

Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data

GibetTaniHicham, El amraniChaker

AbdelmalekEssaadi University, Morocco

*Corresponding author: GibetTaniHichamgibet.tani.hicham@gmail.com

Abstract

Cloud Computing and Big Data are the upcoming Information Technology (IT) computing models. These groundbreaking paradigms are leading IT to a new set of rules that aims to change computing resources delivery and exploitation model, thus creating a novel business market that is exponentially growing and attracting more and more investments from both providers and end users that are looking forward to make profits from these innovative models of computing. In the same context, researchers and investigators are wrestling time in order to develop, test and optimize Cloud Computing and Big Data platforms, whereas several studies are ongoing to determine and enhance the essential aspects of these computing models especially compute resources allocation. The processing power scheduling is crucial when it comes to Cloud Computing and Big Data because of the data growth management and delivery design proposed by these new computing models, that requires faster responses from platforms and applications. Hence originates the importance of developing high efficient scheduling algorithms that are compliant with these computing models platforms and infrastructures requirement.

keywords

Cloud Computing; Big Data; Scheduling Algorithms; Cloud Computing Simulation; Round Robin; First Come First Served

INTRODUCTION

Cloud Computing introduces a new Information Technology (IT) service delivery model that aims to reshape the world of Computing. In fact, a Cloud user needs only a light terminal to access any desirable IT resource. The Cloud proposed resources ranges from documents processing, email or storage to complicated data treatment and management system such as Enterprise Resource Planning, Database Management Systems, Integrated Development Environment or Application hosting, and all of that is presented as a service and charged in a pay-as-you-use, pay-as-you-go or even by subscription.

Recently, we cannot talk about Cloud Computing without tackling the subject of data growth management, in other terms Big Data. This newly born computing model is on an everyday development in order to cope with the changes going on the IT world thus adapting to the world Cloud orientation.

The central focus on these computing models are to raise profits for both providers and users and this is by reducing the costs relative to IT infrastructures, platforms management and scaling up as well as promoting accuracy and auditability. With the Cloud Computing outstanding business growth, the investigations on Cloud Computing resources allocation and scheduling algorithms are becoming a major issue because of the Cloud delivery model and whereas countless researches are carried out each year all over the world.

Under the same theme, this paper presents an ongoing exploration of Central Processing Unit (CPU) allocation and task scheduling on the Cloud in order to elaborate an optimal design that guarantee data treatment celerity and efficiency while respecting Big Data fundamentals.

I CPU ALLOCATION AND SCHEDULING

The allocation and scheduling of Compute resources (CPU) is a major Cloud Computing factor that will affect massively the most significant characteristics of Quality of Service (QoS) related to applications and platforms response time, thus comes the importance of choosing high-enhanced CPU allocation algorithms that suits the cloud infrastructure and meets the customer requirements.

The major Compute resources scheduling algorithms are the following:

- First Come First Served Scheduling :

One of the basic scheduling algorithms and the foundation of the most used scheduling algorithms. The algorithm has a simple design of processing tasks based on their arrival order [2][3].

- First Come First Served with Shortest Job First Scheduling:

This algorithm is one version of the previous algorithm, which adds a new rule that arranges tasks in an ascending way based on their burst time and then queuing them for execution [2][3][4].

- Round Robin Scheduling:

The Round Robin algorithm was designed based on the distribution of CPU time among the scheduled tasks. On the same context, all the tasks get on a queue list whereas each task get a small unit of CPU time (Quantum, usually 10-100 milliseconds) [2].

- Min-Min Scheduling:

The mechanism of Min-Min scheduling algorithm work on determining the minimum completion time for each task submitted for execution. All tasks go through two steps: In the first step, the algorithm looks for a set of tasks with minimum completion time and then begins the second step to appoint the task selected in the previous step to the right resource (Selective algorithm) [4][5].

- Max-Min Scheduling:

The Max-Min scheduling is a selective algorithm that functions exactly as for the Min-Min scheduling but this time the algorithm chooses the task with the higher completion time and continues the work on a descending order [4][5].

- Priority Scheduling:

The Priority Scheduling algorithm assigns a fixed priority (score) to every task, and the scheduler arranges the tasks in the ready queue based on their priority (score) and executes the ones with the higher priority and so on [6].

