APTIKOM Journal on Computer Science and Information Technologies
Vol. 4, No. 3, 2019, pp. 112~124
ISSN: 2528-2417
DOI: 10.11591/APTIKOM.J.CSIT.148

An approach towards development of a migration enabled improved datacenter broker policy

Debashis Das[1], Sourav Banerjee[2],Ayan Kundu[3], Swagata Chandra[4], Saptarshi Pal[5], Utpal Biswas[6]
[1,2,3,4,5]Kalyani Government Engineering College, Kalyani, India-741235
[6]University of Kalyani, Kalyani, India
*Corresponding author, debashis2124@gmail.com[1], mr.sourav.banerjee@ieee.org[2], ayankundu35@gmail.com[3], chandraswagta9@gmail.com[4], mail4riju93@gmail.com[5], utpal01in@yahoo.com[6]

Abstract

Cloud computinghas left its remarkable note on the computing world over the last few years. Through its effectiveness, litheness, scalability & availability cloud computinghas changed the nature of computer system deployment. The Quality of Service (QoS) of a cloud service provider (CSP) is an important element of research interest which includes different critical issues such as proper load, minimization of waiting time, turnaround time, makespan and suppressing the wastage of bandwidth of the system. The Datacenter Broker (DCB) policy helpsassigning a cloudletto a VM. In present study, we proposed an algorithm, i.e., Migration enabled Cloudlet Allocation Policy (MCAP) for allocation of cloudlets to the VMs in a Datacenter by taking into accounttheload capacity of VMs and length of the cloudlets. The experimental results obtained using CloudSim toolkit under extensive loads that establish performance supremacy of MCAP algorithm over the existing algorithms.

Keywords*: allocation policy, load balancing, makespan, migration enabled cloudlet allocation policy, quality of service*

1. Introduction

Distributed environments [1] have left its waves in the pooled platform to utility-based models. The cutting-edge technology used for this platform is cloud computing [2].It enables the delivery of computer resources through the network. It follows an on-demand service model where the users are charged based on their usage. There are various types of cloud service available such as -Infrastructure as a Service (IaaS), Software as a Service (SaaS), Platform as a Service (PaaS), Database as a Service (DaaS), and Identity as a Service (IdaaS). Though cloud computing [3, 4] offers the scalability, usability, and flexibility to acquire or relinquish resources, it also allocates resources of varying configuration to the best suit of as per the requirements of theapplication. Thisgives the customers more efficient manage over the resources by the efficient scheduling techniques [5]. So that resources of the cloud computing system [6] are utilized efficiently. Cloud computing [7] refers to applications and services that run on a distributed network using virtualized [8, 9] resources and accessed by common internet protocols and networking standards. Computers in a cloud environment dynamically allocate one or more unified computing resources established through internet connection between the cloud service providers and cloud users. It empowers the concept of location independence anywhere in the world. It delivers the user with sophisticated infrastructure even where suchhigh-level infrastructure is impossible to setup. Although cloud computing [10] eliminates the barrier controlling of various computing resources and reduce the cost of this high ended infrastructure. Cloud computing is thus a business unit where companies provide computation power, huge storage, and various other software services to users through the internet without knowing where the resources are physically situated [11].The background elements in this system such as virtual machine (VM) allocationpolicies [12], load balancing [13], load sharing [14], process migration, and distributed shared memory access are completely abstracted from the customer's perspective. The customers can only access and control the cloud-based applications as well as infrastructure by logging in to a cloud interface. The cloud service providers deliver better service and performance than the software programs installed locally on end-user computers. Allocatingcloudlets to VMs [15] isan inspiring issue in a heterogeneous cloud like environment. To make the environment more dexterous, it requires an efficient allocation policy by which more efficiency will be acquired. There are many existing cloudlet allocation policies in cloud computing to allocate the cloudlets to the different resources [16] or VMs. The cloudlet

Received August 9, 2019; Revised Septembert 13, 2019; Accepted October 16, 2019

allocation policy plays a key role to improve the overall system performance maximizing throughput and minimizing the completion time and makespan [17] of the entire cloud system. Moreover, a proper allocation policy may lead to a proper load balancing [18] of the host(s) and allocates cloudlets to the suitable resources [18] or VMs that may lead to improve the QoS [18-20] of the overall system.

2. CloudSim Toolkit

Several Grid simulators [21, 22] such as Sim-Grid, GridSimand GangSim are suitable for simulating the grid application in a distributed environment but cannot deliver the infrastructure and application-level requests arising from cloud computing environment. The grid simulators cannot distinguish the multi-layer services (such as, SaaS, PaaS, and IaaS, etc.) which are essential in cloud computing environment. Therefore, cloud environment modelling and simulation toolkits support for enabling real-time response of services between cloud customer and cloud service providers. TheCloudSim [23] is a cloud environmentmodelling and simulation toolkit [24] that provisions real-time response of services between customers and CSP. An open source CloudSim framework [25] shown in Figure 1 developed on Grid- Sim toolkit [26] conveys support for economically resource management and cloudlet scheduling, [26]cost management, bandwidth management etc. One of the important factors that make cloud computing infrastructure different from grid computing infrastructure is the huge deployment of virtualized infrastructure. The CloudSim toolkit [27] is advantageous for the developers to implement their own policies in the domain of cloud by extending relevant classes of CloudSimtoolkit. Clouds contain the virtualization layer that integratesexecution, management and hosting environment for application services which is unmanageablein grids. The several modules of CloudSimtoolkit have been mentioned in Figure 1 which is relevant to this research.

