Int. J. Advance Soft Compu. Appl, Vol. 8, No. 2, July 2016
ISSN 2074-8523

Dynamic Batch Mode Cost-efficient Independent Task Scheduling Scheme in Cloud Computing

R. Kanniga Devi[1], K. Vimala Devi[2], and S. Arumugam[3]

[1]Department of Computer Science and Engineering
Kalasalingam University Anand Nagar, Krishnankoil-626126, Tamil Nadu, India
e-mail: rkannigadevi@gmail.com
[2]Department of Information Technology
PSR Engineering College, Sivakasi-626140, Tamil Nadu, India.
e-mail: k.vimaladevi@gmail.com
[3]National Centre for Advanced Research in Discrete Mathematics
Kalasalingam University Anand Nagar,
Krishnankoil-626126, Tamil Nadu, India.
e-mail: s.arumugam.klu@gmail.com

Abstract

Cloud Computing is a paradigm of large scale distributed computing which is a repackaging of various existing concepts/ technologies such as utility computing, grid computing, autonomic computing, virtualization and Internet technologies. Cloud computing uses the Internet technologies for delivery of resources as a service to users on demand. As it is a developing technology, various issues such as resource provisioning, security, energy management and reliability need to be addressed. This work focuses on resource provisioning issue in the context of scheduling schemes adapted by cloud environment. Job scheduling is a very challenging task in cloud computing because of its complex distributed architecture. Many algorithms have been proposed using different scheduling techniques. In this paper a cost-efficient dynamic batch mode scheduling approach has been proposed based on assignment rule. The proposed work is compared with other popular scheduling approaches such as round robin, Min-Min, and Max-Min and the results show that our approach improves performance of the system by means of reducing the cost of running all jobs and reducing the average Virtual Machine (VM) utilization. CloudSim, a simulation tool has been used.

Keywords: *Assignment rule, Cloud Computing, CloudSim toolkit, Graph model, Scheduling.*

1 Introduction

Task scheduling is one of the critical activities performed in cloud computing environments. Task Scheduling is complicated in cloud computing environment due to its abstract heterogeneous architecture, dynamic behaviour and resource heterogenity. Since cloud computing has evolved from grid computing, distributed computing and parallel computing paradigms, the scheduling algorithms developed for these systems can also be applied in cloud with suitable modifications. The task scheduling is defined as the mapping of tasks to resources which may be distributed over the cloud network. Many research works have been done on task scheduling for improving resource utilization, reducing cost of running jobs, improving the quality of service , maintaining fairness among the jobs, maintaining excellent system throughput, and minimizing makespan (the total length of the task schedule) by reducing waiting time of tasks and balance the load across resources.

Tasks can be classified into dependent tasks and independent tasks where dependent tasks have precedence constraints and independent tasks have no dependencies among the tasks and have no precedence constraints to be followed during scheduling. Dependent tasks can be handled by workflow scheduling with the help of DAG (Directed Acyclic Graph).

The task scheduling algorithms can be broadly classified as static or dynamic based on the time at which the scheduling or assignment decisions are made. In the case of static scheduling, information regarding all the resources in the cloud and the complete set of tasks is assumed to be available by the time the task is scheduled on the cloud. But in dynamic scheduling, a prior knowledge of the resources needed by the task and the environment in which it would be executed is unavailable as the jobs arrive in a real time mode. Hence dynamic scheduling is suitable for cloud since it deals with on-demand resource provisioning.

The dynamic scheduling algorithms can be used in two fashions namely on-line mode and batch mode. In online mode, a task is scheduled onto a machine as soon as it arrives. Each task is scheduled only once and the scheduling result cannot be changed. Hence, on-line mode of dynamic scheduling can be applied, if the arrival rate of the tasks in the real time is low. However, in batch mode the tasks are collected into a set that is examined for scheduling at prescheduled times. While online mode considers a task for scheduling only once, batch mode considers a task for scheduling at each scheduling event until the task begins execution. Since the cloud environment is a heterogeneous system and the arrival rate of requests is too high, the batch mode scheduling is more appropriate for a cloud environment. Some examples of batch mode algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Priority scheduling, Random algorithm, Min-Min algorithm and Max-Min algorithm. Most fit task scheduling algorithm (MFTF) is an example of on-line mode scheduling algorithm. Since scheduling problem is identified as NP- complete due to the heterogenity of resources, many heuristic based algorithms have been proposed.

