

Optimization of Resource Cost and Service Charge Scheme in Cloud Computing

Kshitija S. Sontakke¹, Karan S. Mashal²

¹ME Student, Dept .of Computer Engineering, RMD Sinhgad School of Engineering, Warje, Pune - 411058, India

²Assistant Professor, Dept of Computer Engineering, RMD Sinhgad School of Engineering, Warje, Pune - 411058, India

Abstract: Cloud computing is a form of internet based and on demand computing where information and shared resources are provided to the customer on-demand. From the cloud service providers' point of view profit is the most imperative factor and it is fundamentally decided by the arrangement of a cloud service platform under given market requirements. A single long-term renting scheme is normally choose to arrange a cloud platform, which leads to major resource waste and cannot assure the quality of service. To overwhelm the drawbacks of single renting scheme, Double Resource Renting Round Robin scheme is designed which is the combination of short-term and long-term renting. Double resource renting RR scheme not only minimize the resource waste but also guarantees the quality of service to the customers. In which M/M/m+D queuing model is used for job scheduling. Double resource renting round robin scheme not only increase the profit of the cloud service providers but also provides the Qos to the customer by using load balancing round robin algorithm.

Keywords: Cloud computing, service-level-agreement, profit, service charge, queuing model, load balancing, round robin

1. Introduction

Cloud computing is kind of internet based computing that depends on shared pool of physical and virtual resources, instead of using local or personal hardware and software.

Cloud computing delivers services and computing resources to the consumer on demand in pay-per-use manner anytime and anywhere, but for accessing these services consumer must connected to the high speed internet. Many areas like e-commerce, banking has been adopted Cloud due to its benefits. It offers secure, fast and convenient data storage.

A service provider can assemble different multiserver systems [2] for various application areas which is used to execute special kind of service requests and applications. Fig. 1. Shows three tiers of a cloud computing environment, i.e. Infrastructure providers, service providers, and customers. An infrastructure provider is responsible for the maintenance of basic software and hardware facilities. A cloud service provider rents resources from the infrastructure providers and deliver services to customers. A customer sends its request to a cloud service provider and pays for it based on the quantity and the quality of the provided service. [1] Organizations no need to buy computing re-sources to handle the capacity peaks. In the cloud computing for increasing the profit first should know about the cost and revenue. Profit maximization must consider the consumer satisfaction also the cost of the cloud includes the energy consumption cost and renting cost. For maximizing the profit of the cloud service providers we have to reduce the renting cost and increase the service price to the consumer and also provide quality of services to the consumer.

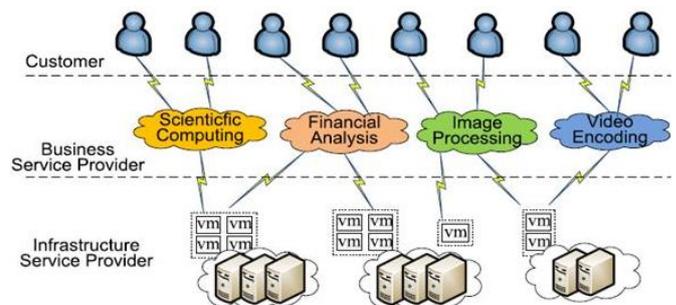


Figure 1: Three tier Cloud Environment

2. Literature Survey

In the cloud computing environment for maximizing the profit of the cloud service provider's optimal multiserver configuration problem is analyzed. In [2] paper multiserver system is considered as an M/M/m queuing model, such that problem of optimization can be developed and solved analytically. They have considered power consumption and two server speed model viz idle speed model and the constant speed model. The probability density function of the waiting time of a newly arrived service request is derived. The calculation of the expected service charge to a service request is done. The expected net business gain in one unit of time is obtained. They have demonstrated numerical calculations of the optimal server size and speed. Their pricing model consider many elements such as service quantity, application environment workload, multiserver system configuration, SLA, service quality, consumer fulfillment, low quality service penalty, renting charge, energy consumption charge, and service provider's profit as well as margin.

In [3] paper they have proposed a cloud framework which considers Service Level Agreement, Optimization of resources and Admission Control along with cloud structure and marketing strategies for avoiding loss, retain profit,

maximal returns, which eventually bring on maximal profit in the cloud organization. They have also considered the factor of minimization of cost and optimal output in cloud enterprise and defined Multi Objective Bat Algorithm which provide best way to obtain maximal optimization to maximize the profit in cloud business. In this paper different profit maximization factors are considered for infrastructure provider, service provider and customer.