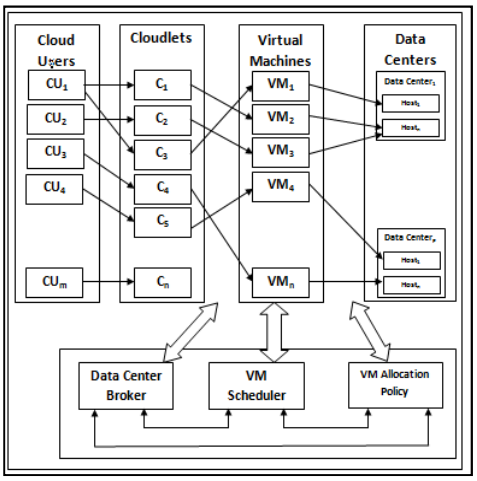

Figure 1. Cloudsim work style [18]

2.1 Cloud Information Service (CIS)

CIS [24] maps user requests to suitable cloud providers. The CIS and DCB of CloudSimgovernor resource discovery and information interaction. It is the core of simulated scheduling [2, 29].

2.2 Datacenter Broker (DCB)

This class contains a broker, which is responsible for interacting between cloud users and CSP depending on users' bandwidth and QoS necessities. The broker deploys cloudlets into the hosts. User-developed scheduling algorithms and/or techniques are designed and incorporated in the DCB module.

2.3 VM Scheduler

VM scheduler is an abstract class inherited by a Host constituent that represents the scheduling policies such as space-shared, time-shared etc. which are required for allocating processing power to the VMs for better performance.

2.4 VM Allocation Policy

VM Allocation Policy is used to select an obtainable host in a Datacenter, which encounters the capacity, storage, and bandwidth availability requirement for the disposition of a VM. The DCB allocates the cloudlets to the VM as per its requirements.

3. Research Method

The performance of the proposed MCAP policy has been compared with two other existing allocation policies- a) Round Robin Allocation (RRA) [7] Policy and b) Conductance Algorithm (CA) [18]. The major disadvantages of these two are larger makespan, larger waiting time, higher turn-around time of the VMs as well as the host which is present in the Datacenter. The proposed MCAP provides better results than the existing policies which are discussed below over the above-mentioned parameter

3.1. The Round Robin Allocation (RRA) Policy [7]

Round Robin Allocation (RRA) Policy [30-32] allocates the cloudlet to first available VM. For example, consider there are fivecloudlets (C_0, C_1, C_2, C_3, and C_4) and three VMs (VM_0, VM_1, and VM_2) present in the system. Table 1 illustrates the allocation fashion. According to this policy, cloudlet C_0 allocated to VM_0, C_1 allocated to VM_1, C_2allocated to VM_2, C_3 allocated to the VM_0 and C_4 a llocated to VM_1.

Table 1. Cloudlet binding with VM

Cloudlet	Virtual machine (VM)
C_0	VM_0
C_1	VM_1
C_2	VM_2
C_3	VM_0
C_4	VM_1

3.2. Conductance algorithm (CA)

It [18] considers each VM as a pipe. It calculates the *Conductance* (processing power) as per (1) of each VM as the ratio of its capacity to the sum of the capacity of all the VMs present in a System.

$$Conductance_j = MIPS_j / \sum_{j=1}^{n} MIPS_j \quad (1)$$

After the calculation of conductance, multiply the conductance of that particular VM with the length of the cloudlet list. To determine the strip length, (2) is used. It determines the number of cloudlets the VM can process.

$$Striplength_j = conductance\ j \times (Length\ of\ the\ cloudlet\ list) \quad (2)$$

The existing policies do not consider the procedure for finding out the minimum makespan of the VM(s) as well as the host(s) in DC.

4. Proposed MCAP Algorithm

We shall make the following modification to DCB in order to improve its performance. At first, VMs are allocated based on Remaining Load Capacity (RLC) inside a Datacenter using their processing capability characteristic i.e. Millions of Instructions per Second (MIPS).

$$Present\ Load = (Total\ MI\ of\ the\ Cloudlets\ waiting\ in\ the\ local\ queue\ of\ the\ VM / Capacity\ of\ the VM) \quad (3)$$

In the second step, the cloudlets are migrated from a VM to another VM following the proposed MCAP discussed below in order to improve the overall performance of the system.

4.1. Cloudlet Allocation Strategies

The DCB module is upgraded with this newly designed allocation policy (MCAP) and allocates cloudlet(s) to a suitable VM. The proposed allocation policy (MCAP) considers the Maximum allowable Load Capacity (LC_{max}) index and the Remaining Load Capacity (RLC) index for each VM. The maximum permissible amount of load that can be allocated to a VM determines the maximum allowable load capacity (LC_{max}) index of a VM. The MCAP allocation policy highlights Remaining Load Capacity (RLC) index. This parameter can be defined as the remaining amount of allocable load to a VM. This parameter defines how many cloudlets can be allocated to that VM so that it performs within its optimal capacity. The RLC value is stored in a table, i.e., RLC index table which is dynamic in nature. The MCAP policy estimates the LC_{max} of every VM and initially the RLC value is same as LC_{max} for every VM. The VMs are arranged hierarchically following the RLC value in decreasing order and allocates cloudlet to a VM with maximum RLC (VM_{max}) value.

4.2. Working principle with demonstration

In MCAP algorithm, an example is cited in Figure (2-5) to illustrate the policy simply within this short study. In Figure 2, at a certain instance, two VMs ($VM_0$and VM_1) are there and three cloudlets (C_0, C_2, C_3) have been allotted to VM_0 and a cloudlet C_1 is allotted to VM_1 based on MCAP algorithm. After allocation of the cloudlets the RLC values of those VMs are 50 and 70 respectively. There are four cloudlets (C_4, C_5, C_6, and C_7) in the GQ, waiting for allocation. The sizes of those cloudlets (C_4, C_5, C_6, and C_7) are assumed 10,30,35,32 respectively. The MCAP policy allocates a cloudlet to a VM with maximum RLC (VM_{max}) among all the VMs. Here, the VM_1 will be appropriately chosen which is mentioned in Figure 3.

