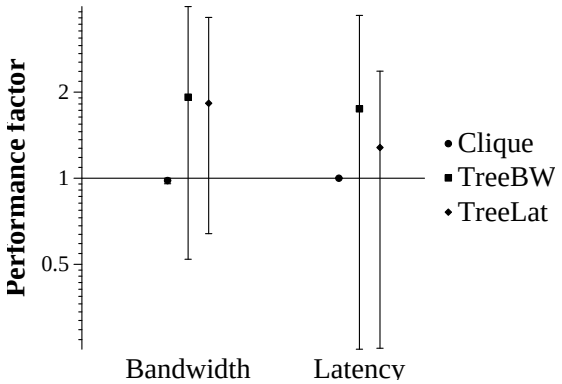
The Grid'5000 project[2] aims at building a highly reconfigurable, controllable and monitorable experimental Grid platform gathering 9 sites geographically distributed in France. Its main purpose is to serve as an experimental testbed for research in Grid Computing.

We have performed all latency and bandwidth measurements on this platform, and passed the results to the reconstruction algorithms. We evaluate the graphs produced with a graphical evaluation, and end-to-end measurements.

A. Graphical Evaluation

The topologies reconstructed by the spanning tree algorithms are shown on Figure 1. We can observe that the result is quite close to the original platform graph, though some links are missing, as it was expected. We can also note that the latency-based reconstruction does not look as good as the bandwidth one: it has added one link that is not in the original platform. This is certainly because latency measures are less stable; furthermore, as a Grid network, the infrastructure of Grid'5000 is more focused on optimizing bandwidth than latency.

[2] http://www.grid5000.fr



2: End-to-end tests on the Grid'5000 platform.

B. End-to-End Evaluation

The end-to-end evaluations were performed thanks to simulations: we simulated latency and bandwidth measurements on the reconstructed platforms, and compared them to the real measurements. The results are shown on Figure 2. As expected, the clique algorithm gets very good results, since it specifies all the values. On the other hand, tree-based algorithms tend to over-estimate the latencies. This can be explained by the fact that most paths are longer in the reconstructed tree, since some shortcuts are missing. However, the minimum ratio shows that some latencies are under-estimated: this comes from the fact that the routing in Grid'5000 is not done to optimize latency, and our algorithms have discovered paths that have lower latency than the actual paths used in Grid'5000.

The tree-based algorithms tend to over-estimate bandwidths as well. It is important to know though that in the simulation, the bandwidth measurements depend on a "window-size" parameter that describes how latency may limit the available bandwidth. This parameter is a constant for the whole simulation, while it seems that these values are different on the different clusters of Grid'5000. It is thus not possible, for now, to reproduce all real-life latency/bandwidth values in the simulator.

V. Experiments on Simulation

The evaluation of topologies cannot be based purely on end-to-end measurements: they are too biased towards cliques, which do not accurately represent the actual topology of the network. To perform the more informative applicative measurements, we need to use only simulations. We present here two types of experiments: the first one is based on a modeling of a real network architecture, while for the second one we have generated synthetic platforms using GridG [25].

(a) Real platform (b) TreeLat (c) TreeBW

3: Topologies reconstructed by the spanning tree algorithms on the Renater platform.

A. Renater

Renater[3] is the French public network infrastructure that connects all major universities. We have created a model of a part of this network, by selecting a dozen meaningful nodes and the corresponding links, which we have annotated with bandwidth and latency values available on the Renater information website. The original and reconstructed topologies are shown on Figure 3. Once again, the reconstructed graphs are very close to the original one, but since latency measures are much more stable in simulations, the result of TreeLat is just as good as TreeBW. Of course, since these algorithms build trees, we expect that these platforms will not model interferences accurately.

Figure 4 shows the evaluation of the reconstructed topology through simulation. The plots show the minimum, maximum and geometric means on a logarithmic scale.