- GeneticScheduling:

The genetic algorithm is very simple. It starts with a population of random individuals, each corresponding to a particular candidate solution to the problem. Then, the best individuals survive, mate, and create offspring, originating a new population of individuals. This process is repeated a number of times, and usually leads to better and better individuals [7].

II ROUND ROBIN CPU SCHEDULING ALGORITHM

To the best of our knowledge, the Round Robin Scheduling algorithm is one of the simplest and most used scheduling algorithm up to this moment. The concept of this algorithm is to share the CPU time among all scheduled tasks on a ready queue. The most important aspect of the Round Robin algorithm is the time slice (Time Quantum) that will be allocated to each task submitted for execution.

While the time quantum is a decisive characteristic on the Round Robin algorithm, several proposed Round Robin based algorithms are suggesting static time quantum that segments the CPU time among all submitted tasks, nevertheless, a static time quantum is not always the best solution.

A more viable alternative is the use of dynamic time quantum that adapts the CPU time slices to the tasks changes happening on the ready queue for execution. Under the same topic, the Round Robin based algorithm proposed on this paper uses a dynamic time quantum and adds a smarter layer to the existing algorithm in order to adjust the CPU time to different situations.

III LITERATURE REVIEW ON ROUND ROBIN ALGORITHM

Several researchers and investigators working on improving the CPU response time of the Round Robin algorithm concentrate their effort on the CPU time-sharing design. Following are the most remarkable carried improvement on this algorithm:

A. Abdulrazaq, E. A. Saleh and B. S. Junaidu [8] proposed a Round Robin based algorithm, which calculates a dynamic time quantum based on the average burst time of tasks on a queue list and then allocates the time quantum to every task on the queue. The algorithm introduces a test to verify if the time quantum was enough for task termination and reallocate the necessary time to complete the task execution in the opposite case. This mechanism reduced the tasks context switches thus improved the algorithm performance.

N. R. Siva, S.V.N. Srinivasu and K. R. Rama [9] proposed a Round Robin time quantum calculated from the average burst time of tasks on the ready queue and it is dynamically adjustable to those tasks that require greater time than the allotted time quantum.

N. Abbas, K. Ali and K. Seifedine [10] introduced a new-based Round Robin algorithm that calculates the time quantum dynamically as the average burst time of tasks on the ready queue without any task arrangement and assigns the time quantum based on the tasks arrival time to the ready queue.

M. Rakesh, H. S. Behera, P. Khusbu, D. Monisha and P. Lakshmi [11] proposed a Round Robin based algorithm that calculates a dynamic time quantum based on Priority otherwise the algorithm adapts the tasks execution to the Shorter Job First algorithm.

M. K. Mishra and K. Abdul Kadir [12] introduced a new perception to the simple Round Robin, which compares the remaining burst time of a task to the algorithm static time quantum after the first allocation. If the remaining time is less than one time quantum the CPU will reallocate the time quantum to the task otherwise it is sent to a waiting queue.

IV THE PROPOSED MODIFIED ROUND ROBIN ALGORITHM

The design that we propose for the modified Round Robin based algorithm (Smarter Round Robin, SRR) begins with a logical organization of submitted tasks based on their burst time and queuing them on a waiting list.

The smarter layer we talked about previously steps in during the time quantum calculation and adapts the CPU time to several situations:

- In case of a waiting list with three or less tasks:

- The proposed algorithm uses the Shorter Job First Algorithm design and allocates the time necessary for each task until its execution

- In case of a waiting list with more than three tasks

- If the total count of tasks on the list is a pair number
 - The proposed algorithm dynamically calculates a time quantum as the average of tasks burst time.
- If the total count of tasks on the list is an impair number
 - The proposed algorithm dynamically calculates a time quantum as the median of tasks burst time

For each of the conditions that treats the case of a waiting list with more than three tasks, the proposed algorithm injects a test to verify if there are more than three reoccurring tasks with the same burst time. If it is the case, the time quantum is set to the value of the repetitive task with highest burst time on the waiting list.

