ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

Dynamic Steady State Distribution in Cloud

S. Selvi, Dr B. Kalaavathi

Abstract— Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources like networks, servers, storage, applications, and services. A Cloud Computing Server will handle the Cloud computing users jobs which are different from user to user and with different QOS requirements. Job scheduling system is responsible to select the best suitable resources by taking some static and dynamic parameters restrictions of Users jobs into consideration. The main objective is to determine a proper sequence where tasks are executed while obeying to some (transaction logic) constraints as centralized or decentralized, static or dynamic, or a hybrid. In general, dynamically scheduling the jobs and distributing to the cloud users is a very high overhead process since it has to meet the QOS requirements of the users even in a continuous change in the traffic rate, thereby making the maximum profit for cloud computing service provider. This can be achieved by credit manager analyzer which handles the large amount of incoming jobs effectively by distributing the required resources from the job pool in a steady state manner and hence obtaining an optimum queue size dynamically.

Keywords- **Cloud Computing, Job pool, Transaction logic, QoS, credit manage.**

I. INTRODUCTION

Cloud computing is one of the emerging technology in internet. During the job scheduling system, dynamic steady state distribution of jobs is one of the significant and serious issue to be solved in the Cloud Computing environment. This distribution mainly involved in the perfect job scheduling of the user's job which varies according to their respective Qos requirements demands. From the side of cloud computing user's they would think how the cloud computing resource will satisfy their demands and the money they would pay for the resources. Cloud computing user's may also concentrate on the due time of finishing the jobs even when there is a continuous change in the traffic rate and the computing capacity. While, from the side of Service providers they always think of gaining the maximum profit by providing the requested resources of the user and also it is a great need to have a dynamic steady distribution of jobs. This paper focuses on how to deal with the distribution of jobs dynamically using credit traffic manager.

II. RELATED WORK

Cloud computing is coming up as a computing platform for providing resources to the user's in an efficient way. This platform also plays an important role in scheduling of the jobs. This job scheduling system is also responsible for providing the user's demanded resources dynamically even in the continuous change of traffic rate by considering some of the static and dynamic parameters. Nowadays, we can find many research work have done on Job Scheduling in Grid Computing. A reference [1-7] provides a view for the roles of job scheduling in Grid computing environment. The topologies presented for cloud or grids are classified into centralized and decentralized schedulers [1]. Most works are on centralized schedulers as there is an implementation complexity in decentralized schedulers. Reference [2] describes how to model and evaluate the performance of hierarchical job scheduling. Reference [3] shows an iterative scheduling algorithms based on the grids. [4] Presents a novel stochastic algorithm for QoS-constrained job scheduling in a web service oriented grid.

References [5-8] describe the approach of QoS performance analysis with dynamic scheduling system of Cloud Computing service. Reference [9] gives brief description of how the queuing parameters are built related to it fundamentals. Reference [10] explains how to meet cloud computing user's QoS requirements.

Most related research papers just rarely describes how the differential service-oriented guarantees job scheduling system in cloud computing. Except from this only a very few papers care about how to provide efficient resources to the users and to gain maximum profit for the cloud computing service providers by minimizing system costs. In this paper we explain how to provide dynamic distribution of jobs irrespective of the change in the traffic flow with the modeling of system which has a credit manager analyzer in order to control the flow. Section A describes the queuing model of jobs using Quasi Birth Death Process. Section B describes the parameters required for scheduling. Section C highlights on adapting folding algorithm for handling credit manager to obtain steady distribution of jobs from the queue to the resources.

III. SYSTEM FOR DYNAMIC STEADY STATE DISTRIBUTION OF JOBS USING CREDIT MANAGER

Cloud computing environment is considered as a powerful server in the system point of view. This will handle the jobs submitted to the server. Considering the user's job have priorities (where each priority will be of different one) to be processed by the server. So the jobs are classified into different classes with different priorities.

