Improved Round Robin Scheduling Algorithm With Varying Time Quantum

*

1st FIAD Alaa
SIMPA LAB
USTO-MB
Oran,Algeria
alaa.fiad@univ-usto.dz

2rd Mekkakia Maaza Zoulikha
SIMPA LAB
USTO-MB
Oran,Algeria
zoulikha.mekkakia@univ-usto.dz

3nd BENDOUKHA Hayat
SIMPA LAB
USTO-MB
Oran,Algeria
hayat.bendoukha@univ-usto.dz

***Abstract*—Cloud computing is a distributed computing model; which refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. Task scheduling is one of the most important issues in cloud environment, such as the tasks must be affected to the appropriate virtual machines considering different factors at the same time, which leads to considering the task scheduling problem an NP-complete problem. This article presents an improved ROUND ROBIN (RR) CPU scheduling algorithm with varying time quantum. The appropriate algorithm proven better than traditional RR and IRRVQ algorithms. The results show that the waiting time and turnaround time have been reduced in the proposed approach compared to traditional RR and IRRVQ**

***Index Terms*—Waiting Time, Burst Time, Round Robin Scheduling, Turnaround Time.**

I. Introduction

As users begin to use clouds to store or run their applications, these systems become sensitive to workload issues. The optimal and fastest selection of the virtual machine for a specific task becomes an important problem, because an improper scheduling of tasks can reduce the system performance. The proposed algorithm is an improved version of ROUND ROBIN algorithm in which we focus on reducing the waiting time and turnaround time of tasks and maximizing the resource utilization.

The rest of the paper is organized as follows Section II presents the background of cloud computing, Section III focuses on some related work. While, in Section IV the proposed technique is presented with multiple examples and finally, the conclusions is presented in Section V

II. BACKGROUND

Cloud computing can be defined as a new style of computing in which scalable and often virtualized resources are provided as services on the Internet.This cloud model is composed of three service models, and four deployment models [1].

A. Deployment models

A deployment model defines the purpose of the cloud and the nature of how the cloud is located. The NIST definition for the four deployment models is as follows:

- Public cloud: The public cloud infrastructure is available for public use alternatively for a large industry group and is owned by an organization selling cloud services.
- Private cloud: The private cloud infrastructure is operated for the exclusive use of an organization. The cloud may be managed by that organization or a third party. Private clouds may be either on- or off-premises.
- Hybrid cloud: A hybrid cloud combines multiple clouds (private, community of public) where those clouds retain their unique identities, but are bound together as a unit. A hybrid cloud may offer standardized or proprietary access to data and applications, as well as application portability.
- Community cloud: A community cloud is one where the cloud has been organized to serve a common function or purpose. It may be for one organization or for several organizations, but they share common concerns such as their mission, policies, security, regulatory compliance needs, and so on. A community cloud may be managed by the constituent organization(s) or by a third party [2].

B. Service models

- SaaS (Software-as-a-Service) in this service, users use applications remotely from the cloud.
- Infrastructure-as-a-service (IaaS) provides virtual machines, virtual storage, virtual infrastructure and other hardware resources with guaranted processing power.
- Platform-as-a-Service (PaaS) is similar to IaaS, but also includes operating systems and required services for a particular application. In other words, PaaS is IaaS with a custom software stack for the given application [3].

C. SCHEDULING IN CLOUD COMPUTING

Scheduling is one of the most important activities of cloud computing. There are various task scheduling algorithms in the cloud environment, whose primary purpose is to attribute correctly the tasks arrived in the system to the resources available to obtain the minimum waiting time of tasks. Among these algorithms, we finds: FCFS, Round Robin, Min-Min, Max-Min, PSO, genetic algorithm...

- FCFS: First come First serve basis means that task that come first will be execute first.
- Round-Robin algorithm (RRA): it means that each task will be executed a time slice calling time quantum. In this Scheduling algorithm time is to be given to resources in a time slice manner.
- Min-Min Algorithm: Min-Min algorithm selects the smaller tasks to be executed first.
- Max-Min algorithm: Max-Min algorithm selects the bigger tasks to be executed first [4].

