Service Delivery Improvement for the Cloud Service Providers and Customers

Sourav Banerjee
Dept of Computer Sc and Engineering
Kalyani Govt Engg College
Kalyani, Nadia, West Bengal, 741235, India

Dipunsu Mandal
System programmer, CIRM department
University of Kalyani
Kalyani, Nadia, West Bengal, 741235, India

Mainak Adhikari
Dept of Computer Sc and Engineering
University of Kalyani
Kalyani, Nadia, West Bengal, 741235, India

Utpal Biswas
Dept of Computer Sc and Engineering
University of Kalyani
Kalyani, Nadia, West Bengal, 741235, India

ABSTRACT

Cloud computing is one of the newest technology. Today lots of business organizations and educational institutions using Cloud environment, but one of the most important thing is to improve the Quality of Service (QoS) of the Cloud service provider (CSP), there are so many parameters affecting QoS, waiting time is one of them. If waiting time can be optimized QoS may get improved. QoS improvement means how fast it shares the resources for the client machines. There are lots of algorithms available but the waiting time is the important factor in this case. The algorithm is said to be best if it requires very less waiting time to share resources to its client machines. In this paper we propose an algorithm which requires optimum waiting time as well as it does not suffer from starvation in comparison to other algorithms. Here we implement the algorithm in M/M/S queueing model where k number of clients send request to the job scheduler and this scheduler selects resources from the resource pool and using those algorithms job scheduler give permission to the Cloud Users (CU) how they use the resources in less waiting time. In Cloud environment there are numbers of resources present inside CSP in Resource Pool module and there are number of clients send request to the Cloud Service Provider (CSP) and a dedicated Job Scheduler, inside CSP handles those resources in a very efficient manner. In cloud environment cloud user sends the request to cloud service provider which selects the resources for the user using scheduling algorithm. In this paper we describe how the job scheduler handles those resources for the users using our proposed algorithm which is Improved Round Robin scheduling algorithm (IRRA).

Keywords: Cloud computing, Queuing model, Cloud service provider, Cloud user, Improved Round Robin Algorithm, Quality of Service.

1. INTRODUCTION

Cloud computing [1,2,3,4] is the process of delivering computing as a service rather than a product, whereby shared resources, software, and information are provided to users and other devices as a utility over the network (typically the Internet). Basically it is a Service Oriented Architecture (SOA) by nature. Cloud environment allows users to use applications without installation and access their personal files at any computer with Internet access. Cloud computing provides computation, software applications, data access, data management and storage resources without requiring cloud users to know the location and other details of the computing infrastructure. End users access cloud based applications through a web browser or a light weight desktop or mobile app while the business software and data are stored on resources at a remote location. Cloud application providers [6,7] strive to give the same or better service and performance than if the software programs were installed locally on end-user machines. Cloud environment [5] has now been using in lot of fields like in IT industries as well as in other industries. So the quality of service needs to be improved. There are lots of algorithms available to give services to the client side, but we must need to choose the best one among them which give better performance. Now in case of better performance means how fast and fluently cloud resources give the permission to the client machines to handle the resources after sending the request from the client side. So here we concentrate to reduce the waiting time of the overall system. The algorithm whose waiting time is optimum will lead to the system stabilization and QoS maximization [8] and which carries suitable disadvantages must be used in Cloud Service Provider zone.

Now we briefly discuss about our model. Our proposed model mainly comprises of two parts- CU and CSP. Cloud user mainly sends request to the cloud service provider for the resources. CSP first stores the request in to the waiting queue and then perform the scheduling operation using job scheduler [9,13]. Job scheduler is mainly responsible for sharing the resources among the Cloud Users within a suitable time period. So the algorithm which may provide service as well as the resource allocation in optimal time period without causing starvation, is chosen as the best one for the total system. Our proposed IRRA algorithm mainly being executed in job scheduling module of CSP and it shares resources in optimal waiting time. In the next section we discuss our proposed algorithm for job allocation operation.

2. PROPOSED WORK

In this section we discuss our proposed algorithm related to resource allocation as well as scheduling mechanism. But before starting our discussion we first elaborate about queueing model where we apply that algorithm.