The ratios are plotted in the same way for applicative measurements on Figure 4b. We can observe here that although end-to-end measurements are quite close to the original ones (the minimum and maximum can be quite far off, but the geometric means are close to 1 because most values are accurate), the differences in the topologies yield very bad results for the applicative running time. This is especially true for applications which perform several communications in parallel, like PMM or ALL2ALL. The platforms produced by both spanning tree algorithms create additional interferences, and thus lead to running times that can be more than twice the original value. On the other hand, reconstructing the platform as a clique removes all interferences between parallel communications, which leads to a much smaller predicted running time.

The third part of the figure shows the result of the interference tests. Tree-based algorithms correctly detect almost all of the actual interferences, but also add a large number of interferences that are not present in the original platform. The clique algorithm does the opposite, since it detects almost no interference — neither real nor false.

B. GridG

The GridG synthetic platform generator [25] allows the study of various types of platforms, which may be different from the ones we can access and thus test directly. In this experiment, we have generated 15 different platforms, using GridG's default parameters, each of them containing about 40 hosts. The results are shown on Figure 5, and are quite similar to the ones obtained with the Renater platform. This indicates that the classical tree- and clique-based algorithms are not suited to discovering real network topologies.

VI. Conclusion

In this work, we have designed a thorough evaluation framework for topology reconstruction algorithms. We have developed ALNeM, an application-level measurement and reconstruction infrastructure, which is freely available[4]. We have used this framework to evaluate classical reconstruction algorithms (namely spanning trees and cliques) and shown both through real experiments and simulations that none of these algorithms is fully satisfying in a Grid context. Our future work is to propose new practical algorithms and evaluate them within the same framework.

References

[1] I. Foster, *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004.

[2] A. Legrand, H. Renard, Y. Robert, and F. Vivien, "Mapping and load-balancing iterative computations on heterogeneous clusters with shared links," *IEEE Trans. Parallel Distributed Systems*, vol. 15, no. 6, pp. 546–558, 2004.

[3] P.-K. Chouhan, H. Dail, E. Caron, and F. Vivien, "Automatic middleware deployment planning on clusters," *International Journal of High Performance Computing Applications*, vol. 20, no. 4, pp. 517–530, Nov. 2006.

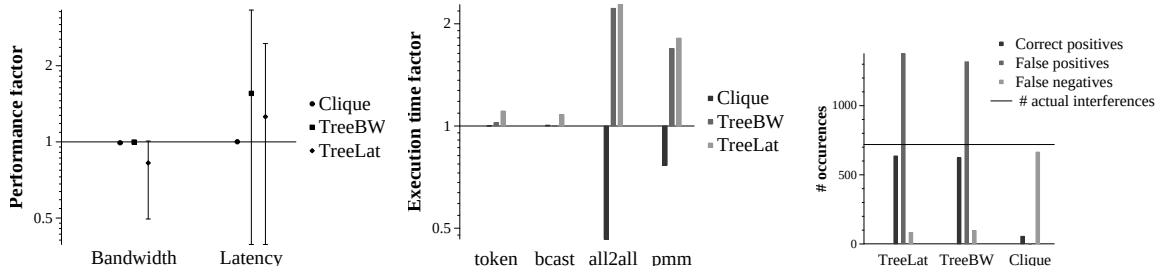
[4] T. Kielmann, R. F. H. Hofman, H. E. Bal, A. Plaat, and R. A. F. Bhoedjang, "MagPIe: MPI's collective communication operations for clustered wide area systems," *ACM SIGPLAN Notices*, vol. 34, no. 8, pp. 131–140, 1999.

[5] M. Quinson, "GRAS: A research & development framework for grid and P2P infrastructures," in *18th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2006)*, 2006.

[6] P. Dinda, T. Gross, R. Karrer, B. Lowekamp, N. Miller, P. Steenkiste, and D. Sutherland, "The architecture of the remos system," in *10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10)*, Aug. 2001.

