Analysis of Different Variants in Round Robin Algorithms for Load Balancing in Cloud Computing

Subasish Mohapatra
Dept. of CSE,
NIT, Rourkela

Subhadarshini Mohanty
Dept. of CSE, ITER, SOA University

K. Smruti Rekha
Dept. of CSE, ITER, SOA University

Pooja Samal
Dept. of CSE,
CET, Bhubaneswar

Pranati Mishra
Dept. of CSE,
CET, Bhubaneswar

ABSTRACT

Cloud computing is the emerging internet-based technology which emphasizes commercial computing. Cloud is a platform providing dynamic pool resources and virtualization. Based on a pay-as-you-go model, it enables hosting of pervasive applications from consumer, scientific, and business domains. To properly manage the resources of the service provider we require balancing the load of the jobs that are submitted to the service provider. Load balancing is required as we don’t want one centralized server’s performance to be degraded. A lot of algorithms have been proposed to do this task. In this paper we have analyzed of various policies utilized with different algorithm for load balancing using a tool called cloud analyst. Basically we have compared different variants of RR for load balancing.

Keywords

Cloud computing, Virtual machine, Cloud service provider, Cloud Analyst, CloudSim, Cloud Service Broker.

1. INTRODUCTION

Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. Cloud computing is attractive to business owners as it eliminates the requirement for users to plan ahead for provisioning, and allows enterprises to start from the small and increase resources only when there is a rise in service demand [1]. Cloud computing can be classified as a new paradigm for the dynamic provisioning of computing services supported by state-of-the-art data centers that usually employ Virtual Machine (VM) technologies for consolidation and environment isolation purposes [2]. Cloud computing delivers an infrastructure, platform, and software (applications) as services that are made available to consumers in a pay-as-you-go model. In industry these services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) respectively [3]. Many computing service providers including Google, Microsoft, Yahoo, and IBM are rapidly deploying data centers in various locations around the world to deliver Cloud computing services. The goal of load balancing is improving the performance by balancing the load among these various resources (network links, central processing units, disk drives…) to achieve optimal resource utilization, maximum throughput, maximum response time, and avoiding overload [4]. Load balancing is a relatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time [5]. Dividing the traffic between servers, data can be sent and received without major delay. Different kinds of algorithms are available that helps traffic loaded between available servers [2B]. A basic example of load balancing in our daily life can be related to websites. Without load balancing, users could experience delays, timeouts and possible long system responses. Loadbalancing solutions usually apply redundant servers which help a better distribution of the communication traffic so that the website availability is conclusively settled [5]. There are many different kinds of load balancing algorithms available, which can be categorized mainly into two groups. The following section will discuss these two main categories of load balancing algorithms.

1.1 Static Algorithms

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each server has been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [6].

1.2 Dynamic Algorithms

Dynamic algorithms designated proper weights on servers and by searching in whole network a lightest server preferred to balance the traffic. However, selecting an appropriate server needed real time communication with the networks, which will lead to extra traffic added on system. In comparison between these two algorithms, although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result [6]. However; dynamic algorithm predicated on query that can be made frequently on servers, but sometimes prevailed traffic will prevent these queries to be answered, and correspondingly more added overhead can be distinguished on network.

The remaining of this paper is organized as follows. Section 2 lists the related work. Section 3 describes the algorithm and defines the environment parameters. Section 4 experiments. Section 5 analyses of the experiment results. Section 6 gives the conclusions.