Queueing model [5] is used to approximate a real queueing situation or system, so the queueing behavior can be analyzed mathematically. Queueing models allow a number of useful steady state performance measures to be determined. Concept of queueing theory concept comes from Kendall's notation. Kendall's notation is a standard notation for classifying queueing systems into different types. Kendall's notation mainly described by the notation A/B/C/D/E. Where **A-** Distribution of interarrival times of customers; **B-** Distribution

of service times; **C-** Number of servers; **D-** Maximum total number of customers which can be accommodated in system,i.e. system capacity; **E-** Queuing discipline [5]. Let's take an example where- M/M/m/K/N- this would describe a queuing system with an exponential distribution for the interarrival times of customers and the service times of customers, m servers, a maximum of K customers in the queueing system at once, and potential customers in the calling population. There are lots of models are available like M/M/1, M/M/2, etc. Here we describe our algorithm in M/M/1 model. An **M/M/1 queue** represents the queue length in a system having a single server, where arrivals are determined by a Poisson process and job service times have a distribution. An M/M/1 queue is a stochastic process whose state space is the set {0, 1, 2, 3...} where the value corresponds to the number of customers in the system, including any pattern currently in service. Mathematical formula of M/M/1 queueing model is shown in figure.2 and diagrammatically it is shown in figure 3.

$$Q = \begin{pmatrix} -\lambda & \lambda & & \\ \mu & -(\mu+\lambda) & \lambda & \\ & \mu & -(\mu+\lambda) & \lambda \\ & & \mu & -(\mu+\lambda) \quad \lambda \end{pmatrix}$$

Fig1: Mathematical formulation of M/M/1 Queueing model

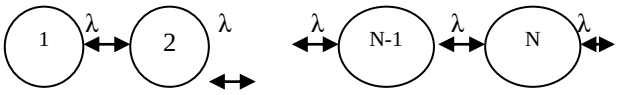


Figure 2: Diagram of M/M/1 Queueing model

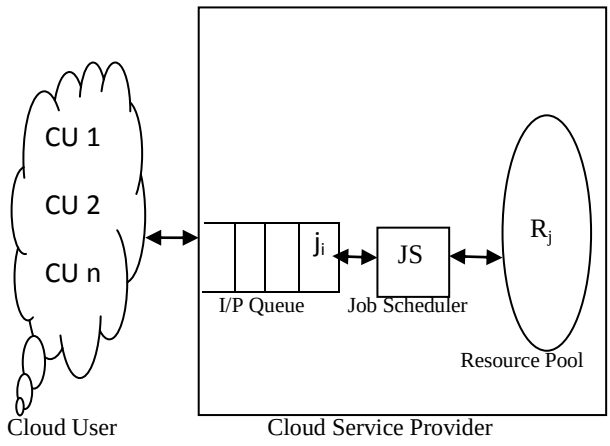


Figure 3: Cloud queuing model

For easy description we propose our scheduling algorithm based on M/M/1 queueing model.

In Figure 3 we design our propose model where we mainly apply our algorithm. In this model n number of cloud users send request to the cloud service provider (CSP). CSP then stores the request to the queue and then the operation must be performed by the job scheduler. Job scheduler, empowered with IRRA is mainly responsible for choosing the resources for the cloud users within suitable time period as well as optimal waiting time.

Our proposed scheduling algorithm is Improved Round Robin algorithm (IRRA) and the model is M/M/1/∞/IRRA. In this model the administrator must set the time quantum before starting the algorithm and the administrator has the right to change or set the time quantum value.

First we describe how our algorithm actually works in the following steps-

Step 1. Set a time quantum (which may vary depending upon scheduling capacity) and then start the execution of jobs.

Step 2. Use one comparator to find out the shortest burst time job and stamp it with highest priority, follow this procedure for the rest of the jobs

Step 3. Execute first highest priority process to the time quantum, then execute next highest priority process and so on until all processes being executed and continue until termination.

Step 4. Finally calculate waiting time of every Jobs and average waiting time.

Here we consider an example for better understanding.