Here is an illustration of how the proposed Smarter Round Robin CPU Scheduling works:

Smarter Round Robin

First, all the processes are sorted in ascending way based on the tasks burst time (the time needed for execution) and sent to the waiting list

1. $nt \rightarrow$ number of tasks
2. $i \rightarrow$ counter

Second, if the tasks count on the waiting list is equal or less than three, send the waiting list tasks to the ready list

3. While (WQ != NULL and $nt \leq 3$) {
4. $RQ \leftarrow WQ$

// Applying Shorter Job First algorithm

5. For $i=1$ to nt
6. {
7. $T_i \leftarrow T_i$
8. Send task T_i to finish list
9. $i \leftarrow i + 1$

//WQ: Waiting Queue List
//RQ: Ready Queue List
// TQ: Time Quantum
// T_i : Burst Time of Task i
}
}

Third, if the tasks count is greater than three, find duplicates and send them to duplicate list

10. While (WQ != NULL and $nt > 3$) {
11. If (DL != NULL)

//DL: Duplicates List

12. Send Tasks from Waiting List to Ready List
13. $RQ \leftarrow WQ$
14. $TQ \leftarrow T_{n-1}$

//n: Task counter on the Duplicates List

```

15. For i=1 to nt
{
    If TQ >= Ti
    Ti ← Ti
    Send task Ti to Finish list
    i ← i + 1
    Else
    Ti ← TQ
    Send task Ti to Waiting list
    i ← i + 1
}
16. Go to First step
Fourth, if the tasks count on the waiting list is a pair number and greater than three, send tasks on waiting list to ready list
17. RQ ← WQ
18. TQ ←  $\sum Ti / nt$ 
19. For i=1 to nt
20. {
    If TQ >= Ti
    Ti ← Ti
    Send task Ti to Finish list
    i ← i + 1
    Else
    Ti ← TQ
    Send task Ti to Waiting list
    i ← i + 1
}
}
21. Go to First step
Fifth, if the tasks count on the waiting list is an impair number and greater than three, send tasks on waiting list to ready list
22. RQ ← WQ
23. TQ ←  $T(nt+1 / 2) / 2$ 
24. For i=1 to nt
25. {
26. If TQ >= Ti
27. Ti ← Ti
    Send task Ti to Finish list
    i ← i + 1
28. Else
    Ti ← TQ
    Send task Ti to Waiting list
    i. i ← i + 1
}
29. Go to First step
}

```

V SMARTER ROUND ROBIN ALGORITHM IMPLEMENTATION AND EVALUATION

In order to assess and evaluate the proposed modified Round Robin Algorithm, we considered a set of experiments tested and implemented on CloudSim simulation toolkit and compared to the algorithms integrated with this Cloud simulation platform, simple Round Robin (RR) and First Come First Served (FCFS). We also compared the results with the Shorter Job First (SJF) algorithm and a previously proposed Round Robin algorithm with a dynamic time quantum calculated as the Average Burst Time (Round Robin ABT) [2]. The implementation with Cloud Computing and Big Data real infrastructures is planned and with will be done shortly.

The evaluation metrics were set to the following:

- Average Turnaround Time: The average amount of time necessary to carry out the execution of tasks.
- Average Waiting Time: The average amount of time spent by tasks on the waiting queue.
- Average Response Time: The average amount of time from task submission until the first response.
- Number of Context Switches: The number of switches from one task to another on the ready queue.

The experiments were conducted on a set of tasks randomly submitted for execution in order to assess the effectiveness of these algorithms and pinpoint their flaws.

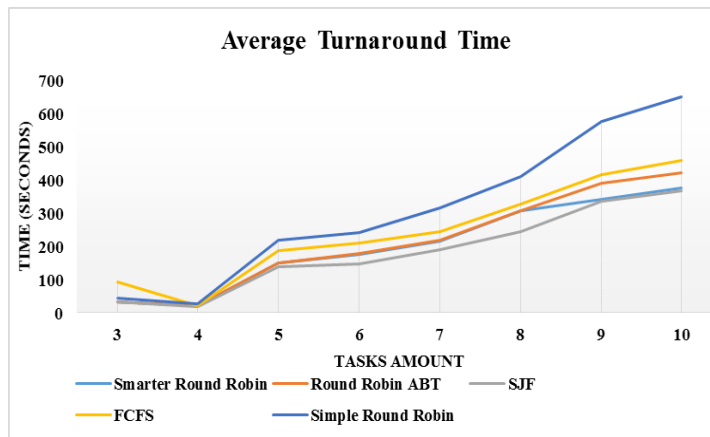


Figure 1: Average Turnaround Time Evaluation

“Figure 1” represents the performance of the evaluated algorithms on regards to the turnaround time. The Smarter Round Robin algorithm proposed on this paper gives remarkable results regarding the necessary time to carry out the execution of each task thus promoting celerity.