In the [4] paper they have presented the idea of cloud computing with load balancing. It helps in proper usage of resources and improves performance of the system. Several load balancing algorithms are also described such as Max-Min algorithm, Opportunistic Load Balancing, Round Robin Algorithm, Min-Min algorithm, Honeybee Foraging Behavior, Task Scheduling based on LB, Shortest Response Time First, Ant Colony Optimization which consists of many factors like better utilization of resources, better response time, scalability, high performance.

In [5] paper they have discussed about pricing in cloud, aspects of pricing models and different types of pricing models such as genetic model for pricing, pay-as-you go model in cloud computing markets.

Multi objective genetic algorithm is proposed in the [6] for solving multivariable optimization problem. This system provides a new cloud brokering mechanism with cloud service discovery using this optimization technique. This paper mainly focus the IaaS layer of the cloud computing. In this system user sends job to the service provider and service provider provides services to the user. Here the user requirement is to acquire services within deadline and within minimum cost. At the same time service providers aim is to increase the income for that he should give the quality of service to the user by minimizing customers cost, providers idle time and customers waiting time and maximize service providers benefit. [7]

3. Proposed System

3.1 Queuing Model

The cloud service platform is made up of multiserver system with service request queue. Cloud assigns resources for job in the form of VM. Users submit their jobs to the cloud which uses a job queuing system. [1] All the jobs are scheduled by scheduler in the centralized manner and allocated to the different Virtual Machines. Queuing model is one kind of service request queue. An M/M/m+D queueing model [2] is constructed for our multi-server system with changing system size. Where, M/M/m is Kendall's notation of the queueing model in which the first part represents the input process, the second part represents the service distribution, and the third part represents the number of servers. When incoming service request cannot be handled instantly after arriving into the system, they are firstly put into the queue until any available server handles it. This queueing model works as a first come first serve manner [6].

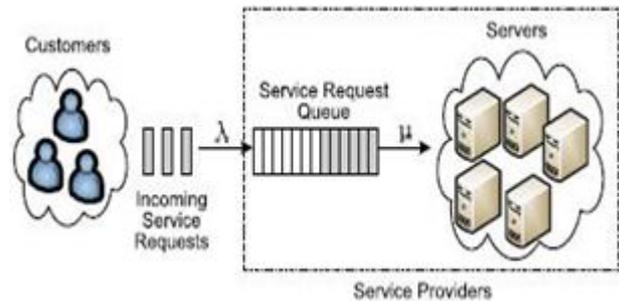


Figure 2: M/M/m Queuing Model

3.2 Load Balancing

Workloads are distributed over multiple computing resources such as computers, computer clusters, network links, disk drives or CPU using load balancing. Load balancing is used to maximize throughput, optimize resource use, avoid overload and minimize response time of any single resource. Using many components with load balancing instead of a single component may increase availability and reliability via redundancy. For improving the overall performance of the system load balancing distribute larger processing load to the smaller processing node. [4] Load balancing is used to distribute dynamic local workload equally between all the nodes. We can obtain high satisfaction of the user with proper usage of resources through load balancing. Proper balancing of load and high usage of resources helps in reducing consumption of resources. Without load balancing many problems occur such as delays, timeouts and long system responses. There are several algorithms available for load balancing such as Min Min algorithm, first come first serve algorithm, Max Min algorithm, Round robin algorithm and so on. From which we are to using the Round Robin algorithm for balancing the workload of the virtual machines.

3.3 System Architecture

A double resource renting round robin scheme is also known as double quality guaranteed round robin scheme (DQG-RR) is a novel renting scheme proposed for service providers which combines long-term renting and short-term renting, which not only assure quality-of-service requirements under the varying system workload, but also reduce wastage of resources. Fig.3. shows system architecture of Double Quality Guaranteed Round Robin scheme. In which client needs to first create account and logged into the system. Client account information will be stored into the database. After logging into the system client need to initially send a Storage allocation space to the server. And also select priority for the service request.

