International Journal of Grid and High Performance Computing
Volume 10 • Issue 1 • January-March 2018

An Enhanced Task Scheduling in Cloud Computing Based on Deadline-Aware Model

Mokhtar A. Alworafi, University of Mysore, Mysore, India
Suresha Mallappa, University of Mysore, Mysore, India

ABSTRACT

Cloud computing is the latest in distributed computing technology. The delivery mechanism between the service provider and users depends on Service Level Agreement (SLA). SLA contains Quality of Service (QoS), which has some constraints such as deadline to achieve user satisfaction. In this article, the authors propose a Deadline-Aware Priority Scheduling (DAPS) model to minimize the average makespan, and maximize resource utilization under deadline constraint. In the proposed model, the tasks are sorted based on length priority in ascending order and labeling the VM's state as successful which achieves the deadline constraint, and then mapping the tasks to the suitable VM that has minimum processing time. The authors compared their proposed model to the existing algorithms GA, Min-Min, SJF and Round Robin. The proposed model outperforms other algorithms by reducing the average of makespan, mean of total average response time, number of violations, violation ratio, and failure ratio, while increasing resource utilization, and guarantee ratio for tasks that meet deadline constraint.

KEYWORDS

Cloud Computing, DAPS, Deadline Constraint, Makespan, Resource Utilization

1. INTRODUCTION

Recently, cloud computing has become a turning point of resource computing where resources are provided as services on demand as per the users' request (Tripathy, 2016). Cloud computing has high scalability and reliability in obtaining services and it can fulfill the requirements of mass data storage (Qu et al., 2016). Cloud computing services is classified as: 1) Infrastructure as a Service (IaaS); 2) Platform as a Service (PaaS); and 3) Software as a Service (SaaS) (Singh et al., 2016).

Cloud service provider desires to meet the QoS requirements of cloud service users by providing sufficient amount of resources (Singh and Chana, 2015). Amazon EC2 is a better example of the cloud services, where a number of services are hosted. The Gmail services from Google is also examples that is used in daily life (Sharma and Peddoju, 2014). Cloud computing techniques are widely used in marketing, where it has a pay-as-you-use feature services (Yao et al., 2016). Cloud computing provides accessibility to applications and a large number of available virtualized resources with low management effort and low cost (Bousselmi et al., 2016). Virtualization is the base of cloud computing that efficiently offers resources to the users (Sing et al., 2016) by virtualization, the cloud data center can host thousands of physical machines which in turn are composite of various virtual machines that offer demand environments for any consumers (Mao et al., 2016). As the cloud user base grows,

DOI: 10.4018/IJGHPC.2018010103

the cloud service providers face the challenge of adjusting to the demands of this growth in business and technical dimensions (Sharkh et al., 2016). The service provider is responsible for mapping the tasks to available resources. The mapping of the task by the scheduler undergoes many constraints or conditions (Atiewi et al., 2016).

The most emphasized research point in cloud computing is scheduling, where the cloud users require more efficient scheduling technique which optimize and improve the overall performance metrics in the system. In cloud computing, task scheduling is done at two levels: first, at system-level scheduling, where VMs are deployed on available physical nodes, and second, at user-level scheduling, where tasks are assigned to suitable VMs (Saxena et al., 2016).

The system-level scheduling deals with the management of resource issues within the data centers while user-level scheduling handles issues between users and cloud providers (Masdari et al., 2016).

A schedule is robust, if it works to absorb some uncertain degree in the execution time of task. Strong task scheduling is most required in mission-critical and time-critical tasks. Here, meeting the deadline is most important (Poola et al., 2014).

In task scheduling algorithm, with an emphasis given to deadline, needs to achieve the deadline of each task which is being implemented, prior to its burst time. Depending on that, the algorithm can specify tasks that should be executed in less time. The deadline specific notion has been defined in several ways, but the most important point of every definition is the task scheduling within the time specification (Perret et al., 2013). The objective of task scheduling in cloud computing is to execute tasks with less makespan (completion time) (Baxodirjonovich and Choe, 2015).

Some of the task scheduling algorithms already in use are Genetic algorithm, Min-Min, SJF and Round Robin, we will be clarifying them as follows:

Genetic Algorithm (GA) is an efficient optimization algorithm that imitates the progression of natural evolution. The idea of GA is a biological evolution process in chromosomes. This idea is based on the survival of the fittest, where it obtains better solutions by recombining with the other (Madni et al., 2016). The Min-Min algorithm is still the basis of the present scheduling algorithms in cloud era, in which a set of unmapped list of tasks are found in cloud resources with minimum completion time. The algorithm selects the task with minimum size, and then assigns it to the corresponding resource. Finally, that task is removed from the set. The Min-Min procedure is repeated until all the tasks are mapped (Chen et al., 2013).

The Shortest-Job-First (SJF) algorithm is a priority scheduling algorithm. In this algorithm, the priority is the inverse of the next CPU burst where, the task with longer CPU burst has lower priority and vice versa (Lakshmi and Srinivasu, 2016).

In Round Robin algorithm, the next VM will be allocated to the task in the queue without considering the VM load, meaning this policy does not take into account the capabilities of resources, length of the tasks and priority. So, the longer tasks with higher priority are completed with the higher response times (Devi and Uthariaraj, 2016).

Our contribution in this paper, we propose Deadline-Aware Priority Scheduling (DAPS) model to schedule tasks and distribute them among the available cloud resources and thereby minimize the makespan based on deadline constraint with minimum completion time.

This paper is structured as follows: Related work is presented in Section 2. Section 3 presents problem definition. Proposed work of our model is presented in Section 4. Section 5 presents experimental results and analysis. Performance evaluation is presented in Section 6. We conclude the paper in Section 7.

2. RELATED WORK

In task scheduling, there are many works performed to schedule the tasks over the resources regarding some constraints like deadline. One of these works as in Poola, Deepak et al. (2014) presented a robust scheduling algorithm with resource allocation policies. Using this algorithm and policies,