III. RELATED WORKS

Several studies and researches have been carried out in this field:

- First-Come-First-Serve: in this algorithm, the tasks are received and queued until the resource are available; once that's the case the tasks are mapped to them depending on their arrival time. No other criteria are considering for this technique [5].
- Round Robin (RR): its similar to FCFS, but a small time unit called the time quantum or time slice is defined. Each task run in the processor during this time quantum [6].
- Min-Min: In this approach the tasks are collected into a set called meta-task (MT). It has two phases. In the first phase the expected execution time of each task are calculated. In the second phase a task with the overall minimum expected completion time in MT are chosen and mapped to the corresponding resource, then the task are removed from MT and this operation are repeated until the MT are empty. Its gives better makespan compared to FCFS and RR, but the QOS in not consider.
- Max-Min: Its similar to Min-Min algorithm, but in phase 2 the Max-Min chose the task with maximum expected completion time and assign it to corresponding resource. Max-min algorithm does better than Min-min algorithm in cases when the number of short task is more than the longer ones [7].
- Improved Max-Min heuristic model for task scheduling in cloud: its an improved version of Max-Min algorithm. The scheduler schedule by assign tasks with maximum execution time to the resource which produce minimum completion time rather than original Max-min assigns task with maximum completion time to the resource which provides minimum execution time.it gives better makespan than Max-Min, but no criteria are consider like : scalability, availability, stability [8].
- RASA algorithm: its a combination between to known algorithms Max-Min and Min-Min. RASA uses the advantages of the both algorithms and covers their disadvantages. The remaining tasks are assigned to their appropriate resources by one of the two strategies, alternatively. For instance, if the first task is assigned to a resource by the Min-min strategy, the next task will be assigned by the Max-min strategy. In the next round the task assignment begins with a strategy different from the last round. Its gives better result than Max-Min and Min-Min algorithms. But this study is only concerned with the number of the resources to be odd [9].
- IRRVQ algorithm: its a combination between SJF and RR scheduling algorithms with varying time quantum. Initially the processes are arranged in the ascending order of their burst time in the ready queue. CUP is allocated to the first process in the ready queue and time quantum is put equal to his burst time. Processes are arranged in the ascending order after each cycle, the operation are repeated until the ready queue is empty. The IRRVQ shows his superiority compared to the traditional RR [10].

IV. PROPOSED APPROACH

The proposed approach is an improved version of a task scheduling algorithm ROUND ROBIN (RR). Initially, tasks arriving in the ready queue are arranged in the ascending order of their burst time, using the medium of tasks, the ready queue is divided into two sub-queues as follow: Tasks having burst time less than the medium are stored in the first queue (light task queue). And the rest of tasks in the second queue (heavy task queue). The tasks in the first queue are executed first following these steps: The CPU is allocated to tasks using RR scheduling with a small time unit, called time quantum equal to the burst time of the medium task, after each cycle the tasks in the queue are organized in ascending order of the remaining burst time and the process is repeated, until the queue is empty. After all the tasks in the first queue have been executed, the same steps are applied to schedule the tasks of the second queue. In what follows we present the algorithmic description of the approach:

```
1.Arrange the tasks in the ascending order
2.Find the medium burst time
3. For all tasks
4. If MBT ⟩ burst time of the task then put in light task queue
5. Else put in heavy task queue
6. End for
// The tasks in the first queue are executed first following these steps://
7. Quantum time == MBT
7. For all light stacks do
8. The CPU is allocated to tasks using RR scheduling (Quantum time)
9. For all heavy tasks do
10. The CPU is allocated to tasks using RR scheduling (Quantum time)
```

V. RESULTS AND DISCUSSION

A. Case 1

Five processes P1, P2, P3, P4 and P5 have been considered. The processes are arriving at time 0 with burst time 15, 32, 10, 26 and 20 respectively. The processes P1, P2, P3, P4 and P5 are arranged in the ascending order of their burst time in the ready queue which gives the sequence P3, P1, P5, P4 and P2. The ready queue is divided into two sub-queues using medium burst time . The first sub-queue contains the processes P3, P1 and P5, the second one contains P4 and P2.

Initially the processes of the first sub-queue are executed: The time quantum value is set equal to the burst time of the process medium i.e. 15. CPU is allocated to the processes P3, P1 and P5 for a time quantum of 15 milliseconds (ms). After the first cycle, the burst time of processes P3, P1 and P5 are equal to 0, 0 and 5 respectively. The processes P3 and P1 are finished execution so they are removed from the ready queue. The time quantum value is set to the burst time of P5 i.e. 5. The CPU is allocated to process P5 for a time quantum of 5 ms. After second cycle; the burst time of P5 is 0 so the process is removed from the ready queue.

The first sub-queue is empty so the second sub-queue begins execution: The time quantum value is set equal to the burst time of medium process, in this case we have two processes P4 and P2 so we calculate the average of their burst time so its equal to 29. CPU is allocated to the processes P4 and P2 from the ready queue for a time quantum of 29 ms. After third cycle, the burst time of P4 and P2 are 0, 3 respectively. The process P4 has finished execution, so its removed from the ready queue. Now only P2 in the ready queue, so CPU is allocated to P2 for a time quantum of 3 ms. The waiting time of P1 is 10 ms, for P2 is 71 ms, 0 ms for P3, 45 ms for P4, 25 ms for P5. The average waiting time is 30.2 ms. With same number of processes with same arrival and CPU burst times, the average waiting time is 54.2 ms for time quantum 10 in RR, and 46.2 ms for time quantum 10, 5, 5, 6, 6 in IRRVQ. The average turnaround time is 50.8 ms in proposed algorithm while average turnaround time is equal to 74.8 for time quantum 10 in RR, and 66.8 for time quantum 10, 5, 5, 6, 6 in IRRVQ.

