DOI: 10.1051/matecconf/20165702009

Optimization of Task Scheduling Algorithm through QoS Parameters for Cloud Computing

Monika[1], Abhimanyu Jindal[2,a]

[1]*Assistant Professor, U.I.E.T., PU, Chandigarh, India*
[2]*Research Student, U.I.E.T., PU, Chandigarh, India*

Abstract. Cloud computing is an incipient innovation which broadly spreads among analysts. It furnishes clients with foundation, stage and programming as enhancement which is easily available by means of web. A cloud is a sort of parallel and conveyed framework comprising of a gathering of virtualized PCs that are utilized to execute various tasks to accomplish good execution time, accomplish due date and usage of its assets. The scheduling issue can be seen as the finding an ideal task of assignments over the accessible arrangement of assets with the goal that we can accomplish the wanted objectives for tasks. This paper presents an optimal algorithm for scheduling tasks to get their waiting time as a QoS parameter. The algorithm is simulated using Cloudsim simulator and experiments are carried out to help clients to make sense of the bottleneck of utilizing no. of virtual machine parallely.

1 Introduction

Cloud computing is as of late a blasting area in distributing computing that conveys powerfully versatile administrations on interest over the Internet through virtualization of equipment and programming. Cloud computing provide on demand services such as SaaS (Software as a service), IaaS (Infrastructure as a service) and PaaS (Platform as a service) to the clients. The customers can access all required hardware, software, platform, applications, infrastructure and storage with the ownership of just a web association. They access required resources and services on pay-per-use basis. The growth of cloud has reduced the performance of the resources. Aside from numerous approaches to upgrade the throughput and productive resource usage one way is task scheduling. Through task scheduling we can deal with the resource use which thus expands the throughput of the whole system. Task scheduling is the process of assigning tasks to the resources of the cloud efficiently so as to improve performance of the cloud in terms of QoS parameters. Scheduling is an NP- hard problem [1]. Allocation of tasks to the resources should be done carefully so as to minimize execution time and cost of the customer requests. If tasks are submitted according to length, priority or performance, the performance of the system may be improved significantly. A proficient task scheduling algorithm enhances the system throughput and helps service provider to provide good QoS. To get maximum benefit, scheduling is needed with distribution of load on the available resources[2]. The performance of scheduling algorithm is usually measured in terms of turnaround time, execution time, waiting time, cost etc. which must be minimized for improved performance of system. Using so as to minimize execution time and cost brought about heterogeneous resources over cloud is an imparative issue. Cost is another parameter which needs to be kept in mind while working on cloud and should me minimum [3]. Cloud systems have limited resources and they are very much costly. Scheduling algorithm not only improves QoS parameters but also reduce the expenses of the customer by efficiently scheduling tasks on the resources. In order to get high throughput, various issues need to be considered. To get most extreme advantage as far as make-span, planning of tasks is fundamental with dispersion of burden on processors and run- time allocation of resources [4]. Cloud performance may be increased by giving quick response to the users whenever they require dynamically adjusting resources of the system. As fast as the use of cloud services is increasing, the security issues have also been increased which needs to be handled carefully while working with cloud [5].

Energy consumption of system need also be reduced for efficient working. The size of cloud base is growing quick which bring about colossal energy utilization and working expense. Cloud workload is additionally a critical issue in scheduling. Provisioning of suitable resources to cloud workload relies upon nature of service necessities of applications. Cloud infrastructure should also support other applications such as big data and soft real-time applications which are the base of today's internet [6]. Where all the scheduling algorithms aim at improving the performance of system and maximizing the profit of service provider, customer satisfaction and feed back is also important. Thus a resource scheduling mechanism powered with a relevance feedback network

[a] Corresponding author: erabhi.puchd@gmail.com

is required [7]. Hence, a high performance task scheduling algorithm is required for improved throughput of the system. In this paper, an optimal algorithm for task scheduling is proposed which execute tasks according to their priorities in the system.

The rest of the paper is organized as follows. The section 2 describes some similar work related to the task scheduling, section 3 describes the proposed work, section 4 describes the experiment and results and the paper is concluded in section 5.

