# An Integration of Round Robin with Shortest Job First Algorithm for Cloud Computing Environment

*Dr. Thomas Yeboah*[1]
*HOD, Department of Computer Science Christian Service University College*
*tyeboah@csuc.edu.gh*

*Prof. Odabi I. Odabi*[2]
*WellSpring University*
*Benin-City*
*Nigeria*
*odabiiodabi@gmail.com*

*Kamal Kant Hiran*[3]
*HOD, Department of IT*
*Sikkim Manipal University*
*Accra*
*kamal@smughana.com*

***ABSTRACT: Cloud computing is a recent scientific revolution in information technology that promises a delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet). Delivery of services is provided to computers and other devices upon request. In other words, it is a technology based on the internet and central remote servers to maintain data and applications. This technology allows consumers and enterprises to use applications without the need of installing them or allowing access to their personal files at any computer with internet access. It has many merits except that there are some crucial issues that need to be resolved in order to improve reliability of cloud environment. These issues are related with the job Scheduling, fault tolerance and different security issues in cloud environment. One of the most popular job scheduling algorithms that gives satisfactory load balancing is Round Robin (RR). This paper proposes an enhancement to the traditional RR, namely Round Robin with Shortest Job First (RRSJF). The enhanced version of RR algorithm is based on the integration of round-robin with shortest job first algorithm that selects for processes based on shortest job first in a round robin fashion to give optimal selection of job. A simulation has been carried out using CloudSim simulator V 3.0 to test the performance of the proposed scheme in terms of different evaluation metrics such as average turnaround, average waiting time and context switches. This Scheduling algorithm gives better result compared to Round Robin (RR).***

***KEYWORDS: Round Robin, Shortest Job First, Scheduling Algorithm, performance, cloud computing, load balancing, Waiting time, Response time***

## I. INTRODUCTION

Cloud computing is a general term for anything that involves delivering hosted services over the Internet.

Basically, two kinds of cloud are identified in cloud environment. The first type is public cloud. For this type service may be sold to anyone on the Internet. Here, Amazon Elastic Compute Cloud (EC2) [2], Google App Engine [4] are large public cloud providers. The second type of the cloud is the private cloud. It is a proprietary network that supplies hosted services to a limited number of clients (end-users). Cloud computing enables users to remotely run services such as Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) [3].

In cloud computing Load balancing is the process of distributing the load among various virtual machines of a cloud computing system to improve both resource utilization and job response time while also avoiding a situation where some of the virtual machines are heavily loaded while other virtual machines are idle or doing very little work. Load balancing ensures that all the virtual machines in the cloud system or every node in the network does approximately equal amount of work at any instant of time. Job Scheduling is a fundamental load balancing function that determines which process run, when there are multiple processes. Job scheduling is important because it impacts resource utilization and other performance parameters. In cloud computing, there exists a number of job scheduling algorithms like First Come First Serve (FCFS), Shortest Job First Scheduling (SJF), Round Robin scheduling, Priority Scheduling etc. This paper proposes an enhancement to the traditional RR, namely Round Robin with Shortest Job First (RRSJF) to establish an effective load balancing that can be implemented in cloud environment to ensure that all Virtual Machines are busy in their assigned jobs and none of the Virtual Machines gets Idle.

## II. RELATED WORKS

Efficient job scheduling to datacenters in cloud computing environment is done with the main aim of achieving load balancing.