The comparison result of RR, IRRVQ and proposed algorithm is shown in Table 2 and Fig. 1.

Fig. 2 show the Gantt chart representation of RR with time quantum 10.

Fig. 3 show the Gantt chart representation of IRRVQ with time quantum 10, 5, 5, 6, 6 respectively.

Fig. 4 show the Gantt chart representation of proposed algorithm.

TABLE I
PROCESSES WITH THEIR ARRIVAL AND BURST TIME (CASE1).

Processes	Arrival time	Burst time
P1	0	15
P2	0	32
P3	0	10
P4	0	26
P5	0	20

TABLE II
COMPARISON OF RR, IRRVQ AND PROPOSED ALGORITHM (CASE1).

Algorithm	Time Quantum	Average Waiting Time (ms)	Average Turnaround Time (ms)
RR	10	54.2	74.8
IRRVQ	10, 5, 5, 6, 6	46.2	66.8
Proposed approach	15, 5, 29, 3	**30.2**	**50.8**

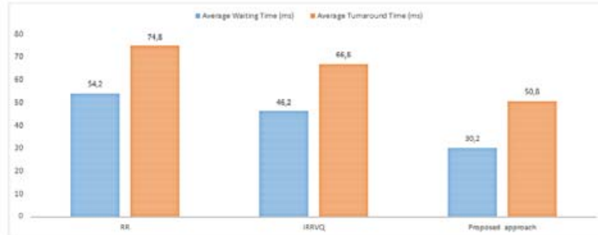

Fig. 1. Comparison of RR, IRRVQ and proposed algorithm (Case1).

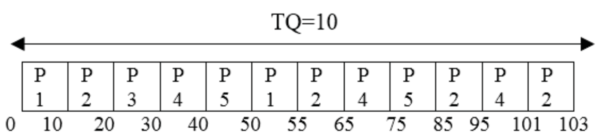

Fig. 2. Gantt chart representation of RR with TQ=10 (Case1).

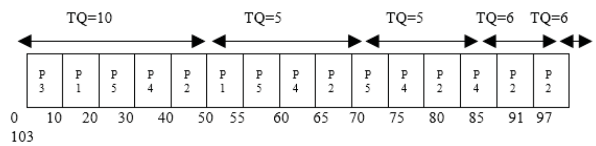

Fig. 3. Gantt chart representation of IRRVQ (Case1).

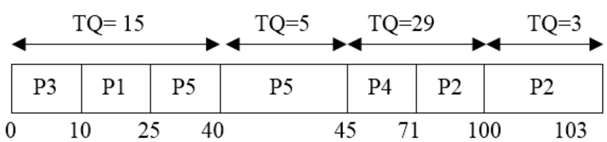

Fig. 4. Gantt chart representation of proposed algorithm (Case1).

B. Case 2

In this case processes are arriving at different time. Five processes P1, P2, P3, P4 and P5 arriving at time 0, 4, 10, 15, 17 respectively with burst time 7, 25, 5, 36, 18. The time quantum value is set equal to the burst time of the first process in the ready queue i.e. 7. CPU is allocated to the process P1 from the ready queue for a time quantum of 7 ms. After first

TABLE III
PROCESSES WITH THEIR ARRIVAL AND BURST TIME (CASE 2).

Process	Arrival Time	Burst Time
P1	0	7
P2	4	25
P3	10	5
P4	15	36
P5	17	18

TABLE IV
COMPARISON OF RR, IRRVQ AND PROPOSED APPROACH (CASE2).

Algorithm	Time Quantum (TQ)	Average Waiting Time (ms)	Average Turnaround Time (ms)
RR	7	25.6	43.8
IRRVQ	7, 11, 7, 11	19.4	37.6
Proposed approach	7, 25, 11.5, 6.5, 36	**17**	**35.2**

Fig. 5. Comparison of RR, IRRVQ and proposed algorithm (Case2).

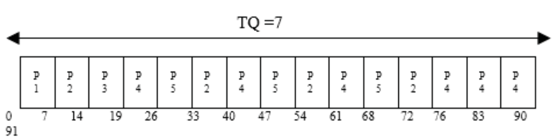

Fig. 6. Gantt chart representation of RR with TQ = 7 (Case2).

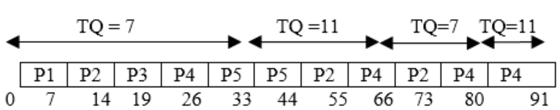

Fig. 7. Gantt chart representation of IRRVQ (Case2).