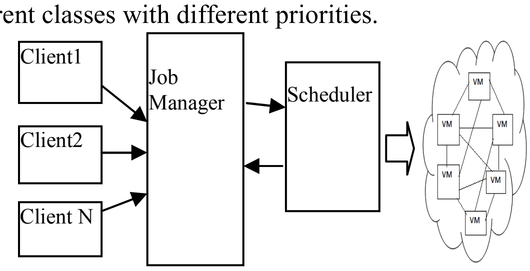

Fig. 1. An Illustration for Cloud Job Scheduling

The Figure1 illustrates the overview of the scheduling mechanisms in cloud. The Clients from different locations sends their request to the Job Manager, which acts as a

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

priority classifier of classifying the incoming jobs into. Highest, Middle and Lowest priority, then it is send to the scheduler for providing their service.

A. *Queuing model for processing of jobs*

In this model above, Users jobs are classified into N different classes by their different priorities. Each class i (i□ [1, N]) is with a priority. The small number of i is, the higher priority of the class is. Class 1 has the highest priority in the queue. We assumed that Users jobs in different classes with different priority. And they come to the server with a Poisson distribution at a certain rate, while the process time to each job by the server is in accord with a general distribution. Thus, we can build M/G/1 queuing model with non-preemptive system (See Figure.2). The issue of job scheduling in the Cloud Computing environment is turned into a queue scheduling problem for M/G/1 with non preemptive system.

To measure the characteristics of Users jobs and Service Provider's computing resources, we assume that Users jobs in the same class with priority are submitted to the Cloud according to Poisson distribution with rate I λ, and job scheduling system in the Cloud will assign some resources in the Cloud to process each job with a general service time distribution of mean i/t and second moment 2i/t .In each class with same priority, Users jobs are process by the order of its arrival (First In First Out).

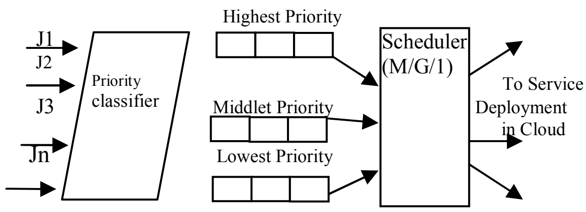

Fig. 2. Queuing Model for Differential Cloud Computing Service

For the Users jobs in each class are in accord with Poisson distribution at the rate of i\ λ, we can easily get the total rate λ of all Users jobs that submitted to the Cloud Computing environment. As for, each service rate i μ for each class is in accord with a general distribution, with the mean i t and second moment 2i t . We can use the following figure (Fig..3) to depict them.

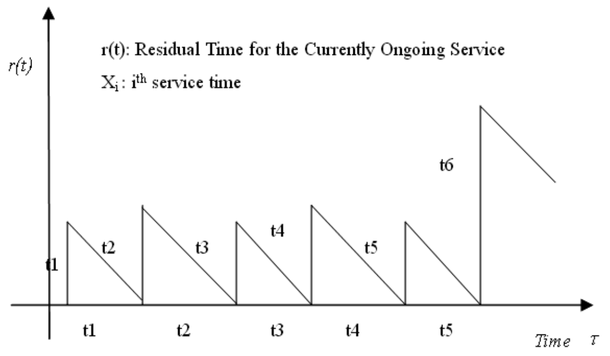

Fig. 3.The Service time for Each Class in a Cloud Computing Environment

B. *Assumption of parameters*

The service rate of the Cloud Computing environment is μ, the traffic intensity of the Cloud Computing environment is ρ,

The traffic intensity of jobs in each class

$$\bar{t} = \frac{1}{\mu} = \sum_{i=1}^{n} \frac{[(\lambda]_i * t_i)}{\lambda}$$

$$\rho = \frac{\lambda}{\mu} = \sum_{i=1}^{n} \rho_i$$

To make the system equilibrium, the cloud job scheduling system can only work provided that ≤1.

The mean residual service time R for all jobs in the cloud can be got by

$$R = \sum_{i=1}^{n} \rho_i \left(\frac{\overline{t_i^2}}{2 \cdot \bar{t}_i} \right) = \frac{1}{2} \sum_{i=1}^{n} \lambda_i \overline{t_i^2}$$

For a job in class 1, the waiting time is,

C. *Quasi Birth Death Process*

With a suitable ordering of states, the transition matrix of a QBD has the block partitioned from

$$Q = \begin{bmatrix} Q_1 & Q_0 & 0 & 0 & \cdots \\ Q_2 & Q_1 & Q_0 & 0 & \cdots \\ 0 & Q_2 & Q_1 & Q_0 & \cdots \\ 0 & 0 & Q_2 & Q_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ . & . & . & . & \cdots \end{bmatrix}$$

When the QBD is in state (k,i) we say that it is in the level k and phase i.

