Visualization of Parallel Applications: Results of an International Collaboration

Lucas Mello Schnorr[1 2], **Guillaume Huard**[2], **Philippe O. A. Navaux**[1]

[1] Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[2]INRIA Moais research team, CNRS LIG Laboratory, Grenoble University
ZIRST 51, avenue Jean Kuntzmann - 38330 - Montbonnot Saint Martin - France

{lmschnorr,navaux}@inf.ufrgs.br, Guillaume.Huard@imag.fr

Abstract. *The performance visualization of parallel applications is a necessary step for a better comprehension of application behavior and its correspondence with the execution environment. In the scope of grid environments, there are new challenges for the visualization of parallel applications, mainly related to the visual observation of the network influence and the scalability of the visualization techniques. We present in this paper two proposals that address these problems. We also present the details of the international collaboration that allowed the effective development of this work.*

1. Introduction

The visualization analysis of parallel applications appeared almost at the same time the first graphical screens were available. As parallel and distributed systems evolved, so did the visualization techniques. Today, there are considerable challenges to the visual analysis of parallel applications, especially when we consider grid environments characteristics and their influences over applications. Grid platforms, such as the Grid'5000 [Capello et al. 2007], have highly hierarchical network interconnections that might influence parallel applications behavior. Current visualization techniques are not able to properly identify problems of applications that are caused by the network. Another characteristic of Grids is their size, making the development of larger applications possible. In this context, traditional visualization techniques are not able to visualize applications with thousands or even millions of processes.

We show here two proposals to address these issues. The first one is focused on the network influence over parallel applications, where our proposal resides in a new three dimensional visualization technique that is able to show the network topology and its relation with the application's components behavior. The second one uses an information visualization technique called Treemap [Johnson and Shneiderman 1991] to the analysis of parallel applications with hierarchical organization of monitoring data. This second approach attacks the problem of visualization scalability present in most of available visualization tools.

The thesis of the first author is developed within the scope of a collaboration between the Federal University of Rio Grande do Sul (UFRGS), located at Porto Alegre, Brazil, and the Grenoble Institute of Technology (INPG), in France. The collaboration had the financial support by the CAPES/Cofecub Project 4602/06-4, with an established *co-tutelle* aggreement between the two universities. During 18 months, the first author was located in France at the INRIA MOAIS Project Team, part of the CNRS Lig Laboratory. The main results of the joint work were published in the IEEE/ACM Grid'2008 [Schnorr et al. 2008] and the IEEE/ACM CCGrid'2009 conferences [Schnorr et al. 2009]. The collaborations between Brazil and Grenoble in the performance visualization area started in the middle of the 90's, with the thesis of Benhur de Oliveira Stein, advised by Jacques Chassin de Kergommeaux. The current collaboration is an effort in keeping joint works in this area.

The rest of the paper is divided in 4 sections. Section 2 briefly lists related work. Section 3 presents the proposal of an abstract model for the two new visualization techniques. Section 4 presents the implementation of the model in the prototype *3va*. We end the paper with a conclusion.

2. Contextualization

Several visualization tools were developed in the last 25 years to the analysis of parallel applications. A relevant set of tools is composed of ParaGraph (1990), Paradyn (1995), Vampir (1996), Pajé [Stein et al. 2000] and ParaProf (2003). Most of these tools are focused in the analysis of parallel applications executed in homogeneous and controlled environments, such as clusters. If we consider their visualization techniques in grid platforms, we observe the lack of support for important characteristics of grids that might influence the analysis. Next section details these problems and our approach to them.

3. Visualization Techniques Developed

Parallel application analysis, when applied to distributed environments such as grids, poses a new set of problems to be solved. These problems are related to how the grid is organize and its characteristics, such as the heterogeneity of hardware and software, dynamism of resources and larger size. The techniques developed to visualize parallel applications behavior were mostly focused in cluster-like environments, viewed as a more controlled and homogeneous environment.

Considering this context of the need of visualization techniques for the analysis of grid applications, we isolated two key problems. The first one is related to the importance of analysing applications taking into account network characteristics. The topology, bandwidth and latency of the network might have an important influence over the behavior of the programs. This happens especially in network-bound applications, where the amount of data being transferred or the need for low-latency communications is determinant. By analysing related work, we can observe that none of existing visualization techniques for application analysis show in a clear form the correlation between application and network use. The only tool that had in some way this analysis was ParaGraph, but it was built only for small-scale parallel applications.

The second problem is related to the size of grid parallel applications and the visualization techniques used for their analysis. Grids might scale up to thousands and possibly millions of resources. The Grid'5000 platform [Capello et al. 2007], for instance, was conceived to be composed of 5000 processing cores. Even if this project limitation exists, the addition of new resources can continue indefinitely. Therefore, the potential utilization of these resources increases resulting in larger applications that might attain millions of processes. Most of the visualization techniques that attack the problem of analysing applications of this size work by grouping, reducing and filtering out processes behavior to later plot them using classical visualization techniques.

With the main goal of trying to solve these problems, we have proposed two visualization techniques. The first one, focused in the problem of the network influence over the application, resulted in the proposal of a three dimensional visualization technique that uses two dimensions to render a network topology and the third dimension as timeline [Schnorr et al. 2008]. This allows developers to be able to track up when the limitations of the network influence the application. The second visualization technique proposed is focused in the problem of analysing large-scale applications. Our proposal in this case uses an information visualization technique called Treemap [Johnson and Shneiderman 1991], combined with a hierarchical organization of monitoring data, to depict the behavior of processes in different time intervals [Schnorr et al. 2009].