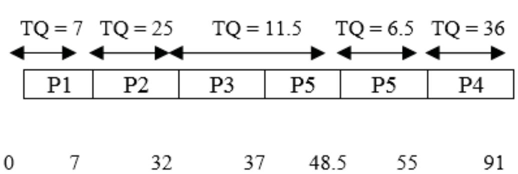

Fig. 8. Gantt chart representation of proposed algorithm (Case2).

cycle, the burst time of P1 is 0 and P2 has arriving in the ready queue. The process P1 has finished execution, so its removed from the ready queue. The time quantum value is set equal to the burst time of P2 i.e. 25. CPU is allocated to P2 for a time quantum of 25 ms. After second cycle, the burst time of P2 is 0 and P3, P4, P5 have arriving in the ready queue. The process P2 has finished execution so its removed from the ready queue. The processes P3, P4 and P5 are arranged in the ascending order of their remaining burst time in the ready queue. The ready queue is divided into two sub-queues using medium burst time. The first sub-queue contains P3 and P5, and the second sub-queue contains P4.
Initially the processes of the first sub-queue are executed: The time quantum value is set equal to the burst time of the process medium i.e. 11.5. CPU is allocated to the processes P3 and P5 for a time quantum of 11.5 ms. After third cycle, the remaining burst time for P3 and P5 are 0 and 6.5 respectively. The process P3 has finished execution, so its removed from the ready queue. The time quantum value is set equal of the burst time of P5 i.e. 6.5. CPU is allocated to the process P5 for a time quantum of 6.5 ms.
The first sub-queue is empty so the second sub-queue begins execution: The time quantum value is set equal to the burst time of the only process in the queue P4 i.e. 36. CPU is allocated to the process P4 for a time quantum of 36 ms. The waiting time of P1 is 0 ms, 3 ms for P2, 22 ms for P3, for P4 is 40 ms and 20 ms for P5. The average waiting time is 17 ms. With the same example, the average waiting time is 25.6 ms for time quantum 7 in RR, and 19.4 ms for time quantum 7, 11, 7, 11 in IRRVQ. The average turnaround time is 35.2 in proposed algorithm while its equal to 43.8 for time quantum 7 in RR and 37.6 for time quantum 7, 11, 7, 11 in IRRVQ.
The comparison result of RR, IRRVQ and our algorithm is shown in Table 4 and Fig. 5.
Fig. 6 show the Gantt chart representation of RR with Time Quantum 7. Fig. 7 show the Gantt chart representation of IRRVQ with time quantum 7, 11 7, 11 respectively. Fig. 8 show the Gantt chart representation of proposed algorithm.

VI. CONCLUSION AND FUTURE WORK

The paper presents an improved version of Round Robin task scheduling algorithm with varying time quantum. The results show that the proposed algorithm gives better performance than conventional RR and IRRVQ algorithms. The waiting time and turnaround time have been reduced in our algorithm. This will help to improve the performance of the system.

REFERENCES

[1] Mell, Peter, & Tim. (2011, September 28). The NIST Definition of Cloud Computing. Retrieved January 8, 2020, from https://csrc.nist.gov/publications/detail/sp/800-145/final.
[2] Sosinsky, B. (2011). Cloud computing bible. Indianapolis, IN: Wiley Publishing.
[3] Escalante, A. (2010). Handbook of Cloud Computing. Berlin: Springer US. doi: https://doi.org/10.1007/978-1-4419-6524-0

[4] Sharma, Shimpy. (2014). Different Scheduling Algorithms in Different Cloud Environment. IJARCCE. 3.
[5] Rahul,M. (2015). A Brief Review of Scheduling Algorithms in Cloud Computing. Asian Journal of Technology & Management Research.
[6] Arya, G. P., Nilay, K., & Prasad, D. (2018). An Improved Round Robin CPU Scheduling Algorithm based on Priority of Process. International Journal of Engineering & Technology, 7(4.5), 238-241. doi:10.14419/ijet.v7i4.5.20077
[7] Etminani, K., & Naghibzadeh, M. (2007). A Min-Min Max-Min selective algorihtm for grid task scheduling. 2007 3rd IEEE/IFIP International Conference in Central Asia on Internet. doi:10.1109/canet.2007.4401694
[8] Devipriya, S., & Ramesh, C. (2013). Improved Max-min heuristic model for task scheduling in cloud. 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE). doi:10.1109/icgce.2013.6823559
[9] Parsa (2009). RASA: A New Grid Task Scheduling Algorithm. International Journal of Digital Content Technology and its Applications.
[10] Kumar Mishra, M. and Rashid, F. (2014). An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum. International Journal of Computer Science, Engineering and Applications, 4(4), pp.1-8.