Chapter 110
Adaptive Threshold Based Scheduler for Batch of Independent Jobs for Cloud Computing System

TAJ ALAM
Jaypee Institute of Information Technology, Noida, India

PARITOSH DUBEY
Jaypee Institute of Information Technology, Noida, India

ANKIT KUMAR
Jaypee Institute of Information Technology, Noida, India

ABSTRACT

Distributed systems are efficient means of realizing high-performance computing (HPC). They are used in meeting the demand of executing large-scale high-performance computational jobs. Scheduling the tasks on such computational resources is one of the prime concerns in the heterogeneous distributed systems. Scheduling jobs on distributed systems are NP-complete in nature. Scheduling requires either heuristic or metaheuristic approach for sub-optimal but acceptable solutions. An adaptive threshold-based scheduler is one such heuristic approach. This work proposes adaptive threshold-based scheduler for batch of independent jobs (ATSBIJ) with the objective of optimizing the makespan of the jobs submitted for execution on cloud computing systems. ATSBIJ exploits the features of interval estimation for calculating the threshold values for generation of efficient schedule of the batch. Simulation studies on CloudSim ensures that the ATSBIJ approach works effectively for real life scenario.

DOI: 10.4018/978-1-7998-5339-8.ch110

INTRODUCTION

A distributed system is the interconnection of distributed computational and functional resources. They help in meeting the demand of executing the large-scale, high-performance computational jobs. These systems can be realized as homogenous or heterogeneous systems. They are quite useful for different scientific and business applications (Jiang & Huang, 2012; Jiang et al., 2011; Ye et al., 2013). Heterogeneous Distributed Systems (HDS) exhibits more visibility owing to the practical application of participation of various resources belonging to possibly different administrative domains to solve computational problems. Various types of distributed systems are grids (Liu et al., 2005), peer-to-peer systems (Rao et al., 2010), ad-hoc networks (Xue et al., 2006), cloud computing systems (Gkatzikis & Koutsopoulos, 2013), pervasive computing systems (Lukowicz et al., 2012) and online social network systems (Jiang & Jiang, 2014). Grid Computing (Liu et al., 2005) is build up for highly computational intensive scientific applications, but cloud computing (Gkatzikis & Koutsopoulos, 2013) goes one step ahead by providing dynamic resource provisioning and resource sharing through virtualization. Cloud computing is based on the principle of shifting the computing from traditional desktop to the Internet. Cloud computing provides cost-efficient server-based computing. The model is often being referred as "pay as you go model" i.e. the users can rent virtual resources called Virtual Machines and pay for what they actually use. Based on the abstraction level of the services, cloud delivery models can be classified into three types: Infrastructure as a Service (IaaS) Platform as a Service (PaaS), and Software as a Service (SaaS). There are four kinds of cloud deployment model: public cloud, private cloud, community cloud, and hybrid cloud.

Scheduling the jobs on the computational resources is one of the prime concerns in such systems. Scheduling the tasks over computational nodes has been established to be an NP-complete problem. Scheduling can be defined as the mapping of job or sub-jobs on a set of resources such that the application parallelism inherent in the job can be exploited over the available hardware parallelism (Vidyarthi et al., 2009; Alam & Raza, 2016; Alam & Raza, 2018; Xhafa & Abraham, 2010). There are various types of applications that are executed on these systems. These applications range from web services, the scientific computation to data storage types to name a few. In general, any application that is executed on these systems can be divided into a number of tasks. The performance of such system regarding application execution thus becomes dependent on the allocation of these tasks on the available nodes with optimization of QoS parameters (Attiya & Hamam, 2006; Jiang & Jiang, 2009; Li & Lan, 2005).

In general, the performance of the scheduler can be enhanced by simply increasing the capacity of the machines or by adding more computational machines to the system. But, this may not be the necessary solution for performance gain as an adverse situation of the unbalanced system may come up where already available machines may be overloaded. When there is an uneven workload distribution on computational machines, adding more machines or increasing the capacity may worsen the situation further. In such cases, the only acceptable solution is redistributing the load in a balanced way. Load Balancing is a variant of scheduling with the objective of maximizing system utilization with minimization of makespan. It is one of the primary concerns in cloud computing. Cloud platform has the advantage of being able to be quickly scaled up and down at any point in time. This dynamic environment demands an efficient load balancing for customer satisfaction, optimization of the rate of revenue, along with minimization of turnaround time and improved utilization. The challenging issue with the above-mentioned goal is to dispatch incoming tasks to VMs in a heterogeneous cloud environment (Gkatzikis & Koutsopoulos, 2013).

Related Content