2 Related work

The cloud technology is growing fast due to its ease in use and its performance. The performance of cloud is increased due to various scheduling algorithms developed in the past few years. The performance of scheduling algorithm is defined on the basis of various QoS parameters i.e. execution time, arrival time, waiting time, deadline, cost, scalability and many more. The main role in scheduling is played by a broker that is responsible for allocation of tasks to the VMs (Virtual Machines) while the system is executing user requests.

One of the traditional methods of task scheduling is done by assigning priority to the tasks or VMs. In [8] authors proposed an algorithm PISA (Priority Impact Scheduling Algorithm) which schedules user tasks according to the priority given by the customer itself. The results of the algorithm show that the high priority tasks executed successfully with a higher ratio than FIFO algorithm by 26.7%. In [9] authors have proposed an algorithm for priority scheduling in which both the tasks and VMs are assigned with priorities and then tasks are scheduled on the VMs. The QoS value of tasks is gathered and then the system assigns priorities to the tasks. The task with low QoS value is given highest priority, next task with 2nd lowest QoS value is assigned 2nd highest priority and so on. The list of Vms is assigned priorities on the basis of MIPS. The VM with highest MIPS is assigned highest priority then the VM with 2nd highest MIPS value and so on. The highest priority task is submitted to the highest priority VM and so on. As the last VM is allocated, the next task is allocated on the first VM again. A similar algorithm is proposed in [10] which allocates tasks on VMs based on their priority. The priority may be assigned on the basis of any parameter such as length, user requirement or on the basis of previous performance statistics.

In [11] authors have proposed CDB task scheduling algorithm which schedules tasks based on the cost of execution of task that is calculated using deadline and length of the cloudlet. The algorithm performs well in achieving the deadlines of the tasks. In [12] Budget and Deadline Constrained Heuristic is presented which is based upon Heterogeneous Earliest Finish Time and results in reduction of the cost and execution time of the system. S. Sindhu and S. Mukherjee [13] proposed two algorithms LCFP and SCFP which schedule the cloudlets on the VMs according to their length. In LCFP, longest cloudlet is scheduled first in the system which results n reduction in the reduction in makespan of the system. In SCFP, shorter cloudlets are schedulet first which reduce the flowtime of the overall execution. Zhanjie Wang and Xianxian Su proposed a dynamically hierarchical resource allocation algorithm which dynamically divides the nodes and tasks into various levels according to their computing power and the newly arriving task joins the node of its own level only[14]. This algorithm reduced the makespan of the system. Another algorithm proposed by the authors of [15] is integrated with priority and SJF(Shortest job first) schemes and it reduced the waiting time and processing time by 30%.

Reference [16] proposed some mathematical models which aims to reduce execution time of the tasks by efficiently allocating VMs. Similar work is done in [17, 18, 19] where scheduling is done in order to reduce execution time and arrival time. Authors in [20] proposed a Dynamic task scheduling scheme DGS which allocates computing tasks to the virtual machines using greedy strategy and the scheme results in reduction in the completion time and improvement in the resource utilization. A Bin-Balancing Algorithm is proposed in [21] which aims at reducing the energy consumption by the system taking into consideration the deadlines of the tasks. Sukhpal Singh and Inderveer Chana proposed a framework in [22, 23] which reduces the execution time of the system along with the cost and energy consumption.

3 Proposed work

3.1 Introduction

The cloud customers want their task to be done within lesser time while at the same time taking into consideration the priority and waiting time of each task. However, the existing algorithms do not provide proper information about the waiting time of the tasks. The waiting time which is the time a task waits in the queue before execution depends upon the no. of VMs and tasks in the system that are currently being utilized in execution process. Each task in the cloud environment experiences a different waiting time according to the length of the task currently being executed by the VM. In the proposed work, the task of allocation of cloudlets on the VMs is done by the broker and it is responsible for execution of each task according to its priority.

3.2 Problem statement

The main problem is to get the waiting time of the tasks with that of the execution time so that actual performance of the algorithm may be defined. The current algorithms do not consider the length of the tasks during their execution which results in starvation of the small length task in some cases.

Task is defined as Ti = {Ii, Li, Pi}, Where Ii is the ID of the ith task, Li is the length and Pi is the priority assigned to the task.

