Using MyGrid to Run Bag of Tasks Applications on Computational Grids

Walfredo Cirne Daniel Paranhos Lauro Costa Elizeu Santos-Neto
Nigini Abílio Nazareno Andrade Francisco Brasileiro Jacques Sauvé

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Campina Grande – PB – 58109.970
{walfredo,danielps,lauro,elizeu,nigini,nazareno,fubica,jacques}@dsc.ufcg.edu.br

Fabrício Alves Barbosa da Silva

UniSantos – Universidade Católica de Santos
R. Dr. Carvalho de Mendonça nº 144, sala 405
Santos – SP – CEP 11070-906.
fabricio@unisantos.br

Carla Osthoff Barros

Laboratório Nacional de Computação Científica
osthoff@lncc.br

1 Introduction

Bag of Tasks (BoT) applications are those parallel applications whose tasks are independent of each other. Despite their simplicity, BoT applications are used in a variety of scenarios, including data mining, massive searches (such as key breaking), parameter sweeps [1], Monte Carlo simulations, fractal calculations (such as Mandelbrot), and image manipulation applications (such as tomographic reconstruction [6]). Moreover, due to the independence of its tasks, BoT applications can be successfully executed over widely distributed computational grids, as has been demonstrated by SETI@home [5]. In fact, one can argue that BoT applications are the applications most suited for computational grids, where communication can easily become a bottleneck for tightly-coupled parallel applications.

However, very few users of BoT applications are currently using computational grids, despite the potential dramatic increase in resources grids can bring to bear for problem resolution. We believe that this state of affairs is due to (i) the complexities involved in using grid technology, and (ii) the slow deployment of existing grid infrastructure. Today, one must commit considerable effort to make an application run efficiently on a grid. The user, who is ultimately interested in getting the application's results, seldom has the training or the inclination to deal with the level of detail needed to use current grid infrastructure. Furthermore, the existing grid infrastructure is not ubiquitously installed yet. Users often have access to resources that are not grid-ready.

This paper is a short description of MyGrid, a system designed to change this state of affairs. MyGrid aims to *easily* enable the execution of BoT application on *whatever resources the user has available*. MyGrid chooses a different trade-off compared to existing grid infrastructure. It forfeits supporting arbitrary applications in favor of supporting only BoT applications (which are relevant and amenable to execution on grids). By focusing on BoT applications, MyGrid can be kept simple to use, simple enough to be a solution for *real users*, who want to run their applications *today* and don't really care for the underlying grid support they might use.

This is not to say, however, that MyGrid is a replacement for existing grid infrastructure. MyGrid uses grid infrastructure whenever available. It simply does not depend on it. MyGrid can be seen more as a representative of the user in the grid. It provides simple abstractions through which the user can easily deal with the grid, abstracting away the non-essential details. It schedules the application over whatever resources the user has access to, whether this access is through some grid infrastructure (such as Globus [2]) or via simple remote login (such as `ssh`). MyGrid's scheduling solution is a particularly interesting contribution because it uses task replication to achieve good performance relying on no information about the grid or the application [4]. Note that not needing information for scheduling simplifies MyGrid usage. MyGrid is open source software, available at http://dsc.ufcg.edu.br/mygrid.

2 Design Goals

We intend MyGrid to be a production-quality solution for users who want to execute BoT applications on computational grids today. Note that "today" implies that we cannot assume that some new software will be widely deployed. Our design goals were thus established with this vision in mind. We want MyGrid to be *simple*, *complete*, *encompassing*, and *secure*.

By *simple*, we imply that MyGrid should be as close as possible to an out-of-the-box solution. The user wants to run her application. The least she gets involved into grid details, the better. Towards this goal, we worked on minimizing the installation effort. This is important because if the user had to manually install MyGrid on many machines, the simplicity of the solution would suffer.

Complete means that MyGrid must cover the whole production cycle, from development to execution, passing through deployment and manipulation of input and output. This goal is key for MyGrid to be useful in practice. In order to support all activities within the production cycle of a BoT application, MyGrid provides the notion of *working environment*, which consists of a small set of abstraction that enables the user to manipulate her files on the Grid.

Due to their loosely coupled nature, BoT applications can potentially use a very large number of processor. Therefore, we do not want MyGrid to preclude the user from using a given processor. We want MyGrid to be *encompassing*, in the sense that all machines the user has access to can be utilized to run her BoT applications. An important consequence of this goal is that MyGrid must be a user-centric solution, i.e. MyGrid cannot assume that some given software is installed in a machine for it to be employed in the computation. That is, we cannot assume that a particular grid infrastructure (e.g. Globus [2]) is ubiquitously installed on all machines a user has access to. Note also that, simplified installation also helps here. It does not suffice that the user potentially can employ all machines she has access to. It has to be simple to do so.