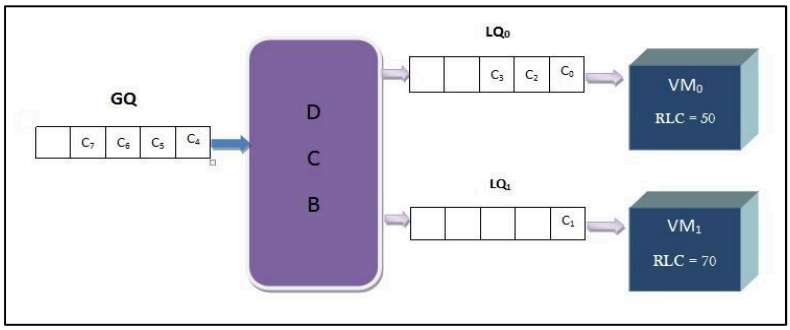

Figure 2. Initial situation- before allocation of cloudlets C_4, C_5, C_6 and C_7

After allocation of the cloudlets, the RLC of VM_1 will be decreased and the RLC index table will be modified according to the updated RLC value. The Figure 3 shows, the DCB despatches C_4 to the VM_1guided by the MCAP policy. Another parameter, known as cloudlet count (cld_count), will represent the number of cloudlets present in the LQ of a VM.

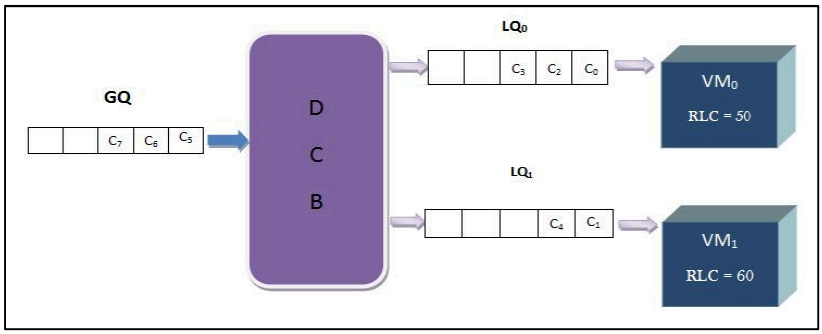

Figure 3. After allocation of cloudlet C_4

$$RLC = ((LC_{max} - \text{Present load of the VM})/LC_{max} \text{of VM}) * 100\% \quad (4)$$

All the logical estimations regardingcloudlet allocation is done in the DCB as prior operation. The DCB finalizes the cloudlet to VM mapping for the arrived cloudlet batch. If the total length of all the cloudlets is equivalent to LC_{max}, the RLC of that VM becomes zero percent. If the allocation of a cloudlet to a VM makes the VM exhausted,i.e.the RLC of that VM becomes negative andthe DCB decides to reallocate or migrate a cloudlet to the other VM. These can be realized through an example stated below.

The VM with negative RLC is termed as victim VM. The DCB selects a cloudlet with minimum length from the LQ of that victim VM for migration. The scenario is explained in the Figure 4.

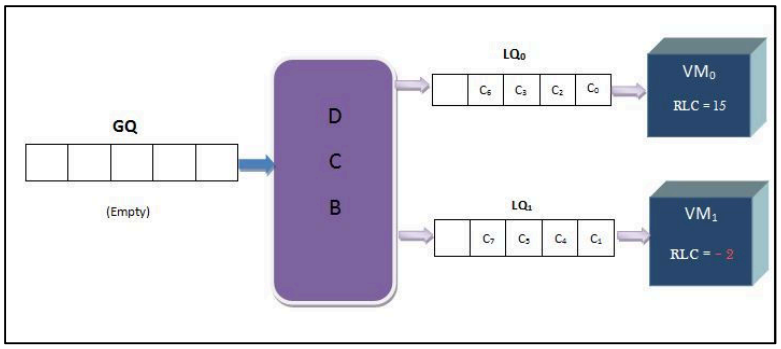

Figure 4. After allocation of cloudlets C_4, C_5, C_6 and C_7 to the VM

Here, in Figure 4, the VM_1 is termed as victim VM. The DCB searches for a cloudlet with minimum length from the LQ of VM_1 and finds C_4. So, the DCB migrate C_4 to VM_0which is mentioned in Figure 5. As the RLC of VM_0 is 15 and the length of C_4 is 10. After reallocating C_4 to VM_0 the RLC of VM_0 becomes 5 and the RLC index table is updated according to it. This is described in Figure 5.

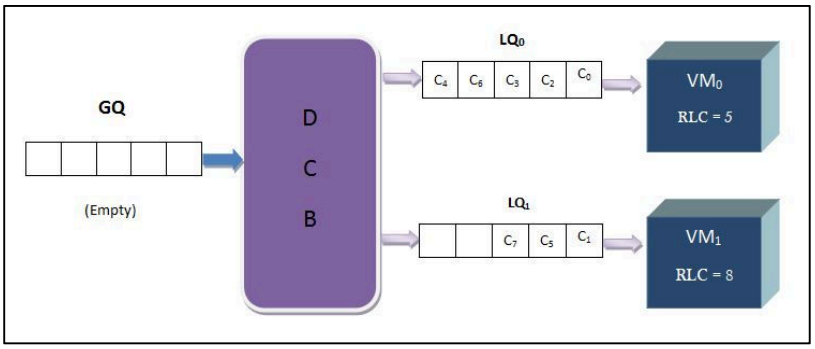

Figure 5. After migration of C_4 from the LQ_1 of VM_1 to the LQ_0 of VM_0

The proposed MCAP algorithm under discussion is equipped with cloudlet migration. Here the cloudlets are migrated to make more balanced cloud environment that ultimately gives better QoS. Based on the RLC value of the VMs the cloudlets have been allocated and a mechanism of cloudlet migration for balancing loads [25] on each VM is deployed. The VM with maximum RLC is denoted as VM_{max}.If more than one such VM_{max} exists then the VM which has the lowest number of cloudlets in its local queue (LQ) is selected. The VM with minimum RLC is denoted as VM_{min} and if more than one such VM exists, then the VM with the highest number of cloudlets in its LQ is selected among VM_{min}. The cloudlet with minimum length has the maximum probability to get accommodated in a VM. If the capacity of VM_{min} exceeds LC_{max} (RLC of VM_{min} becomes negative) then the cloudlet with minimum length (cld_{min}) is transferred from VM_{min} to VM_{max}. This mapping happens logically in the DCB. Based on which the DCB allocates or migrates a cloudlet to a VM. Thus, the RLC values of VMs stay balanced. The entities like

VM_{max}, VM_{min} and cld_count can be calculated from the VM descriptor block in Table 2. To avoid infinite migration, a parameter known as swapcount is used which is mentioned in Table 3. Here threshold limit is considered as 16 [30].

