Chapter 4

Moldable Job Allocation for Handling Resource Fragmentation in Computational Grid

Kuo-Chan Huang
National Taichung University of Education, 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 to 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 paper adopts moldable job allocation policies 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 moldable job allocation policies are also compared to the multi-site co-allocation policy, which is another approach usually used to deal with the resource fragmentation issue. The results indicate that the proposed moldable job allocation policies can further improve the system performance of a heterogeneous computational Grid significantly.

DOI: 10.4018/978-1-60960-603-9.ch004
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.

In a computational Grid environment, a common practice is to allocate an entire parallel job onto a single participating site. However, this kind of allocation sometimes runs into a situation called resource fragmentation. The following is an example. Assume a Grid consisting of 4 computing sites each equipped with 32 processors. After a sequence of job allocations, at some moment the amounts of leftover processors for the four sites are 4, 2, 4, 6 in order. At the moment, a new job requiring 10 processors is submitted into the Grid. Apparently, there is no site being able to accommodate the job for immediate execution. It has to wait in queue. However, carefully inspecting the leftover processors reveals that some combinations among the four sites have a total amount of leftover processors larger than the requirement of the incoming job. For example, site 3 and site 4 add up to exactly 10 processors. Site 1, site 2, and site 3 together can make it, too. This is what we called resource fragmentation in Grid environments. This paper tries to deal with resource fragmentation through moldable job allocation.

Most current parallel application programs have the moldable property (Dror, Larry, Uwe, Kenneth, & 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 job allocation policies proposed in this paper take advantage of the moldable property of parallel programs to improve the overall system performance.

This paper develops moldable job allocation policies 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 moldable job allocation policies are also compared to the multi-site co-allocation policy, which is another approach usually used to deal with the resource fragmentation issue. The results indicate that the proposed moldable job allocation policies are effective as well as stable under different system configurations and can tolerate a wide range of runtime estimation errors.

RELATED WORK

This paper 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 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.
Related Content

Pervasive Grids: Challenges and Opportunities
www.igi-global.com/chapter/pervasive-grids-challenges-opportunities/36402?camid=4v1a

Peer-to-Peer Desktop Grids Based on an Adaptive Decentralized Scheduling Mechanism
www.igi-global.com/article/peer-peer-desktop-grids-based/38975?camid=4v1a

Information Security in Data and Storage Grids through GS3
www.igi-global.com/chapter/information-security-data-storage-grids/58749?camid=4v1a

Adaptive Self-Organization in Distributed Tree Topologies
www.igi-global.com/article/adaptive-self-organization-in-distributed-tree-topologies/117168?camid=4v1a