A Scheduling Approach in Cloud Environment using Deadline and VMs MIPS and Bandwidth

Anamika* and Pawan Luthra**
*Research Student
anamika3534@gmail.com
**Assistant Professor
pawanluthra81@gmail.com

Abstract: Cloud computing is the new innovation which is presently being utilized by PC users. Cloud registering gives more solace in the advanced time as there is no compelling reason to convey substantial gadgets with you since everything is accessible over the Internet. A cloud is gathering of number of virtual machines which can deal with different procedures to execute parallel. As cloud is growing there is a huge increase in number of cloud users. When there are various undertaking demands over the cloud then there is a need to choose which assignment will be executed first so that QOS parameters give better results. Scheduling of the cloudlets implies discovering the suitable and the required resources for that assignment and after that schedules the cloudlet onto that resource to achieve the objective for that cloudlet. Credit based scheduling algorithm considers only cloudlet length and cloudlet priority to schedule the cloudlet but in this approach a calculation has been characterized expecting cloudlet length, cloudlet priority, cloudlet dynamic submission time and cloudlet deadline, VM MIPS and VM bandwidth to decrease response time and waiting time using CloudSim 3.0.3 simulator.

Keywords: Cloud computing, scheduling, deadline, cloudlet, bandwidth.

Introduction

Cloud computing implies controlling, getting to and arranging the applications online and benefiting applications, infrastructure and data storage over the Internet. The clients are not required to introduce a product on their PCs, they can have the administrations of programming over the system through cloud. It is less costly in light of the fact that clients don't have to buy the entire framework, they will pay according to their utilization. Virtualization is a method which is utilized behind the cloud computing. It permits to share single physical object of a resource or application among numerous number of clients. Cloud design comprises of two parts. Front end is the customer piece of cloud computing framework which is made out of uses and framework that are fundamental to get to the cloud computing stage for instance web browser. Back end is cloud comprising of the considerable number of cloud resources crucial to give cloud computing benefits that are data storage, virtual machines, security systems, servers, deployment models. In public cloud model the administrations are effortlessly open to overall population for example Google. In private cloud model the clients connected with the specific association can get to the administrations of private cloud. The hybrid cloud model is combination of public and private cloud. Decisive cloudlets are taken care of in private cloud and indecisive are taken care of in public cloud. In community cloud gathering of associations is permitted to get to its administrations. Cloudlet scheduling is the procedure of dispensing cloudlets onto accessible resources in time taking after the limitations portrayed by the client and the cloud supplier. Research on cloudlet scheduling in cloud computing has turned into an interesting issue. The main motive of cloudlet scheduling algorithm is to allocate resources to cloudlet and minimize completion time, maximize resource utilization [1]. Load balancing is one of the fundamental challenge in cloud computing. It is a method which is required to circulate the dynamic workload over different virtual machines to guarantee that no single virtual machine is overloaded. Load balancing procedures help in ideal usage of resources and thus in upgrading the performance of the framework. The objective of load balancing is to minimize the resource consumption which will facilitate decrease power usage and carbon emanation rate that is the critical need of cloud computing [2]. Despite the fact that cloud computing has an extensive advantages over customary registering models, yet it raises serious security worries that point of confinement its across the board selection loss of administration, information security and unreliable information deletion are a few examples [3]. Datacenters associated with a cloud system are typically geographically distributed, yet connected together with dedicated high-bandwidth communication links. Latest trends demonstrate that cloud computing is developing to traverse more globally distributed datacenters. For geo-distributed datacenters, there is an expanding requirement for calculations to place assignments crosswise over datacenters, by together considering information and computation [4]. Power management approaches which mean to diminish total energy consumption in datacenters challenges in both hardware and resource administration policies. The ideal usage level of a host to execute a specific number of cloudlet to minimize energy consumption of the host is required [5]. Cloud computing offers diverse resources over network appears an option answer for software testing. In any case, how to assess and allot these effective assets to particular testing projects are still required further examination [6]. It is critical to precisely appraise the reliability of a cloud computing framework keeping in mind the end goal to better reduce faults and accordingly successfully use its execution to accomplish the SLA of cloud users [7]. Hence an optimal cloudlet scheduling algorithm is needed to decrease the completion time and increase resource utilization. In this paper an algorithm has been designed in which cloudlets get scheduled on the cloud resources according to cloudlet priority, cloudlet length, cloudlet dynamic submission time and cloudlet deadline, VM (Virtual Machine) bandwidth and VM MIPS. The rest of the paper is sorted out as follows. The segment 2 specify some related work on cloudlet scheduling and part 3 portrays the proposed work, part 4 describes experimental setup and results and part 5 concludes the whole process.