The proposed MCAP allocation policy needs two different data structures to track arriving cloudlet and status of the VMs. These are as follows:

4.2.1.VM descriptor block

VM descriptor block is such a data structure where all the information about a VM is stored in the datacenter broker.

This data structure containsVM_id, cld_count, RLC, utilization as an attribute.

Table 2. VM descriptor block

VM_id	cld_count	RLC	utilization

VM_id: - It is a unique identifier of a VM
cld_count:- It specifies the number of cloudlets allocated to this VM.
RLC: - It specifies the load capacity remaining in a VM after allocation of cloudlet(s).
Utilization: - It specifies how much of the VM is utilized.

4.2.2.Cloudlet descriptor block

This data structure contains cld_id, corresponding VM ID, cloudlet length as its attributes.

Table 3. Cloudlet descriptor block

cld_id	VM_id	Length	swapcount

cld_id:- It is a unique identifier of a cloudlet.
VM_id:- It specifies the VM which is allocated for a particular cloudlet.
length: - It specifies the length of a cloudlet.
swapcount: - It specifies the number of migrations of a cloudlet from one VM to another VM.

4.3. Algorithm: Algorithm of the MCAP algorithm

The algorithm of the MCAP work is described here.

Input:

a) Enter the number of VMs
b) Enter the number of cloudlets

Algorithm:

InMCAP algorithm, the cloudlets are distributed among all the VMs specified by users based on remaining load capacity and migration.

1. Start.
2. Enter the number of VM and cloudlets
3. For all cloudlets i
    a. Allocate cloudlets i to the VM_{Max}.
    b. Set the parameter of VM descriptor block (Table 2) and cloudlet descriptor block (Table 3).
4. While VM_{Min} among VMs is negative
    a. if swapcount of $cld_{min}$of VM_{min} is less than 16 using cloudlet descriptor block.
        i. Migrate cld_{min}fromVM_{Min} to VM_{Max}using VM descriptorblock
    ii.Setthe parameter of VM descriptor block (Table 2) and cloudlet descriptor block (Table 3).
    b. else
        i. Terminate process showing overloaded message.
5. Bind Cloudlets to VM.
6. Stop.

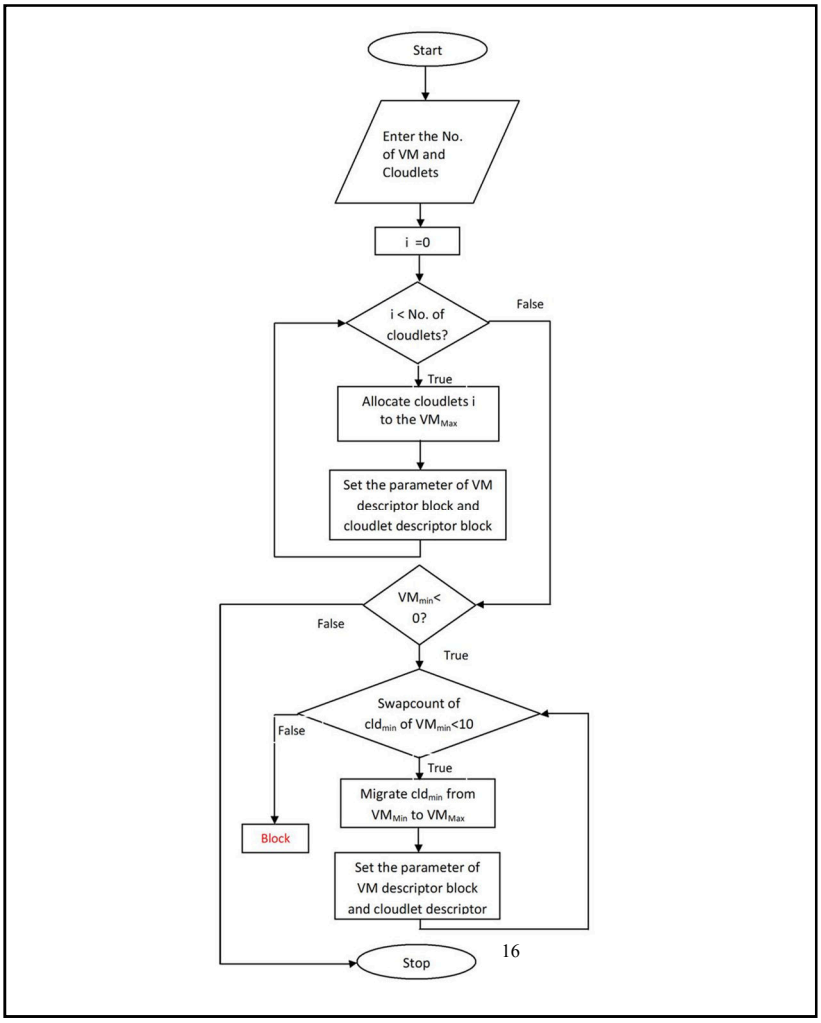

Figure 6. Flowchart:Flowchart of MCAP algorithm

5. Result & Analysis

MCAP algorithm is implemented in CloudSim 3.0.3 and compared its results with existing algorithm like Round Robin. We have taken some specific no. of cloudlets having predefined length running in certain VM for our algorithm as well as for Round Robin. The complete dataset is shown in Table 4.