Round Robin scheduling scheme distributes the tasks over the available Virtual Machines(VMs) in a round manner where each task is equally handled. Min-Min scheduling scheme assigns priority to the task that requires the shortest execution time and calculates its expected completion time on a VM and assigns to a VM which complete the task at the earliest time. Max-Min scheduling scheme is similar to Min-Min but it sets priority to the task that requires the longest execution time rather than the shortest execution time.

The task scheduling model adopted in this work is independent, dynamic and batch mode task scheduling.

The remainder of the paper is organized as follows. Section 2 is devoted to related works. We formally describe the model and based on this model we propose a cost-efficient scheduler in Section 3. Simulation study is discussed in Section 4 and we conclude our research in Section 5.

2 Related Work

This section gives various scheduling schemes classified as independent and dependent task scheduling and algorithms prevalent in clouds.

2.1 Independent tasks scheduling

Kim et al. [1] suggest a model for estimating the energy consumption of each virtual machine without dedicated measurement hardware. This model estimates the energy consumption of a virtual machine based on in-processor events generated by the virtual machine. Based on this estimation model they also propose a virtual machine scheduling algorithm that can provide computing resources according to the energy budget of each virtual machine.

Tsai et al. [2] propose an improved differential evolution algorithm(IDEA) based on the proposed cost and time models on cloud computing environment. The proposed IDEA combines the Taguchi method and a differential evolution algorithm(DEA).This multi-objective optimization approach is applied to find the Pareto front of total cost and makespan.

Somasundaram et al.[3] design and develop a CLOUD Resource Broker (CLOUDRB) for efficiently managing cloud resources and completing jobs for scientific applications within a user-specified deadline. It is implemented and integrated with a Deadline-based Job Scheduling and Particle Swarm Optimization (PSO)-based Resource Allocation mechanism. It is minimizing both execution time and cost based on the defined fitness function.

Palmieri et al. [4] use game theory and autonomous agents for effective Multi-User Task Scheduling. They present a novel uncoordinated fully distributed scheduling scheme for federated cloud organizations, based on in-dependent, competing, and self-interested job/task execution agents, driven by optimum social welfare.

Xu et al. [5] propose an economic model for dynamic scheduling. The resource allocation using the fairness constraint and the justice function has been compared. The first constraint is to classify user tasks by QoS preferences, and establish the general expectation function in accordance with the classification of tasks to restrain the fairness of the resources in selection process. The second constraint is to define resource fairness justice function to judge the fairness of the resources allocation.

Wang et al.[6] propose a new multi-objective bi-level programming model based on MapReduce to improve the energy efficiency of servers. They formulate the problem as an integer bi-level programming model. In order to solve the model efficiently, specific design encoding and decoding methods are introduced. Based on these, a new effective multi-objective genetic algorithm based on MOEA/D is proposed.

Wu et al.[7] propose a scheduling algorithm for the cloud datacenter with a dynamic voltage frequency scaling technique. The scheduling algorithm can efficiently increase resource utilization; hence, it can decrease the energy consumption for executing jobs.

Saurabh Kumar Garg et.al.[8] propose near-optimal scheduling policies that exploit heterogeneity across multiple data centers for a cloud provider. They consider a number of energy efficiency factors (such as energy cost, carbon emission rate, workload, and CPU power efficiency) which change across different data centers depending on their location, architectural design, and management system.

2.2 Dependent tasks scheduling

Mezmaz et al. [9] investigate the problem of scheduling precedence-constrained parallel applications on heterogeneous computing systems (HCSs) like cloud computing infrastructures. This work pays much attention to energy consumption. They propose a parallel bi-objective hybrid genetic algorithm that takes into account makespan, and energy consumption. The method is based on dynamic voltage scaling (DVS) to minimize energy consumption.