Related Work

The main motive of cloudlet scheduling in cloud computing is to schedule cloudlets onto the available resources so that the resources are utilized and user requests are satisfied. The datacenter broker plays the main role for scheduling cloudlets onto virtual machines as it knows in advance both the cloudlets and available datacenters. The performance of cloudlet scheduling is evaluated by calculating execution time, response time, waiting time, turnaround time and cost etc by assuming QOS parameters like cloudlet length, cloudlet priority, cloudlet deadline, VM MIPS and VM bandwidth.

The paper [8] defines scheduling algorithm according to cloudlet priority and completion time. The priority of the cloudlet is determined by indicated properties like client benefits, need expected by the client, length of the assignment and its workload. At that point cloudlet with least completion time is assigned first so that every one of the cloudlet could be finished in less time and load on the framework is balanced. In the paper [9] novel static planning calculation has been proposed taking into account time-sharing virtual machines. It can advance the makespan of cloudlets inside of a datacenter. In [10] two level scheduling algorithm is used taking into account load balancing. The methodology considers both the dynamic requirements of client and load over the cloud resources, thus it meets both the dynamic client requirements and expansion usage of resources. The paper [11] highlights another scheduling strategy with suitable load adjusting system that aides in circulating the cloudlets to the VMs just as prone to their capacity which makes the framework more dynamic, alive, and adjusted. This reduces the completion time of the cloudlets and in addition decreases the makespan of the VMs and the hosts of a datacenter. The algorithm presented in [12] ensures the resource utilization and the completion of cloudlets before its deadline by utilizing two methodologies EDF (Earliest Deadline First) and LWF (Largest Weight First).If the cloud has enough resources to execute then the VM will be apportioned to the occupation and occupation will be expelled from the holding up line and machine status will be changed to busy from idle. But if the required resources are not accessible then this calculation will utilize the LWF as backfilling algorithm, to choose the cloudlet with most extreme waiting time. In [13] authors presented a heuristic scheduling strategy which schedules the cloudlets on resources and aims at improving overall performance of the system. Paper [14] presented a heuristic which aims at reducing the cost of the system by addressing all the servers. The authors of [15] also presented a monitoring scheme of resources to check the usage of resources and migrate them dynamically to improve the overall throughput of the system. Similar algorithm is proposed in [16] which too aim at utilizing the resources and minimize the processing time of the customer requests. Beloglazov and R. Buyya [17] proposed a resource management policy which monitors the resources of the system, migrates them from one cloudlet to another dynamically and switches the idle resources off. This policy saved the energy of the system to a significant extent. Authors of [18] also proposed an approach for resource management which saves the wastage of resources in cloud. D. Warneke and O. Kao [19] presented a framework Nephele which schedules the cloudlets on the different types of VMs by exploiting the resources of the cloud dynamically. In [20] authors have proposed a scheduling algorithm which reduces the turnaround time of the cloudlets in heterogeneous resource environment. The algorithm utilizes the greedy approach for allocation of cloudlets on the resources of cloud. Jun Wu [21] proposed an algorithm BATS (Blocking-aware two-speed) which synchronize the cloudlets on the shared resources dynamically to saves the energy of the resources and aids the cloudlets to meet their deadlines. Another energy efficient scheduling algorithm VILCF(Voltage Island Largest Capacity First) [22] schedules the periodic real-time cloudlets on the multi-core processors and utilizes the resource energy efficiently. A similar work is done in [23] to improve the energy efficiency of system. The authors in [24] proposed a cloudlet scheduling model that minimizes the energy consumption by reducing the number of VMs. In [25] authors proposed an Adaptive and hierarchical cloudlet scheduling scheme (AHS) for dynamically scheduling cloudlets in cloud. The authors in [26] proposed a multi-objective scheduling (MOS) scheme which aims at minimizing makespan and at the same time reducing the resource cost and preserving fault tolerance. In [27] the authors state that deadline of a task is an important factor and highest priority is assigned to the task having earliest deadline. In [28] the authors describe the scheduling algorithm in which tasks are kept in queue and task with least completion time gets executed first and cost matrix is being generated.