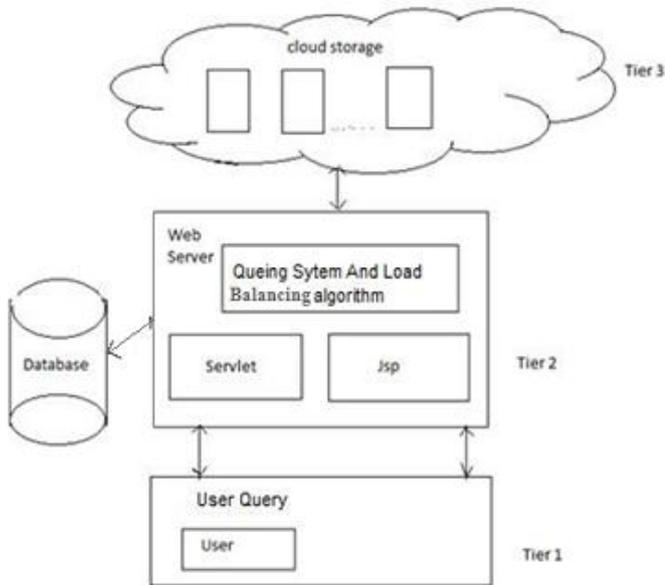


Figure 3: System Architecture of DQG-RR Scheme

Every client request will be stored in a queue in the first come first serve (FIFO) manner and processes by Business Service Provider. Business service provider acts as a job scheduler in between client and infrastructure service provider. Once the business service provider approved the request from the client, then the client can request for an Infrastructure Storage Space. Then BSP logged into the system and can run DQG algorithm on incoming service request. After that round robin algorithm is performed for that queuing system is given as a input to the Round robin algorithm. Then round robin algorithm attach time slice to each client request and check the workload of each virtual machine. If workload of one virtual machine increases then request to that virtual machine is get transferred and processed by another virtual machine. Load will be evenly distributed between all the virtual machine. Processing time and priority of service request are considered for RR scheduling. We are improving QOS of cloud service provider by balancing workload of VMs. Using round robin load balancing algorithm we are maximizing the profit of our cloud. BSP can also calculate optimal server size, optimal server speed, and optimal server size and speed. A multi-server system is designed as an M/M/m+D queuing model and the performance indicators such as the ratio of requests, the average service charge, that require short term servers are analyzed. For maximizing the profit of cloud service providers our pricing model takes some elements into considerations such as the quantity of a service, multiserver system configuration,[2] the workload of an application environment, SLA, customer satisfaction, service quality, low-quality service penalty, the renting charge, and margin and profit of service provider.

3.4 Algorithms

3.4.1 Double Quality Guaranteed Algorithm

- 1: A multiserver system with m servers is running and waiting for the events as follows
- 2: A queue Q is initialized as empty
- 3: **Event** – A service request arrives
- 4: Search if any server is available

- 5: **if true then**
- 6: Assign the service request to one available server
- 7: **else**
- 8: Put it at the end of queue Q and record its waiting time
- 9: **end if**
- 10: **End Event**
- 11: **Event** – A server becomes idle
- 12: Search if the queue Q is empty
- 13: **if true then**
- 14: Wait for a new service request
- 15: **else**
- 16: Take the first service request from queue Q and assign it to the idle server
- 17: **end if**
- 18: **End Event**
- 19: **Event** – The deadline of a request is achieved
- 20: Rent a temporary server to execute the request and release the temporary server when the request is completed
- 21: **End Event**

3.4.2 Finding Optimal Size Algorithm

Algorithm 2 Finding the optimal size

- Input:** $s, \lambda, \bar{r}, a, P^*, \alpha, \beta, \gamma, \delta, \xi,$ and D
Output: the optimal number Opt_size of fixed servers
- 1: $Profit_max \leftarrow 0$
 - 2: find the server size m using the analytical method
 - 3: $m_i^* \leftarrow \lfloor m \rfloor, m_u^* \leftarrow \lceil m \rceil$
 - 4: $Profit_i \leftarrow Profit(m_i^*, s), Profit_u \leftarrow Profit(m_u^*, s)$
 - 5: **if** $Profit_i > Profit_u$ **then**
 - 6: $Profit_max \leftarrow Profit_i$
 - 7: $Opt_size \leftarrow m_i^*$
 - 8: **else**
 - 9: $Profit_max \leftarrow Profit_u$
 - 10: $Opt_size \leftarrow m_u^*$
 - 11: **end if**