MyGrid should not weaken the user's security; otherwise it probably will not be used. Consequently, we want MyGrid to be *secure* in the sense that the user's security will not be compromised by a security problem in one of the many machines that compose the user's Grid.

3 MyGrid Approach

MyGrid attains its goals by (i) providing simple abstractions that enable the user to describe the application without having to know details of each grid machine (such as the file system organization), and (ii) introducing Work Queue with Replication (WQR), a scheduling heuristics that attains good performance without relying on information about the grid or the application, although consuming a few more cycles. Note that not depending on information makes WQR much easier to deploy in practice (since we do not have to monitor processors throughout the grid) and also much more user friendly (since users do not have to estimate a priori the execution time of their tasks).

MyGrid architecture was designed to enable the user to benefit from "whatever resources she has access to". This is done by defining the *Grid Machine Interface* as the minimal set of services that must be available in a machine for it to be used as a grid machine. These services are:

Service
Task start-up on a grid machine (remote execution)
Cancellation of a running task
File transfer from the grid machine to the home machine
File transfer from the home machine to the grid machine

Table 1 – Grid Machine Interface

There can be many ways to implement the Grid Machine Interface. Actually, Grid Machine Interface is a *virtualization* [3] of the access services to a grid machine. One way to implement the Grid Machine Interface lets the user furnish MyGrid with scripts that implement the four services listed in Table 1. In this case, MyGrid uses the *Grid Script* module to access the machine. Note that Grid Script enables the user to inform MyGrid on how to access a given machine in a very flexible way. As long as the user is able to translate "having access to a given machine" into "providing scripts can encapsulate such access", MyGrid will be able to use the machine.

4 Demo

During the WRNP2, we intend to demo MyGrid running integer factoring, scheduling simulations and a dynamic molecular application. The scheduling simulations are those used to study WQR itself. The dynamic molecular application is from Paulo Bisch's group at UFRJ. It uses using THOR (a package for modeling and molecular dynamics developed at UFRJ) to study interactions among HIV-1's protease and a family of its mutants with protease inhibitors.

The grid we going to use will include at least 58 processors, distributed throughout the following 7 independent computing sites:

- Carcara/LNCC/Petropolis/Brazil - 1 linux processor
- ApeLab/USCD/San Diego/USA - 20 solaris procecessors
- LSD/UFCG/Campina Grande/Brazil- 8 linux processor
- NACAD/UFRJ/Rio de Janeiro/Brazil - 1 linux processor
- NCE/UFRJ/Rio de Janeiro/Brazil - 4 linux processors
- LCP/UFRJ/Rio de Janeiro/Brazil - 14 linux processor
- GridLab/UCSD/San Diego/USA - 10 linux processors

Acknowledgments

We would like to thank HP, CNPq (grant 55.2153/02-8) and RNP for the financial support. Without it, it would have been impossible to carry out this work. We'd also like to thank Márcio Vasel, Nigini, Luciana and Nazareno for their great work in different parts of MyGrid.

References

[1] D. Abramson, J. Giddy and L. Kotler. *High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?* IPDPS'2000, pp. 520-528, Cancun Mexico, IEEE CS Press, USA, 2000. http://ipdps.eece.unm.edu/2000/papers/Abramson.pdf

[2] I. Foster and C. Kesselman. *The Globus Project: A Status Report*. Proceedings of IPPS/SPDP '98 Heterogeneous Computing Workshop, pg. 4-18, 1998. http://www.globus.org/research/papers.html

[3] I. Foster, C. Kesselman, J. Nick, S. Tuecke. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. June 22, 2002. http://www.globus.org/research/papers.html

[4] D. Paranhos, W. Cirne, and F. Brasileiro. *Trading Cycles for Information: Using Replication to Schedule Bag-of-Tasks Applications on Computational Grids*. Submitted for publication. http://walfredo.dsc.ufcg.edu.br/resume.html#publications

[5] SETI@home web page. http://www.seti.org/science/setiathome.html

[6] S. Smallen, W. Cirne and J. Frey et al. *Combining Workstations and Supercomputers to Support Grid Applications: The Parallel Tomography Experience*. Proceedings of the HCW'2000 - Heterogeneous Computing Workshop. 2000. http://walfredo.dsc.ufcg.edu.br/resume.html#publications