Proposed Work

Introduction

Cloudlet scheduling in cloud environment means deciding which resource will be available to which cloudlet at a particular time. There are various QOS parameters which are considered while scheduling cloudlets over datacenters e.g. cloudlet priority, cloudlet length, cloudlet deadline, VM bandwidth, VM MIPS. While scheduling cloudlets the main motive is to reduce response time, cost, execution time, waiting time and turnaround time. Deadline is the final time by which the cloudlet must complete its execution, waiting time is the time for which the cloudlet waits in the ready queue, response time is the time by which the cloudlet gives its first response in case of time sharing system and dynamic submission time is the time at which the cloudlet gets submitted to the broker. The existing algorithm only calculates the makespan while considering two QOS parameters cloudlet length and cloudlet priority. But the proposed algorithm calculates and reduces waiting time and response time while considering cloudlet length, cloudlet priority, cloudlet dynamic submission time, cloudlet deadline, VMs bandwidth and VMs MIPS.

Problem Statement

The proposed algorithm considers QOS Parameters like cloudlet length, cloudlet priority, cloudlet deadline, cloudlet dynamic submission time, VMs bandwidth and MIPS. The cloudlet having lower length, higher priority and earlier deadline gets scheduled first onto the VM that is having highest value of bandwidth and MIPS. The main motive to use dynamic submission time is to reduce requests on a single resource by various cloudlets. Hence there are less number of cloudlet requests for resources and the scheduler can take quick decisions and waiting time is reduced.

Assigning Priority and Deadline to Cloudlets

The proposed algorithm assigns priority and deadline to the cloudlet as defined by the user. In the simulation process unique priority and deadline are randomly generated by the system.

Sorting Cloudlets according to QOS parameters

Credit is assigned to each task according to cloudlet length, cloudlet priority and cloudlet deadline; to assign credit to cloudlets according to deadline Algorithm 1 is followed. To assign credit according to cloudlet length and cloudlet priority CBSA [29] is followed; firstly average length is calculated and compared with individual cloudlet length and then credit is assigned to each cloudlet in the range of 1 to 5 following CBSA [29] and the total credit is calculated by multiplying all three credits and cloudlets are sorted according to descending order of total credit.

Algorithm 1: To sort cloudlets according to credit Assigned

```
Create tasks form i to n with length Li.
Assign priority Pi.
Assign deadline Di.
Assign credit to task based on length and priority according to CBSA
To Assign credit according to deadline
For(i to n)
{
   If(deadline<=9)
      {
           deadline factor=10
      }
   else
      {
          deadline factor=100
       }
Deadline[i]=deadline/deadline factor
}
Total credit=length credit*priority credit*deadline credit;
Sort tasks in descending order according to total credit
```

Creation of VMs

MIPS and bandwidth are defined for VMs and they are sorted in descending order according to combined value of VMs bandwidth and MIPS.

Binding of cloudlets to VMs

Cloudlet with the lower value of credit gets bind to VM with highest MIPS and bandwidth, cloudlets are submitted to the broker with the difference in submission time and then they have to wait lesser time in the queue as a result of which waiting time and response time has been reduced.