Table 4. Properties of cloudlets

cld_id	length (MI)
0	1100
1	500
2	200
3	2500
4	1550
5	900
6	2000
7	800
8	650
9	2200

Table 5. Processing capability of VM

VM_id	MIPS
0	10
1	10
2	10

5.1. Comparison between RRA andMCAP Algorithm

Table 6. Comparison based on turnaround time

cld_id	RRA	MCAP
0	110.1	110.1
1	50.1	50.1
2	20.1	70.1
3	360.1	320.1
4	205.1	265.1
5	110.1	90.1
6	560.1	520.1
7	285.1	170.1
8	175.1	235.1
9	780.1	485.1

Table 7. Comparison based on average turnaround time

Number of cloudlets	RRA	MCAP
10	265.6	231.6

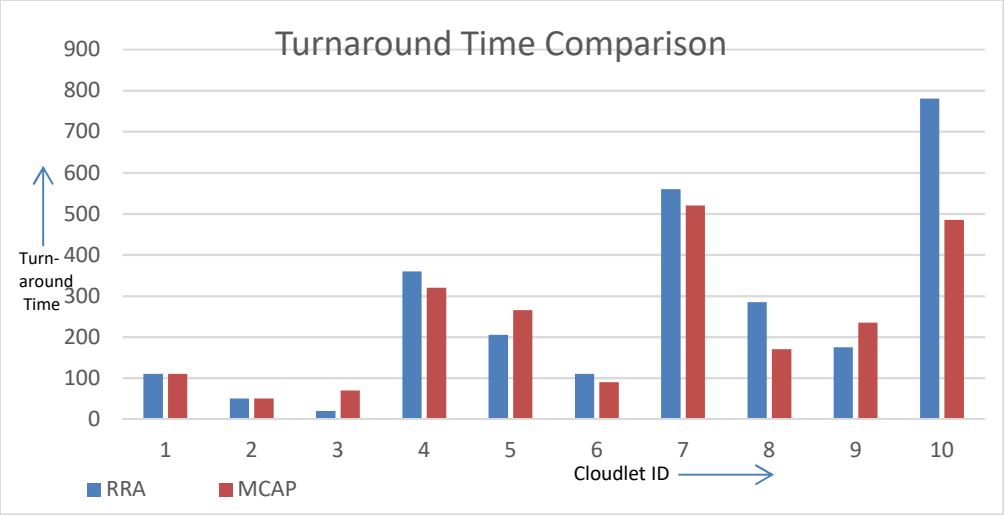

Figure 7. Comparison between RRA and MCAPbased on turnaround time

Table 8. Comparison based on waiting time

cld_id	RRA	MCAP
0	0.2	0.2
1	0.2	0.2
2	0.2	50.1
3	110.1	70.1
4	50.1	110.1
5	20.1	0.2
6	360.1	320.1
7	205.1	90.1
8	110.1	170.1
9	560.1	265.1

Table 9. Comparison based on average waiting time

Number of cloudlets	RRA	MCAP
10	141.63	107.63

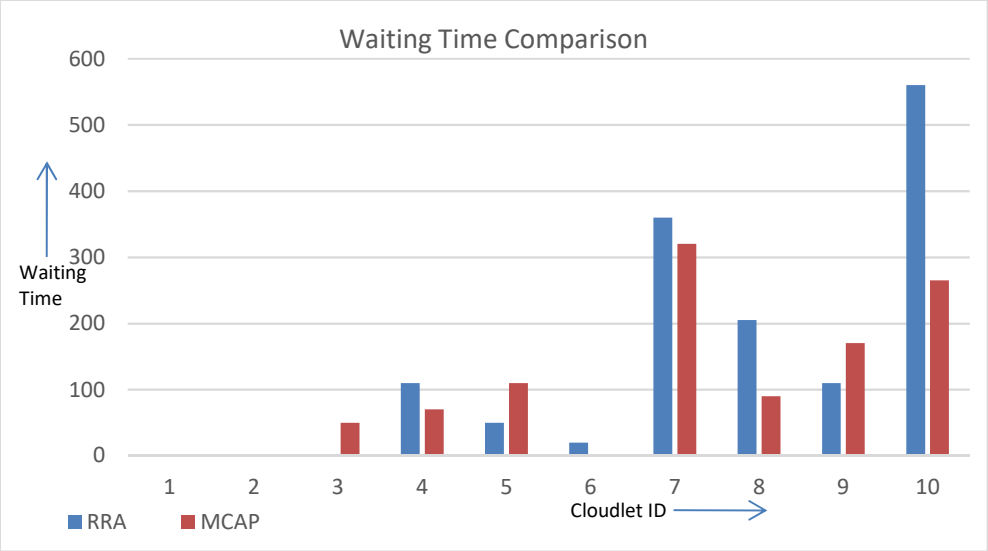

Figure 8. Comparison between RRA and MCAP based on waiting time

Table 8. Comparison based on Makespan

Algorithm	Makespan
RRA	780.1
MCAP	520.1

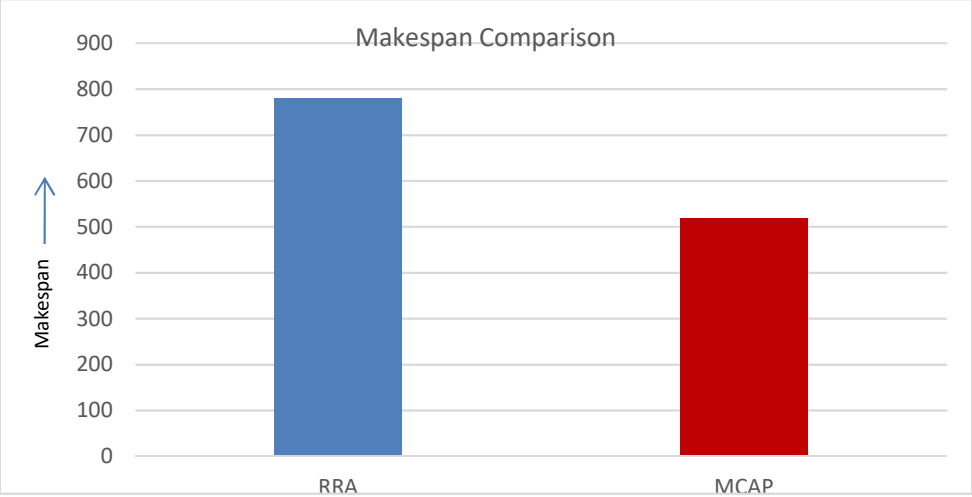

Figure 9. Comparison between RRAand MCAPbased on makespan

Table 11. Performance analysis

	RRA	MCAP	Improvement (%)
Avg. Turn-around Time	265.6	231.6	12.8
Avg.WaitingTime	141.63	107.63	24.01
Makespan	780.1	520.1	33.33