3.3 Assigning priorities for tasks

The proposed algorithm give priority to the task according to its length. The task of smallest length is assigned highest priority and the task of largest length is assigned the lowest priority. The length of the task is set previously and each one have a different length.

3.4 Virtual machine selection

The cloud broker maintains the list of tasks and the VMs. It sorts the list of tasks in ascending order of their assigned priorities. After the sorting, the final list of tasks is bound with the list of VMs. The first task in the task's list is allocated to the first VM in the VM's list, second task in the task's list to the second VM in the VM's list and so on. As soon as the allocation process reach the last VM, the next task will be allocated to the first VM of the VM's list and the process of allocation will be repeated for the rest of the tasks.

```
Algorithm 1: Parallelized task scheduling algorithm
Create a list of tasks (n)
Create a list of VMs (m)
Define priority Pi for each task Ti
Sort task list in descending order of priority
i⟵ 0 to n
j⟵ 0 to m
for (j⟵ 0 to m )
    {
    If (j<=m)
        {
        for (i⟵ 0 to n)
        {
           bind Ti to VMj;
           j++ ;
           i++ ;
        }
        }
     Else /* when last VM has been allocated*/
        {
           j=0 ; /* Repeat from first VM again*/
        }
    }
```

4 Experiment and results

In order to obtain the results of the algorithm, the simulation was done using CloudSim 3.0.3 simulator which was run over Eclipse Juno. The operating system used was Windows 7 with Core i3 processor. In this simulation scenario, fifteen VMs and fifty tasks were involved. All the tasks were of varied lenghts and have a priority assigned with them. Initially, all the tasks were allocated to only one VM and then the experiment was repeated by increasing number of VMs one by one (eg. Two VMs, three VMs, and so on). The results obtained from the experiment are shown in Fig. 1.

In Fig. 1, Y-axis is the waiting time and X-axis is the number of VMs. The results show that the waiting time reduces to approximately half when number of VMs is two for performing same work. However, the experimental outcomes shows that the decreasing curve has stopped while using more than ten VMs which means it has met the bottleneck of using VMs parallely. In cloud, users should use ten VMs at a time when the target is to reduce the waiting time for the tasks.

Table 1. Experimental setup.

OS	Window 7
Processor	Core i3
Memory	2 GB
Simulator	3.0.3
Platform	Eclipse

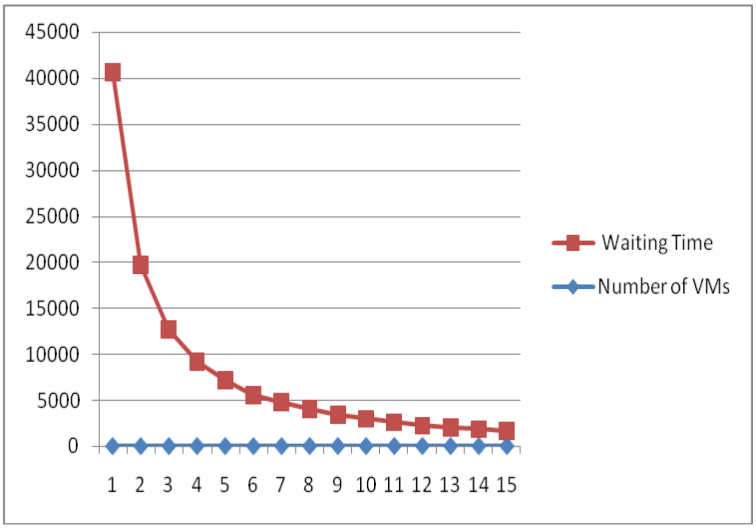

Figure 1. Output.

5 Conclusion and future work

Scheduling is the backbone of a cloud environment. It makes productive utilization of the resources while at the same time, considering the QoS parameters, is an important issue, since cloud infrastructure have limited resources and they are costly too. In this regard, an efficient scheduling algorithm is required so that the entire work might be accomplished in least time and could bring about lesser expense. In this study, an algorithm is presented for parallel processing of tasks and the outcomes gave us a bottleneck of utilizing number of VMs concurrently with the help of waiting time as a QoS parameter. The waiting time is reduced significantly by utilizing more number of VMs and scheduling tasks on the VMs according to their assigned priority.

