

Inexpensive and Scalable High Performance Computing: the OurGrid Approach¹

Francisco Brasileiro
fubica@dsc.ufcg.edu.br

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Laboratório de Sistemas Distribuídos
Av. Aprígio Veloso, s/n, Bloco CO
58.109-970, Campina Grande - PB, Brazil

High performance computing over public resources and computational grids has recently become a reality with several voluntary computing and grid initiatives currently in operation. However, the mainstream solutions available to implement such systems require substantial computational infrastructure support to allow the aggregation of a large number of resources spread over many administrative domains. To make high performance computing affordable even for those users that cannot rely on having a highly qualified computing support team, we have proposed, implemented and deployed OurGrid. OurGrid is a peer-to-peer grid middleware that supports the automatic creation of large computational grids for the execution of embarrassingly parallel applications. It has been used to support the OurGrid Community (<http://status.ourgrid.org/>), a public free-to-join grid and the ShareGrid, “a collaborative project, coordinated by TOPIX (TOrino Piemonte Internet eXchange) in the framework of the Innovation Development Program and funded by Regione Piemonte, aimed at providing a computing and storage service to the academic research community” (<http://dcs.di.unipmn.it/>).

OurGrid takes opportunistic computing, pioneered by the Condor project [1,2], to a new dimension, potentially allowing thousands of peers to form a collaborative community for the exchange of idle computing capacity [3]. This is possible, first of all, because OurGrid incorporates an incentive mechanism – named the “Network of Favors” – that makes in the best interest of each peer to donate as much idle resources as possible, discouraging free-riding [4]. Also important is its ability to discover idle resources in a very efficient way, thanks to the use of NodeWiz, a highly scalable and fault-tolerant peer-to-peer Grid Information Service that is specialized in answering multi-attribute range queries [5,6]. In this large scale, heterogeneous and very dynamic grid, attain accurate information about resources and applications to perform appropriate application scheduling is an important issue. OurGrid solves this problem by trading cycles for information. Its Workqueue with Replication (WQR) scheduler uses no information about applications or resources and instead uses judicious replication on the assignment of tasks to resources so to achieve performance comparable to the traditional bin-packing schedulers that require full and accurate information about the system [7]. Finally, OurGrid provides a security portfolio that encompasses mechanisms with different levels of assurance and deployment difficulty, leaving to the user the option for selecting the correct tradeoff between security guarantees and cost, therefore, welcoming a very large spectrum of users to join the community.

Since the OurGrid Community appeared as a production grid, many users have benefited from its technology. Applications executing over the OurGrid Community

¹ This project is developed in cooperation with Hewlett-Packard Brazil R&D.

concerns different areas including several kinds of simulations, molecular dynamics [8], climate forecast [9], hydrological management [9], medical images processing [10], and data mining [11]. These applications share two common characteristics: they have high processing demand and can be parallelized as a Bag-of-Tasks application. OurGrid applications can be categorized by the way they interact with the middleware. There are two main approaches: script-based and embedded applications. Script-based applications are the simplest ones. These applications comprise the execution of several tasks, in which each task can be defined as the execution of a program that reads input data from a file and outputs its result to another file. In this case the application is written as a text file that contains entries for all tasks that comprise the application, each entry being described as follows: i) a stage-in command used to transfer input data and executable code to the grid machine; ii) the command that will be executed at the grid machine; iii) a stage-out command that transfers the output of the execution to the user's machine; and, iv) a validation command that runs a sanity check on the results produced, so to detect bogus responses generated by faulty or malicious resources. In most cases it is possible to use a script to automate the generation of the application description file, since there may be hundreds or thousands of tasks per application. Embedded applications incorporate calls to the OurGrid middleware Java API. The use of grid resources is transparent to the application users. All the grid complexity is hidden from them. This approach is normally used when the application is more complex than those exemplified before. When using the OurGrid API directly, users are able to access sophisticated features that are not available through the simpler script-based interaction mode.

References

- [1] D. Thain, T. Tannenbaum, M. Livny. Distributed Computing in Practice: The Condor Experience. *Concurrency and Computation: Practice and Experience*, Vol. 17(2-4), pp. 323-356, February-April 2005.
- [2] A. R. Butt, R. Zhang, Y. C. Hu. A Self-Organizing Flock of Condors, *Journal of Parallel and Distributed Computing*, Vol. 66 (1), pp. 145-161, January 2006.
- [3] W. Cirne, F. Brasileiro, N. Andrade, L. Costa, A. Andrade, R. Novaes, M. Mowbray. Labs of the World, Unite!!!, *Journal of Grid Computing*, Vol. 4 (3), pp. 225-246, September 2006.
- [4] N. Andrade, F. Brasileiro, W. Cirne, M. Mowbray. Automatic Grid Assembly by Promoting Collaboration in Peer-to-Peer Grids, *Journal of Parallel and Distributed Computing*, Vol. 67(4), pp. 957-966, August 2007.
- [5] S. Basu, S. Banerjee, P. Sharma, S.-J. Lee. Nodewiz: Peer-to-Peer Resource Discovery for Grids, *Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005)*, IEEE Computer Society, Washington, DC, USA, pp. 213-220, May 2005.
- [6] F. Brasileiro, L. B. Costa, A. Andrade, W. Cirne, S. Basu, S. Banerjee. A Large Scale Fault-tolerant Grid Information Service, *Proceedings of the 4th International Workshop on Middleware for Grid Computing (MGC 2006)*, ACM Press, New York, NY, USA, p. 14, November 2006.
- [7] W. Cirne, F. Brasileiro, D. Paranhos, L. Góes, W. Voorsluys. On the Efficacy, Efficiency and Emergent Behavior of Task Replication in Large Distributed Systems, *Parallel Computing*, Vol. 33(3), pp. 213-234, April 2007.
- [8] C. Veronez, C. Osthoff, P. Pascutti. HIV-I Protease Mutants Molecular Dynamics Research on Grid Computing Environment. *Proceedings of Brazilian Workshop on Bioinformatics (WOB 2003)*, pp. 161-164, October 2003.

- [9] William Voorsluys, Eliane Araújo, Walfredo Cirne, Carlos O. Galvão, Enio P. Souza, Enilson P. Cavalcanti. Fostering Collaboration to Better Manage Water Resources, *Concurrency and Computation: Practice and Experience*, Vol. 19(12), pp. 1609–1620, August 2007.
- [10] M. Oliveira, P. Marques, W. Cirne. Grid Computing in the Optimization of Content-Based Medical Images Retrieval, *Radiologia Brasileira*, Vol. 40, pp. 255-262, July 2007.
- [11] Fabrício A. B. da Silva, Sílvia Carvalho, Hermes Senger, Eduardo R. Hruschka, Cléver R. G. de Farias. Running Data Mining Applications on the Grid: a Bag-of-Tasks Approach, *Proceedings of the International Conference on Computational Science and its Applications (ICCSA 2004)*, Springer Berlin, Heidelberg , Germany, pp. 168-177, May 2004.