IGI PUBLISHING

ITJ 4535

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

This paper appears in the publication, International Journal of Grid and High Performance Computing, Volume 1, Issue 1
edited by **Emmanuel Udoh © 2008, IGI Global**

Adaptive Processor Allocation for Moldable Jobs in Computational Grid

Kuo-Chan Huang, National Taichung University, Taiwan

Po-Chi Shih, National Tsing Hua University, Taiwan

Yeh-Ching Chung, National Tsing Hua University, Taiwan

ABSTRACT

In a computational grid environment, a common practice is try to allocate an entire parallel job onto a single participating site. Sometimes a parallel job, upon its submission, cannot fit in any single site due to the occupation of some resources by running jobs. How the job scheduler handles such situations is an important issue which has the potential to further improve the utilization of grid resources as well as the performance of parallel jobs. This article develops adaptive processor allocation policies based on the moldable property of parallel jobs to deal with such situations in a heterogeneous computational grid environment. The proposed policies are evaluated through a series of simulations using real workload traces. The results indicate that the proposed adaptive processor allocation policies can further improve the system performance of a heterogeneous computational grid significantly.

Keywords: adaptive processor allocation; computational grid; job scheduling; moldable property

INTRODUCTION

Most parallel computing environments running scientific applications adopt the space-sharing approach. In this approach, the processing elements of a parallel computer are logically partitioned into several groups. Each group is dedicated to a single job, which may be serial or parallel. Therefore, each job has exclusive use of the group of processing elements allocated to it when it is running. However, different running jobs may have to share the networking and storage resources to some degree.

Most current parallel application programs have the *moldable* property (Dror, Larry, Uwe, Kenneth and Parkson, 1997). It means the programs are written in a way so that at runtime they can exploit different parallelisms for execution according to specific needs or available resource. Parallelism here means the number of processors a job uses for its execution. The *moldable* property raises an interesting question whether it is possible to design special processor allocation policies, taking advantage of this property, to improve the overall system performance.

This article develops adaptive processor allocation policies based on the moldable property of parallel jobs for both homogeneous parallel computers and heterogeneous computational grid environments. The proposed policies require users to provide estimations of job execution times upon job submission. The policies are evaluated through a series of simulations using real workload traces. The effects of inexact runtime estimations on system performance are also investigated. The results indicate that the proposed adaptive processor allocation policies are effective as well as stable under different system configurations and can tolerate a wide range of estimation errors.

RELATED WORK

This article deals with scheduling and allocating independent parallel jobs in a heterogeneous computational grid. Without grid computing, local users can only run jobs on the local site. The owners or administrators of different sites are interested in the consequences of participating in a computational grid, whether such participation will result in better service for their local users by improving the job turnaround time. A common load-sharing practice is to allocate an entire parallel job to a single site which is selected from all sites in the grid based on some criteria. However, sometimes a parallel job, upon its submission, cannot fit in any single site due to the occupation of some resources by running jobs. How the job scheduler handles such situations is an important issue which has the potential to further improve the utilization of grid resources as well as the performance of parallel jobs.

Job scheduling for parallel computers has been subject to research for a long time. As for grid computing, previous works discussed several strategies for a grid scheduler. One approach is the modification of traditional list scheduling strategies for usage on grid (Carsten, Volker, Uwe, Ramin and Achim, 2002; Carsten Ernemann, Hamscher, Streit and Yahyapour, 2002a, 2002b; Hamscher, Schwiegelshohn, Streit and Yahyapour, 2000). Some economic based methods are also being discussed (Buyya, Giddy, & Abramson, 2000; Carsten, Volker and Ramin, 2002; Rajkumar Buyya, 2002; Yanmin et al., 2005). In this article, we explore non economic scheduling and allocation policies with support for a speed-heterogeneous grid environment.

England and Weissman (2005) analyzed the costs and benefits of load sharing of parallel jobs in the computational grid. Experiments were performed for both homogeneous and heterogeneous grids. However, in their works simulations of a heterogeneous grid only captured the differences in capacities and workload characteristics. The computing speeds of nodes on different sites are assumed to be identical. In this article, we deal with load sharing issues regarding heterogeneous grids in which nodes on different sites may have different computing speeds.

For load sharing there are several methods possible for selecting which site to allocate a job. Earlier simulation studies in the literature (Hamscher et al., 2000; Huang and Chang, 2006) showed the best results for a selection policy called *best-fit*. In this policy a particular site is chosen on which a job will leave the least number of free processors if it is allocated to that site. However, these simulation studies are performed based on a computational grid model in which nodes on different sites all run at the same speed. In this article we explore possible site selection policies for a heterogeneous computational grid. In such a heterogeneous environment nodes on different sites may run at different speeds.

In the literature (Barsanti and Sodan, 2007; John, Uwe, Joel and Philip, 1994; Sabin, Lang, and Sadayappan, 2007; Srividya, Vijay, Rajkumar, Praveen and Sadayappan, 2002; Sudha, Savitha and Sadayappan, 2003; Walfredo and Francine, 2000, 2002) several strategies for scheduling moldable jobs have been introduced.