ISSN 0146-4116, Automatic Control and Computer Sciences, 2018, Vol. 52, No. 3, pp. 208–219. © Allerton Press, Inc., 2018.

An Approach Towards Development of a New Cloudlet Allocation Policy with Dynamic Time Quantum[1]

Sourav Banerjee[a, *], Akash Chowdhury[b], Swastik Mukherjee[b], and Utpal Biswas[c]

[a] *Kalyani Government Engineering College, Kalyani, Nadia, 741235 India*

[b] *Institute of Science and Technology, Paschim Medinipur, 721201 India*

[c] *University of Kalyani, Kalyani, Nadia, 741235 India*

**e-mail: mr.sourav.banerjee@ieee.org*

Received April 26, 2017; in final form, January 24, 2018

Abstract—Cloud computing is one of the most emerging technologies which has created a revolution in the High performance Computing (HPC) domain. The term Quality of Service (QoS) plays a vital role in the formation of more flexible integration of various technologies. The Waiting Time (WT), Turnaround Time (TAT), Context Switching (CS) and Makespan (MS) are the primary parameter that has great impact on the scheduling of cloudlets. The Proposed algorithm has improved the resource utilization system of the existing Round Robin Algorithm (RRA) and Improved Round Robin Cloudlet Scheduling Algorithm (IRRCSA) by introducing the concept of dynamically calculated Time Quantum (TQ) for each virtual machine (VM) according to the allocated cloudlets. This new approach in cloudlet scheduling drastically reduced average WT, average TAT and Number of CS of the VMs, which further enhanced the capability of cloud service providers (CSPs) to provide better QoS.

Keywords: cloud computing, QoS, cloudlet, cloudlet scheduling, OTQ, RET

DOI: 10.3103/S0146411618030033

1. INTRODUCTION

Cloud computing, being one of the most emerging distributed technologies, needs constant progress in its architectural or computational infrastructure to revolutionize the idea more flexibly. The rapid growth of the technology depends totally on the continuous improvements in terms of cost efficiency and QoS. The high level of abstraction in cloud computing has given cloud users (CUs) the flexibility and the modularity to the extent of efficient on demand service. The virtualization of resources and wide availability of services has made cloud computing popular and unique, thus dragging the attention of the researchers to enhance its ability in various ways. Thus, major attention of the improvement lies in the domain of cloudlet scheduling which can bring drastic changes in terms of flexibility and cost efficiency. Cloudlet scheduling and resource allocation are the primary concerns to extend the capability of the cloud computing. Thus the Proposed algorithm is based on the optimal cloudlet scheduling policy integrating the IRRCSA [14] model. The common data sets emphasize the rate of improvement more efficiently.

1.1. Our Contribution

Several researches are in the process for shaping cloud computing in more efficient and advanced form. While several researchers are working on the security and integrity issues, many are focusing on the infrastructural improvement and Service Level Agreement (SLA) [1]. The Proposed scheduling algorithm focuses on the enhancement in QoS [2], improving the total system throughput by decreasing the average WT, average TAT and number of CS remarkably, by calculating and assigning a specific time quantum termed as Optimal Time Quantum (OTQ) [3] for the deployed VMs. As cloud computing focuses on arranging resources and processing the cloudlets more quickly based on demand, it is necessary to improve cloudlet-scheduling policies with more flexibility towards dynamicity.

[1] The article is published in the original.

1.2. Organization

The rest of the paper is organized as follows—Section 2 describes the different relevant works done on this area and the motivation behind this research. Section 3 concentrates on the main part of the research that depicts evaluation of Proposed algorithm with formulations, algorithm and flowchart. Experimental results are tabulated in the Section 4. Comparison and Analysis section is depicted in Section 5, which comprises of two sub sections – Tabular analysis and Graphical analysis. Future research directions and conclusion are discussed in Section 6 and 7 respectively.

2. RELATED WORK

Several researchers are working in the cloud computing [4, 31] domain all over the world for making any possible improvement in system functionality and efficiency of VMs running in the hosts of cloud data centers. The developed cloudlet scheduling algorithms [5, 21] are simulated in frameworks like renowned CloudSim 3.0.3. GridSim Toolkit [6] has the capability to give users provision of modeling and simulating differently configured Grid resources [7] and networks in order to examine newly developed algorithms and working techniques with having full control of the simulated environment. But it lacks support for cloud computing [8] based infrastructure and application requirements and does not provide real-time tradeoffs among CUs and CSPs. Some known grid simulators are GridSim, GangSim [9] and SimGrid [10].

2.1. CloudSim Framework

CloudSim is an open source framework that offers simulation environment for resource management and application scheduling cloudlets [11, 12]. CloudSim 3.0.3 is developed on the famous CloudSim [13] framework. It comprises classes for simulating cloud infrastructure that describes cloud elements like data centers, VMs [32], processing units, applications and various provisioning and scheduling management policies. CloudSim 3.0.3 is widely used for testing new cloudlet scheduling algorithms.

CloudSim modules that are used in the research are listed below [14].

(1) VM Scheduler – it is an abstract class that any host component implements to schedule VMs to Processing Elements (PE) according to the stated policies (space shared, time shared) [15, 16].