The above transition matrix looks like a block version of the transition matrix

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ \mu & -(\mu+\lambda) & \lambda & 0 & \cdots \\ 0 & \mu & -(\mu+\lambda) & \lambda & \cdots \\ 0 & 0 & \mu & -(\mu+\lambda) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix}$$

For the M/M/1 queue, so it is reasonable to ask what properties carry over from the M/M/1 queue in a "block sense" to a QBD.

The single server queue is stable if and only if λ < μ.

Let x be the solution to

X [Q0 + Q1 + Q2] = 0:

Then the QBD is positive recurrent if

xQ0e' < xQ2e',

Null recurrent if

xQ0e'= xQ2e',

And transient if

xQ0e' > xQ2e'.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

D. Dynamic steady state distribution of jobs

Dynamically scheduling the jobs and distributing to the cloud users is a very high overhead process since it has to meet the QOS requirements of the users even in a continuous change in the traffic rate, thereby making the maximum profit for cloud computing service provider. This can be achieved by credit manager analyzer which handles the large amount of incoming jobs effectively by distributing the required resources from the job pool in a steady state manner and hence obtaining an optimum queue size dynamically. Basically the credit manager is used to smooth out any burstiness of the incoming jobs entering into the job pool. It can handle a large amount of jobs without any delays. The steady state distribution of jobs in the job pool is achieved by using a special type of algorithm called Folding Algorithm.

IV. JOB'S ARRIVAL

The parameters which included are arrival rate and service rate. Customers arrive at the rate of lambda. The jobs have limited space to be queued. Here there are many credit banks (0, 1…. K) and K+1 discrete levels with the level of i. The probability that a customer requires j stages of service is (j = 1... K***),*** which are independent of others.

V. NO JOBS SUBMITTED TO THE JOB POOL (NO RESOURCES ARE REQUESTED TO PROVIDE)

If there are no jobs submitted to the job pool it may be because there is no requesting of resources or the user may be blocked due to limited queue capacity, then the credit bank content will go up to Δ. When a customer is blocked then it will be considered as in the stage of (k+1) of service.

VI. CREDIT MANAGER IN SCHEDULING

The following scenario is assumed.

a) Jobs arrive to the queue by the arrival rate of λ.

b) Let the buffer size be n-1.

c) There are k+1 discrete levels of bank (0, 1… k).

d) A customer requires of k stages of service.

e) The time spent in each service is exponential with mean 1/k.

f) For a job to have k level of service, the job should obtain at least (k/r) of credit i.e. Cmax=(k/r)

g) If no jobs enter the service or the job does not reaches the Cmax, then the jobs have to wait until it reaches the sufficient level though it may be at the head end of the queue. The job increases exponentially with a fixed rate.

VII. ADAPTATION OF AN ALGORITHM FOR HANDLING CREDIT MANAGER

The transition matrix in [2, 3] has a form slightly different from the P in our problem. We find that it is possible to apply the folding algorithm to our problem by adding an adaptation phase before proceeding to the first reduction phase. For the matrix P with even number of entries, using the reduction procedure in [2, 3], the resulting child matrix becomes

$$P_1 = \begin{bmatrix} A_0^{(1)} & U^{(1)} & & & \\ D^{(1)} & A^{(1)} & U^{(1)} & & \\ & \ddots & \ddots & \ddots & \\ & & D^{(1)} & A^{(1)} & U^{(1)} \\ & & & D^{(1)} & A_2^{(1)} \end{bmatrix}$$

Where

$$A_0^{(1)} = A_1 - D_0 A_0^{-1} U_0 - UA^{-1}D$$
$$A^{(1)} = A - DA^{-1}U - UA^{-1}D$$
$$A_2^{(1)} = A_2 - DA^{-1}U$$
$$U^{(1)} = -UA^{-1}U$$
$$D^{(1)} = -DA^{-1}D$$

