International Journal of Multidisciplinary Approach and Studies ISSN NO:: 2348 – 537X

Resource Management System for Computational Grid

S. Selvi,
Associate Professor, Department of Electronics and Communication Engineering, Dr. Sivanthi Aditanar college of Engineering,Tamilnadu, India.

ABSTRACT

Computing grids are emerging as a new computing paradigm for solving challenge applications in science, engineering and medical field. The high performance computing (HPC) resources such as parallel computers or large-scale clusters act as the building blocks of computational grids. The efficient scheduling of independent jobs on the distributed Grid resources is one of the challenging issues in Grid computing. It is a very complex problem since resources are geographically distributed having different usage polices and may exhibit highly non uniform performance characteristics, heterogeneous in nature, and have varying loads and availability. In this paper, the detail about Resource Management System(RMS) is described which will increase the usability, acceptance and performance of HPC machines.

Key words: *Computational grid, high performance computing, Resource management, Scheduling*

1. INTRODUCTION

The term grid was introduced in the late 1990's by Ian Foster and Carl Kesselmann, but the idea of sharing distributed computing power can also be found in the earlier meta-computing concepts of the 1980's [1-3]. However the grid ignited broad interest in the scientific and public press as next boost to computing technology. The grid stands today for a new approach that promises to give a wide range of scientific area the computing power they need. The grid is expected to offer transparent access to resources of very different nature as for instance CPU, network, data or software.

It is a difficult task for the user to find and utilise such resources as they are provided by different owner with individual access policies. The access policies are enforced by the local resource management system. As a result, these grid configurations have a high degree of heterogeneity in the technological features of the local RMS as well as the enforced polices. This leads to major differences between grid resource management systems and the local resource management systems. The Grid RMS has to deal with many heterogeneous resources in a highly dynamic environment while it has no exclusive control over any resource. In contrast, the local RMS typically manages only one or a few resource types in a static configuration. These resources reside within a single administrative domain in which the RMS has exclusive control over the resources.

Thus prior accessing a remote resource, a negotiation process is required in which an agreement is reached on the resource offering and utilization. This process is manually not feasible in a large scale Grid environment with many potentially available heterogeneous resources. Therefore the Grid infrastructure must provide services for automatic resource brokerage that take cares for the resource selection and negotiation process. Today virtualization became a common technique for IT systems in many application environments. Virtualization introduces an abstraction layer on top of resources, so that physical characteristics are hidden from the user. At present CPU, network, and storage virtualization are the most widespread forms. Virtualization also provides new and powerful features in light of service based infrastructures or Grids. In the area of grid computing, virtualization gained more and more interest, but less in terms of service consolidation and or increasing server utilization. Instead virtualization allows addressing multiple problems in grid systems, like coping with the heterogeneity of grid resources, the difference in software stacks, enhanced features in resource management like a more general check in pointing or migration models. Adopting virtualization in smart ways gets us closer to real grid computing with more flexibility in the type of applications and the resources to use[4,5].

2. CURRENT GRID SCHEDULING SYSTEMS

This section deals with the overview of current grid scheduling systems.

2.1 Portable Batch System (PBS)

The portable batch system, PBS is a workload management solution for HPC systems and LINUX cluster[6]. PBS includes many novel approaches to resource management and job scheduling, such as the extraction of scheduling policy into a single separable, completely customizable mobile. The purpose of the PBS system is to provide additional controls over initiating or scheduling execution of batch jobs and to allow routing of those jobs between different hosts. The batch system allows a site to define and implement policy as to what types of resources and how much of each resource can be used by different jobs. The batch system also provides a mechanism with which a user can ensure that a job will have access to the resources required to complete that job.

The batch system comprises several components, typical interaction between the components is based upon the client server model; with clients make (batch) requests to servers and the servers performing work on behalf of the clients. A batch server is a persistent process or set of processes, such as daemon. The server manages a number of different objects, such as queues or jobs, each object consisting of a number of data items or attributes. It also provides batch services such as creating, routing, and executing, modifying or deleting jobs for batch clients.

2.2 Condor

The goal of the condor project is to develop, implement, deploy and evaluate mechanisms and polices that support high throughput computing by using distributed environments that can deliver large amounts of processing capacity over long periods of time[7].

When a user submits a job to condor, the system finds an available machine on the network and begins running the job on that machine. If condor detects that a machine running a condor job is to no longer available, condor can checkpoint the job, move it to a different machine that would be idle, and continue the job on the new machine from precisely where that job left off. In addition to using condor as an example of scheduling, examples from two systems derived from the condor system are cactus and the European Data Grid(EDG) project.

2.3 Cactus and Condor

Parts of condor are being used very successfully in the GrADS project to schedule applications in cooperation with the Cactus Computational toolkit. Cactus is an open source

Problem solving environment designed for scientists and engineers with a modular structure to enable parallel computation across different architectures and collaborate code development between different groups.

2.4 EDG Resource Broker

The Workload Management System (WMS) is the component of the Data Grid middleware responsible for managing the Grid resources in such a way that applications are conveniently, efficiently and effectively executed. The core component of the EDG WMS is the Resource Broker (RB)[8]. Given a job description, the RB tries to find the best match between the job requirements and the resources available on the Grid, using information from the Condor match making techniques. The description of a job is expressed in the job description language, which is based on Condor class ads. Security, communication and job submission are layered on top of standard Globus Toolkit components.

3. CONCLUSION

This paper defines the steps a user currently follows to make a scheduling decision across multiple administrative domains. Different types of resource management systems are described.

4. REFERENCES

i Dong, F., Akl, Selim.G.: Scheduling Algorithms for Grid Computing: State of the Art and Open Problems.2006-504.Ontario, Kingston: Queen's University, School of Computing. (2006)

ii Foster, I., Kesselmann, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers. USA (1999)

iii Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications. 15(3),200-220 (2001)

iv Foster, I.: The Grid: A New Infrastructure for 21st Century Science. Physics Today. 55 (2), 42-47 (2002) v Foster, I., Iamnitchi, A.: On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In: Proc. of 2nd International Workshop on Peer-to-Peer Systems (IPTPS’03), Berkeley, CA, USA,(2003)

vi http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.87.7172

vii http://research.cs.wisc.edu/htcondor/doc/condorgrid.pdf

viii http://www.powershow.com/view1/22560a-Dc1Z/EDG_Resource_Broker_for_the_Glue_Schema_powerpoint_ppt_presentation