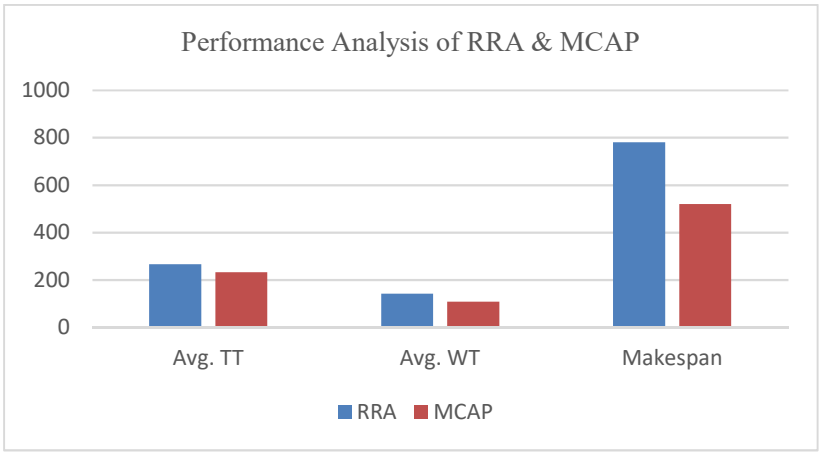

Figure 10. Comparison between RRA and MCAP based on Avg. Turn-around time, Avg. Waiting Time and Makespan

5.2. Comparison between CA and MCAP Algorithm

Table 12. Comparison based on turnaround

cld_id	CA	MCAP
0	200.1	110.1
1	50.1	50.1
2	20.1	70.1
3	690.1	320.1
4	355.1	265.1
5	90.1	90.1
6	220.1	520.1
7	195.1	170.1
8	115.1	235.1
9	440.1	485.1

Table 13. Comparison based on average Turnaround time

Number of cloudlets	CA	MCAP
10	237.6	231.6

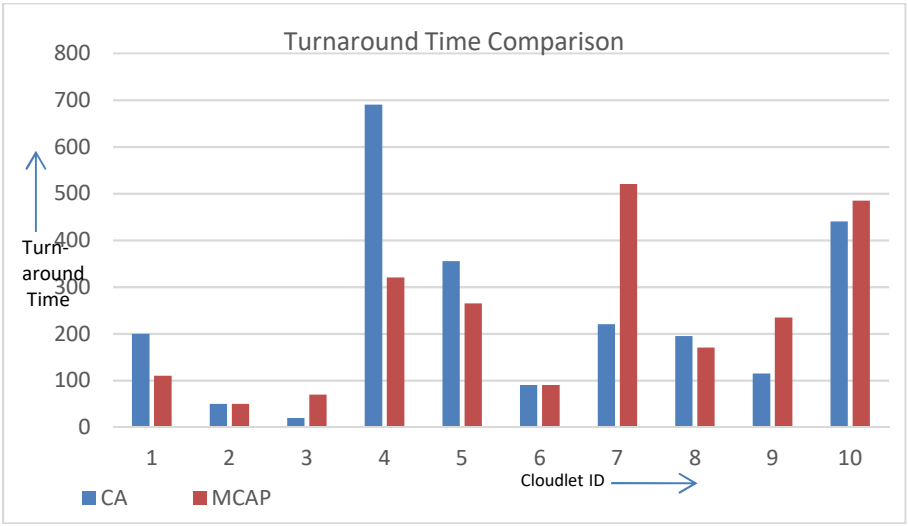

Figure 11. Comparison between CA and MCAP based on turnaround time

Table 14. Comparison based on waiting time

cld_id	CA	MCAP
0	90.1	0.2
1	0.2	0.2
2	0.2	50.1
3	440.1	70.1
4	200.1	110.1
5	0.2	0.2
6	20.1	320.1
7	115.1	90.1
8	50.1	170.1
9	220.1	265.1