3.4.3 Finding Optimal Speed Algorithm :

Algorithm 3 Finding the optimal speed

- Input:** $m, \lambda, \bar{r}, a, P^*, \alpha, \beta, \gamma, \delta, \xi,$ and D
Output: the optimal server speed Opt_speed
- 1: $Profit_max \leftarrow 0$
 - 2: find the server speed s using the analytical method
 - 3: $s_i^* \leftarrow s_i, s_u^* \leftarrow s_{i+1}$ if $s_i < s \leq s_{i+1}$
 - 4: $Profit_i \leftarrow Profit(m, s_i^*), Profit_u \leftarrow Profit(m, s_u^*)$
 - 5: **if** $Profit_i > Profit_u$ **then**
 - 6: $Profit_max \leftarrow Profit_i$
 - 7: $Opt_speed \leftarrow s_i^*$
 - 8: **else**
 - 9: $Profit_max \leftarrow Profit_u$
 - 10: $Opt_speed \leftarrow s_u^*$
 - 11: **end if**

3.4.4 Finding Optimal Size and Speed Algorithm :

Algorithm 4 Finding the optimal size and speed

Input: $\lambda, \bar{\tau}, a, P^*, \alpha, \beta, \gamma, \delta, \xi,$ and D
Output: the optimal number Opt_size of fixed servers and the optimal execution speed Opt_speed of servers

- 1: $Profit_max \leftarrow 0$
- 2: find the server size m and speed s using the analytical
- 3: $m_i^* \leftarrow [m], m_u^* \leftarrow [m]$
- 4: find the optimal speed s_i^* and s_u^* using Algorithm 3 with server size m_i^* and m_u^* , respectively
- 5: $Profit_i \leftarrow Profit(m_i^*, s_i^*), Profit_u \leftarrow Profit(m_u^*, s_u^*)$
- 6: **if** $Profit_i \leq Profit_u$ **then**
- 7: $Profit_max \leftarrow Profit_u$
- 8: $Opt_size \leftarrow m_u^*, Opt_speed \leftarrow s_u^*$
- 9: **else**
- 10: $Profit_max \leftarrow Profit_i$
- 11: $Opt_size \leftarrow m_i^*, Opt_speed \leftarrow s_i^*$
- 12: **end if**

3.4.5 Round Robin Algorithm

- 1) Round RobinVmLoad Balancer maintains an index of VMs and state of the Vms (busy/available). At start all VMs have zero allocation
- 2) a) The datacentercontroller receives the user request / cloudlets
 b) It stores the arrival time and burst time of the user requests
 c) The requests are allocated to VMs on the basis of their states known from the VM queue
 d) The RoundRobinVmLoadBalancer will allocate the time quantum for user request execution.
- 3) a) The RoundRobinVmLoadBalancer will calculate the turn- around time of each process
 b) It also calculates response time and average waiting time of user requests.
 c) It decides the scheduling order.
- 4) After the execution of cloudlets, the Vms are de-allocated by the RoundRobinVm- LoadBalancer
- 5) The DataCenterController checks for new/pending/waiting request in queue
- 6) Continue from step-2 [8]

3.5 Mathematical Model

Input: Set of Parameters.

Output:

- 1) Load Degree
- 2) Load Degree Average

Define a load Parameter Set: $P=fp1,p2....pkg$

$k=$ represent the total no. of parameters

Compute the load degree as:

$$Load_Degree(N) = \sum_{i=1}^k aiPi \tag{1}$$

Where,

$$ai \sum_{i=0}^n ai = 1 \tag{2}$$

eq(2) represent weights that may differ for different kinds of jobs.

N represents current node $i=1...k$

$i=1...k Pi= (1 <= i <= k ;Pi \in [0,1])$ – User Request

Average Cloud partitioning load Degree:

$$Load_Degree_Avg = \left(\sum_{i=1}^k Load_Degree(Ni) \right) / n \tag{3}$$

Classify Load

Input:

- 1) Load Degree.
- 2) Load Degree Average

Output:

Status Information:

Idle:

Load_Degree(n)=0

There is no job being processed by this node so the status is changed to Idle.

Normal:

$0 < Load_Degree(n) \leq Load_Degree_High$

The node is normal and it can process other jobs.

Overloaded:

$Load_Degree_High \leq Load_Degree(n)$

The node is not available and cannot receive jobs until it returns to the normal. [9]

4. Results

We have implemented this project on cloud from cloudzones.org. We have created three instance of virtual machines on cloud and performed round robin load balancing on that virtual machines. An Apache Tomcat provides deployment environment for the application. A backend tier, Mysql database server is used. Fig.4. represents time graph for DQG scheme(existing scheme) and Round Robin load balancing scheme(proposed scheme), Which shows Round robin load balancing require less time to execute as compared to DQG scheme.



Figure 4: Time Graph

Fig.5. Represents memory graph for DQG scheme (existing scheme) and Round Robin load balancing scheme (proposed scheme), Which shows Round robin load balancing scheme requires less memory than DQG scheme.