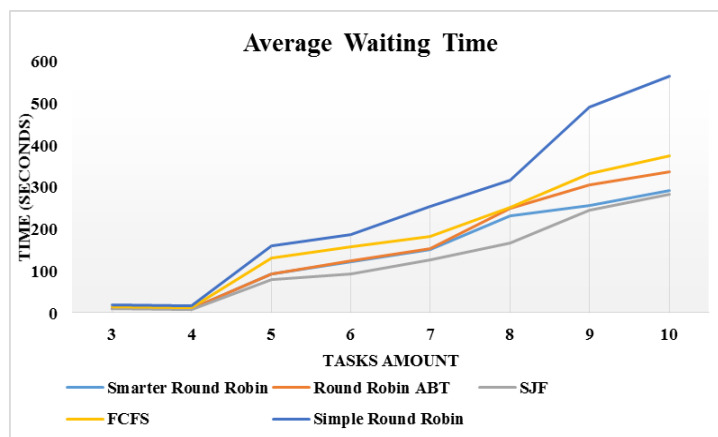


Figure 2: Average Waiting Time Evaluation

“Figure 2” is a depiction of the time each task spends on the waiting list before going into action. The proposed Smarter Round Robin algorithm gives proximate results to the Shorter Job First algorithm that takes on the lead of the evaluated algorithms.

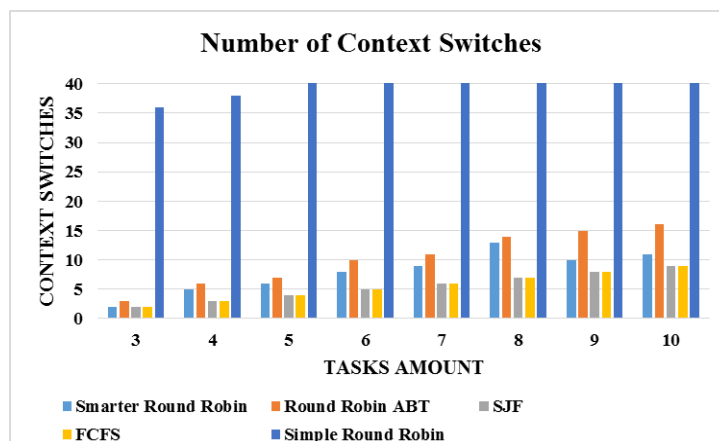


Figure 3: Context Switches Evaluation

“Figure 3” illustrates the number of context switches for each of the evaluated algorithms. The Smarter Round Robin algorithm enhanced the results of the previous proposed version of this algorithm [2] and reduced the amount of context switches thus reducing the power consumption.

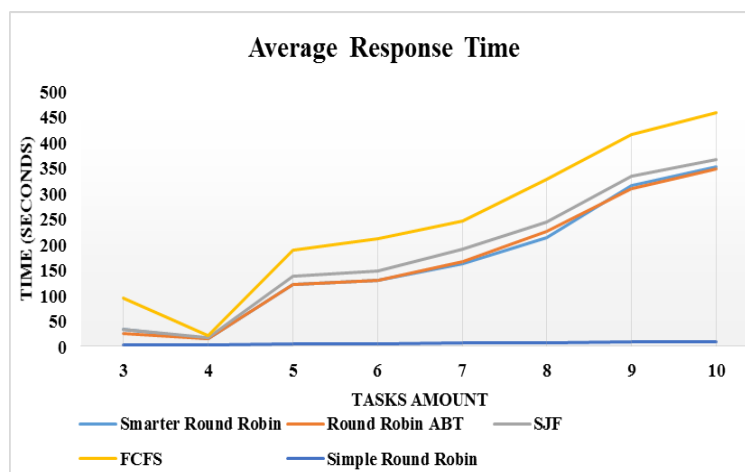


Figure 4: Average Response Time Evaluation

“Figure 4” shows how the Smarter Round Robin algorithm gave a remarkable average response time in comparison with the other scheduling algorithms.