(2) Cloudlet Scheduler – it is an abstract class that schedules suitable cloudlet from the local queue (LQ) of the VM obeying certain stated scheduling policies for execution of the cloudlets.

(3) VM Allocation Policy – it is an abstract class that comprises the policies required for allocation of VMs to suitable hosts in datacenters.

(4) Datacenter Broker or DCB – it is a class having encapsulated broker properties, capable of taking care of user requirements and handling services between service providers and users. For efficient cloudlet execution new improved cloudlet scheduling and allocation algorithms are implemented in DCB method [22].

2.2. Scheduling Algorithms

RRA – in this algorithm [18, 19] first there is a fixed amount of time taken named as the time quantum (TQ). Each cloudlet in the LQ of the VMs gets executed for one TQ. After execution that cloudlet is preempted and sent back to the tail of the LQ. At this moment, the next cloudlet in the LQ is given the chance for execution for one TQ, which after execution is sent back to the tail of the LQ of the VM as done in the case of previous cloudlet. This process goes on until all the cloudlets in the LQ have finished executing for their respective expected execution time [20] (ET). Let's assume there are three cloudlets C_0, C_1, C_2 in the LQ with ETs as 25, 10, 12 respectively. Then according to the RRA the Gantt chart is given below.

C_0	C_1	C_2	C_0	C_1	C_2	C_0	C_2	C_0	C_0

0 5 10 15 20 25 30 35 37 42 47

The main drawback of this algorithm is high average WT, average TAT and number of CS. Suppose the remaining execution time of a cloudlet is 1 s. Even if the remaining execution time of that cloudlet is 1 s, it has to wait for an entire round in which all other cloudlets will execute for 1 TQ each, after which that cloudlet will be permitted to execute for its remaining 1 s.

IRRCSA – this algorithm [14] has a fixed TQ and the cloudlets get a chance to get executed for one TQ every time. After execution of any cloudlet for one TQ the Remaining Execution Time (RET) of that cloudlet is calculated and checked whether it is less than or equal to TQ or not. If condition is true then that cloudlet immediately gets chance of execution for another TQ, else it is sent back to the tail of the LQ. This process continues until all the cloudlets in the LQ have completed their execution. The Gantt chart according to the above-assumed cloudlets is depicted below.

C_0	C_1	C_1	C_2	C_0	C_2	C_2	C_0	C_0	C_0

0 5 10 15 20 25 30 32 37 42 47

The main drawback of this algorithm is its fixed TQ. A fixed TQ leads to a very high number of CS and accordingly increases the average WT and average TAT. The TQ of a VM should depend on the ETs of all the cloudlets allocated to it. Every VM will be assigned with different cloudlets. Therefore the TQ should not be fixed for all the VMs, but should be dynamic depending on the cloudlets allocated to the respective VMs.

2.3. Research Motivation

In the Proposed work the drawbacks of the existing scheduling algorithms that are discussed in Section 2.2 have been improved in terms of average WT, average TAT and number of CS.

It has been observed that –

$$\text{Number of CS} \infty (\text{TQ})^{-1}. \tag{1}$$

$$\text{Average Response Time (ART)} \infty \text{TQ}. \tag{2}$$

According to equation (1) the number of CS increases if TQ gets decreased whereas according to equation (2) increase in TQ results in increase in Average Response Time (ART) for cloudlets thus tending the time slicing approach towards First Come First Serve (FCFS) scheduling policy.

So, obtaining a specific TQ that is assigned dynamically to a particular VM and where that obtained TQ results in having the minimum difference between itself and the largest cloudlet ET among all the cloudlets in the LQ of that VM, is recognized as the optimized solution for reducing the number of CS as well as the ART.

The proposed algorithm that makes an approach towards acquiring and assignation of such a TQ termed as OTQ that fulfills the aforementioned requirement, along with all the related concepts are discussed in detail in Section 3.

3. PROPOSED WORK

The proposed work is an additional improvement strategy over the existing IRRCSA [14] using optimal time quantum (OTQ). The use of OTQ in cloudlet scheduling policy has brought a great improvement in terms of increasing the QoS of CSP [28, 29]. The QoS of CSP [30] encompasses various issues like – WT, TAT, [23, 24] CS, MS etc. This work highlights the improvement of WT, TAT and CS, which play as key factor for improving system performance. The lesser the WT, TAT and CS are acquired, the greater the efficiency is achieved. The proposed model is depicted in Fig. 1.

Herein, ET is another factor that leaves important impact on the WT, TAT and CS. ET can be calculated as

$$\text{ET}_i = \text{MI}_i/\text{MIPS}_n, \tag{3}$$

where, ET_i and MI_i are the ET and Million Instruction of the ith cloudlet respectively. MIPS_n are the million instructions per second of the nth VM where the ith cloudlet is allocated.