Table 15. Comparison based on average waiting time

Number of cloudlets	CA	MCAP
10	113.63	107.63

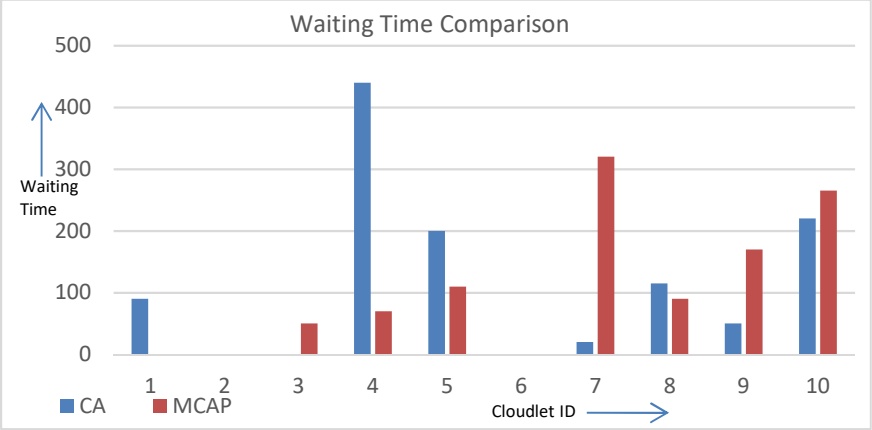

Figure 12. Comparison between CA and MCAP based on waiting time

Table 16. Comparison based on Makespan

Algorithm	Makespan
CA	690.1
MCAP	520.1

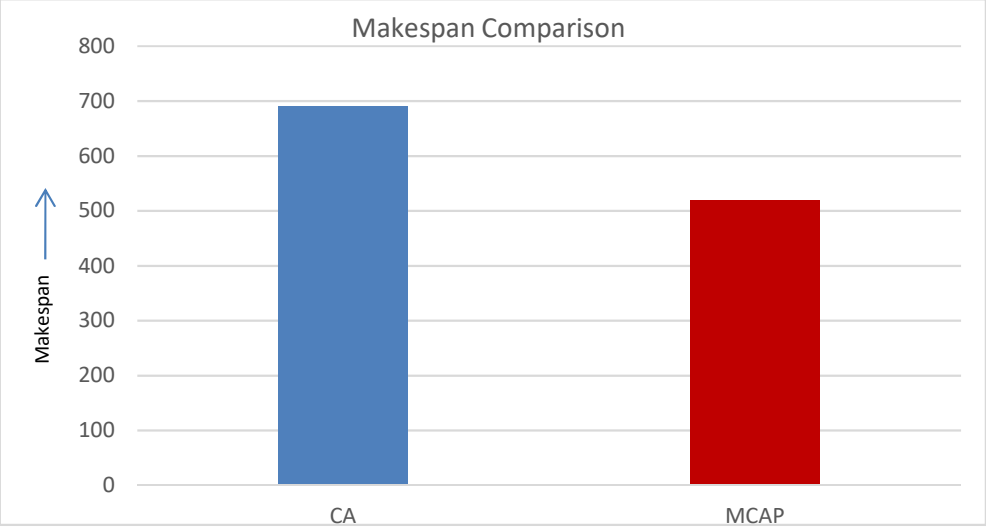

Figure 13. Comparison between CA and MCAP based on makespan

Table 17. Performance analysis

	CA	MCAP	Improvement (%)
Avg.Turn-around Time	237.6	231.6	2.52
Avg.WaitingTime	113.63	107.63	5.28
Makespan	690.1	520.1	24.63

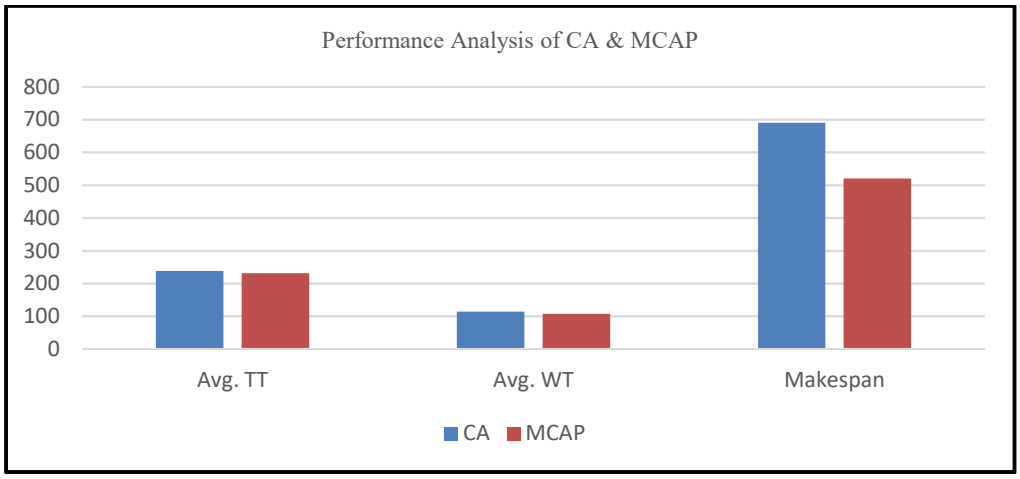

Figure 14. Comparison between CA and MCAP based on Avg. Turn-around time, Avg. Waiting Time and Makespan

6. Conclusion

The objective of the load balancing is to assign the cloudlets effectively to the computing resources optimally. The migration enabled cloudlet allocation is highly demanding nowadays due to its adaptive nature. Depending upon remaining load capacity the cloudlets are allocated. This is a challenging task in a large high-performance computing environment like cloud. The MCAP algorithm takes into account the capacity of VMs unlike most of the allocation strategies. The obtained result sets and rate of improvement in performance are the evidence of the efficacy and sustainability of the proposed MCAP algorithm.

References

[1]. Sosinsky B. Cloud Computing Bible, 1st ed., Indianapolis, Indiana: Wiley Publishing, Inc. 2011: ch.1,4,5: 1-18, 58-68, 73-77.
[2]. Calheiros R N, Ranjan R, Cesar A, Rose F D, Buyya R. CloudSim: A Novel Framework for modelling and Simulation of Cloud Computing Infrastructures and Services. 2009.
[3]. Pallis G. Cloud Computing: The New Frontier of Internet Computing. Internet Computing. IEEE. 2010: 14(5), 70-73. doi: 10.1109/MIC.2010.113.
[4]. Zhang Q, Cheng L, Boutaba R. Cloud Computing: State-Of-The-Art and Research Challenges. *J Internet Serv Appl. The Brazilian Computer Society*. 2010. 7-18. doi: 10.1007/s13174-010-0007-6.
[5]. Dikaiakos M D, Pallis G, Katsa D, Mehra P, Vakali A. Cloud Computing: Distributed Internet Computing for IT and Scientific Research. Internet Computing, IEEE. 2009: 13(5):10-13. doi: 10.1109/MIC.2009.103.
[6]. Sahoo J, Mohapatra S, Lath R.Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues.Computer and Network Technology (ICCNT). Second International Conference on. 2010: 222-226, 23-25. doi: 10.1109/ICCNT.2010.49.
[7]. Parsa S, Entezari-Maleki R. RASA: A New Task Scheduling Algorithm in Grid Environment. *World Applied Sciences Journal 7 (Special Issue of Computer & IT)*. IDOSI Publications. 2009: 152-160. ISSN: 1818.4952.
[8]. Alwabel A, Walters R, Wills G. Desktop CloudSim: Simulation of Node Failures in the cloud. in Proceedings of The Sixth International Conference on Cloud Computing, GRIDs, and Virtualization. IARIA. 2015: 14-19. ISBN: 978-1-61208-388-9.
[9]. Buyya R, Calheiros R N, Grozev N. cloudsim 3.0 API.Cloudbus.org. 2015. [Online]. Available at: http://www.cloudbus.org/cloudsim/doc/api/.
[10]. Buyya R, Calheiros R N, Grozev N. The CLOUDS Lab: Flagship Projects-Gridbus and Cloudbus. Cloudbus.org. 2015. [Online]. Available at: http://www.cloudbus.org/middleware/.
[11]. Panchal B, Prof Kapoor R K. Dynamic VM Allocation using Clustering in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2013. 3: 143-150.

