Job Resource Ratio Based Priority Driven Scheduling in Cloud Computing

Pinal Salot[1] Purnima Gandhi[2]
[1,2] Alpha College of Engineering, GTU, Gujarat, India

Abstract— Cloud Computing is an emerging technology in the area of parallel and distributed computing. Clouds consist of a collection of virtualized resources, which include both computational and storage facilities that can be provisioned on demand, depending on the users' needs. Job scheduling is one of the major activities performed in all the computing environments. Cloud computing is one the upcoming latest technology which is developing drastically. To efficiently increase the working of cloud computing environments, job scheduling is one the tasks performed in order to gain maximum profit. In this paper we proposed a new scheduling algorithm based on priority and that priority is based on ratio of job and resource. To calculate priority of job we use analytical hierarchy process. In this paper we also compare result with other algorithm like First come first serve and round robin algorithms.

Key Words: Priority based scheduling, Job scheduling, cloud computing, optimization algorithm.

I. INTRODUCTION

There has been various types of scheduling algorithm exist in distributed computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR). By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is suitable example of On-line mode heuristic scheduling algorithm.

1) *First Come First Serve Algorithm:*
Job in the queue which come first is served. This algorithm is simple and fast.

2) *Round Robin algorithm:*
In the round robin scheduling, processes are dispatched in a FIFO manner but are given a limited amount of CPU time called a time-slice or a quantum. If a process does not complete before its CPU-time expires, the CPU is preempted and given to the next process waiting in a queue. The preempted process is then placed at the back of the ready list.

3) *Most fit task scheduling algorithm:*
In this algorithm task which fit best in queue are executed first. This algorithm has high failure ratio.

4) *Priority scheduling algorithm:*
The basic idea is straightforward: each process is assigned a priority, and priority is allowed to run. Equal-Priority processes are scheduled in FCFS order. The shortest-Job-First (SJF) algorithm is a special case of general priority scheduling algorithm. An SJF algorithm is simply a priority algorithm where the priority is the inverse of the (predicted) next CPU burst. That is, the longer the CPU burst, the lower the priority and vice versa. Priority can be defined either internally or externally. Internally defined priorities use some measurable quantities or qualities to compute priority of a process.

A. *Scheduling Process*

Scheduling process in cloud can be generalized into three stages namely–

– Resource discovering and filtering – Datacentre Broker discovers the resources present in the network system and collects status information related to them.

– Resource selection – Target resource is selected based on certain parameters of task and resource. This is deciding stage.

– Task submission -Task is submitted to resource selected.

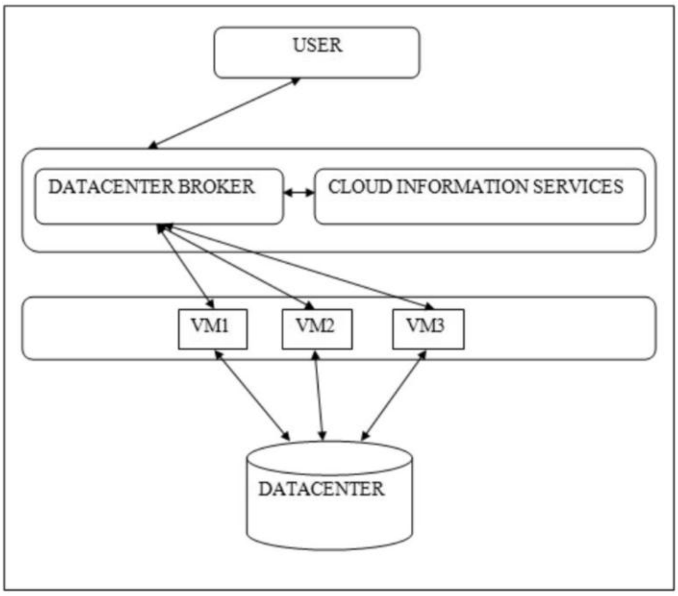

Fig. 1. Scheduling Process

II. RELATED WORK

The Following scheduling algorithms are currently prevalent in clouds.

A. *Resource-Aware-Scheduling algorithm (RASA):*

Saeed Parsa and Reza Entezari-Maleki [2] proposed a new task scheduling algorithm RASA. It is composed of two traditional scheduling algorithms; Max-min and Min-min. RASA uses the advantages of Max-min and Min-min algorithms and covers their disadvantages. Though the deadline of each task, arriving rate of the tasks, cost of the task execution on each of the resource, cost of the communication are not considered. The experimental results show that RASA is outperforms the existing scheduling algorithms in large scale distributed systems.