In the proposed algorithm, the cloudlets in the LQs of each VM gets executed for a specific time span termed as OTQ that is dynamically calculated and assigned for each VM. In a particular VM, after every execution of cloudlets till one OTQ, the total ET of the cloudlets gets decreased by the OTQ assigned to that VM and thus according to equation (4) it produces the RET for those cloudlets.

$$\text{RET}_i = \text{ET}_i - \text{OTQ}_n. \tag{4}$$

Here RET_i is the RET of the ith cloudlet which was currently executed and OTQ_n is OTQ calculated and assigned for the nth VM where the ith cloudlet is getting executed. The ith cloudlet is allowed to continue its execution for another OTQ if the calculated RET_i is less than or equal to the OTQ_n, else it is sent back to the tail of the LQ of the VM and the next cloudlet in the LQ i.e., $(i + 1)$th cloudlet gets the chance for

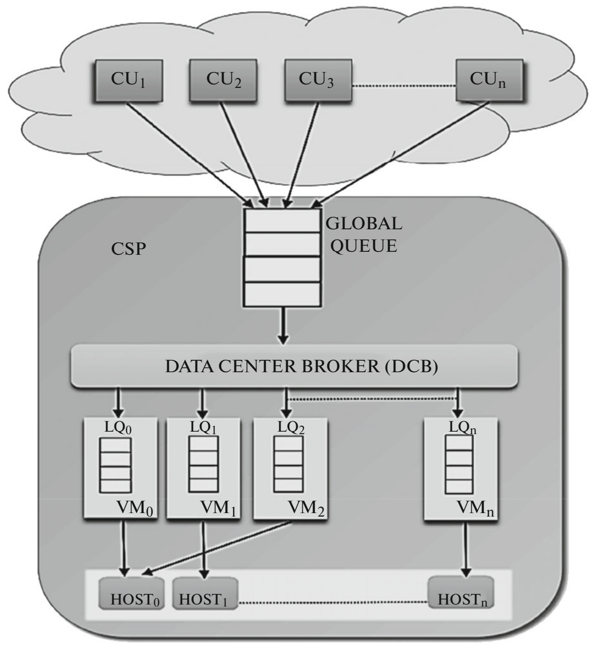

Fig. 1. Proposed model.

execution for one OTQ and after its execution again it's RET i.e., RET_{i+1} is calculated according to equation (4), and compared with OTQ_n to determine its chances for continuation of execution. This process continues until the LQ of the VM gets emptied.

In the Proposed approach, the dynamic acquiring of OTQ is based specifically on the ETs of cloudlets residing on the LQ of each VM. To calculate the OTQ for a particular VM, the Average Execution Time (AET) is deduced from equation (5) according to the ETs of cloudlets residing in the LQ of that VM.

$$\mathrm{AET} = \frac{\sum_{\mathrm{i}=0}^{K} \mathrm{ET\ of\ Cloudlet}}{K}. \tag{5}$$

Here K is the total number of cloudlet residing in LQ of the VM and i is referred to each of that K number of cloudlets residing in the LQ of that VM.

By equation (6) a Median value (M') of the AET and highest ET among the K number of cloudlets of the VM is calculated. The ceiling value of M' is considered as the OTQ as depicted in equation (7).

$$\mathrm{M'} = \frac{(\mathrm{AET} + \mathrm{Highest\ ET\ in\ the\ LQ\ of\ the\ VM})}{2}, \tag{6}$$

$$\mathrm{OTQ} = \lceil \mathrm{M'} \rceil. \tag{7}$$

Let us assume, a particular VM, having three cloudlets (Cl_1, Cl_2, and Cl_3) in the LQ with ETs as 30, 50 and 33 respectively. Then according to equation (5)

$$\mathrm{AET} = \frac{(30 + 50 + 33)}{3} = 37.67.$$

Here Cl_2 as the highest ET in LQ, the Median value (M') is obtained by equation (6) as,

$$\mathrm{M'} = \frac{37.67 + 50}{2} = 43.83.$$

Therefore according to equation (7),

$$\mathrm{OTQ} = 44.$$

After calculating the required OTQ for a particular VM each cloudlet in that VM is allowed to get executed for one OTQ. After execution of a cloudlet for one OTQ the RET for that cloudlet is calculated according to equation (4) and it is checked whether that,

$$\mathrm{RET} \le \mathrm{OTQ}.$$

Table 1. CloudSim compared to Grid simulators [17]

Grid simulators	CloudSim
1. Supports simulations of distributed application on Grid platforms. 2. Cannot differentiate or isolate clearly the multi-layer service abstractions of cloud computing. 3. No support for Virtualization enabled resource modeling and environment for application.	1. A framework for generic simulation of distributed cloud applications involving cloud computing. 2. Capable of isolating differences between IaaS, SaaS and PaaS service abstractions. 3. Supports virtualized infrastructure and modeling of datacenter type environments.

If the condition is true then that cloudlet gets higher priority to be allowed to get executed immediately for another OTQ. In the proposed approach, average WT, average TAT and number of CS has decreased drastically which is a clear indication of increased system throughput and improved QoS.