If the matrix *P* is odd,

$$P_1 = \begin{bmatrix} A_0^{(1)} & U^{(1)} & & & & \\ D^{(1)} & A^{(1)} & U^{(1)} & & & \\ & \ddots & \ddots & \ddots & & \\ & & D^{(1)} & A^{(1)} & U^{(1)} & \\ & & & D^{(1)} & A_2^{(1)} & U_b^{(1)} \\ & & & & D_b^{(1)} & A_b^{(1)} \end{bmatrix}$$

Where

$$A_0^{(1)} = A_1 - D_0 A_0^{-1} U_0 - UA^{-1}D$$
$$A^{(1)} = A - DA^{-1}U - UA^{-1}D$$
$$A_2^{(1)} = A - DA^{-1}U$$
$$U^{(1)} = -UA^{-1}U$$
$$D^{(1)} = -DA^{-1}D$$
$$A_b^{(1)} = A_2, U_b^{(1)} = U, D_b^{(1)} = D$$

We see that the child matrix in each ***case*** is exactly the proposed form for the folding algorithm, so that the algorithm can now be applied directly to our problem. Similarly, for the expansion phase, some modification of the algorithm is needed in order to obtain the result.

VIII. CONCLUSION AND FUTURE WORK

Thus, the continuous change of traffic flow in a cloud computing environment can be regulated by using credit manager analyzer which can handle large amount of incoming data effectively in a steady state manner. It can also helps to achieve Quality of Service and also queue size can be dynamically optimized. In future, we have an idea of improving the security in scheduling of the incoming jobs.

REFERENCES

[1]. Hamscher, V., et al., Evaluation of Job-Scheduling Strategies for Grid Computing. LECTURE NOTES IN COMPUTER SCIENCE, 2000: p. 191-202.

[2]. Zhiguang, S. and L. Chuang. Modeling and Performance Evaluation of Hierarchical Job Scheduling on the Grids. in Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on. 2007.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

[3]. Bayati, M., et al. Iterative Scheduling Algorithms. in INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. 2007.

[4]. Patel, Y. and J. Darlington. A Novel Stochastic Algorithm for Scheduling QoS-Constrained Workflows in a Web Service-Oriented Grid. in Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on. 2006.

[5]. Weidong, H., Y. Yang, and L. Chuang. Qos Performance Analysis for Grid Services Dynamic Scheduling System. in Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on. 2007.

[6]. Afzal, A., A.S. McGough, and J. Darlington, Capacity planning and scheduling in Grid computing environments. Future Generation Computer Systems 2008. 2008(24): p. 404-414.

[7]. Kiran, M., et al. A prediction module to optimize scheduling in a grid computing environment. in Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on. 2008.

[8]. Rasooli, A., M. Mirza-Aghatabar, and S. Khorsandi. Introduction of Novel Rule Based Algorithms for Scheduling in Grid Computing Systems. in Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on. 2008.

[9]. Hock, N.C., Queuing Modelling Fundamentals. JOHN WILEY&SONS, 1997.

[10] Lu, C., Queue Theory. Beijing YouDian Publish House, 2007.

[11] Luqun Li, An Optimistic Differentiated Service Job Scheduling System for Cloud computing Service users and providers 2009, Third International Conference on Multimedia and Ubiquitous Engineering.

[12] Ming-shan Kwok and Raymond W.Yeung, Analysis of a Credit Manager for traffic Regulation in a high speed network system with Finite Buffer 1994, IEEE Conference.

AUTHOR BIOGRAPHY

S. Selvi is pursuing her Ph.D. Degree in the area of Cloud Computing in Anna University of Chennai. She had completed PG degree in Velalar College of Engineering and Technology, Erode and UG degree in Kongu Engineering College, Perundurai.She has published more 6 papers in National Conferences and 2 papers in International Conference. She is currently working as an Assistant Professor in Angel College of Engineering, Tirupur. She has co ordinate various workshops, seminars and guest Lectures. She also attended various workshops and seminars.