Algorithm 2: To Sort VMs according to bandwidth and MIPS and bind cloudlet to VM

```
Generate VMS for j to n

Assign bandwidth and MIPS

  For(j to n)

    {

      Sort VMs according to bandwidth and MIPS in descending order

     }

Dynamic submission time difference = 4ms +5( i) ;

bindcloudletToVm (cloudlet_i, VM_j)

When VM in VM list are finished then next cloudlet is assigned to first VM in VM list.
Repeat until all the Cloudlets in cloudlet list get executed.
```

Experimental Setup and Results

The simulation was done on CloudSim 3.0.3 simulator on eclipse platform. The operating system used for this simulation was Window 7 with core 2 duo processor with 3GB ram and 320 GB hard disk. In this simulation 50 cloudlets having length 1000-50000 are processed over 10 VMs starting from one VM and then increasing the number of VMs one by one with MIPS ranges from 300 to 1000. Waiting time and response time has been calculated for varying number of VMs.

Table 1. Comparison Table

No. of VMs	Waiting Time(in ms)		Response Time(in ms)	
	CBSA	Proposed Algorithm	CBSA	Proposed Algorithm
1	416	324	441	341
2	202	127	227	155
3	130	63	156	91
4	94	40	115	59
5	73	29	99	56
6	59	20	84	52
7	49	11	74	44
8	41	8	67	40
9	35	6	61	37
10	31	4	54	35

As form the comparison table it is clear that waiting time and response time is improved as compared to CBSA [29].