Table 1. Process table

Jobs	Burst Time
J_1	20
J_2	12
J_3	08
J_4	16
J_5	04

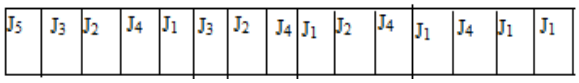
Consider the table 1 where we consider five Jobs and the burst time. Now we use the comparator to find out the minimum time quantum of the Job and assign it as priority1 and continue the process which declares in table 2.

Table 2. Process table

Jobs	Burst Time	Priority
J_1	20	5
J_2	12	3
J_3	08	2
J_4	16	4
J_5	04	1

Next we make the Gantt chart of those processes according to the algorithm. Now Job J_5's burst time is minimum so it may execute first, then J_3 and so on until all the Jobs are executed serially according to their schedule.

Gantt Chart-



0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60

Next we find out the waiting time of each process and then average waiting time of all processes which are shown below-

Waiting time of Job J_1-> (16+ (32-20) + (44-36) + (52-48) = 40

Waiting time of Job J_2-> (8+ (24-12) + (36-28)) = 28

Waiting time of Job J_3-> (4+ (20-8)) = 16

Waiting time of Job J_4-> (12+ (28-16) + (40-32) = 32

Waiting time of Job J_5-> 0

Average waiting time of Jobs => (40+28+16+32+0) / 5 => 116 / 5 => 23.2

So the waiting time of IRRA is very less which we will show in next section and it has lots of other advantages are available compare to other scheduling algorithms' like it will never suffer from starvation and for that reason QoS will be improved which describe in next section.

3. PERFORMANCE ANALYSIS

In this section we describe other two important scheduling algorithms of M/M/1 model and also describe how our propose IRRA algorithm much better than the other algorithms.

Now we first describe Shortest Job First Scheduling algorithm. The procedure of execution of this algorithm in brief is given below-

Step 1. First use one comparator to find out the minimum burst time Job and it execute first, then next lowest burst time Job execute and so on until all the Jobs are executed fully.

Step 2. Then find out the waiting time of all the Jobs and finally find out average waiting time of all the Jobs.

Now again we consider the table 1 and we see that Job J_5 has lowest burst time, so that Job execute first, then Job J_3 has lowest burst time, so it execute next. In this way all the Jobs are finished their execution. Now we first design the Gantt chart of SJF algorithm and find out average waiting time of all the processes.

Gantt chart:-

0 4 12 24 40 60

Waiting time of Job J_1-> 40

Waiting time of Job J_2-> 12

Waiting time of Job J_3-> 4

Waiting time of Job J_4-> 24

Waiting time of Job J_5-> 0

Average waiting time of Jobs => (0+4+12+24+40) / 5 => 80 / 5 => 16

Advantage- SJF scheduling algorithm's advantage is that the waiting time of all the Jobs are very low in comparison to the other algorithms.

Disadvantages- Generally, SJF scheduling algorithm has two disadvantages. One of them is that it is non preemptive scheduling and second one is starvation problem that may put a great impact in QoS improvement procedure. Starvation problem means one long burst time Job must wait long time or starve for other short term Jobs. So this scheduling is not very useful.

Now we consider another scheduling algorithm which is Round Robin algoritham.The procedure of execution of this algorithm is given below-

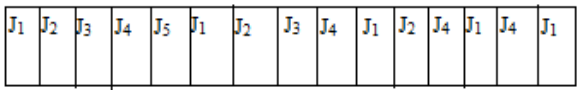
Step 1. First we set one time quantum (which may very) and then perform the execution of the Jobs.

Step 2. Then first Job start its execution up to this time quantum, then execute next Job up to the time quantum and so on until all the Jobs finish its execution.

Step 3. Finally we calculate the waiting times of each algorithm and finally we calculate the average waiting time of all the Jobs.

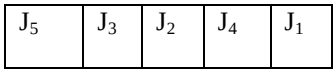
Now again we consider the table 1 and first we set the time quantum (here we set it as 4) and then start the execution of the Jobs. Now first Job J_1 is executed first and it execute till 4 unit, then Job J_2 execute 4 units and so on until all processes execute fully. Now we first design Gantt chart and find out the average waiting time of those Jobs.

