Chapter 3
A Novel System Oriented Scheduler for Avoiding Haste Problem in Computational Grids

Ahmed I. Saleh
Mansoura University, Egypt

ABSTRACT

Scheduling is an important issue that must be handled carefully to realize the "Just login to compute" principle introduced by computational grids. Current grid schedulers suffer from the haste problem, which is the inability to schedule all tasks successfully. Accordingly, some tasks fail to complete execution as they are allocated to unsuitable workers. Others may not start execution as suitable workers are previously allocated to other tasks. This paper introduces the scheduling haste problem and presents a novel high throughput grid scheduler. The proposed scheduler selects the most suitable worker to execute an input grid task. Hence, it minimizes the turnaround time for a set of grid tasks. Moreover, the scheduler is system oriented and avoids the scheduling haste problem. Experimental results show that the proposed scheduler outperforms traditional grid schedulers as it introduces better scheduling efficiency.

INTRODUCTION

Recently, due to the dramatic development of network technologies and the popularity of the Internet, grid computing has become an appealing research area (Jens, Martin, & Roman, 2009; Lee, Squicciarini, & Bertino, 2009). Computational grids are the next generation of computer clusters. They aim to maximize the utilization of resources owned by a set of distributed heterogeneous systems (He, Jarvis, Spooner, Bacigalupo, Tan, & Nudd, 2005; (Sacerdoti, Katz, Massie, & Culler, 2003). Moreover, grids can be considered as the recent instances of metacomputing (Wolski, Spring, & Hayes, 1999). The primary goal of grid computing is to provide a transparent access to geographically distributed heterogeneous resources owned by different individuals or organizations (Jen & Yuan, 2009). Hence, the grid provides hardware and software infrastructure to create an illusion of a virtual supercomputer that exploit the computational power aggregated from a huge

DOI: 10.4018/978-1-4666-2065-0.ch003

set of distributed workers (Buyya, 1999). This allows the execution of tasks whose computational requirements exceed the available local resources. However, although the notion of grid computing is simple and attractive, its practical realization poses several challenges and open problems that need to be addressed (Tsai & Hung, 2009; Creel & Goffe, 2008). These challenges include resource discovery, failure management, fault tolerance, resource heterogeneity, reliability, scalability, security, and more importantly the scheduling of incoming tasks among available grid resources (Tseng, Chin, & Wang, 2009).

Scheduling is the major puzzle in developing a grid based computing paradigm (Tseng, Chin, & Wang, 2009; Iavarasan, Thambidurai, & Mahilmannan, 2005). It involves the matching of task or application requirements with the available resources (Tseng, Chin, & Wang, 2009). Scheduling in grids can be carried out in three different phases which are; (i) resource discovery, (ii) scheduling, and (iii) executing (Li & Hadjinicolaou, 2008). However, to achieve the expected potentials of the available resources, efficient scheduling algorithms are required (Daoud, & Kharma, 2008). Unluckily, scheduling algorithms previously employed in computer clusters can't be used in grids as they run on homogenous and guaranteed resources over the same LAN. A Scheduler used in a computer cluster only manages such cluster; hence, it owns the resources with no ability to discover new ones (Sacerdoti, Katz, Massie, & Culler, 2003). Also it assumes both the availability and stability of resources. On the other hand, scheduling in grids is significantly complicated as a result of grid heterogeneity and dynamic nature (Kiran, Hassan, Kuan, & Yee, 2009). Unlike the cluster scheduler, a grid scheduler should have the ability to discover new computing resources over multiple administrative domains (Yan, Shen, Li, & Wu, 2005). The dynamic nature of grids is a result of both the network connectivity and grid resources. The network may be unreliable as it can't guarantee its bandwidth. Moreover, grid resources change both availability and capability over time as they may join or leave the grid without any notification (Shah, Veeravalli, & Misra, 2007; Kousalya, & Balasubramanie, 2008).

Two alternative views may be considered when developing a grid scheduling system. The first is the User View (UV) while the other is the System View (SV). On one hand, the user aims to achieve the maximum Quality of Service (QoS); hence, he asks the scheduler to elect the best currently available resources for executing his task. On the other hand, the grid system tries to manage the available resources in a way for achieving the maximum QoS for all users not for a specific one. Based on those alternative views, grid schedulers can be categorized into two major categories, which are; (i) Task Oriented Grid Schedulers (TOGS), and (ii) System Oriented Grid Schedulers (SOGS). The former supports the user's demands, and therefore, it tries to minimize the execution time of each input task. The latter category supports the system's demands, and therefore, it tries to introduce a high throughput computing. Unlike TOGS, SOGS aims to maximize the processing ability of the system over the long run. Also, it introduces a better resource management scheme by allocating the grid task to the most suitable resource, not the best available one.

To the best of our knowledge, all the proposed grid scheduling algorithms were task oriented ones (Aggarwal, Kent, & Ngom, 2005), which usually suffer from the haste problem. The haste problem is the ability of the scheduler to present a good scheduling performance in the present; however an overall degraded performance is presented in the long run. Moreover, implementing an efficient SOGS has not been addressed yet. Hence, scheduling in grids is still more complex than the proposed solutions. Many hurdles stand in the way of achieving the maximum utilization of grid resources (Liu, Yang, Lian, & Wanbin, 2006). Accordingly, scheduling in grids is still an elusive problem that attracts the interests of many researchers (Aggarwal, & Kent, 2005; Hsin, 2005).