3.1. Proposed Algorithm

Step 1: START
Step 2: VMs are created and stored in vm_list
Step 3: A batch of cloudlets are stored in the cloudlet_list
Step 4: All the cloudlets are assigned to the VM in FCFS manner
Step 5: for n=1 to N // N is the total number of VMs present in vm_list
    Step 5.1: for i = 1 to K { // where K=Number of cloudlets allocated in n^{th} VM
        $ET_i = MI_i / MIPS_n$ // ET_i=ET of i^{th} cloudlet and MI_i=MI of i^{th} cloudlet
}
    Step 5.2: SetET_{max}= 0
for i = 1 to K {
        If (ET_{max}< ET_i){
ET_{max} = ET_i//where ET_{max} is the maximum ET among the cloudlets allocated in n^{th} VM
        }
}
    Step 5.3: Set AET = 0, Sum_{ET} = 0
    for i=1 to K{
        $Sum_{ET} = Sum_{ET} + ET_i$
}
AET = Sum_{ET} /K
    Step 5.4: Set $OTQ_n = \lceil (ET_{max} + AET)/2 \rceil$ // where OTQ_n = OTQ calculated for n^{th} VM
Step 6: for n=1 to N{
While (LQ_n != null){ // where LQ_n is LQ of n^{th} VM
        for i = 1 to K{
            i^{th} cloudlet gets executed for one OTQ_n
            $RET_i = ET_i - OTQ_n$
            If (RET_i<= OTQ_n){
                i^{th} cloudlet is allowed to execute for another OTQ_n
            }
            Else {
                i^{th} cloudlet context switch and $i+1^{th}$ cloudlet gets chance for execution
            }
}
 }
    }
Step 7: END

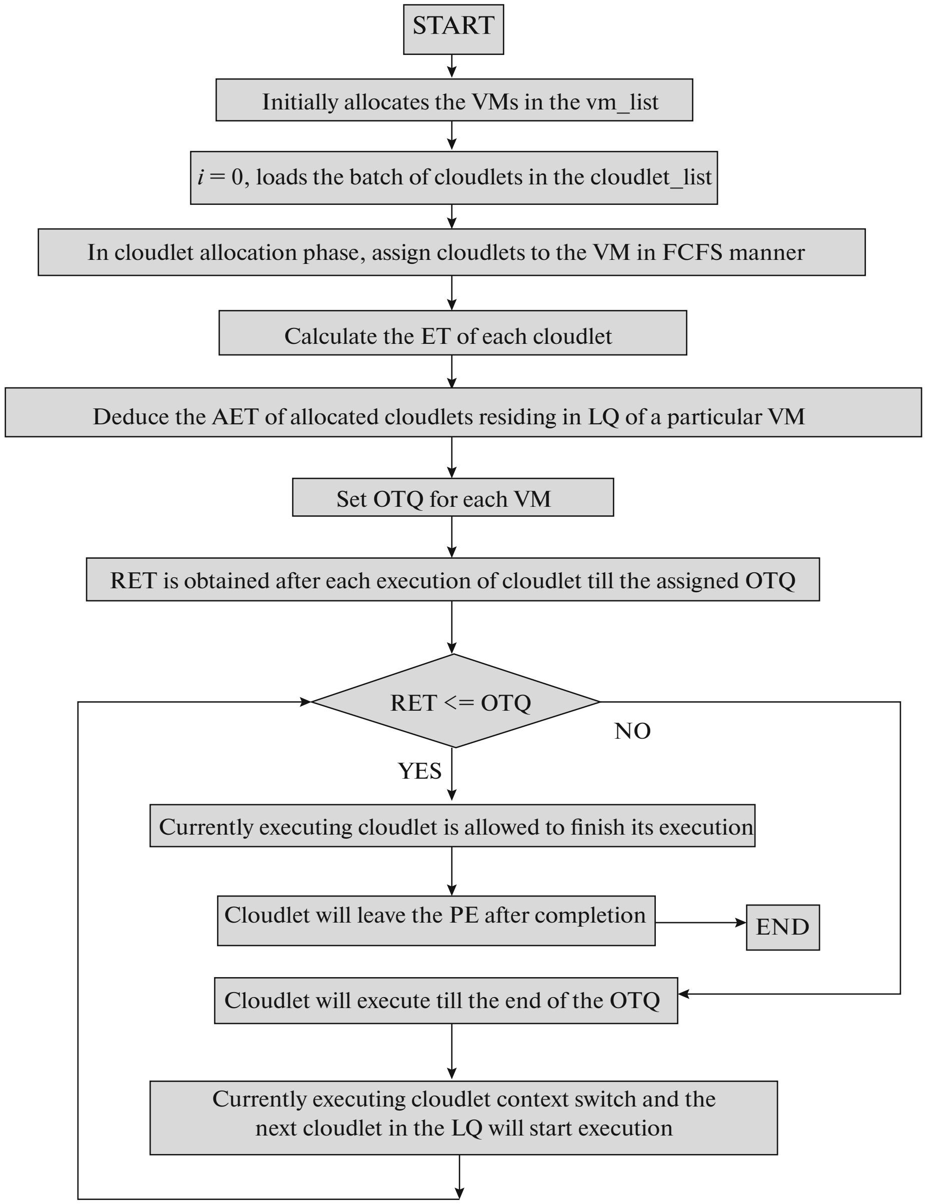

Fig. 2. Flowchart of the proposed work.

3.2. Flowchart

Flowchart of the proposed algorithm is depicted in Fig. 2.

