Journal of King Saud University – Computer and Information Sciences (2017) **29**, 473–483

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Development and analysis of a three phase cloudlet allocation algorithm

Sudip Roy [a], **Sourav Banerjee** [a,*], **K.R. Chowdhury** [a], **Utpal Biswas** [b]

[a] *Kalyani Government Engineering College, Kalyani, Nadia, India*
[b] *University of Kalyani, Kalyani, Nadia, India*

Received 15 September 2015; revised 11 January 2016; accepted 11 January 2016
Available online 28 March 2016

KEYWORDS

Cloud computing;
Cloudlet allocation;
Cloud service provider;
Cloudsim

Abstract Cloud computing is one of the most popular and pragmatic topics of research nowadays. The allocation of cloudlet(s) to suitable VM(s) is one of the most challenging areas of research in the domain of cloud computing. This paper highlights a new cloudlet allocation algorithm which improves the performance of a cloud service provider (CSP) in comparison with the other existing cloudlet allocation algorithms. The proposed Range wise Busy-checking 2-way Balanced (RB2B) cloudlet allocation algorithm optimizes few basic parameters associated with the performance analysis. An extensive simulation is done to evaluate the proposed algorithm using Cloudsim to attest its efficacy in comparison to the other existing allocation policies.

1. Introduction

Cloud computing is the newest trend in the field of computer science and it is said to be the future of modern technology. Cloud computing is popular mostly for its special ability to utilize shared resources most efficiently. The allocation of the cloudlets to the suitable resources known as the virtual machines or VMs (Fu and Zhou, 2015) is an essential requirement in cloud computing environment. In a typical Cloud environment there is a module known as datacenter broker (DCB) which controls the entire datacenter including the cloudlet allocation to VMs. So, like any normal computing performance optimization and improvement of the allocation algorithm is always a possibility.

Engineering an efficient cloudlet allocation algorithm (Zhang et al., 2007) is a challenging research area and many such policies have been proposed, analyzed and compared on heterogeneous parallel computing environments. A new mechanism had been introduced, known as effective aggregated computing power (EACP) (Radulescu and Van Gemund, 1999) that improves the performance. The Adaptive weighted factoring (AWF) (Carino and Banicescu 2008) is used for scheduling parallel loops. The dynamic loop scheduling with reinforcement learning (Rashid et al., 2008) (DLS-with-RL) is very much effective for use in time stepping scientific applications with many steps. The scheduling (Aziz and El-Rewini, 2008) policies for Grid environment use several methods which are similar yet different to the mechanisms of cloudlet allocation policies. Genetic Algorithms (Pop, 2008) are also used for

---

[*] Corresponding author.
E-mail addresses: sanosuke009@gmail.com (S. Roy), mr.sourav.banerjee@ieee.org (S. Banerjee), papanthegenius@gmail.com (K.R. Chowdhury), utpal01in@yahoo.com (U. Biswas).

Peer review under responsibility of King Saud University.

ELSEVIER **Production and hosting by Elsevier**

http://dx.doi.org/10.1016/j.jksuci.2016.01.003
1319-1578

scheduling. The Opportunistic Load Balancing (OLB) heuristic (Braun et al., 2008) (Makelainen et al., 2014) (Iordache et al., 2007) chooses a cloudlet from the batch of cloudlets arbitrarily and allocates it to the next VM which is estimated to be available, not considering the cloudlet's expected execution time on that VM, resulting in very poor makespan (Wang et al., 2006). The Minimum Execution Time (MET) (George Amalarethinam and Muthulakshmi, 2011) allocates each cloudlet chosen arbitrarily to the VM with the least possible execution time resulting in the severe imbalance of load across the VMs. The Minimum Completion Time (MCT) (George Amalarethinam and Muthulakshmi, 2011) heuristic allocates each cloudlet to the VM with the minimum completion time for that cloudlet. It literally combines the advantages of OLB and MET. The QoS (Quality of Service) guided Min–min (He et al., 2003) assigns cloudlets which require higher bandwidth. The QoS priority grouping scheduling (Dong et al., 2006) gives importance to deadlines. The QoS Sufferage (Ullah Munir et al., 2007) considers network bandwidth as a major factor and schedules tasks based on their bandwidth requirement. The Grid-JQA (Khanli and Analoui, 2007, 2008) scheduling solution uses an aggregation formula which combines the parameters together with weighting factors to calculate QoS. The proposal of a dissimilar and new scheduling algorithm (Afzal et al., 2008) that tries to minimize the cost of the execution as well as satisfying the QoS constraints, views the scheduling environment as a queuing system. Another user oriented scheduling algorithm which uses an advanced reservation and resource selection techniques (Elmroth and Tordsson, 2008) minimizes the execution time of individual cloudlets without considering the make span. The multiple resources scheduling (MRS) (Benjamin Khoo et al., 2007) algorithm considers both the system capabilities and the resource requirements of cloudlets as majority factors. In cloud computing environment, most of the allocation policies load some specific resources comparatively more heavily, leaving other resources either idle or least loaded (Livny and Melman, 2011). As a result, load balancing is an important issue in cloud computing which affects the performance of the cloud service provider.