Figure 1 depicts the proposed component model used to generate the new visualizations techniques for parallel applications. The input of the model is depicted at right, composed by the traces from the parallel applications and the resources description. The two outputs are at right, as a form of a 3D Visualization (bottom right) or a Treemap View (top right). The flow of data occurs from left to right, mainly influenced by new monitoring data arriving or changes in the resources description. Configurations and interaction mechanisms act from the visualizations to the middle components.

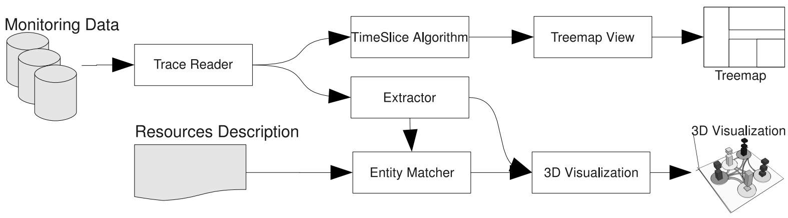

Figure 1. Model of the proposed solutions.

4. The *3va* Prototype

The model created for the proposed visualization techniques is implemented in a prototype called *3va*, that stands for Three Dimensional Visualization Analysis. The prototype is developed in the Objective-C and C++ languages, with each component implemented separately. The trace reader was implemented using the API defined by the DIMVisual library [Schnorr et al. 2006], and the Pajé Simulator [Stein et al. 2000]. The two visualizations components (Treemap and 3D), were implemented from scratch using the Wxwidgets and Ogre3D libraries. The components that implemented most of the algorithms proposed in the model (TimeSlice Algorith, Extractor and Entity Matcher) were fully implemented using the Objective-C language. Two resources description were used in this work: network topology and hierarchical structure. The monitoring data input was mostly composed of traces from parallel applications developed with the KAAPI library [Gautier et al. 2007], although other types of traces, such as from MPI applications, can be easily adapted to the be used within *3va*. Figure 2 shows the 3D visualization generated by the prototype.

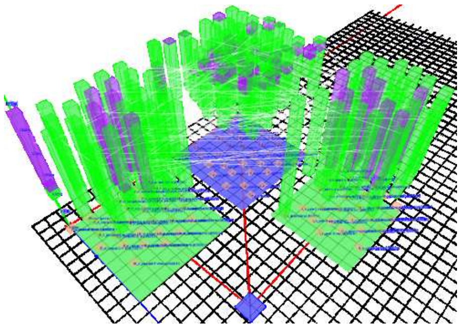

Figure 2. 3D visualization generated by the prototype *3va*.

5. Conclusion

Grid applications have particular characteristics that current visualization tools are not able to show. This paper focused on two of them: the visual analysis of the network topology influence over the parallel applications and the visualization scalability. Our approach to these issues is to use a three dimensional visualization to be able to show network topology and application behavior together; and to use hierarchical organization of monitoring data with the treemap technique to be able to analyze larger applications. We have developed a prototype that implements these approaches and generated results with the analysis of KAAPI applications executed in the Grid'5000 platform. As future work, we intend to refine the mechanisms used to create the visualizations to turn more easy the analysis. This work is developed within the scope of a collaboration Brazil/France, funded by a CAPES/Cofecub project. We support the continuity of this type of collaborations because of the positive aspects of this international environment to research and the know-how acquired over the years by both sides in the performance visualization area.

References

Capello, F., Plateau, B., Priol, T., Vandromme, D., Cappello, F., Dayde, M., Desprez, F., Jeannot, E., Jegou, Y., Lanteri, S., Melab, N., Namyst, R., Quinson, M., Richard, O., and Primet, P. V.-B. (2007). Grid5000. http://www.grid5000.fr. Last accessed April, 24, 2009.

Gautier, T., Besseron, X., and Pigeon, L. (2007). Kaapi: A thread scheduling runtime system for data flow computations on cluster of multi-processors. In *Proceedings of the international workshop on Parallel symbolic computation*, pages 15–23, New York, NY, USA. ACM.

Johnson, B. and Shneiderman, B. (1991). *Tree-Maps: a space-filling approach to the visualization of hierarchical information structures*. IEEE Computer Society Press Los Alamitos, CA, USA.

Schnorr, L. M., Huard, G., and Navaux, P. O. A. (2008). 3d approach to the visualization of parallel applications and grid monitoring information. In *Proceedings of the 9th IEEE/ACM International Conference on Grid Computing*, Washington, DC, USA. IEEE Computer Society.

Schnorr, L. M., Huard, G., and Navaux, P. O. A. (2009). Towards visualization scalability through time intervals and hierarchical organization of monitoring data. In *The 9th IEEE International Symposium on Cluster Computing and the Grid, CCGRID*. IEEE Computer Society.

Schnorr, L. M., Navaux, P. O. A., and de Oliveira Stein, B. (2006). Dimvisual: Data integration model for visualization of parallel programs behavior. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, pages 473–480, Washington, DC, USA. IEEE Computer Society.

Stein, B., de Kergommeaux, J. C., and Bernard, P.-E. (2000). Pajé, an interactive visualization tool for tuning multi-threaded parallel applications. *Parallel Computing*, 26:1253–1274.