Table 2. A comparative study of existing scheduling policies with the proposed work

	RRA	IRRCSA	PROPOSED
Nature of Allocation	Dynamic	Dynamic	Dynamic
Nature of TQ	Static	Static	Dynamic
Advantages	Simpler to implement	Lower WT, TAT and number of CS than RRA, thus Improved RRA	Lower WT, TAT and number of CS than RRA and IRRCSA, thus achieved better performance than RRA and IRRCSA
Disadvantages	Low static TQ increases number of CS	Low static TQ increases number of CS	Dynamic TQ makes implementation complex

Table 3. Properties of Cloudlets

Cld_id	MI
0	9000
1	16000
2	11000
3	6000
4	15000
5	8000
6	12000
7	17000
8	10000
9	7000

Table 4. Properties of VMs

VM_id	MIPS
0	300
1	600
2	400
3	500

Table 5. Allotted cloudlets to VM0, VM1, VM2, VM3 with their expected execution time (ET)

VM_0			VM_1			VM_2			VM_3		
Cld_id	MI	ET	Cld_id	MI	ET	Cld_id	MI	ET	Cld_id	MI	ET
0	9000	30	1	16000	27	2	11000	28	3	6000	12
4	15000	50	5	8000	13	6	12000	30	7	17000	34
8	10000	33	9	7000	12						

Table 6. Waiting time, turnaround time and number of context switching in VM_0, VM_1, VM_2 and VM_3

VM_0			VM_1			VM_2			VM_3		
Cld_id	WT	TAT	Cld_id	WT	TAT	Cld_id	WT	TAT	Cld_id	WT	TAT
0	0	30	1	0	27	2	0	28	3	0	12
4	30	80	5	27	40	6	28	58	7	12	46
8	80	113	9	40	52						
Average WT		**36.66**	***Average WT***		**22.33**	***Average WT***		**14**	***Average WT***		**6**
Average TAT		**80**	***Average TAT***		**39.66**	***Average TAT***		**43**	***Average TAT***		**29**
Number of CS		**2**	***Number of CS***		**2**	***Number of CS***		**1**	***Number of CS***		**1**

3.3. Comparison of Existing Scheduling Algorithms with Proposed Work

In Tables 1, 2, given below, the proposed work is compared with the existing cloudlet-scheduling algorithm that is mentioned in Section 2.2.

4. EXPERIMENTAL RESULTS

This section considered ten cloudlets and four VMs for example. Table 3 represents cloudlet properties, which comprise the cloudlet ids (Cld_id), and their respective MIs. Table 4 shows the properties of VMs comprising the VM ids and their respective MIPS. In Table 5, cloudlets allocated to each VM (0, 1, 2, 3) are shown respectively. In VM_0 according to the allocated cloudlets and equation (7) we have obtained the OTQ to be 44. In VM_1 according to the allocated cloudlets and equation (7) we have obtained the OTQ to be 22. In VM_2 according to the allocated cloudlets and equation (7) we have obtained the OTQ to be 30. In VM_3 according to the allocated cloudlets and equation (7) we have obtained the OTQ to be 29. Table 6 depicts the WT, TAT and Number of CS obtained in VM_0, VM_1, VM_2 and VM_3.

5. COMPARISON AND ANALYSIS

In this section the results of RRA, IRRCSA and the Proposed algorithms will be compared with each other and analyzed in three sub Sections – 5.1. Tabular analysis, 5.2. Graphical analysis, 5.3. Comparison of proposed work with SRDQ.

In Section 5.1, Tables 7–9 depicts comparison of the average WT, average TAT and number of CS for all the four VMs – VM_0, VM_1, VM_2, and VM_3 in case of RRA, IRRCSA and proposed algorithm. Table 10 shows a summary of the improvements achieved in the Proposed algorithm in comparison to RRA and IRRCSA. In Section 5.2, Figs. 3–5 depicts the comparison of WT, TAT and number of CS in all the four VMs in case of RRA, IRRCSA and proposed algorithm. In Section 5.3.2, Fig. 6 depicts the comparison of proposed algorithm and SRDQ with respect to average WT, average TAT and number of CS. Figure 7 shows the rate of improvements obtained from the proposed algorithm on comparison to SRDQ.

5.1. Tabular Analysis

Table 7. WT for RRA, IRRCSA and proposed in VM_0, VM_1, VM_2, VM_3

	VM_0			Avg. WT	VM_1			Avg. WT	VM_2		Avg. WT	VM_3		Avg. WT
Cld_id	**0**	**4**	**8**		**1**	**5**	**9**		**2**	**6**		**3**	**7**	
RRA	50	63	65	**59.33**	25	25	28	**26**	25	28	**26.5**	10	12	**11**
IRRCSA	40	63	55	**52.67**	25	15	23	**21**	20	28	**24**	5	12	**8.5**
PROPOSED	0	30	80	**36.66**	0	27	40	**22.33**	0	28	**14**	0	12	**6**

