MobiGrid*– Framework for Mobile Agents on Computer Grid Environments

Rodrigo M. Barbosa[1], **Alfredo Goldman**[1]

[1]Department of Computer Science – Institute of Mathematics and Statistics – University of São Paulo
Rua do Matão, 1010 - Cidade Universitária – 05508-090 São Paulo, SP

rodbar@ime.usp.br, gold@ime.usp.br

Abstract. *This project focuses on the implementation of a framework for mobile agents support within a grid environment project, namely InteGrade. Our goal is to present a framework where time consuming sequential tasks can be executed using mainly the idle cycles of a network of personal workstations. The mobile agents may be used to encapsulate long processing applications (tasks). These agents can migrate whenever the local machine is requested by its user, since they are provided with automatic migration capabilities. Our framework also provides to the user a manager that keeps track of the agents submitted by him.*

1. Motivation: Mobile Agents on Computational Grids

The *grid* [Foster and Kesselman, 1999] paradigm was created to face two major problems of *clusters* and supercomputers: high prices and waste of resources. The *computational grid* idea was clearly inspired by *clusters*, in the sense that we have many computers interconnected by a network in order to provide together greater computational power. The idea is to provide computational resources similarly to the way we get power supply. The waste problem is addressed in a way that whenever a computer is idle, its computational power can be supplied to the *grid*. On this context, the idea of mobile agents can be interesting. They can be used to encapsulate opportunistic applications, which can use small slices of the available computational time of personal workstations, migrating to another machine whenever the local user requests his machine, always preserving the processing already done. On this way, mobile agents can be considered as a complementary tool to decrease even more the idle time of a *grid*.

This text proposes our solution to provide a mobile agents environment in a *grid* and is organized in the following way: Section 1 explains the motivations to provide mobile agents support in a *grid*; Section 2 describes our project objectives; Section 3 gives a general overview of the framework; Section 4 provides ideas for future work and concludes this paper.

2. Objectives

In an IBM's pioneer report *Mobile Agents: Are They a Good Idea?* [Chess et al., 1995], its authors analyze the potential of mobile agents and introduce, among other ideas, the possibility of employing mobile agents in order to use idle computational resources. Another

*Project supported by the State of São Paulo Research Foundation - FAPESP. Process number: 02/11165-6

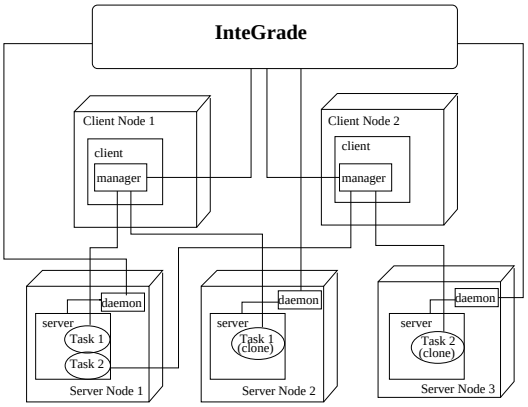

Figure 1: General architecture of the framework

report, *e-Gap Analysis* [Fox and Walker, 2003], which makes a study about the way scientists do science nowadays with technology support - what is named *e-Science* - points, among other gaps, the absence of mobile agents support in the existent grid infrastructures.

This project consists in the implementation of a mobile agents framework for InteGrade [Goldchleger et al., 2004]. The main idea of our framework is to allow an efficient utilization of computational resources for large sequential or for embarrassing parallel applications.

On this way, mobile agents can be used in a complementary way to InteGrade applications, allowing an even better utilization of computational resources. Among the applications that could be executed using mobile agents, there are loosely coupled parallel applications like SETI@home [SETI@home, 2004], or sequential applications that demand long processing time. Our goal is to provide a framework to use the grid idle resources not used by InteGrade.

3. Framework Overview

The main idea of our framework is to provide the programmer with a programming environment for long running applications, which we call *tasks*. The framework is being implemented in Java over Aglets [Aglets, 2004]. We will mention all the components of it at a high level:

1. *task*: long running application, encapsulated in a mobile agent. On the implementation of this *task*, the programmer must take care of the *task* state, since the standard Java environments for mobile agents provide weak migration [Illmann et al., 2000]. To help the programmer, our framework provides a special method. This method is used in order to inform the framework when the *task* has reached a consistent point, being ready for migration or clonation;
2. *manager*: it is the component responsible for registering *tasks*. The two most important functions of the *manager* are:
   (a) migration: when a *task* is submitted, the *manager* queries the InteGrade infrastructure searching for an idle machine which has a chance of remaining on this state for a given time. With this information, the *manager* dispatches the *task* to such a machine. A similar procedure is used when the *task* needs to migrate;
   (b) *liveness*: the *manager* also creates a clone of the *task* and dispatches it to another machine. In the context of our work, a *task* that is being executed

in more than one machine has *liveness*. When one of the clones dies, the *manager* makes a copy of the still alive clone and dispatches it to another machine.

3. *server*: this is the server installed on each machine that provides resources to the framework. It provides a execution environment for the *tasks*. When the machine is requested by the local user, the *server* asks the evacuation of the *tasks* that are being hosted by it;
4. *daemon*: it verifies whether the machine is idle or not. When the machine is idle, it communicates that to InteGrade and turns on the *server*. When the machine is requested by its local user, the *daemon* informs the *server*, which evacuates the *tasks* and terminates;
5. *client*: component that provides the user with tools to submit *tasks* to the framework. It also provides a host environment for the *manager*.

In Figure 1, we have an overview of our framework. Each one of the client nodes hosts a *manager*, which communicates with InteGrade. The *manager* of Client Node 1 manages Task 1 and its clone. The *manager* of Client Node 2 manages Task 2 and its clone. Observe also that the *daemons* communicate with InteGrade.

4. Conclusion

The idea of mobile agents in InteGrade is very instigating and interesting. Our framework, in the future, can be used to implement many kinds of applications from embarrassing parallel applications, like SETI@home, to applications that need many kind of resources not available in a single InteGrade node. Given the opportunistic characteristic of mobile agents, InteGrade can reach near zero idleness on its nodes, what is impossible nowadays. Another interesting point is our concern on building a framework that is transparent to the machine user, so he will not face a performance loss. A strong feature of our framework is its portability, since it is written almost all in Java. In the future, our framework can be extended to serve a grid infrastructure that manages different kinds of resources available on several machines.

References

Aglets (2004). Site of the Aglets project. `http://aglets.sourceforge.net`. Last visit on February, 2004.

Chess, D., Harrison, C., and Kershenbaum, A. (1995). Mobile Agents: Are They a Good Idea? Technical report, T.J. Watson Research Center.

Foster, I. and Kesselman, K. (1999). *The Grid: Blueprint for a new Computing Structure*, chapter 2,3,5,11. Morgan Kaufmann Publishers.

Fox, G. and Walker, D. (2003). e-Science Gap Analysis. Technical report, Indiana Univesity and Cardiff University.

Goldchleger, A., Kon, F., Goldman, A., Finger, M., and Bezerra, G. C. (2004). InteGrade: Object-Oriented Grid Middleware Leveraging Idle Computing Power of Desktop Machines. *Concurrency and Computation: Practice and Experience*, 16:449–459.

Illmann, T., Weber, M., Kargl, F., and Krüger, T. (2000). Migration in java: Problems, classification and solutions. In *Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA'00)*.

SETI@home (2004). Site of the SETI@home project. `http://setiathome.ssl.berkeley.edu/`. Last visit on February, 2004.