[12]. Casanova H. Simgrid: A toolkit for the simulation of application scheduling, Cluster Computing and the Grid. Proceedings. First IEEE/ACM International Symposium on. 2001:430-437.doi: 10.1109/CCGRID.2001.923223.
[13]. Ostermann S, Plankensteiner K, Prodan R, Fahringer T. GroudSim: An event-based simulation framework for computational grids and clouds. Euro-Par 2010 Parallel Processing.Workshops. No. 261585. 2011: 305-313.
[14]. Kaur P, Prof Dr. Kaur P D. Efficient and Enhanced Load Balancing Algorithms in Cloud Computing. *International Journal of Grid Distribution Computing*. SERSC. 2015. 8(2): 9-14.ISSN: 2005-4262 IJGDC.
[15]. Marc B, Leser U. Dynamiccloudsim: Simulating Heterogeneity in Computational Clouds. *Future Generation Computer Systems*. 2015: 1-22.
[16]. Randles M, Lamb D, Taleb-Bendiab A. A comparative study into distributed loadbalancing algorithms for cloud computing. 2010 IEEE 24thinternational conference on advancedinformationnetworking and application workshops. 2010:551-556, 20-23. doi: 10.1109/WAINA. 2010.85.
[17]. Somani R, Ojha J. A Hybrid Approach for VM Load Balancing in Cloud Using CloudSim. *International Journal of Science. Engineering and Technology Research (IJSETR)*.2014. Vol.3(6): 1734-1739. ISSN: 2278-7798.
[18]. Chatterjee T, Ojha V K, Adhikari M, Banerjee S, Biswas U. (2014) VáclavSnášel, Design and Implementation of an Improved Datacenter Broker Policy to Improve the QoS of a Cloud. in Proceedings of the Fifth Intern. Conf. on Innov. In Bio-Inspired Comput. AndAppl, Springer International Publishing Switzerland. IBICA 2014. 2014: 281-290. doi: 10.1007/978-3-319-08156-4_28.
[19]. Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency Computat. Pract. Exper. John Wiley & Sons Ltd. 2011. 24(13): 1397-1420.doi: 10.1002/cpe.1867.
[20]. Chaisiri S, Lee B, Niyato D. Optimization of Resource Provisioning Cost in Cloud Computing. IEEE Trans. Serv. Comput. IEEE. 2012; 5(2):164-177. doi: 10.1109/TSC.2011.7.
[21]. Feng G, Garg S, Buyya R, Li W. Revenue Maximization Using Adaptive Resource Provisioning in Cloud Computing Environments. 2012 ACM/IEEE 13th International Conference on Grid Computing. IEEE. 2012: 192-200.doi: 10.1109/Grid. 2012.16.
[22]. Javadi B, Thulasiraman P, Buyya R. Cloud Resource Provisioning to Extend the Capacity of Local Resources in the Presence of Failures. IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems. IEEE. 2012: 311-319.doi: 10.1109/HPCC.2012.49.
[23]. Yu H, Lan Y, Zhang X, Liu Z, Yin C, Li L. Job Scheduling Algorithm in Cloud Environment. International Conference on Computational and Information Sciences. IEEE. 2013: 1652-1655. doi: 10.1109/ICCIS.2013.432.
[24]. Calheiros R N, Ranjan R, Beloglazov A, Rose C D, Buyya R. (2010) CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience. John Wiley & Sons Ltd. 2010. 41(1): 23-50. doi: 10.1002/spe.995.
[25]. Li K, Xu G, Zhao G, Dong Y, Wang D. Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization. Sixth Annual Chinagrid Conference. IEEE. 2011: 3-9.doi: 10.1109/ChinaGrid.2011.17.
[26]. Ru J, Keung J. (2013) An Empirical Investigation on the Simulation of Priority and Shortest-Cloudlet-First Scheduling for Cloud-Based Software Systems. 2013 22nd Australian Software Engineering Conference. IEEE: 78-87, doi: 10.1109/ASWEC.2013.19.
[27]. Wickremasinghe B, Calheiros R N, Buyya R. CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications. Advanced Information Networking and Applications (AINA). 2010 24th IEEE International Conference on. 2010: 446-452, 20-23. doi: 10.1109/AINA.2010.32.
[28]. Katyal M, Mishra A. Application of Selective Algorithm for Effective Resource Provisioning in Cloud Computing Environment. *IJCCSA*. 2014. 4(1): 1-10. doi: 10.5121/ijccsa.2014.4101.
[29]. Buyya R, Ranjan R, Calheiros R N. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities. In: Proceedings of the 7th High Performance Computing and Simulation Conference. (HPCS 2009. ISBN: 978-1-4244-4907-1. IEEE Press. New York, USA). Leipzig, Germany. 2009.
[30]. Forouzan B A. Data Communications and Networking. 3rd Ed. New Delhi: Tata McGraw-Hill. 2004: 497-560.
[31]. Martino B D, Esposito A, Nacchia S.Comput Sci Res Dev A semantic model for business process patterns to support cloud deployment. Springer Berlin Heidelberg. 2017. 32(3–4): 257–267. https://doi.org/10.1007/s00450-016-0333-4.
[32]. Oberg M, Woitaszek M, Voran T, Tufo H M. A system architecture supporting high-performance and cloud computing in an academic consortium environment. Computer Science - Research and Development. Springer-Verlag. 2011; 26(3-4): 317-324, doi:10.1007/s00450-011-0172-2.