Table 8. TAT for RRA, IRRCSA and proposed in VM_0, VM_1, VM_2, VM_3

	VM_0			Avg. TAT	VM_1			Avg. TAT	VM_2		Avg. TAT	VM_3		Avg. TAT
Cld_id	**0**	**4**	**8**		**1**	**5**	**9**		**2**	**6**		**3**	**7**	
RRA	80	113	98	**97**	52	38	40	**43.33**	53	58	**55.50**	22	46	**34**
IRRCSA	70	113	93	**92**	52	28	35	**38.33**	48	58	**53**	17	46	**31.50**
PROPOSED	30	80	113	**74.33**	27	40	52	**39.66**	28	58	**43**	12	46	**29**

Table 9. Number of CS in RRA, IRRCSA and proposed in VM_0, VM_1, VM_2, VM_3

Algorithm	VM_0	VM_1	VM_2	VM_3
RRA	20	9	11	5
IRRCSA	17	6	9	3
PROPOSED	2	2	1	1

Table 10. Summary table for rate of improvement (%)

Performance Metric	VM_id	**Rate of improvement (%)** (in comparison to RRA)	**Rate of improvement (%)** (in comparison to IRRCSA)
WT	0	38.21%	28.11%
TAT	0	23.37%	19.20%
CS	0	90%	88.23%
WT	1	14.11%	–5.95%
TAT	1	8.46%	–3.35%
CS	1	77.77%	6.67%
WT	2	47.16%	41.66%
TAT	2	22.52%	18.86%
CS	2	90.90%	88.88%
WT	3	45%	29.41%
TAT	3	4.70%	7.93%
CS	3	80%	66.67%

5.2. Graphical Analysis

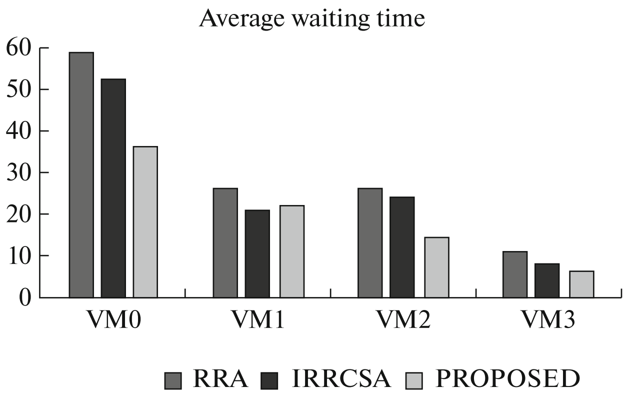

Fig. 3. Average WT for RRA, IRRCSA and proposed in VM0, VM1, VM2 and VM3.

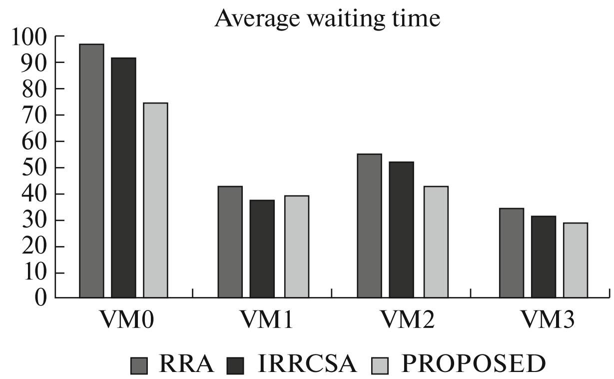

Fig. 4. Average TAT for RRA, IRRCSA and proposed in VM_0, VM_1, VM_2 and VM_3.

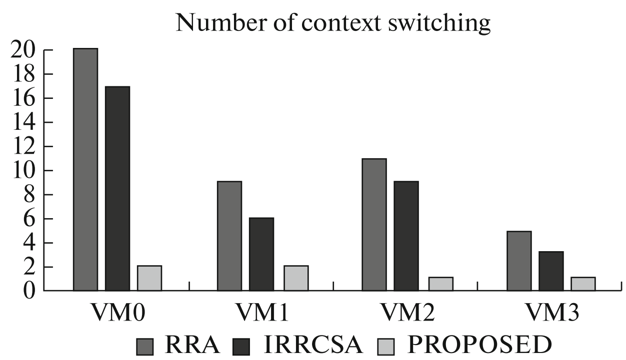

Fig. 5. Number of CS for RRA, IRRCSA and proposed in VM_0, VM_1, VM_2 and VM_3.

5.3. Comparison of Proposed Algorithm with SRDQ

The proposed algorithm and SRDQ are compared on the basis of average waiting time, average turnaround time and number of context switching required for the execution of all the cloudlets. Table 11 depicts the dataset considered for comparison. In Section 5.3.1, Table 12 depicts the performance analysis of the proposed algorithm and SRDQ with respect to average WT, average TAT and number of CS.

Table 11. Dataset

Cld_id	Burst time/Expected execution time (ET)
C1	12
C2	8
C3	23
C4	10
C5	30
C6	15
Total ET	**98**

According to the proposed algorithm and equation (5) of it, for the cloudlets 1–6—average execution time (AET) = 16.33.

According to equation (6) and (7), for cloudlets 1–6 – OTQ = 24.