B. *RSDC (Reliable Scheduling Distributed in Cloud computing):*

Arash Ghorbannia Delavar,Mahdi Javanmard , Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi[1] proposed a reliable scheduling algorithm in cloud computing environment. In this algorithm major job is divided to sub jobs. In order to balance the jobs the request and acknowledge time are calculated separately. The scheduling of each job is done by calculating the request and acknowledges time in the form of a shared job. So that efficiency of the system is increased.

C. *An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment:*

Dr. M. Dakshayini, Dr. H. S. Guruprasad [3] proposed a new scheduling algorithm based on priority and admission control scheme. In this algorithm priority is assigned to each admitted queue. Admission of each queue is decided by calculating tolerable delay and service cost. Advantage of this algorithm is that this policy with the proposed cloud architecture has achieved very high (99%) service completion rate with guaranteed QoS. As this policy provides the highest precedence for highly paid user service-requests, overall servicing cost for the cloud also increases.

D. *A Priority based Job Scheduling Algorithm in Cloud Computing:*

Shamsollah Ghanbari, Mohamed Othman proposed a new scheduling algorithm based on multi – criteria and multi - decision priority driven scheduling algorithm. This scheduling algorithm consist of three level of scheduling: object level, attribute level and alternate level. In this algorithm priority can be set by job resource ratio. Then priority vector can be compared with each queue. This algorithm has higher throughput and less finish time.

E. *Extended Max-Min Scheduling Using Petri Net and Load Balancing:*

El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy[5] has proposed a new algorithm based on impact of RASA algorithm. Improved Max-min algorithm is based on the expected execution time instead of complete time as a selection basis. Petri nets are used to model the concurrent behavior of distributed systems. Max-min demonstrates achieving schedules with comparable lower makespan rather than RASA and original Max-min.

F. *An Optimistic Differentiated Job Scheduling System for Cloud Computing:*

Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagar, Juhi Bansiwal[6] has proposed a differentiated scheduling algorithm with non-preemptive priority queuing model for activities performed by cloud user in the cloud computing environment. In this approach one web application is created to do some activity like one of the file uploading and downloading then there is need of efficient job scheduling algorithm. The QoS requirements of the cloud computing user and the maximum profits of the cloud computing service provider are achieved with this algorithm.

G. *Improved Cost-Based Algorithm for Task Scheduling:*

Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam [7] proposed an improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in cloud. The improvisation of traditional activity based costing is proposed by new task scheduling strategy for cloud environment where there may be no relation between the overhead application base and the way that different tasks cause overhead cost of resources in cloud. This scheduling algorithm divides all user tasks depending on priority of each task into three different lists. This scheduling algorithm measures both resource cost and computation performance, it also Improves the computation/communication ratio.

H. *Performance and Cost evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling:*

Ioannis A. Moschakis and Helen D. Karatza has proposed a gang scheduling algorithm with job migration and starvation handling in which scheduling parallel jobs, already applied in the areas of Grid and Cluster computing. The number of Virtual Machines (VMs) available at any moment is dynamic and scales according to the demands of the jobs being serviced. The aforementioned model is studied through simulation in order to analyze the performance and overall cost of Gang Scheduling with migrations and starvation handling. Results highlight that this scheduling strategy can be effectively deployed on Clouds, and that cloud platforms can be viable for HPC or high performance enterprise applications.

III. ANALYTICAL HIERARCHY PROCESS

In this section we explain the Analytical Hierarchy Process briefly. It is a multi-criteria decision-making (MCDM) and multi-attribute decision-making (MCDM) model. Basically architecture of AHP is consisted of three levels which are objective level, attributes level and alternatives level respectively. The foundation of AHP is comparison matrix which can be shown as Eq. (1).

$$\text{aij} = \begin{cases} \frac{1}{aji} & i \neq j \\ 1 & i = j \end{cases} \tag{1}$$

Each entry in the matrix *A* is positive ( ). Also *A* is a square matrix ( ). For any arbitrary comparison matrix such as *A* we can compute a vector of weights such as associated with *A*. Relationship between *A* and can be shown as Eq.(2)

$$A = \begin{cases} aij = \frac{wi}{wj} & i \neq j \\ 1 & i = j \end{cases} \quad (2)$$

An essential step in AHP is to calculate vector of weights. Vector of weights can be computed through the Eq. (3)

$$A_w = \gamma_{max}\omega \quad (3)$$

Actually Eq. (3) γ_{max} is denoted the principal eigenvalue of A and ω is denoted the corresponding eigenvector. If A is absolutely consistent then $\gamma_{max} = 1$. In this case A will be consistent. consistency ratio (CR) can be defined as Eq. (4).

$$\text{CR} = \frac{CI}{RI} \quad (4)$$

In Eq. (4), *RI* is denoted the random index(*RI*), *RI* can be calculated randomly based on rank of comparison matrix.