Su et al. [10] propose that large programs may be decomposed into multiple sequences of tasks that can be executed on multiple VMs in a cloud. Such sequences of tasks can be represented as a directed acyclic graph (DAG), where nodes are tasks and edges are precedence constraints between tasks. They present a cost-efficient task scheduling algorithm using two heuristic strategies.

Lee et al. [11] propose a scheduling algorithm which attempts to maximize profit within the satisfactory level of service quality specified by the service consumer, the development of a pricing model, the application of this pricing model to composite services with dependency consideration, the development of two sets of service request scheduling algorithms, and the development of a prioritization policy for data service aiming to maximize the profit of data service.

Mustafizur Rahman et.al.[12] define the workflow scheduling problem and describe the existing heuristic based and meta heuristic-based workflow scheduling strategies in grids. Then, they propose a dynamic critical-path-based adaptive workflow scheduling algorithm for grids, which determines an efficient mapping of workflow tasks to grid resources dynamically by calculating the critical path in the workflow task graph at every step. Finally, they outline a hybrid heuristic combining the features of the proposed adaptive scheduling technique with meta heuristics for optimizing execution cost and time as well as meeting the users requirements to efficiently manage the dynamism and heterogeneity of the hybrid cloud environment.

Hamid Mohammadi Fard et.al.[13] propose a generic multi-objective optimization frame-work supported by a list scheduling heuristic for scientific workflows in heterogeneous Distributed Computing Infrastrucutres. The algorithm approximates the optimal solution by considering user-specified constraints on objectives in a dual strategy: maximizing the distance to the users constraints for dominant solutions and minimizing it otherwise. The algorithm for a four-objective case study is comprising makespan, economic cost, energy consumption, and reliability as optimization goals.

Saeid Abrishami et.al.[14] propose Partial Critical Paths (PCP), algorithm for the cloud environment and propose two workflow scheduling algorithms: a one-phase algorithm which is called IaaS Cloud Partial Critical Paths (IC-PCP), and a two-phase algorithm which is called IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2). Both algorithms have a polynomial time complexity which make them suitable options for scheduling large workflows.

Joel J.P.C. Rodrigues et.al. [15] propose a novel media-aware flow scheduling architecture with the aim of improving the multimedia quality and increasing the networks lifetime. In order to avoid interfering with the multimedia applications delay requirements, this work also proposes to analyze frames delay and jitter. The proposal has proven to improve the multimedia quality and decrease the trans-mission delay in a controllable manner, and thus the tradeoffs between QoS, lifetime, and delay requirements can be achieved according to the considered scenario.

Some of the aforementioned scheduling algorithms are static approaches that do not consider the dynamic characteristics of resources in cloud environment. Since the problem in this paper involves cost, we need a cost-efficient resource selection model that can cope with the dynamic environment. This work is distinct from the related works by proposing a dynamic batch mode independent task scheduling mechanism for cloud environment. The solution proposed in this paper is based on assignment problem which schedules tasks to the best available resources, thereby reducing cost of utilizing the resources.

3 Proposed Work

3.1 Rule of Assignment

The assignment problem is one of the combinatorial optimization problems in operations research.

Let $C = \{c_1, c_2, \ldots c_m\}$, the set of m cloudlets.

Let $w(c_i)$ be the weight of i^{th} cloudlet , assigned based on one of the physical characteristics of the cloudlet, namely length, which is treated as a priority, where $w(c_1) \leq w(c_2) \leq .... \leq w(c_m)$.

Let $V = \{m_1, m_2, \ldots m_n\}$, the set of n virtual machines. Let $w(m_j)$ be the weight of j^{th} virtual machine assigned based on one of the physical characteristics of the virtual machine, namely size, where $w(m_1) \geq w(m_2) \geq .... \geq w(m_n)$.

A cloudlet c_i is assigned to a virtual machine m_j , where i is the smallest integer and j is the largest integer, such that $w(c_i) \leq w(m_j)$.