According to the proposed algorithm, the Gantt chart obtained is as follows –

Cl1	Cl2	Cl3	Cl4	Cl5	Cl5	Cl6
0 — 12 — 12	12 — 8 — 20	20 — 23 — 43	43 — 10 — 53	53 — 24 — 77	77 — 6 — 83	83 — 15 — 98

5.3.1. Tabular Performance Analysis of Proposed Algorithm and SRDQ

Table 12. Performance analysis of the proposed algorithm and SRDQ with respect to average WT, average TAT and number of CS

Algorithm	**Average waiting time (WT)**	**Average turnaround time (TAT)**	**Number of context switching (CS)**
SRDQ	41.806	58.09	8
PROPOSED	35.16	51.50	5
Percentage of Improvement (%)	**15.90%**	**11.34%**	**37.50%**

5.3.2. Graphical Analysis of Proposed Algorithm and SRDQ

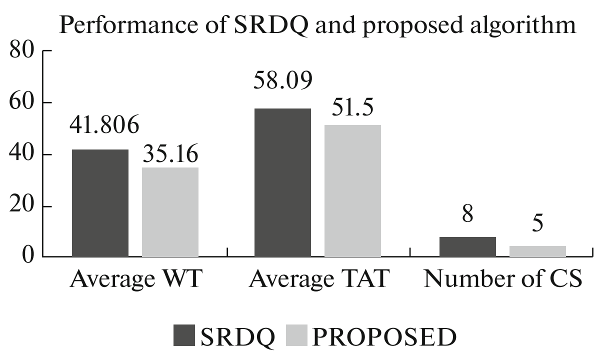

Fig. 6. Performance of SRDQ and proposed algorithm with respect to average WT, average TAT and number of CS.

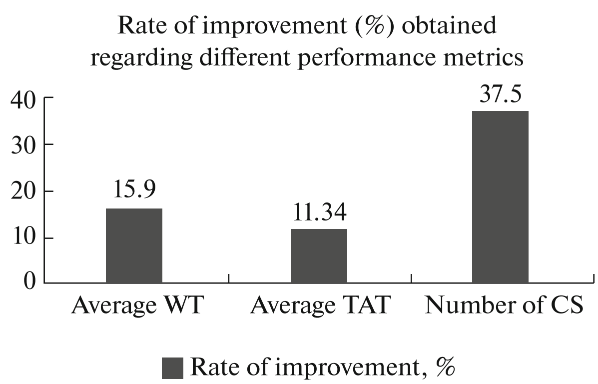

Fig. 7. Rate of improvement (%) obtained by proposed algorithm with respect to SRDQ.

6. FUTURE RESEARCH DIRECTIONS

Cloudlet scheduling being one of the most important portions of cloud computing [25–27] needs constant improvement for better accuracy and greater system throughput. The advancement in existing algorithmic approach may give cloud computing another level of extension. The parameters affecting the QoS, like Response Time, Make Span, Energy consumptions should be major areas to focus on.

7. CONCLUSION

The proposed algorithm, which clearly highlights the improvements over the existing IRRCSA and RRA scheduling policy has been thoroughly explained. The proposed algorithm being able to decrease the total number of CS, average WT and average TAT indicates the improvement in total system throughput. However we will extend this work to strengthen its capability of handling dynamic situation, scalability, fault tolerance and integrate it towards acquiring more efficiency and accuracy.