2. RELATED WORK

RR, MRR and TSPBRR had been used in the task scheduling in cloud computing. The research on them received good results and their efficiency had been proved. There are also many related improved work under study. M. Randles etal have proposed comparison of static and dynamic load balancing algorithms for cloud computing [7] [8]. Load balancing in cloud computing system [9] Ram Prasad Padhy, P Goutam Prasad Rao discussed on basic concepts of Cloud Computing and Load balancing and studied some existing load balancing algorithms, which can be applied to clouds. Jiyan et.al, have proposed aresource allocation mechanism with preemptable task execution which increases the utilization of clouds. They have proposed an adaptive resource allocation algorithm for cloud system with preemptable tasks but their approach does not pertain to cost optimization and time optimization [10]. David B. Stewart and Pradeep K. Khosla proposed the maximum-urgency-first algorithm, which can be used to predictably schedule dynamically changing systems [11]. The scheduling mechanism of the maximum-urgency-first may cause a critical task to fail. The modified maximum urgency first scheduling algorithm by Vahid Salmani, Saman Taghavi Zargar, and Mahmoud Naghibzadeh resolves the above mentioned problem [12]. A. Singh etal have proposed MRR, which is superior than RR and has less waiting response time, usually less pre-emption and context switching thereby reducing the overhead and saving of memory space [13]. C. Yashuwaanth proposed a Modified RR(MRR) algorithm which overcomes the limitations of simple RR [14]. R. Mohanty etal. have taken dynamic time quantum which changes with every round of execution, which results show that PBDRR performs better than algorithm MRR [14] in terms of reducing the number of context switches, average waiting time and average turnaround time [15]. H. Casanova et al. [16] and R. Baraglia et al. [17] proposed the heuristic algorithms to solve the scheduling problem based on the different static data, for example, the execution time and system load. Unfortunately, all information such as execution time and workload cannot be determined in advance of dynamic grid environments. M. Katevenis, S. Sidiropoulos, and C. Courcoubetis have propsed Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. [18] M. Shreedhar and G. Varghese have discussed the efficient fair queuing using deficit round robin. [19] Bhathiya Wickrema etal. all present how Cloud Analyst can be used to model and evaluate a real-world problem through a case study of a social networking application deployed on the cloud. We have illustrated how the simulator can be used to effectively identify overall usage patterns and how such usage patterns affect data centres hosting the application [20] [21].

3. SCHEDULING CRITERIA

For the task scheduling based on RR, MRR and TSPBRR, the criteria include the following:

3.1 Context Switch

A context switch is computing process of storing and restoring state of a CPU so that execution can be resumed from same point at a later time. Context switch are usually computationally intensive, lead to wastage of time, memory, scheduler overhead so much of the design of operating system is to optimize these switches.

3.2 Throughput

Throughput is defined as number of process completed per unit time. Throughput will be slow in round robin scheduling implementation. Context switch and throughput are proportional to each other.

3.3 CPU Utilization

We want to keep the CPU as busy as possible.

3.4 Turnaround Time

Turnaround time is sum of periods spent waiting to get into memory, waiting in ready queue, executing on CPU and doing input output. It should be less.

3.5 Waiting Time

Waiting time is the amount of time a process has been waiting in ready queue. The CPU scheduling algorithm does not affect the amount of time during which a process executes or does input-output; it affects only the amount of time that a process spends waiting in ready queue.

3.6 Response Time

Response time is the time it takes to start responding, not the time it takes to output the response. Large response time is a drawback in round robin architecture as it leads to degradation of system performance.

A good scheduling algorithm must possess following Characteristics:

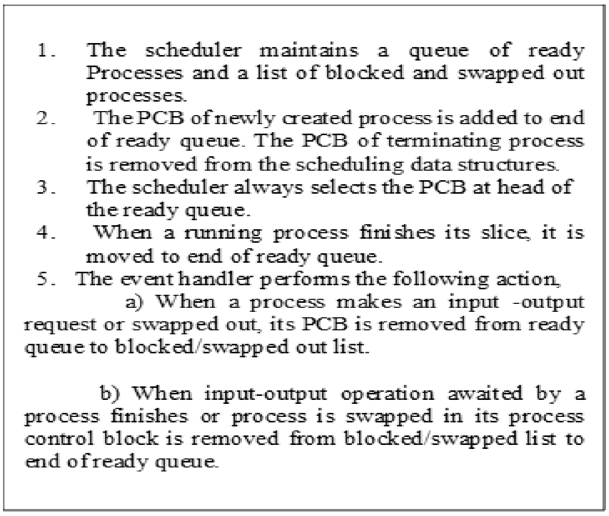
- Minimum context switches.
- Maximum CPU utilization.
- Maximum throughput.
- Minimum turnaround time.
- Minimum waiting time.
- Minimum response time.