In future, investigations will be done to reduce the waiting time of the tasks and an algorithm will be proposed to do as such.

References

1. J.A.J. Sujana, T. Revathi, G. Karthiga, R.V. Raj, Game multi objective scheduling algorithm for scientific workflows in cloud computing, IEEE, 1-6, (2015).
2. K.A. Saranu, S.Jaganathan, Intensified scheduling algorithm for virtual machine tasks in cloud computing, Springer, 283-290, (2014).
3. S. Selvarani, G.S. Sadhasivam, Improved cost-based algorithm for task scheduling in cloud computing," IEEE, 1-5, (2010).
4. S. Banerjee, M. Adhikari, S. Kar, U. Biswas, Development and analysis of a new cloudlet allocation strategy for QoS improvement in cloud, AJSE, Springer, **40**, 1409-1425, (2015).
5. U.A. Kashif, Z.A. Memon, A.R. Balouch, J.A. Chandio, Distributed trust protocol for IaaS cloud computing, IBCAST, IEEE, 275-279, (2015).
6. K. Cheng, Y. Bai, R. Wang, Y. Ma, Optimizing soft real-time scheduling performance for virtual machines with XRT-Xen, IEEE, 169-178, (2015).
7. D. Ding, X. Fan, S. Luo, User-oriented cloud resource scheduling with feedback integration, Springer, 1-22, (2015).
8. Hu Wu, Zhuo Tang, Renfa Li, A priority constrained scheduling strategy of multiple workflows for cloud computing, IEEE, 1086-1089, (2012).
9. A.V. Lakra, D.K. Yadav, Multi-objective tasks scheduling algorithm for cloud computing throughput optimization, ICICCC, **48**, 107-113, (2015).
10. C. Lin, S. Lu, Scheduling scientific workflows elastically for cloud computing, IEEE, 746-747, (2011).
11. Himani, H.S. Sidhu, Cost- deadline based task scheduling in cloud computing, ICACCE, IEEE, 273-279, (2015).
12. A. Verma, S. Kaushal, Cost- time efficient scheduling plan for executing workflows in the cloud, JGC, Springer , **13**, 495-506, (2015).
13. S. Sindhu, S. Mukherjee, Efficient task scheduling algorithms for cloud computing environment, Springer, 79-83, (2011).
14. Z. Wang, S. Su, Dynamically hierarchical resource-allocation algorithm in cloud computing environment, Springer, 2748-2766, (2015).
15. Jia Ru, Jacky Keung, An Empirical investigation on the simulation of priority and shortest-job-first scheduling for cloud based software systems, IEEE, 78-87, (2013).
16. Q.T. Nguyen, N.Q. Hung, N.H. Tuong, V.H. Tran, N. Thoai, Virtual machine allocation in cloud computing for minimizing total execution time on each machine, IEEE, 241-245, (2013).
17. A.K. Das, T. Adhikary, C.S. Hong, An intelligent approach for virtual machine and QoS provisioning in cloud computing, IEEE, 462-467, (2013).
18. R. Achar, P.S. Thilagam, Shwetha D, Pooja H, Roshni, Andrea, Optimal scheduling of computational task in cloud using virtual machine tree, ICEAIT, IEEE, 143-146, (2012).
19. W.J. Wang, Y.S. Chang, W.T. Lo, Y.K. Lee, Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments, Springer, **66**, 783-811, (2013).
20. Liang Ma,Y. Lu, F. Zhang, S. Sun, Dynamic task scheduling in cloud computing based on greedy strategy, Springer, 156-162, (2013).
21. J.M. Tang, L. Luo, K.M. Wei, A heuristic resource scheduling algorithm for cloud computing based on polygons correlation calculation, ICEBE, IEEE, 365-370, (2015).
22. S. Singh, I. Chana, QRSF: QoS aware resource scheduling framework in cloud computing, Springer, **71**, 241-292, (2014).
23. S. Singh, I. Chana, Resource provisioning and scheduling in clouds: QoS perspective, Springer, 1-35, (2016).