We observe that for a cloudlet c_i if there is no m_j with $w(c_i) \leq w(m_j)$, then c_i cannot be executed by any machine in the set V . This assignment rule minimizes the cost of executing the cloudlets.

3.2 Proposed Algorithm

Step 1: Datacenter broker receives new heterogeneous cloudlets/ client requests/tasks to be scheduled.

Step 2: Let $C = \{c_1, c_2, \ldots c_m\}$, the set of m cloudlets. Sort cloudlets in ascending order of their length and let $w(c_i)$ be weight of i^{th} cloudlet, assigned based on one of the physical characteristics of the cloudlet, namely length.

Set *m* to number of cloudlets to be sorted;

Set *w*[*cloudlet*] to the weight of cloudlet;

```
repeat
        flag = false;
        for cloudlet = 1 to m − 1 do
                if w[cloudlet] > w[cloudlet + 1] then
                        swap the cloudlets;
                Set flag = true;
                end if;
```

end do;

$m = m - 1$;

until *flag* = *false* or $m = 1$;

Step 3: Let $V = \{m_1, m_2, \ldots m_n\}$, the set of '$n$' virtual machines. Sort the VMs in descending order of their size and let $w(m_j)$ be weight of j^{th} virtual machine, assigned based on one of the physical characteristics of the virtual machine, namely size.

Set n to number of *VMs* to be sorted;

Set $w[VM]$ to the weight of VM;

repeat

flag = *false;*

for $VM = 1$ to $n - 1$ do

if $w[VM > w[VM + 1]$ then

swap the VMs;

Set *flag* = *true;*

end if;

end do;

$n = n - 1$;

until *flag* = *false* or $n = 1$;

Step 4: choose the smallest i and largest j such that,

$w(c_i) \leq w(m_j)$

Step 5: $w(m_j) \rightarrow w(m_j) - w(c_i)$

Step 6: When m_j finishes c_i

$C \rightarrow C - \{c_i\}$

$w(m_j) \rightarrow w(m_j) + w(c_i)$

Step 7: Repeat from step 4 until $C = \varphi$

3.1 Algorithm Analysis

Let T represent the time complexity of the proposed algorithm.

Then $T = T$ (sorting cloudlets) + T (sorting VMs) + T (assignment rule)

$= O(n \log n) + O(n \log n) + O(n)$

$= O(nlogn)$

4 Simulation Study

The CloudSim toolkit is used to simulate heterogeneous cloud environment. Here the term, cloudlet and task can be used interchangeably. Datacenter maintains virtualized set of physical resources and provide them as services. Datacenter broker schedules each task to the appropriate resources. As the cloudlets (tasks) are submitted by the user, it is the task of the datacenter broker to assign those tasks to the VM. The VM starts running the cloudlets. Here scheduling algorithms come into existence. Each cloudlet c_i requires a different processing capacity for its completion; and this determines the assignment of the cloudlet to a virtual machine. The processing speed of Virtual machine is heterogenous and expressed in terms of million instructions per second (MIPS). The selection of tasks to be scheduled is based on assignment rule.

Three popular and standard batch mode scheduling schemes namely, round robin, Min-Min, Max-Min and proposed approaches for VM selection are analyzed. The proposed approach results in less cost of executing all cloudlets and reduces average VM utilization as compared to other scheduling approaches.

The simulation configuration used in this experiment is shown in the following table:

Table 1.Simulation Parameters

Parameter	**Value**
Configuration of Data center	
Data center architecture	X86
Data center OS	Linux
VMM	Xen
Configuration of Hosts	
No of Hosts	5
MIPS	1000
RAM	10 GB
storage	1 TB
Bandwidth	10000Mbps
Configuration of VMs	
No of VMs	13
size	varying
MIPS	250
RAM	1 GB
Bandwidth	1000Mbps
No of PEs	3
Configuration of Cloudlets	
No of Cloudlets	10-50
Length	Varying(in MI)
File size	300Bytes
Output Size	300Bytes

The proposed algorithm results in a significant reduction in cost and utilization of VMs over other schemes. The result shows that the proposed scheme minimizes the cost of all submitted cloudlets, average VM utilization and the results improve with the increase in cloudlet count.