### A. Round Robin and Its Variants

Round Robin algorithm is one of the most popular and simple algorithms used to implement load balancing in cloud environment. Due to its simplicity and popularity in nature; there has been lot of research carried out to improve performance of this algorithm and therefore there are so many variants of this algorithm.
Singh e.tal [11] proposed a modified round robin (MRR), which is superior than simple Round Robin (RR) and has less waiting response time, usually less pre-emption and context switching thereby reducing the overhead and saving of memory space. In Modified Round Robin (MRR) algorithm, Time Slice (TS) is calculated based on (range× total no of process (N)) divided by (priority (pr) × no of process in queue (p)). Range is calculated by (maximum burst time + minimum burst time) divided by 2. This algorithm improved a scheduling up to some extent but not much. In paper [7] a Priority Based Round Robin was introduced which uses a calculated intelligent time slice. In their research the calculated intelligent time slice allocates a different Time Quantum (TQ) to the individual process according to the priority given to each process. Their results show that PBDRR performs better than MRR algorithm in terms of reducing the number of context switches, average waiting time and average turnaround time. Subasish et. al [12] also proposed a load balancing based algorithm on Time Slice Priority Based Round Robin (TSPBRR). In their algorithm they initially took Time Quantum (TQ) as half of the first request of the Burst Time (1/2BT). It is observed in their research that this algorithm gives considerable improvement over that of MRR but still needs more improvement in response time. A load balancing framework, Weighted Round Robin (WRR) algorithm that allocates request to various virtual machines in a round robin fashion based on the weight without considering the current load on each virtual machine was also proposed in [8]. In their proposed algorithm if a request is allocated to a busy VM it can lead to wait few request wait for a long time that can increase response time. Komal et.al [5] introduced a round robin with server affinity algorithm. This algorithm uses two data structures: *Hash map* – This stores the entry of the last VM and *VM state list*- It stores the allocation status of each VM. In their algorithm when a request is made it first look into the Hash map and if the VM is available, there is no need to run the Round Robin in this case; therefore can save a significant amount of time.

### B. Round Robin Algorithm(RR)

In round Robin Algorithm each process is provided a fix time to execute called quantum. Once a process is executed for given time period. Process is preempted and other process executes for given time period.
For example, table 1-1 shows the execution time of 3processes with quantum number of 4ms.

Table 1-1: 3-Perocess Tasks

| Process | Burst Time |
|---|---|
| P1 | 24 |
| P2 | 3 |
| P3 | 3 |

Using RR algorithm, the processes would be scheduled as shown in Table 1-2

Table 1-2 shows the Gantt chart in a round robin fashion:

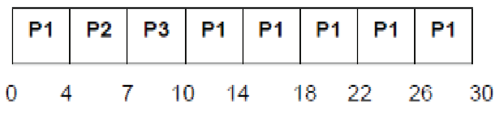

Table 1-2: Gantt chart of Process in Round Robin Fashion.

Average waiting time = (4 + 7 + (10 – 4)) / 3
= 17/ 3 = 5.66 ms.

### C. Shortest Job First Algorithm

Another approach to job scheduling is the shortest job first algorithm. In Shortest Job First (SJF) algorithm, process from the ready queue that has shortest burst time will execute first.
If two processes are having same burst time and arrival time, then FCFS procedure is used to break the tie.
As an example, consider the following set of processes P1, P2, P3, P4 in table 1-3 and their burst times:

Table 1-3: 4-Perocess Task

| Process | Burst Time |
|---|---|
| P1 | 6 |
| P2 | 8 |
| P3 | 7 |
| P4 | 3 |

Using SJF algorithm, the processes would be scheduled as shown in Table 1-4

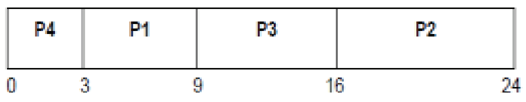

*Fig 1-4: Gantt chart of Process*

Average waiting time = (0 + 3 + 9 + 16) / 4
= 28 / 4 = 7 ms.
Average turnaround = (3 + 9 +16 +24)/4
= 52 / 4 = 13 ms.

## III. PROPOSED ALGORITHM – RRSJF

In this paper the proposed algorithm used Round Robin with shortest Job first scheduling as follows:

i. Allocate data center CPU to every process in round robin fashion, according to given time quantum (say k units) only for one time.
ii. After completion of step 1 process are arranged in increasing order or their remaining burst time in the ready queue. New priorities are assigned according to the remaining burst time of processes; the process with shortest remaining burst time is assigned with highest priority. The processes are executed according to the new priorities based on the remaining bursts time.