The objective of this paper is to improve the existing allocation policies in this domain by devising a new cloudlet allocation algorithm RB2B that focuses mostly on reducing waiting time and make span, at the same time optimizing VM (Marisol García-Valls et al., 2014) utilization to a remarkable amount by distributing the number of cloudlets to the VMs in a most uniform way. The proposed algorithm is incorporated in the datacenter broker (DCB) module. The DCB policy is enhanced with this proposed work and termed as advanced datacenter broker (ADCB) module in this study.

2. Related works

In this paper, few existing allocation policies are taken into account to analyze and compare the advantages of the proposed RB2B. They are described as follows.

2.1. *Min–min (Parsa and Entezari-Maleki, 2009; Kumar and Dutta Pramanik, 2012; El-kenawy et al., 2012)*

Initially a matrix is taken for all unassigned cloudlets. There are two phases in Min–min. In the first phase the set of minimum computation time for each cloudlet in the matrix is calculated and found. In the second phase, the cloudlet with the overall minimum expected computation time is chosen from the matrix and assigned to the corresponding VM. Then the assigned cloudlet is removed from the matrix and the entries of the matrix are modified accordingly. This process of Min–min is repeated until there is no cloudlet left in the matrix, that is, all cloudlets in the matrix are mapped. This algorithm takes $O(mn^2)$ time where m is the number of VMs and n is the number of cloudlets.

2.2. *Max–min (Parsa and Entezari-Maleki, 2009)*

This algorithm is almost similar to Min–Min, but there is a distinct difference in the second phase. This Max–Min first chooses the cloudlet with maximum computation time from the matrix and assigns it to the VM on which the chosen cloudlet gives minimum time to compute. This algorithm also takes $O(mn^2)$ time where m is the number of VMs and n is the number of cloudlets.

2.3. *RASA (Parsa and Entezari-Maleki, 2009)*

This algorithm actually combines the advantages of both Min–min and Max–min. If the number of available VMs is odd, the Min–min algorithm is applied to allocate the first cloudlet, otherwise the Max–min algorithm is applied. The whole process can be divided into a number of rounds where in each round two cloudlets are allocated to appropriate VMs by one of the two strategies, alternatively. The rule is, if the first cloudlet of the current round is allocated to a VM by the Min–min strategy, the next cloudlet will be allocated by the Max–min strategy. In the next round, the cloudlet allocation begins with an algorithm different from the last round. For example if the first round begins with the Max–min algorithm, the second round will begin with the Min–min algorithm. Experimental results show that if the numbers of available resources are odd then starting with applying the Min–min algorithm in the first round gives the better result. Otherwise, it is better to apply the max–min strategy at first. Min–min and Max–min are exchanged alternatively to result in consecutive execution of small and large cloudlets on different VMs and therefore, the waiting time of the small cloudlets in Max–min algorithm and the waiting time of the large cloudlets in Min–min algorithm are ignored. As RASA doesn't consist of any time consuming instruction, the time complexity of RASA (Maheswaran et al., 1999) is $O(mn^2)$ where m is the number of VMs and n is the number of cloudlets.

2.4. *Round Robin Allocation (RRA) (Parsa and Entezari-Maleki, 2009; Bhatia et al., 2010; Banerjee et al., 2015)*

It allocates the cloudlet to first available VM. For example, consider there are four cloudlets (C_0, C_1, C_2, C_3, and C_4) and three

Table 1 RRA allocation style.

Cloudlet	VM
C_0	VM_0
C_1	VM_1
C_2	VM_2
C_3	VM_0
C_4	VM_1