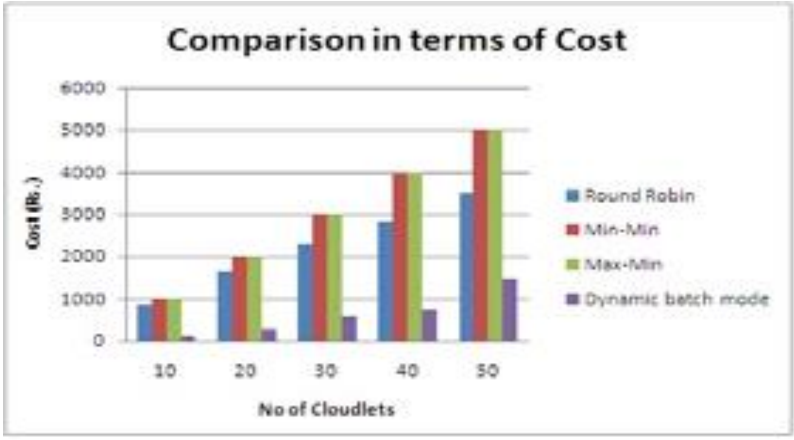

Fig. 1: Proposed scheme Vs. Other scheduling schemes

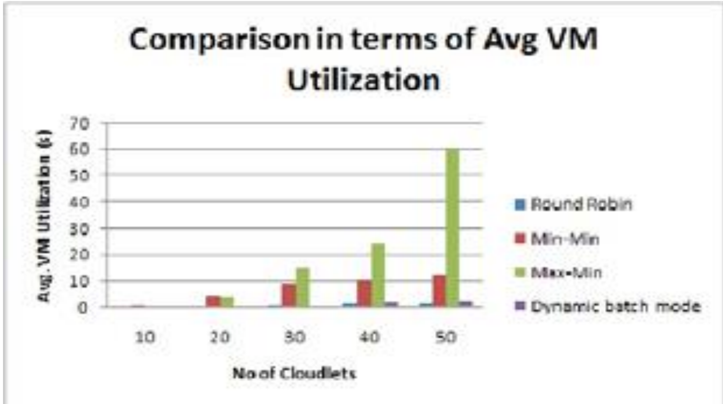

Fig. 2: Proposed scheme Vs. Other scheduling schemes

5 Graph Theoretic Model for a Cloud

Task scheduling in cloud computing environment need to be performed in an efficient manner since users deal with pay-as-you go model. Hence scheduling plays a key role for optimal utilization of resources which minimizes the cost of executing tasks. In this work cloud service provision is described in terms of Virtual machines(VM) and its characteristics and task are described in terms of Cloudlets and its characteristics. The pricing model of service provision is based on the characteristics of VM namely, size.

As cloud consists of thousands of networked nodes running VMs in a virtualized data center, the major elements that exist in any cloud systems can be represented as $C = \{Host, VM\}$ where Host represents set of physical machines and VM represents set of Virtual machines.

As cloud contains set of networked nodes, it is possible to represent the cloud as an undirected graph [16].

The representation of physical machines an undirected graph is given by $PG = (N,E)$ where $N = \{n_i\}$ is the set of all physical machines in the cloud and E is the set of edges that represents physical connection between the physical machines.

The representation of virtual machines as an undirected graph is given by $VG = (V,L)$ where $V = \{v_i\}$ is the set of all virtual machines in the cloud and L is the set of edges that represents virtual connection between the virtual machines.

Using this graph theoretic models various problems in cloud computing can be addressed and results in this directions will be reported in future papers.

6 Conclusion

This paper presents various scheduling schemes prevalent in cloud and proposes a new dynamic batch mode scheme for scheduling tasks based on assignment rule. The complexity of the approach is analyzed and it is experimentally observed that the proposed algorithm reduces cost of running tasks and average VM utilization as compared to other scheduling approaches namely round robin, Min-Min and Max-Min schemes. The proposed algorithm can further be improved by considering some other parameters like make span and load balancing of VMs which are also playing a key role in scheduling cloud tasks and comparing the proposed algorithm with other dynamic batch mode scheduling algorithms such as Berger model. As a future work, the tasks exhibiting dependency among them should also be considered for scheduling. Further, in our future work we propose to take up a practical case study with an intuitive illustration to demonstrate the proposed algorithm in this paper.