The pseudo code of the proposed algorithm is shown below:

### A. Proposed Pseudo code

```
1. First all the processes present in ready
   queue are process in a round robin
   fashion.
   Count → counter set
2. Do {
        if ( TQ >= Min burst (BT) ) then
            BT → TQ
        Counter ++
            }
   While (Ready queue! =NULL)
3. Assign TQ to (1 to n ) process
     For ( 1 to n)
       {
         Pn = $BT_n$-$TQ_n$
           $TQ_n$ → Pn
       }
4. Calculate the remaining burst time of
   the processes
   if ( new process is arrived and BT != 0 )
       Go to Step 1
      Else
      Go to Step 2
   Else
   Go To Step 5
   End if
   End if
   End For
   End Do While
5. Calculate ATT,AWT and CS
   //ATT = Average Turnaround Time,
   AWT = Average Waiting Time, CS =
   No. of Context Switches
6. End
```

## IV. SIMULATION ENVIRONMENT AND RESULT COMPARISON.

### A. Simulation Environment

To analyze and compare the proposed algorithm (*Round Robin with Shortest Job First)* the researcher used Cloudsim tool for the simulations. The cloud environment set up generated was having the following configurations.

Table 1-4: cloud environment setup specifications

| Parameters | capacity |
|---|---|
| VM memory | 512 |
| VM image size | 10,000 |
| Data center – Architecture | X86 |
| Data center – OS | Windows Xp |
| Data center –No. of Machines | 5 |
| Data center – Processor speed | 100MIPS |
| Data center – Number of processors per machine | 5 |

| | |
|---|---|
| Data center – Available BW per Machine | 10,000 |
| Memory per Machine | 2GB |

## B. Comparison and Analysis of Results

During the analysis of the results three cases were considered. In each case the researcher compared the result of the proposed RRSJF method algorithm with Round Robin (RR).

### Case I: Datacenter CPU Burst Time with TQ 5ms

In this case five processes have been defined with Data center CPU burst time and their priorities, these five processes (A, B, C, D, E) are scheduled in round robin fashion and also according to the proposed algorithm as shown in Table1-5.

Table 1-5: 5-Process task Input

| Process No. | Burst Time | Priorities |
|---|---|---|
| A | 22 | 4 |
| B | 18 | 2 |
| C | 9 | 1 |
| D | 10 | 3 |
| E | 4 | 5 |

**Results for Simple Round Robin**

Simple Round Robin does not use priority and five processes has been scheduled using simple Round Robin architecture. The time slice is 5ms.

Table 1-6 shows the Gantt chart when Simple Round Robin (RR) is used:

Table 1-6: Gantt chart for SJF

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | A | B | C | D | A | B | A | B | A | |
| 0 | 5 | 10 | 15 | 20 | 24 | 29 | 34 | 38 | 43 | 48 | 53 | 58 | 61 | 63 |

Number of Context Switches = 13
Average Waiting Time (AWT) = 33.3ms
Average Turnaround Time (ATT) = 48.5ms

**Results for Proposed Algorithm (RRSJF)**

Round Robin with Shortest Job First algorithm proposed consists of two rounds:

**Round 1:** Processes are executed according to their priorities with time quantum given i.e. 5ms.

Table 1-7 shows the Executed CPU burst for first round.

| Process No. | Burst Time | Priorities |
|---|---|---|
| C | 5 | 1 |
| B | 5 | 2 |
| D | 5 | 3 |
| A | 5 | 4 |
| E | 4 | 5 |

Table 1-7: Executed CPU burst for first round.

**Round 2:**

In the second round priorities are assigned to a process based on their remaining CPU Burst Time. The process with least remaining CPU Burst Time is assigned highest priority.

Table 1-8 shows the newly created Executed CPU burst time.

Table 1-8: Remaining Burst Time and Priorities

| Process No. | Remaining Burst Time | Priorities |
|---|---|---|
| C | 4 | 1 |
| D | 5 | 2 |
| B | 13 | 3 |
| A | 7 | 4 |

Table 1-9 shows the Gantt chart when the proposed algorithm (RRSJF) was used:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C | B | D | A | E | C | D | B | A | |
| 0 | 5 | 10 | 15 | 20 | 24 | 28 | 33 | 46 | 63 |

Number of Context Switches = 8
Average Waiting Time (AWT) = 26.3ms
Average Turnaround Time (ATT) = 38. 7ms

### Case II: Datacenter CPU Burst Time with TQ 9ms

Case II- In case II the same problem is considered with varying Quantum Time (QT) of 9ms.