Gantt chart:-



0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60

Waiting time of Job J_1-> (0+ (20-4) + (36-24) + (48-40) + (56-52) = 40



Waiting time of Job J_2-> (4+ (24-8) + (40-28) = 32

Waiting time of Job J_3-> (8+ (28-12)) = 24

Waiting time of Job J_4-> (12+ (32-16) + (44-36)) = 36

Waiting time of Job J_5-> 16

Average waiting time of the Jobs => (40+32+24+36+16) / 5 => 148 / 5 => 29.5

Advantage- Round Robin scheduling algorithm avoided the starvation problem means any long time executable Job must not need to wait infinite time for other short term Jobs, which is one of the best property of scheduling of M/M/1 model.

Disadvantage-The main disadvantage of Round Robin scheduling is that the waiting times of the Jobs are higher comparable to the other algorithms.

IRRA is better than SJF:

SJF scheduling algorithm's main problem is starvation which is improved using IRRA.This is also examined using other algorithms like FCFS, etc.

IRRA is better than RR:

RR scheduling algorithm's main problem is that the average waiting time is maximum in comparison to the other algorithms which is improved by IRRA algorithm.

Advantages of Improved Round Robin Algorithm:

1. IRRA scheduling algorithm reduces average waiting time of the Jobs
2. IRRA scheduling algorithm can overcome the problem of starvation.

4. CONCLUSION AND FUTURE WORK

Last two sections we described about Improved Round Robin algorithm and also describe how it is efficient than other algorithms. We may conclude that the IRRA is one of the most important and improved algorithm in implementing M/M/1 model. IRRA also very simple and less complex algorithm. Using IRRA algorithm job scheduler can easily handle all requests of the client. If more than one client send request for the resource then job scheduler give the chance of all clients to use the resource one after another. But job scheduler does not give permission one client to use the resource all its life time. It gives permission to all the clients to use the resource same amount of time, and if some clients require more than that time quantum then job scheduler first put them in the waiting queue and after some time latter it gives the permission to those clients to execute again and starvation problem must be reduced. In starvation problem job scheduler does not give permission the long term executable process to use the resource, so starvation must arise which is avoided using this algorithm. So IRRA algorithm is very useful for M/M/1 model. So if we implement this algorithm inside cloud service provider then it may help overall service delivery improvement keeping each and every aspect of a stable system.

Here we use the Improved Round Robin algorithm in M/M/1 queueing model for the sake of simplicity where we use only one server. But we are trying to implement this algorithm in job scheduler in such a way that it may share more than one resource. We are going to utilize this algorithm in real world with cloud network domain.

5. REFERENCES

[1] Kaiqi Xiong, Harry Perros "Service Performance and Analysis in Cloud Computing*"* 978-0-7695-3708-5/09 $25.00 © 2009 IEEE page- 693-700

[2] Borja Sotomayor, Rubén S. Montero and Ignacio M. Llorente, Ian Foster , "Virtual Infrastructure Management in Private and Hybrid Clouds" 1089-7801/09/$26.00 © 2009 IEEE

[3] Hongqi Li, Zhuang Wu "Research on Distributed Architecture Based on SOA" 978-0-7695-3522-7/09 $25.00 © 2009 IEEE 670-674

[4] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the Clouds: "A Berkeley View of Cloud computing". Technical Report No. UCB/EECS-2009-28, University of California at Berkley, USA, Feb. 10, 2009

[5] Hock, N.C., "Queueing Modelling Fundamentals". JOHN WILEY&SONS, 1997

[6] Francesco Maria Aymerich, Gianni Fenu1, Simone Surcis "An Approach to a Cloud Computing Network" 978-1-4244-2624-9/08/$25.00 ©2008 IEEE 113 page 113-118

[7] by Xu Lei, Xin Zhe, Ma Shaowu, Tang Xiongyan. "Cloud Computing and Services Platform Construction of Telecom Operator" *Broadband Network & Multimedia* Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on Digital Object Identifier, pp. 864 – 867.

[8] Kaiqi Xiong and Harry Perros, "Service Performance and Analysis in Cloud Computing" , 2009 Congress on **Services –I**

[9] Luqun Li " An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers" Third International Conference on Multimedia and Ubiquitous Engineering page-295-299, 2009

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

[11] Abraham Silberschatz, Peter Galvin, and Greg Gagne "Operating System Concepts", John Wiley & Sons, 2009.