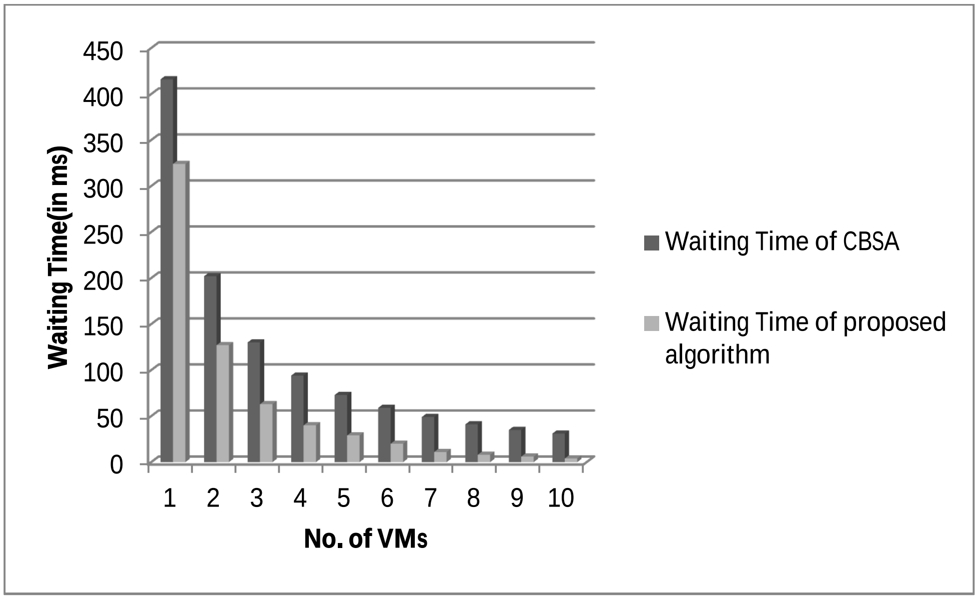

Fig 1.Waiting Time Comparison

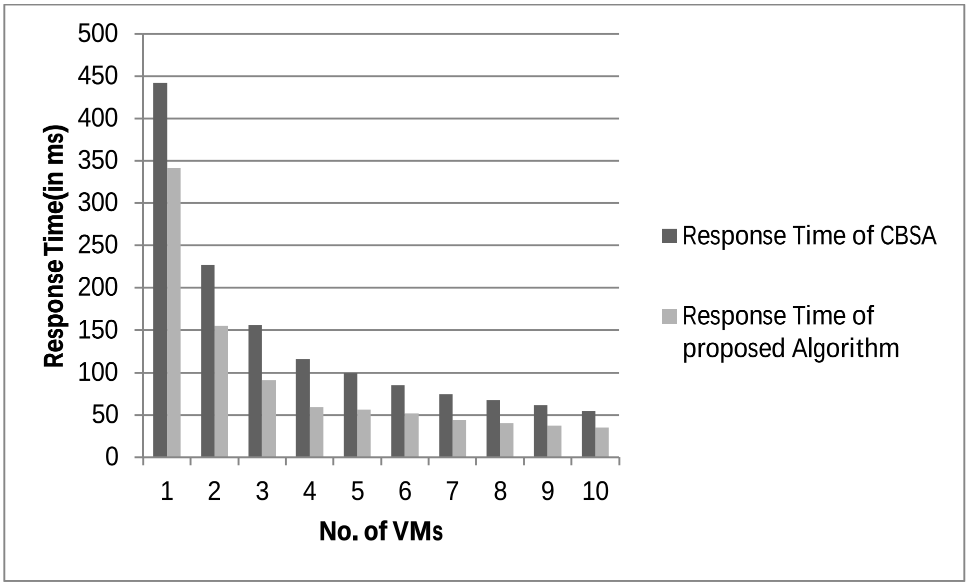

Fig 2. Response Time Comparison

Conclusion

In this paper, a scheduling algorithm is proposed so as to reduce the waiting time and response time of the cloudlets and the experiments conclude that the algorithm outperforms CBSA[29] and the waiting time and response time are reduced significantly through this algorithm. The outputs of the algorithm covey that cloudlet deadline, cloudlet dynamic submission time, VMs MIPS and bandwidth are also important to be considered while working with cloud. Future work would involve improving these factors with the help of more QoS parameters.

References

[1] Liping Zhang, Weiqin Tong, Shengpeng Lu, “Task scheduling of cloud computing based on Improved CHC algorithm,” IEEE, july 2014, pp.574-577.

[2] Velagapudi Sreenivas, Prathap.M, Mohammed Kemal, “Load balancing techniques: Major challenge in Cloud Computing - a systematic review,”IEEE, Feb 2014, pp.1-6.

[3] Fatimah M. Alturkistani, Ahmed Z. Emam, “A Review of Security Risk Assessment Methods in Cloud Computing,” Springer, vol.275, 2014, pp 443-453.

[4] Lingyan Yin, Jizhou Sun, Laiping Zhao, Chenzhou Cui, Jian Xiao, Ce Yu, “Joint scheduling of data and computation in Geo-distributed cloud systems,”15th International Symposium on Cluster, Cloud and Grid Computing IEEE, may 2015 ,pp.657-666.

[5] Seyedmehdi Hosseinimotlagh, Farshad Khunjush, Rashidaldin Samadzadeh, “SEATS: smart energy-aware task scheduling in real-time cloud computing, “Springer, 2015, vol. 71, pp.45-66.

[6] Chorng-Shiuh Koong, Tzu-I Yang, Chang-Chung Wu, “An Implementation of Automatic Software Testing on Cloud Computing Environment,” Springer, 2015,vol.330, pp.1253-1259.

[7] Thanadech Thanakornworakij, Raja F. Nassar, Chokchai Leangsuksun, Mihaela Paun, “A reliability model for cloud computing for high performance computing applications,” Springer, 2013, vol.7640, pp.474-483.

[8] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng and Shengyuan Zhou, “A task scheduling algorithm based on QOS driven in cloud computing,” Science direct, Procedia computer science, 2013, pp.1162-1169.

[9] Xuejun Li,Siyu Zhou,Jian Wang,Xiangjun Liu,Yiwen Zhang,Cheng Zhang,Yun Yang, “Time sharing virtual machine based efficient task level scheduling in scientific cloud workflows,” Springer, 2014, vol.309, pp.121-126.