Table 1-10 shows the Gantt chart when Simple Round Robin (RR) is used:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | A | B | D | A | |
| 0 | 9 | 18 | 27 | 36 | 40 | 49 | 58 | 59 | 63 |

Number of Context Switches = 8
Average Waiting Time (AWT) = 38.3ms
Average Turnaround Time (ATT) = 50. 9ms

Table 1-11 shows the Gantt chart when the proposed algorithm (RRSJF) was used:
Number of Context Switches = 7
Average Waiting Time (AWT) = 28.01ms
Average Turnaround Time (ATT) = 40.6ms

## C. Comparison of Results

The performances of the two algorithms were compared by considering context switches (CS), Average Waiting Time (AWT) and Average Turnaround Time (ATT). Fig. 1-1 shows the performances of the two algorithms:

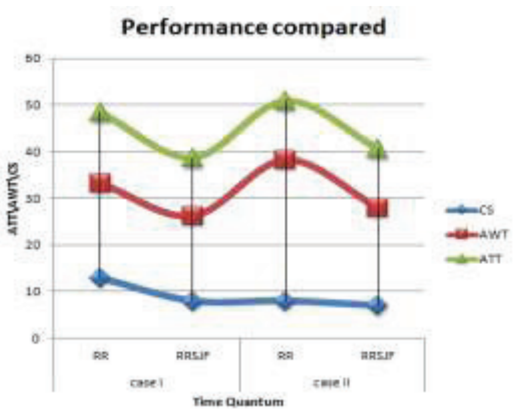

*Fig. 1-1: Performances Compared*

## V. CONCLUSION

The performances of the two algorithms Round Robin and proposed algorithm (RRSJF) were successfully compared and it was concluded that the proposed algorithm is more efficient than the round robin because it has less context switches, less average waiting time and less average turnaround time.

Again, the proposed algorithm reduces problem of starvation as the processes with less remaining CPU burst time are assigned with the higher priorities.

## VI. REFERENCES

[1]. Abbass H.A., “A Single Queen Single Worker Honey-Bees Approach to3-SAT,” The Genetic and Evolutionary Computation ConferenceGECCO2001, San Francisco, USA, 2001.

[2]. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/ (accessed on January 23, 2015)

[3]. Bitam S., BatoucheM., TalbiE-G., “A survey on bee colony algorithms,” 24th IEEE International Parallel and Distributed Processing Symposium, NIDISC Workshop, Atlanta, Georgia, USA, pp. 1-8, 2010.

[4]. Google App Engine. http://code.google.com/appengine/ (accessed on January 23, 2015)

[5]. Komal Mahajan, Ansuyia Makroo, Deepak Dahiya, “Round Robin with Server Affinity: A VM Load Balancing Algorithm for Cloud Based Infrastructure”, http://dx.doi.org/10.3745/JIPS.2013.9.3.379, J Inf Process Syst, Vol.9, No.3, September 2013

[6]. Kun L, Gaochao X, Guangyu Z, Yushuang D, “Cloud Task scheduling based on Load Balancing Ant Colony Optimization” 2011 Sixth Annual China Grid Conference.

[7]. Martin R., David L, Taleb-BendiabA. “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing” 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[8]. Mequanint M, Thomas G.R, ”Wireless Sensor Networks: Scheduling for Measurement and Data Reporting”, August 31, 2005

[9]. Meng X, Lizhen C, Haiyang W, Yanbing B, “A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing” 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications.

[10]. Rodrigo N. C, Rajiv R, Anton B, César A. F, and Rajkumar B “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms”

[11]. Singh et.al, “An Optimized Round Robin Scheduling Algorithm for CPU Scheduling”, International Journal of Computer and Electrical Engineering, Vol. 2, No. 7, 2010, pp. 2383-2385

[12]. Subasish M, Subhadarshini M, Smruti R, “Analysis of Different Variants in Round Robin Algorithms for Load Balancing in Cloud Computing”, International Journal of Computer Applications (0975 – 8887), Volume 69– No.22, May 2013.

[13]. Zehua Z. and Xuejie Z. “A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation”, International Conference on Industrial Mechatronics and Automation, pp-240-243,2010.

[14]. Zhenzhong Z, Haiyan W , Limin X,Li R, “A Statistical based Resource Allocation Scheme in Cloud” 2011 International Conference on Cloud and Service Computing.