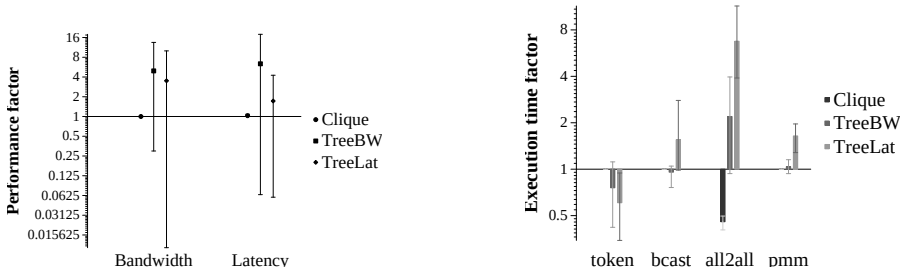
[3] http://www.renater.fr

[4] http://gforge.inria.fr/plugins/scmcvs/cvsweb.php/contrib/ALNeM/?cvsroot=simgrid



(a) End-to-end metrics (b) Applicative metrics (c) Interference metrics

4: Simulated tests on the Renater platform.



(a) End-to-end metrics (b) Applicative metrics

5: Simulated tests on the GridG platforms

[7] N. Miller and P. Steenkiste, "Collecting network status information for network-aware applications," in *INFOCOM'00*, 2000, pp. 641–650.

[8] M. den Burger, T. Kielmann, and H. E. Bal, "TOPOMON: A monitoring tool for grid network topology," in *International Conference on Computational Science (ICCS 2002)*, vol. 2330. Amsterdam: LNCS, Apr. 2002, pp. 558–567.

[9] H. Burch, B. Cheswick, and A. Wool, "Internet mapping project," http://www.lumeta.com/mapping.html, Lumeta Corporation.

[10] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "Idmaps: A global internet host distance estimation service," *IEEE/ACM Transactions on Networking*, Oct. 2001.

[11] E. Ng and H. Zhang, "Predicting internet network distance with coordiantes-based approaches," 2001.

[12] "The cooperative association for internet data analysis," http://www.caida.org/.

[13] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *Measurement and Modeling of Computer Systems*, 1999, pp. 222–223.

[14] R. Wolski, N. T. Spring, and J. Hayes, "The Network Weather Service: A distributed resource performance forecasting service for metacomputing," *Future Generation Computing Systems, Metacomputing Issue*, vol. 15, no. 5–6, pp. 757–768, Oct. 1999.

[15] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal of Supercomputing Applications*, vol. 11, no. 2, pp. 115–128, 1997.

[16] E. Caron and F. Desprez, "Diet: A scalable toolbox to build network enabled servers on the grid," *International Journal of High Performance Computing Applications*, vol. 20, no. 3, pp. 335–352, 2006.

[17] H. Casanova and J. Dongarra, "NetSolve: A Network-Enabled Server for Solving Computational Science Problems," *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, no. 3, pp. 212–223, 1997.

[18] T. Suzumura, H. Nakada, M. Saito, S. Matsuoka, Y. Tanaka, and S. Sekiguchi, "The ninf portal: An automatic generation tool for grid portals," in *Proceeding of Java Grande 2002*, Nov. 2002, pp. 1–7.

[19] G. Shao, F. Berman, and R. Wolski, "Using effective network views to promote distributed application performance," in *International Conference on Parallel and Distributed Processing Techniques and Applications*, June 1999.

[20] B. Lowekamp and A. Beguelin, "ECO: Efficient collective operations for communication on heterogeneous networks," in *10th International Parallel and Distributed Processing Symposium (IPDPS'96)*, 1999.

[21] A. Bestavros, J. Byers, and K. Harfoush, "Inference and labeling of metric-induced network topologies," Boston University, Computer Science Department, Tech. Rep. 2001-010, June 2001.

[22] A. Legrand, M. Quinson, K. Fujiwara, and H. Casanova, "The SimGrid project - simulation and deployment of distributed applications," in *Proceedings of the IEEE International Symposium on High Performance Distributed Computing (HPDC-15)*. IEEE Computer Society Press, 2006.

[23] H. Casanova, "Modeling large-scale platforms for the analysis and the simulation of scheduling strategies," in *18th International Parallel and Distributed Processing Symposium*, Apr. 2004.

[24] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*. SIAM, 1997.

[25] D. Lu and P. Dinda, "Synthesizing realistic computational grids," in *Proceedings of ACM/IEEE Supercomputing 2003 (SC 2003)*, Nov. 2003.