Conclusion

In this paper, an enhanced version of the Round Robin algorithm, called Smarter Round Robin (SRR) has been introduced. The evaluations conducted to test this proposed algorithm showed great benefits of this version in comparison to the other existing scheduling algorithms especially those integrated with CloudSim Simulation toolkit. The improvements we brought to the Round Robin algorithm aims to inject a smarter layer to the existing algorithm in order to adapt the algorithm to different situations that comes with the new delivery model of Cloud Computing as well as reducing the run-time of big data processing. While the Shorter Job First algorithm gave great results in regards to the average turnaround time, average waiting time and number of context switches it still lacks the intelligence and the capability to run tasks in parallel mode or to detect situations where a user executes several instances of the same tasks. The First Come First Served algorithm shows poor performances because of its functioning mechanism that executes the tasks based on their logical arrival time. Consequently, once a time consuming task takes on the lead of the ready

queue list, the other submitted tasks will have to wait for this huge task to finish execution before having the chance to get to the processor thus a long waiting time. The previous proposed version of the Round Robin algorithm [2] used a dynamic CPU quantum based on the average burst time, whereas the time quantum will be recalculated each time there are tasks on the waiting list. This new perception exhibits great benefits but was unable to detect the changes happening on the waiting list, therefore, even if there were only two tasks on the waiting list and their burst time is very low the algorithm instructs the CPU to recalculate the time quantum thus a high number of context switches and average waiting time. Finally, The Round Robin algorithm version integrated with the CloudSim simulation toolkit uses a static time quantum calculated based on the system resources, the static quantum is then allocated for each submitted task. This algorithm is deficient in respect to the average turnaround time, average waiting time and number of context switches, which were always very high; hence, it is not well suited for Cloud Computing nor Big Data.

Our future work will be concentrated on implementing the proposed algorithm with real life Cloud Computing and Big Data platforms as well as on bringing more intelligence to the proposed scheduling algorithms especially neural networks that can have a massive impact on compute resources allocation.

References

- Savitha. P, J Geetha Reddy, A Review Work On Task Scheduling In Cloud Computing Using Genetic Algorithm, *International Journal of Scientific & Technology Research*, Volume 2, Issue 8, August 2013
- G. T. Hicham, EL.Chaker, Cloud Computing CPU Allocation and Scheduling Algorithms using CloudSim Simulator, *International Journal of Electrical and Computer Engineering*, Vol 6, No 4, 2016
- K. Rajveer, K. Supriya, Analysis of Job Scheduling Algorithms in Cloud Computing, *International Journal of Computer Trends and Technology*, volume 9, number 7, Mar 2014
- N. Zanoon, D. Rawshdeh, STASR: A New Task Scheduling Algorithm For Cloud Environment, *Network Protocols and Algorithms*, Vol. 7, No. 2, 2015
- B. Santhosh, Harshitha, A. PrachiKanerla, D. H.Manjaiah, Comparative Study of Workflow Scheduling Algorithms in Cloud Computing, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol.2, Special Issue 5, October 2014
- S. R. Ishwari, G. Deepa, A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems, *International Journal of Innovations in Engineering and Technology*, Vol. 1, Issue 3, Oct 2012
- G. Mohita, G. Savita, Optimized Processor Scheduling Algorithms using Genetic Algorithm Approach, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 6, June 2013
- A. Abdulrazaq, E. A. Saleh, Junaidu B. Sahalu, A New Improved Round Robin (NIRR) CPU Scheduling Algorithm, *International Journal of Computer Applications*, Volume 90 – No 4, March 2014
- G. Siva Nageswara Rao, N. Srinivasu, S.V.N. Srinivasu, G. Rama Koteswara Rao, Dynamic Time Slice Calculation for Round Robin Process Scheduling Using NOC, *International Journal of Electrical and Computer Engineering*, Vol. 5, No. 6, December 2015
- N. Abbas, K. Ali, K. Seifedine, A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average, *International Journal of Computer Science Issues*, Vol. 8, Issue 3, No. 1, May 2011
- M. Rakesh, H. S. Behera, P. Khusbu, D. Monisha, P. Lakshmi, Priority Based Dynamic Round Robin (PBDRR) Algorithm with Intelligent Time Slice for Soft Real Time Systems, *International Journal of Advanced Computer Science and Applications*, Vol. 2, No.2, February 2011
- M. Manish Kumar, K. Abdul Kadir, AN IMPROVED ROUND ROBIN CPU SCHEDULING ALGORITHM, *Journal of Global Research in Computer Science*, Volume 3, No. 6, June 2012