Chapter 14

GRID MULTICRITERIA JOB SCHEDULING WITH RESOURCE RESERVATION AND PREDICTION MECHANISMS

Krzysztof Kurowski, Jarek Nabrzyski, Ariel Oleksiak, Jan Weglarz
Poznan Supercomputing and Networking Center, Poland
naber@man.poznan.pl

Abstract

Grids link together computers, data, sensors, large scale scientific instruments, visualization systems, networks and people. They can provide very large pools of computer resources, enable distributed collaborations and deliver increased efficiency and on-demand computing capabilities. The complexity of Grids on one hand and the requirements towards performance and capability on the other hand call for efficient resource management and scheduling mechanisms. Such mechanisms must take into account not only the hardware and software resources, user jobs and applications, but also policies of the resource owners. Policies usually describe cost models for the resource usage, security mechanisms, quality of service of resource provisioning etc. The problem of scheduling jobs in real Grid environments is very difficult. Due to lack of time characteristics of jobs, and difficulties in characterizing the overall system, traditional OR techniques usually fail or achieve very weak results. Usually, best effort scheduling is the best option. There are, however, some ways to deal with the problems described above.

The main goal of this paper it to present some practical issues of scheduling Grid jobs. Methods and techniques described in the paper are used in a Grid scheduling system, called GRMS (Grid Resource Management System) developed at Poznan Supercomputing and Networking Center. GRMS is widely used in many Grid infrastructures worldwide.

Keywords: Grid computing, Grid resource management and scheduling, multicriteria decision support.

14.1 Introduction

Grid computing can be defined as coordinated resource sharing and problem solving in dynamic, multi-institutional collaborations. More simply, Grid computing typically involves using many resources (compute, data, I/O, instruments, etc.) to solve a single, large problem that could not be performed on any one resource. Often Grid computing requires the use of specialized middleware to mitigate the complexity of integrating of distributed resources within an Enterprise or as a public collaboration.

Generally, *Grid resource management and scheduling* is defined as the process of identifying requirements, matching resources to applications, allocating those resources, and scheduling and monitoring Grid resources over time in order to run Grid applications as efficiently as possible. Grid applications compete for resources that are very different in nature, including processors, data, scientific instruments, networks, and other services. Complicating this situation is the general lack of data available about the current system and the competing needs of users, resource owners, and administrators of the system.

Grids are becoming almost commonplace today, with many projects using them for production runs. The initial challenges of Grid computing–how to run a job, how to transfer large files, how to manage multiple user accounts on different systems–have been resolved to first order, so users and researchers can now address the issues that will allow more efficient use of the resources.

While Grids have become almost commonplace, the use of good Grid resource management tools is far from ubiquitous because of the many open issues of the field. Some of the issues include:

- **Multiple layers of schedulers.** Grid resource management involves many players and possibly several different layers of schedulers. At the highest level are Grid-level schedulers that may have a more general view of the resources but are very "far away" from the resources where the application will eventually run. At the lowest level is a local resource management system that manages a specific resource or set of resources. Other layers may be in between these, for example one to handle a set of resources specific to a project. At every level additional people and software must be considered.

- **Lack of control over resources.** Grid schedulers aren't local resource management systems; a Grid-level scheduler may not (usually does not) have ownership or control over the resources. Most of the time, jobs will be submitted from a higher-level Grid scheduler to a local set of resources with no more permissions than the user would have. This lack of control is one of the challenges that must be addressed.