IV. PROPOSED ALGORITHM

Suppose that $\Psi = \{J_1, J_2 \dots \dots J_m\}$ is a set of jobs that request resources in a cloud environment. Also we assume that $C = \{\ _1, C_2 \dots \dots C_m\}$ is a set of resources available in cloud environment( ). Each job requests a resource with a determined priority. The priority of each job is compared with other jobs separately.

Assume that q^1, q^2, q^d are d comparison matrixes of jobs which are created according to priority of resource accessibilities. For each of comparison matrixes we should compute a priority vector (vector of weights). The priority vector can be obtained by solving Eq. (3). There are several methods for calculating priority vector [1- 5]. An iterative method for solving Eq. (3) can be found in [6].That method solves the Eq. (3) by using numerical methods. Using iterative methods can calculate priority vector (vector of weights) without concerning about consistency problems. In this case we can define a normal matrix of jobs level as Eq. (5).

$$\Delta = [\ _1,\ _2, \dots . \omega_d] \quad (5)$$

It is clear that Δ is a matrix with m (the number of jobs) rows and d (the number of resources) columns. The next step of the proposed algorithm is to make a comparison matrix for resources according to priorities. This matrix determines that which resource has higher priority than others based on decision maker(s). In this case, we will have a matrix with d rows and d columns. Assume that S is comparison matrix for resources level, thus will be defined as priority vector of ***S***. The next step of the algorithm is to calculate *PVS* which is denoted as priority vector of scheduling jobs. *PVS* can be calculated by Eq. (6). Finally, we choose the maximum element of *PVS*, then select corresponding element of Ψ in order to allocate a suitable resource.

$$\text{PVS} = \Delta ._{\gamma} \quad (6)$$

– *Steps of algorithm :*

1) J = set of Jobs
2) C = Set of Resources
3) Make consistent comparison matrix according to priority of resources
4) Compute priority vector for all matrix
5) Make matrix with priority vector name it Δ.
6) For C compute consistent matrix .
7) Compute priority vector and name it Y.
8) Compute PVS. PVS = Δ. Y.
9) Choose a job according to priority value
10) Update list of job according to priority.

V. SIMULATION RESULT

The CloudSim toolkit is used to simulate heterogeneous resource environment and the communication environment [4]. CloudSim(2.1.1) simulator is used to verify the correctness of proposed algorithm. The experiments are performed with Sequential assignment which is default in CloudSim and the proposed algorithm. The jobs arrival is Uniformly Randomly Distributed to get generalized scenario. The configuration of datacenter created is as shown below - Number of processing elements – 1 Number of hosts – 1.

RAM(MB)	2048
Processing Power(MIPS)	1000
VM scheduling	Time Shared with Priority

Table 1. Resources of Virtual Machines

The configuration of Virtual Machines used in this experiment is as shown in Table 2.

Virtual Machines	3
RAM (MB)	256
Processing Power	1000
Processing Element	**1**

Table 2. Configuration of Virtual Machines

Now we will compare result with existing algorithm with this configuration. It will take different finish time with different algorithm. It can be shown by figure 2.

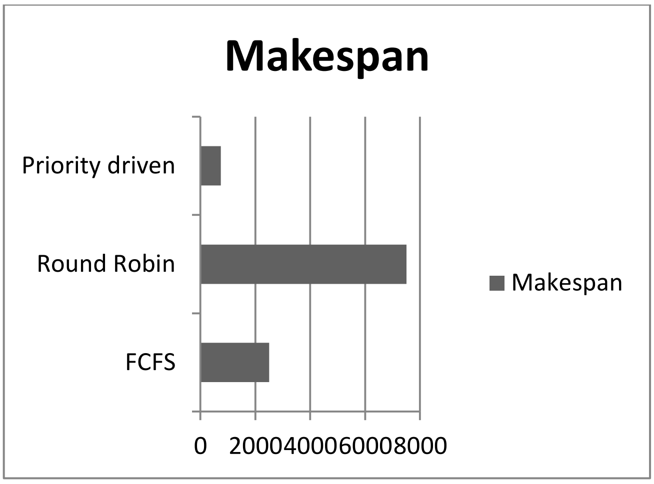

Fig. 2 : Comparison with other algorithm

VI. NUMERIC EXAMPLE

In our example we have 3 resources and 3 jobs. Priority of each job according to resources can be shown in table.