4. METHODOLOGY

In this paper different scheduling algorithms such as Round Robin, modified round robin & time slice priority based round robin has been implemented using cloud Sim platform forexperimentation. Different parameters, Avg response time, meeting time are calculated for evaluating the performance.

4.1 Scheduling based on RR

The scheduling process based on RR experimented in this paper is represented in Fig.1.



```
1.  The scheduler maintains a queue of ready Processes and a list of blocked and swapped out processes.
2.  The PCB of newly created process is added to end of ready queue. The PCB of terminating process is removed from the scheduling data structures.
3.  The scheduler always selects the PCB at head of the ready queue.
4.  When a running process finishes its slice, it is moved to end of ready queue.
5.  The event handler performs the following action,
        a) When a process makes an input -output request or swapped out, its PCB is removed from ready queue to blocked/swapped out list.

        b) When input-output operation awaited by a process finishes or process is swapped in its process control block is removed from blocked/swapped list to end of ready queue.
```

Fig.1. Pseudo code for task scheduling process based on RR

4.2 Scheduling based on MRR

In MRR algorithm, Time Slice (ITS) is calculated which allocates based on (range× total no of process (N)) divided by (priority (pr)× total no of process(p)). Range is calculated by (maximum burst time +minimum burst time) divided by the scheduling process based on MRR experimented in this paper is represented in Fig.3.

```
1)  While(readyqueue! =NULL)
    {
        For iton
      {
         Range=maxbt+minbt/2t
         s=(range×N)/(pr×p)
        }end offor
      }endofwhile
```

Fig.2. Pseudo code for task scheduling process based on MRR

TABLE1. Calculation of time slice for MRR

Pr No	Burst Time (BT)	Priority (Pr)	Range (R)	N	P	Time Slice (TS) R×N/ Pr ×P
P1	25	2	15	5	5	8
P2	5	3	15	5	5	5
P3	15	1	15	5	5	15
P4	8	5	15	5	5	3
P5	10	4	15	5	5	4

N= Total no of process
P= Total no of priority
Range = Maximum BT+Minimum BT /2
Range = 5+25/2 =15
Time slice=Range ×N/ Pr×P
TS = 15 × 5/ 2 ×5 =8

4.3 Scheduling based on TimeSlice Priority Based RR.(TSPBRR)

Let 'TQi' is the time quantum in round i. The number of rounds i varies from 1 to n, where value of i increments by 1 after every round till ready queue is not equal to NULL.

```
1   Calculate TS for all the processes present in
    thereadyqueue.
2.While (readyqueue!=NULL)
    {
    For i=1to ndo
    {
            if( i ==1)
            {
              TQi=½TSi
           }
             Else
             {
              TQi=TQ i-1 +½TQi-1
              }
      If(remaining burst time-TQi) <=2
      TQi=remaining burst time
      } End of For
      } End of while
  3.Average waiting time, average turn around time and no.
of contexts witches are calculated.
End
```

Fig.3. Pseudo code for Task scheduling process based on TSPBRR

The scheduling process based on TSPBRR experimented in this paper is represented in Fig. 3

5. EXPERIMENT

Scheduling algorithms described in the previous section were implemented and tested in Cloud Sim. There are 5 users and 3 resources in the mod for users in the model, each of them requests execution of 100 tasks with different length (in MI) between 1 and 50. The initial information of the resource in the model is described in the table 1.