[10] Yiqiu Fang Fei Wang, Junwei Ge, “A task scheduling algorithm based on load balancing in cloud in cloud computing,” Springer, 2010, vol.6318, pp.271-277.

[11] Soured Banerjee, Mainak Adhikari,Sukhendu Kar, Utpal Biswas, “Development and Analysis of a New Allocation Strategy for QoS Improvement in Cloud,” Springer, May 2015, pp.1409-1425.

[12] SaeMi Shin, , Yena Kim and SuKyoung Lee, “Deadline guaranteed scheduling algorithm with improved resource utilization for cloud computing,” IEEE, 2015, pp.814-819.

[13] Sameer Singh Chauhan, R. C. Joshi, “A waited mean time min-min max-min selective scheduling strategy for independent cloudlets on grid,” IEEE, Feb 2010, pp.4-9.

[14] Benjamin Speitkamp, Martin Bichler, “A mathematical programming approach for server consolidation problems in virtualized datacenters,” IEEE, 2010, vol.3, pp.266-278.

[15] Jaideep Moses, Ravi Iyer, Ramesh Illikkal, Sadagopan Srinivasan, Konstantinos Aisopos, “Shared resource monitoring and throughput optimization in cloud computing datacenters,” IEEE, July 2011, pp.192-196.

[16] S. Gomathi, D.Manimegalai, “An adaptive grouping based job scheduling in grid computing,” IEEE, May 2011, pp.1024-1033.

[17] Anton Beloglazov, Rajkumar Buyya, “Energy efficient allocation of virtual machines in cloud datacenters,” IEEE, May 2010, pp.577-578.

[18] Yagız Onat Yazır et al, “Dynamic Resource Allocation in computing clouds using distributed multiple criteria decision analysis,” IEEE, 2010, pp.91-98.

[19] Daniel Warneke, Odej Kao, “Exploiting dynamic resource allocation for efficient parallel data processing in the cloud,” IEEE, June 2011, vol.22, pp.985-997.

[20] Saumitra Singh, Krishna Kant, “Greedy grid scheduling algorithm in dynamic job submission environment,” IEEE, 2011, pp.933-936.

[21] Jun Wu, “Energy-efficient scheduling of real-time tasks with shared resources,” Elsevier, 2016, pp.179-191.

[22] Jun Liu, Jinhua Guo, “Energy efficient scheduling of real time tasks on multi core processors with voltage island,” Elsevier, 2016, pp.202-210.

[23] QingZhao, Congcong Xiongn, CeYub,ChuanleiZhan,XiZhao, “A new energy aware task scheduling method for data-intensive applications in cloud,” Elsevier, 2016, pp.14-27.

[24] YonghuaXiong, ShaoyunWan, JinhuaShe, MinWu, YongHe, KeyuanJiang, “An energy-optimization-based method of task scheduling for a cloud video surveillance center,” Elsevier, 2016, pp.63-73.

[25] Yizhuo Wanga, Yang Zhang,Yan Su,Xiaojun Wanga,Xu Chen,Weixing Ji, Feng Shi, “An adaptive and hierarchical task scheduling scheme for multi core clusters,” Elsevier, 2014, pp.611-627.

[26] Fan Zhang, ,Junwei Cao,Keqin Li,Samee U. Khand, Kai Hwange, “Multiobjective scheduling of many tasks in cloud platform,” Elsevier, 2014, pp.309-320.

[27] E. Iniya Nehru, Saswati Mukherjee and Abhishek Kumar, “Deadline based priority management in cloud,” Springer, 2015, pp. 745-751.

[28] Mousumi Paul, Goutam Sanyal, “Task scheduling in cloud computing using credit based assignment problem,” IJCSE,2011,pp. 3426-3430.

[29] Antony Thomasa, Krishnalal Ga, agathy Raj V P, “Credit based scheduling algorithm in cloud computing environment,” Elsevier, 2015, vol.46, pp.913-920.