REFERENCES

1. Bianco, P., Lewis, G.A., and Merson, P., *Service Level Agreements in Service-Oriented Architecture Environments (No. CMU/SEI-2008-TN-021),* Carnegie-Mellon Univ. Pittsburgh Pa Software Engineering Inst., 2008. http://www.sei.cmu.edu/.
2. Wu, X., Deng, M., Zhang, R., Zeng, B., and Zhou, S., A task scheduling algorithm based on QoS-driven in cloud computing, *Procedia Comput. Sci.,* 2013, vol. 17, pp. 1162–1169. http://dx.doi.org/. doi 10.1016/j.procs.2013.05.148
3. Nayak, D., Malla, S.K., and Debadarshini, D., Improved round robin scheduling using dynamic time quantum, *Int. J. Comput. Appl.,* 2012, vol. 38, no. 5.
4. Rimal, B.P., Choi, E., and Lumb, I., A taxonomy, survey, and issues of cloud computing ecosystems, in *Cloud Computing: Principles Systems and Applications, Computer Communications and Networks,* Antonopoulos, N. and Gillam, L., Eds., Berlin: Springer, pp. 21–46. doi 10.1007/978-1-84996-241-4_2
5. Shaw, S.B., and Singh, A.K., A survey on scheduling and load balancing techniques in cloud computing environment, *Computer and Communication Technology (ICCCT), 2014 International Conference on,* 2014, pp. 87–95.
6. Buyya, R., Economic-based distributed resource management and scheduling for grid computing, arXiv preprint cs/0204048, 2012.
7. Wilkins-Diehr, N., Special issue: Science gateways—common community interfaces to grid resources, *Concurr. Comput.: Pract. Exp.*, 2007, vol. 19, no. 6, pp. 743–749.
8. Belalem, G., Tayeb, F.Z., and Zaoui, W., Approaches to improve the resources management in the simulator CloudSim, *International Conference on Information Computing and Applications,* 2010, pp. 189–196. doi 10.1007/978-3-642-16167-4_25
9. Dumitrescu, C.L. and Foster, I., GangSim: A simulator for grid scheduling studies, *Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on,* 2005, vol. 2, pp. 1151–1158.
10. Legrand, A., Marchal, L., and Casanova, H., Scheduling distributed applications: The SimGrid simulation framework, *Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on,* 2003, pp. 138–145.
11. Calheiros, R.N., Ranjan, R., De Rose, C.A., and Buyya, R., CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services, arXiv preprint arXiv:0903.2525, 2009.
12. Calheiros, R.N., Ranjan, R., De Rose, C.A.F., and Buyya, R., CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services, in *Technical Report, GRIDS-TR-2009-1,* Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2009.
13. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., and Buyya, R., CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Pract. Exp.,* 2011, vol. 41, no. 1, pp. 23–50.
14. Banerjee, S., Adhikari, M., and Biswas, U., Design and analysis of an efficient QoS improvement policy in cloud computing, *Serv. Oriented Comput. Appl.,* 2017, vol. 11, no. 1, pp. 65–73.
15. El-kenawy, E.S.T., El-Desoky, A.I., and Al-Rahamawy, M.F., Extended max-min scheduling using Petri net and load balancing, *Int. J. Soft Comput. Eng.,* 2012, vol. 2, no. 4, pp. 198–203.
16. Parsa, S., and Entezari-Maleki, R., RASA: A new task scheduling algorithm in grid environment, *World Appl. Sci. J.,* 2009, vol. 7, pp. 152–160.
17. Malhotra, R., and Jain, P., Study and comparison of CloudSim simulators in the cloud computing, *SIJ Trans. Comput. Sci. Eng. Its Appl.,* 2013, vol. 1, no. 4, pp. 111–115.
18. Contributed by techgreek in (2010). Types of scheduling 4th June.
19. Mohan, S., Mixed scheduling, a new scheduling policy, *Proceedings of Insight'09,* 2009.
20. Yang, J., Khokhar, A., Sheikh, S., and Ghafoor, A., Estimating execution time for parallel tasks in heterogeneous processing (HP) environment, *Heterogeneous Computing Workshop, 1994, Proceedings,* 1994, pp. 23–28. doi 10.1109/HCW.1994.324966
21. Roy, S., Banerjee, S., Chowdhury, K.R., and Biswas, U., Development and analysis of a three phase cloudlet allocation algorithm, *J. King Saud Univ., Comput. Inf. Sci.,* 2017, vol. 29, no. 4, pp. 473–483.
22. Bhatia, W., Buyya, R., and Ranjan, R., CloudAnalyst: A CloudSim based visual modeller for analyzing cloud computing environments and applications, *2010 24th IEEE International Conference on Advanced Information Networking and Applications,* 2010, pp. 446–452.
23. Brucker, P., *Scheduling Algorithms,* Berlin: Springer, 5th ed. doi 10.1007/978-3-540-69516-5
24. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., and Warfield, A., Xen and the art of virtualization, *ACM SIGOPS Oper. Syst. Rev.,* 2003, vol. 37, no. 5, pp. 164–177.
25. Xiong, K. and Perros, H., Service performance and analysis in cloud computing, *Services-I, 2009 World Conference on,* 2009, pp. 693–700. doi 10.1109/SERVICES-I.2009.121
26. Sotomayor, B., Montero, R.S., Llorente, I.M., and Foster, I., Virtual infrastructure management in private and hybrid clouds, *IEEE Internet Comput.,* 2009, vol. 13, no. 5. doi 10.1109/MIC.2009.119

27. Adhikari, M., Banerjee, S., and Biswas, U., Smart task assignment model for cloud service provider, *Int. J. Comput. Appl. Adv. Comput. Commun. Technol. HPC Appl.,* 2012, special issue, pp. 43–46.
28. Lei, X., Zhe, X., Shaowu, M., and Xiongyan, T., Cloud computing and services platform construction of telecom operator, *Broadband Network and Multimedia Technology, 2009. IC-BNMT'09. 2nd IEEE International Conference on,* 2009, pp. 864–867. doi 10.1109/ICBNMT.2009.5347793
29. Banerjee, S., Adhikari, M., and Biswas, U., Development of a smart job allocation model for a cloud service provider, *Business and Information Management (ICBIM), 2014 2nd International Conference on,* 2014, pp. 114–119. doi 10.1109/ICBIM.2014.6970946
30. Banerjee, S., Adhikari, M., Kar, S., and Biswas, U., Development and analysis of a new cloudlet allocation strategy for QoS improvement in cloud, *Arabian J. Sci. Eng.,* 2015, vol. 40, no. 5, pp. 1409–1425. doi 10.1007/s13369-015-1626-9
31. Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W., and Lu, C., A cloud computing based network monitoring and threat detection system for critical infrastructures, *Big Data Res.,* 2016, vol. 3, pp. 10–23.
32. Molyakov, A.S., Zaborovsky, V.S., and Lukashin, A.A., Model of hidden IT security threats in the cloud computing environment, *Autom. Control Comput. Sci.,* 2015, vol. 49, no. 8, pp. 741–744.