TABLE2. Calculation of time slice for MRR

Pr No	BurstTime(BT)	Priority(Pr)	Range(R)	N	P	TimeSlice(TS)R ×N/ P r × P
P1	25	2	15	5	5	8
P2	5	3	15	5	5	5
P3	15	1	15	5	5	15
P6	8	5	15	5	5	3
P5	10	4	15	5	5	4

TABLE3. Calculation of time slice for proposed algorithm

ProcessNo	TS	ROUNDS				
		1st	2nd	3rd	4th	5th
P1	8	4	6	9	6	0
P2	5	5	0	0	0	0
P3	15	8	7	0	0	0
P4	3	2	3	3	3	0
P5	4	2	2	5	5	0

Algorithm	AverageTAT	AverageWT	CS
MRR	45.2	37.4	12
TSPBRR	42.4	32.4	13

P1	P2	P3	P4	P5	P1

0 8 13 28 31 35 43

P4	P5	P1	P4	P5	P1

43 46 50 58 60 62 63

Gantt chart for MRR

P1	P2	P3	P4	P5	P1	P3
0	4	9	17	19	21	27

P4	P5	P1	P4	P5	P1
34	37	39	48	51	56	62

Grantt chart for TSPBRR

6. ANALYSIS

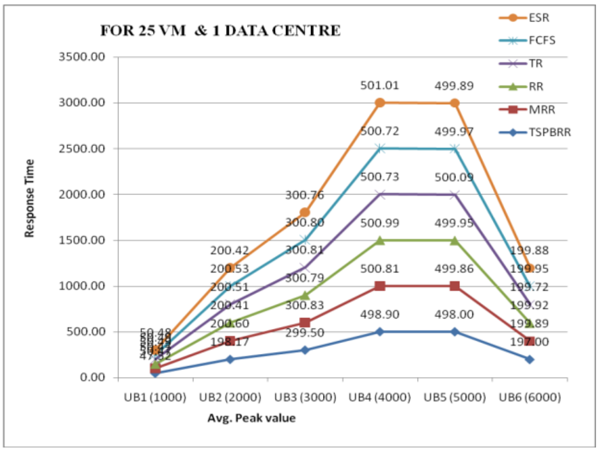


Fig4. Analysis among load balancing policies

Response Time = Fint - Arrt + T Delay (1)

Where, Arrt is the arrival time of user request and Fint is the finish time of user request and the transmission delay can be determined by using the following formulas:

TDelay = T + T (2) latency transfer

Where, T Delay is the transmission delay T latency is the network latency and T transfer is the time taken to transfer the size of data.

7. CONCLUSION

In this paper work it have simulated four different scheduling algorithms along with different variants of round robin algorithm for executing the user request in cloud environment. Each algorithm is observed and their scheduling criteria like average response time, data center service time and total cost of different data centers are found. According to the experiment and analysis round robin algorithm has the best integrate performance. Future work can be based on this algorithm modified and implemented for real time system. Better response time can be expected in this paper if apply some evolutionary algorithms such as PSO, ACO, and ABC instead of classical algorithms.

8. REFERENCES

[1] Qi Zhang, Lu Cheng, Raouf Boutaba, " Cloud Computing : state of –the-art and research challenges", 20th April 2010, Springer, pp. 7-18. .

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A.Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the art ofvirtualization, in: Proceedings of the 19th ACM SymposiumonOperatingSystemsPrinciples,SOSP2003,BoltonLanding,NY, USA, 2003,p.177.

[3] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems, Cloud Computing: Principles, Systems and Applications", Computer Communications and Networks, Chapter 2 , pages 21-46, DOI 10.1007/978-1-84996-241-42,Springer – V erlagLondonLimited, 2010.

[4] A. Khiyaita, H. El Bakkali, M. Zbakh, Dafir El Kettani," Load Balancing Cloud Computing : State of Art", 2010,IEEE.

[5] R. Shimonski. Windows 2000 & Windows Server 2003 Clustering and Load Balancing. Emeryville. McGraw-Hill Professional Publishing, CA, USA (2003), p 2, 2003.