Priority	Resource 1	Resource 2	Resource 3	Priority Vector
Resource 1	1	0.25	3	0.213
Resource 2	4	1	7	0.701
Resource 3	1/3	1/7	1	0.085

Table 3: Priority of resources

Resource 1	Job 1	Job 2	Job 3	Priority Vector
Job1	1	4	3	0.632
Job 2	1/4	1	½	0.137
Job 3	1/3	2	1	0.239

Table 4: Priority of resource 1 according to jobs

Resource 2	Job 1	Job 2	Job 3	Priority Vector
Job1	1	5	3	0.619
Job 2	1/5	1	1/4	0.096
Job 3	1/3	4	1	0.284

Table 5: Priority of resource 2

Resource 3	Job 1	Job 2	Job 3	Priority Vector
Job1	1	1/3	1/7	0.093
Job 2	3	1	½	0.292
Job 3	7	2	1	0.615

Table 6: Priority of resource 3

Comparison matrix can be shown as follows :

$$\begin{bmatrix} 1 & \frac{1}{4} & 3 \\ 4 & 1 & 7 \\ \frac{1}{3} & \frac{1}{7} & 1 \end{bmatrix}$$

For other comparison matrixes in table 2 to 4 we should investigate consistency condition.

According to table 2 to 4 and step 3-5 of proposed algorithm we have:

$$\Delta = \begin{bmatrix} 0.632 & 0.619 & 0.093 \\ 0.137 & 0.096 & 0.292 \\ 0.239 & 0.284 & 0.615 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0.213 \\ 0.701 \\ 0.085 \end{bmatrix}$$

Thus,

$$PVS = \Delta \cdot Y = \begin{bmatrix} 0.575 \\ 0.122 \\ 0.303 \end{bmatrix}$$

Job 1 has highest priority then job 3 has 2nd priority and job 2 has less priority.

VII. CONCLUSION AND FUTURE WORK

Scheduling is one of the most important tasks in cloud computing environment. In this paper we have analyze various scheduling algorithm and tabulated various parameter. Priority is an important issue of job scheduling in cloud environments. In this paper we have proposed a priority based job scheduling algorithm which can be applied in cloud environments. We have named it "PJSC". We can get less finish time than any other algorithm. Job which has maximum priority can be served fast.

The proposed algorithm can be further improved by considering following suggestions –

– By using this algorithm we can also minimize cost also this can be future work.

– Analytic hierarchy process is only method so that we can choose one of the best alternatives from others. Complexity is also future work for proposed algorithm.

REFERENCES

[1] Arash Ghorbannia Delavar,Mahdi Javanmard , Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi "RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012
[2] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.
[3] Dr. M. Dakshayini, Dr. H. S. Guruprasad "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment" International Journal of Computer Applications (0975 – 8887) Volume 32– No.9, October 2011
[4] Shamsollah Ghanbari, Mohamed Othman "A Priority based Job Scheduling Algorithm in Cloud Computing" International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012)
[5] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy "Extended Max-Min Scheduling Using Petri Net and Load Balancing" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012
[6] Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagar, Juhi Bansiwal " An Optimistic Differentiated Job Scheduling System for Cloud Computing" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.1212-1214
[7] Mrs.S.Selvarani1; Dr.G.Sudha Sadhasivam, improved cost-based algorithm for task scheduling in Cloud computing ,IEEE 2010.

[8] Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. By Rajkumar Buyya.
[9] P. Brucker. Scheduling Algorithms. Springer, 4th edition, 2004.
[10] Cloud computing - Wikipedia, the free encyclopedia.htm
[11] Rajkumar Buyya, Chee Shin Yeo and Srikumar Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", The 10th IEEE International Conference on High Performance Computing and Communications, IEEE Computer Society, 2008, pages 5-13.
[12] Armbrust, M., A. Fox, R. Griffith, D. Anthony and Joseph *et al*., 2008. Above the Clouds: A Berkeley View of Cloud Computing. University of California, Berkeley.
[13] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Comp. Syst., vol. 25, no. 6, pp. 599–616, 2009.