A Customer-Oriented Task Scheduling for Heterogeneous Multi-Cloud Environment

Sohan Kumar Pande, Department of CSE, SUIIT, Burla, India
Sanjaya Kumar Panda, Department of CSE, Indian Institute of Technology (ISM), Dhanbad, India & Department of CSE and IT, VSS University of Technology, Burla, India
Satyabrata Das, Department of CSE and IT, V S S University of Technology, Burla, India

ABSTRACT

Task scheduling is widely studied in various environments such as cluster, grid and cloud computing systems. Moreover, it is NP-Complete as the optimization criteria is to minimize the overall processing time of all the tasks (i.e., makespan). However, minimization of makespan does not equate to customer satisfaction. In this paper, the authors propose a customer-oriented task scheduling algorithm for heterogeneous multi-cloud environment. The basic idea of this algorithm is to assign a suitable task for each cloud which takes minimum execution time. Then it balances the makespan by inserting as much as tasks into the idle slots of each cloud. As a result, the customers will get better services in minimum time. They simulate the proposed algorithm in a virtualized environment and compare the simulation results with a well-known algorithm, called cloud min-min scheduling. The results show the superiority of the proposed algorithm in terms of customer satisfaction and surplus customer expectation. The authors validate the results using two statistical techniques, namely T-test and ANOVA.

KEYWORDS

Cloud Computing, Customer Satisfaction, Makespan, Min-Min, Multi-Cloud, Surplus Customer Expectation, T-Test, Task Scheduling

1. INTRODUCTION

Cloud computing provides various services such as infrastructure, platform and software as a service over the Internet (Buyya, Yeo, Venugopal et al., 2009; Durao, Carvalho, Fonseka, & Garcia, 2014). These services are requested by the customers as and when required. In general, the customer requests are represented in the form of applications/jobs/tasks (Tsai, Fang, & Chou, 2013; Li et al., 2012; Panda, & Jana, 2015; Panda, & Jana, 2016). On the contrary, the services are provisioned in the form of various resources such as network, storage, hardware, software and many more (Tsai, Fang, & Chou, 2013). In order to provide the services, the customer requests are mapped with the pool of resources (Li et al., 2012). Therefore, efficient mapping of customer requests to the resources (referred as task scheduling) is a challenging problem which was shown to be NP-Complete (Braun et al., 2001; Maheswaran, Ali, Siegelet al., 1999; Mokotoff, 1999).

Task scheduling is the ordering of $n$ customers’ tasks to the $m$ resources or clouds such that the overall processing time (i.e., makespan) is minimized (Mokotoff, 1999). Note that $n >> m$. Here, the requirements of the customers are varying with respect to the number of resource, cost, deadline etc. On the contrary, the resources are varying with respect to processing speed, capacity, bandwidth, service

DOI: 10.4018/IJCAC.2016100101

Copyright © 2016, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
level etc. Therefore, the performance of the customer task is different from one resource to another resource. It introduces the problem of resource selection for each task in heterogeneous environment like cloud computing systems in which the primary objective is to minimize the makespan. However, minimization of makespan does not necessarily mean customers satisfaction as some tasks dominate the execution of other tasks. Therefore, task scheduling must emphasis on the customer satisfaction. More specifically, it must focuses on the minimization of individual makespan of the tasks rather than the overall makespan of all the tasks.

In this paper, we present the following task scheduling problem. Given a set of \( n \) independent customer tasks and a set of \( m \) clouds, the primary objective is to minimize the individual makespan of all the tasks so that the customer satisfaction is considerably increased. We propose an algorithm called customer-oriented task scheduling (COTS) for the above scheduling problem.

The paper is organized as follows. Section 2 discusses the related work in task scheduling algorithms. Section 3 presents the model and problem description. Section 4 proposes a customer-oriented task scheduling algorithm and analyzes the complexity of the algorithm. Section 5 introduces two performance metrics followed by simulation results in Section 6. We conclude with some future insights in Section 7.

2. RELATED WORK

Task scheduling is a challenging problem from the invention of parallel and distributed computing (Braun et al., 2001; Maheswaran, Ali, Siegel, Hensgen, & Freund, 1999; Ibarra, & Kim, 1977). As the requirements of the customers are rapidly changing, the existing algorithms become infeasible for that working environment. Thus many researchers have proposed various task scheduling algorithms by considering different set of customer requirements (Tsai, Fang, & Chou, 2013; Li et al., 2012; Panda, & Jana, 2015; Panda, & Jana, 2016; Braun et al., 2001; Maheswaran, Ali, Siegel, Hensgen, & Freund, 1999; Ibarra, & Kim, 1977; Ergu, Kou, Peng, Shi, & Shi, 2013; Xhafa, Carretero, Barolli, & Durresi, 2007; Xhafa, Barolli, & Durresi, 2007; Panda, & Jana, 2016; Panda, & Jana, 2015; Panda, & Jana, 2015; Armstrong, Hensgen, & Kidd, 1998; Freund et al., 1998). Ibarra and Kim (1977) have proposed a time-bound algorithm for scheduling \( n \) tasks on \( m \) resources. The algorithm takes \( n \lg n \) for \( m = 2 \) and it produces a scheduling length of at most \((\sqrt{5} + 1)/2\) time of the optimal time. Freund et al. (1998) have called the algorithm \( D \) and algorithm \( E \) of (Ibarra, & Kim, 1977) as max-min and min-min respectively. They have also stated explicitly that the time complexity of these algorithms is \( O(n^2m) \) respectively. Maheswaran, Ali, Siegel, Hensgen and Freund (1999) have proposed three heuristics, namely switching algorithm, \( k \)-percent best and sufferage for heterogeneous computing. The simulation results show the selection of heuristic based on the heterogeneity of tasks and machines. Braun et al. (2001) have compared eleven static heuristics for heterogeneous computing. The comparisons revel that min-min heuristic outperforms all other heuristics in terms of makespan performance metric. Later, Xhafa, Carretero, Barolli and Durresi (2007) have compared the performance of these heuristics in benchmark dataset generated by Braun et al. (2001). The results show that min-min performs well in makespan whereas max-min performs better in resource utilization performance metric.

Li et al. (2012) have recently applied min-min scheduling algorithm in heterogeneous multi-cloud environment and they called the developed algorithm as cloud min-min scheduling (CMMS). The objective of CMMS is to minimize the makespan of a set of applications. However, they have not considered the customer satisfaction in their proposed algorithm. Panda and Jana (2015) have also used min-min algorithm and developed cloud min-max normalization (CMMN) scheduling. The CMMN first normalizes the customer requests using the popular min-max normalization and places the requests into one of the two queues based on a threshold value. However, the process of normalization requires a time complexity of \( O(nm) \). Miriam and Easwarakumar (2010) have proposed
A Proposed Model for Using Cloud Computing and Web2.0 in Deploying E-Learning Ecosystem (ELES)

Towards a Scientific Foundation for Interoperability
[www.igi-global.com/chapter/towards-scientific-foundation-interoperability/45798?camid=4v1a](www.igi-global.com/chapter/towards-scientific-foundation-interoperability/45798?camid=4v1a)