[6] R. X. T. and X. F. Z.. A Load Balancing Strategy Based on the Combination of Static and Dynamic, in DatabaseTechnology and Applications (DBTA), 2010 2nd International Workshop (2010), pp. 1-4.

[7] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010, pp. 551– 556.

[8] A. M. Alakeel, "A Guide to dynamic Load balancing in Distributed Computer Systems", International Journal of Computer Science and Network Security (IJCSNS), Vol. 10, No. 6, June 2010, pages 153-160.

[9] Ram Prasad Padhy (107CS046), PGoutam Prasad Rao (107CS039)."Load balancing in cloud computing system" Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela-769 008, Orissa, India May, 2011.

[10] Jiyin Li, Meikang Qiu, Jain-Wei Niu, YuChen, Zhong Ming "Adaptive Resource Allocation for Preeemptable Jobs in Cloud Systems". IEEEInternational Conference on Intelligent Systems Design and Applications, pp. 31-36, 2010.

[11] David B. Stewart and Pradeep K. Khosla: Real-Time Scheduling of Dynamically Reconfigurable Systems, Proceedings of the IEEE International Conference on Systems Engineering, pp 139-142, August, 1991.

[12] Ahmad, Y.-K. Kwok, M.-Y. Wu, and K. Li, "Experimental Performance Evaluation of Job Scheduling and Processor Allocation Algorithms for Grid Computing on metacomputers," Proc.IEEE 18th Int'l Parallel and Distributed Processing Symp. (IPDPS '04),pp. 170-177, 2004.

[13] A. Singh, P. Goyal, S. Batra : An Optimized Round Robin Scheduling Algorithm for CPU Scheduling, International Journal of Computer and Electrical Engineering (IJCEE), Vol. 2, No. 7,pp 2383-2385, December, 2010.

[14] C. Yaashuwanth and R. Ramesh, "Design of Real Time Scheduler Simulator and Development of Modified Round Robin Architecture for Real Time System", International Journal of Computer and Electrical Engineering (IJCEE), Vol. 10, No. 3, pp 43-47, March, 2010.

[15] R. Mohanty etal, "Priority Based Dynamic Round Robin (PBDRR) Algorithm with Intelligent Time Slice for Soft Real Time Systems", International Journal of Computer and Eletrical Engineering (IJCEE), Vol. 2, No. 2,pp 46-

50,Febuary,2011.

[16] H. Casanova, A. Legrand, D. Zagorodnov and F. Berman, "Heuristics for scheduling parameter Journal of Theoretical and Applied Information Technology © 2005 - 2009 JATIT. All rights reserved. www.jatit.org 115 sweep applications in Grid environments", in Heterogeneous Computing Workshop", 2000,IEEE Computer Society Press, 2000, pp. 349–363.

[17] R. Baraglia, R. Ferrini, and P. Ritrovato, "Astatic mapping heuristics to map parallel applications to heterogeneous computing systems", Research articles. Concurrency and Computation : Practice and Experience,17(13):1579–1605, 2005.

[18] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," IEEE J. Sel. Areas Commun., Vol. 9, No. 8, pp. 1265–1279, 1991.

[19] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," IEEE Trans. Netw., Vol., 4, No. 3, pp. 375–385, 1996.

[20] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities," Proc. of the 7th High Performance Computing and Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.

[21] Bhathiya Wickremasinghe1, Rodrigo N. Calheiros2, and Rajkumar Buyya, "CloudAnalyst: A CloudSim-based Visual Modeller for Analyzing Cloud Computing Environments and Applications", 2010 ,IEEE.

[22] Silberchatz, Galvin and Gagne, 2003. Operating systems concepts.

[23] www- 03.ibm.com/press/us/en/pressrelease/22613.

[24] http://www.amazon.com/gp/browse.html?node=2015900 11