References

[1] Nakku Kim, Jungwook Cho, Euiseong Seo. 2014. Energy-credit scheduler: An energyaware virtual machine scheduler for cloud systems, *Future Generation Computer Systems*, 32,128-137.

[2] Jinn-Tsong Tsai, Jia-Cen Fang, Jyh-Horng Chou.2013. Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm, *Computers and Operations Research*, 40,Issue 12,3045-3055.

[3] Thamarai Selvi Somasundaram, Kannan Govindarajan. 2014. CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud, *Future Generation Computer Systems*, Vol.34, 47-65.

[4] Francesco Palmieri, Luigi Buonanno, Salvatore Venticinque, Rocco Aversa, Beniamino Di Martino. 2013. A distributed scheduling framework based on

selfish autonomous agents for federated cloud environments, *Future Generation Computer Systems*, Vol.29,Issue 6, 1461-1472.

[5] Baomin Xu ,Chunyan Zhao, Enzhao Hu, Bin Hu. 2011. Job scheduling algorithm based on Berger model in Cloud environment, *Journal of Advances in Engineering Software*, vol. 42, 419–425.

[6] Xiaoli Wang, Yuping Wang, Yue Cui.2014. A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing, *Future Generation Computer Systems*, 36,91-101.

[7] Chia-Ming Wu, Ruay-Shiung Chang, Hsin-Yu Chan.2014. A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters, *Future Generation Computer Systems*, 37,141-147.

[8] Saurabh Kumar Garg , Chee Shin Yeo, Arun Anandasivam, Rajkumar Buyya. 2011. Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers, *Journal of Parallel and Distributed Comput*ing, Vol.71,Issue 6, 732-749.

[9] M. Mezmaz, N.Melab, Y. Kessaci, Y.C. Lee, E.-G. Talbi , A.Y. Zomaya, D. Tuyttens. 2011. A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems, *Journal of Parallel and Distributed Computing*, Vol.71,Issue 11, 1497-1508.

[10]Sen Su, Jian Li, Qingjia Huang, Xiao Huang, Kai Shuang, Jie Wang. 2013. Cost-efficient task scheduling for executing large programs in the cloud, *Parallel Computing*, Vol.39,Issues 4-5, 177-188.

[11]Young Choon Lee, Chen Wang, Albert Y. Zomaya, Bing Bing Zhou. 2012. Profit-driven scheduling for cloud services with data access awareness, *Journal of Parallel and Distributed Computing,* Vol.72,Issue 4, 591-602.

[12]Mustafizur Rahman, Rafiul Hassan , Rajiv Ranjan and Rajkumar Buyya.2013. Adaptive workflow scheduling for dynamic grid and cloud computing environment, *concurrency and computation: practice and experience* , 25,1816-1842.

[13]Hamid Mohammadi Fard, Radu Prodan, and Thomas Fahringer.2013. A Truthful Dynamic Workow Scheduling Mechanism for Commercial Multicloud Environments, *IEEE Transactions on Parallel and Distributed Systems*, Vol.24, No.6,1203-1212.

[14]Saeid Abrishami , Mahmoud Naghibzadeh, Dick H.J. Epema. 2013. Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds, *Future Generation Computer Systems* ,29, 158-169.

[15]Joel J.P.C. Rodrigues , Liang Zhou , Lucas D.P. Mendes , Kai Lin , Jaime Lloret. 2012. Distributed media-aware flow scheduling in cloud computing environment, *Computer Communications,* 35, 1819-1827.

[16]P.D. Zegzhda, D.P. Zegzhda and A.V. Nikolskiy. 2012. Using graph theory for cloud system security modeling, MMM-ACNS 2012, , LNCS, 7531, 309–318.