

Coordinated Scheduling in Grid Environments

Ivan Rodero, Francesc Guim, Julita Corbalán, Jesús Labarta
Technical University of Catalonia (UPC), Spain
{irodero, fguim, juli, jesus}@ac.upc.edu

22nd February 2005

Introduction

Grid Computing [IFCKST01] has emerged in recent years providing a way to perform parallel computing in distributed computational resources. Furthermore, the Grid allows executing jobs in different administrative domains and sharing their existent HPC (High Performance Computing) resources. In order to perform job scheduling, job management and resource management at Grid level, usually it is used a metascheduler or a Resource Broker. Typically, the **Resource Broker** is on top of the Grid infrastructure and it tries to respect the local computational resources policies of each administrative domain and its **autonomy**.

In this paper we propose an architecture that allows that the resource broker schedules in **coordination** with the Local Resource Manager and its local schedulers. In particular, we are interested in enabling the broker to obtain information about the dynamic behavior of applications that are running on the local computational resources. Since the Grid technology area is very wide, this project is targeted to High Performance Applications (OpenMP, MPI and MPI+OpenMP) executed on a Grid composed of parallel machines, shared-memory architectures (SMP and CC-NUMA) with a medium to high number of processors.

The proposed architecture is composed for several coordinated components; their characteristics and design goals are exposed on the further sections. The main objective of our system is to perform coordinated scheduling between different levels.

Related Work

To the best of our knowledge, any developed systems have yet been developed implementing coordinated scheduling between Grid level and other lower levels managing MPI+OpenMP applications as we propose in this paper. An emerging research group at GGF is working in the Grid scheduling architecture[GSA05] and it is introducing some concepts of coordinated scheduling but it is only a preliminary work.

Currently, there are available some different projects that are working on external schedulers for queuing systems, such as MAUI [MAUI05]. In fact, our first option was to extend MAUI to accomplish our requirements, however there is a lack of technical information about MAUI and we had some problems with managing the multiprogramming level of applications. Due to this, and since we have to coordinate the scheduler with different components, we decided to implement our own scheduler (eNANOS scheduler).

There are other projects that are currently implementing systems that include both queuing system and scheduler, and that also advanced reservation mechanisms, for instance, the OAR scheduler [OAR05], but they are difficult to extend for achieve our requirements: incorporate mechanisms to interact between the CPU and Grid Scheduling in HPC-Environment.

System Architecture

Our system is composed by: a Grid level Resource Broker, a Local Job Scheduler, a Processor Scheduler, a Performance Monitor, a Predictor System and an Information System as is shown in 1.

The eNANOS Broker is an OGSA-Compliant Resource Broker developed as a Grid Service. It is based on the Globus Toolkit and it is compatible with both Globus Toolkit 2 and Globus Toolkit 3 services. It implements flexible mechanisms that allow it to become compatible with next Globus versions. The main services of this broker are: Resource Discovery, Resource Selection, Resource Monitoring, Job Submission and Job Monitoring. It implements different scheduling and resource selection policies, and provides interfaces that allow to the end user to have control over them. For submitting a job the user must provide a job description and a user multi-criteria files (requirements, recommendations and so on). More information about eNANOS Broker can be found in [IRJCRM+05].

The scheduling of the local system jobs is done by the eNANOS Scheduler. It is the responsible

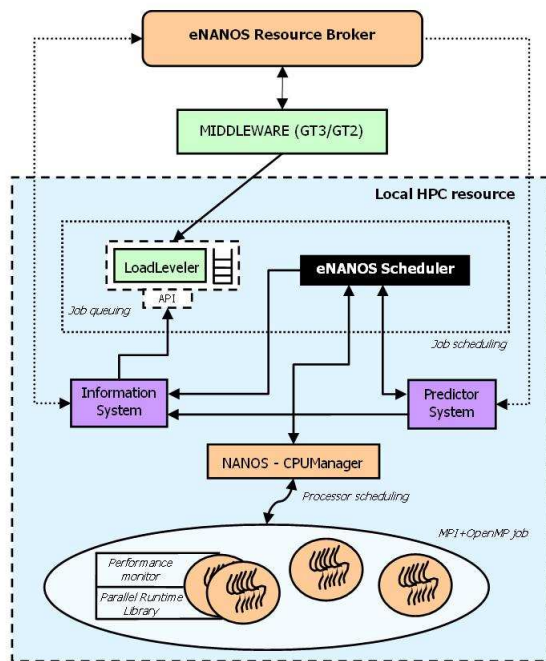


Figure 1: System Architecture

of managing jobs that there are queued in the local queuing system. To carry out the scheduling it obtains information about the local system state and the parallel running applications, it also asks for predictions of possible job executions. This last issue will allow to the eNANOS Scheduler to implement advanced policies. Another important issue of this local scheduler is that it is feed backing information to the Grid Broker about the jobs behavior, allowing it to achieve better performance and throughput thanks to this coordinated scheduling.

The Processor Scheduling is done through the NANOS-CPUManager. It is the responsible of processor scheduling, controlling the jobs multi-programming level and provide information about those jobs to the eNANOS Scheduler. The NANOS-CPUManager is currently implemented for several architectures and SOs, including IBM with AIX 5.

We support that in a HPC execution environment applications should be dynamically monitored to detect its real performance to be able to adjust their scheduling. In our system this monitoring is done by the Performance Monitor.

The Predictor System is the responsible of carry out prediction about the requirements and performance of the jobs. It helps to the eNANOS broker and to the eNANOS scheduler taking decisions when brokering and scheduling.

The Information System provides information about the different kind of entities that are involved on a given host: Jobs, machines, queues and so on. This system collects information from: The Grid environment, the local system (CPU-Manager and

eNANOS Scheduler) and the local queuing system (Load Leveler). It is intended to be a central point of access to information of the overall Grid System. The main goal of it is allow flowing information vertically and bidirectional among all the scheduling entities that are working in our system for achieve better scheduling decisions.

Discussion

In the final version of this paper we will present the explained architecture in more detail. Also we will show results that demonstrate that there is a performance improvement of MPI+OpenMP applications with the current system, and that a coordinated scheduling between all the components of the system allows to achieve a better throughput and to take better scheduling decisions.

We believe that an important issue is that the information should be shared with all the components of the system. If the the Grid broker has more dynamic information about the lower layers of the system it will be able to do a more suitable schedule. By the other hand the local scheduler will be able to get more troughput of the system if it knows more information about the jobs that are coming from the upper layers.

Currently we are working in the information system and predictor system implementation. Thus, the local scheduler has to be finished using the services offered by these components. Also we are working in the coordination between the broker and the local environment. In consequence, we can implement new Grid scheduling policies helped by the information extracted from the local system. Moreover, we plan to implement job migration and QoS mechanisms.

References

- [IFCKST01] Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, 2001.
- [GSA05] Grid Scheduling Architecture Research Group (GSA-RG) site.
- [MAUI05] MAUI Cluster Scheduler site.
- [OAR05] OAR scheduler site.
- [IRJCRM+05] Ivan Rodero, Julita Corbalán, Rosa M. Badia, Jesús Labarta, eNANOS Grid Resource Broker, 3rd European Grid Conference, Amsterdam, February 2005