Fig.6. Represents Quality of service graph for DQG scheme (existing scheme) and Round Robin load balancing scheme (proposed scheme). Which shows Round robin load balancing scheme provide better quality than DQG scheme in terms of processing time that is Round Robin load balancing scheme requires less processing time than DQG scheme.

Fig.7. Represents Quality of Service parameters like Bandwidth, jitter, delay, which shows RR load balancing scheme provide more bandwidth than DQG scheme and requires less delay and jitter than DQG scheme.

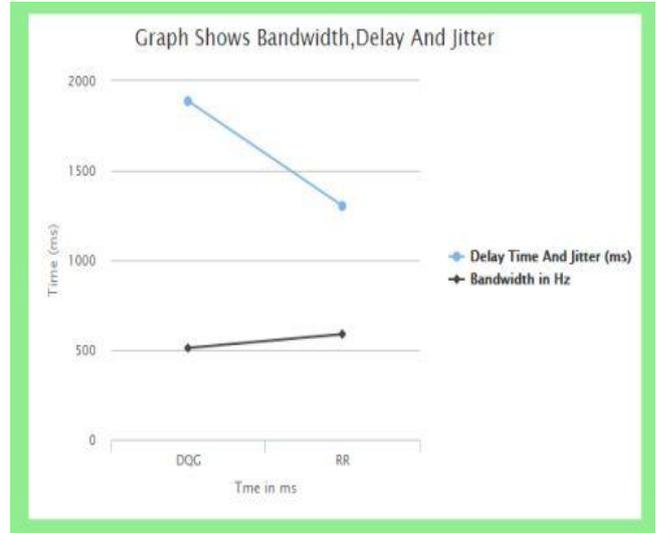


Figure 7: Quality of Service Parameters

Fig.8. Represents profit graph for DQG and RR load balancing Scheme which shows Round Robin load balancing provide more profit than DQG scheme.

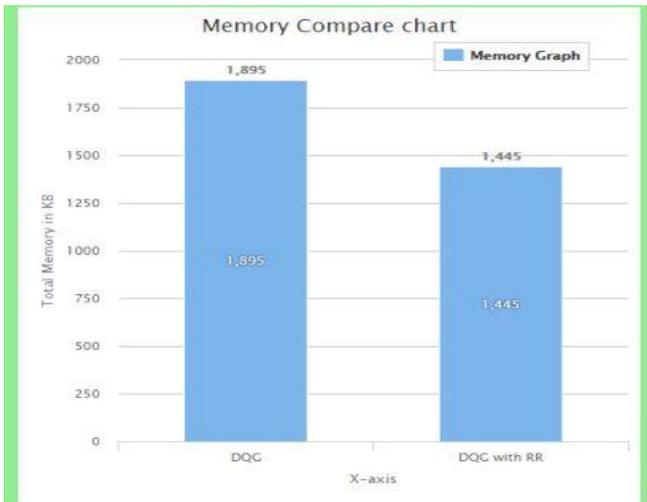


Figure 5: Memory Graph

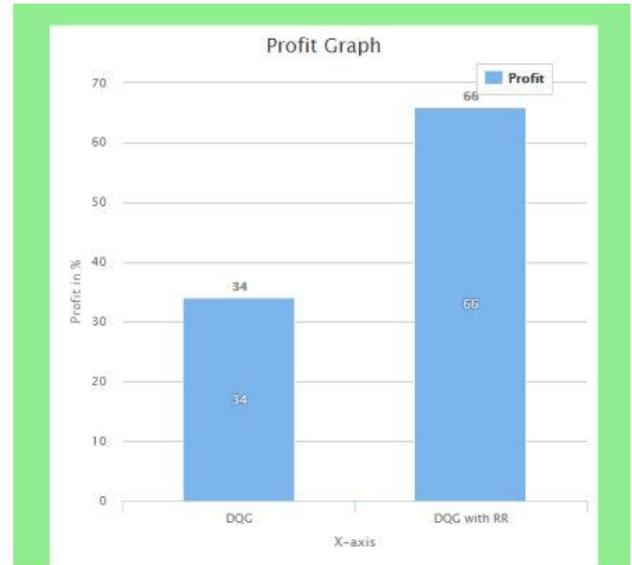


Figure 8: Profit Graph



Figure 6: Quality of Service with Processing Time

5. Conclusion

A novel double quality guaranteed renting round robin scheme is proposed to provide guaranteed quality of service to all requests under the varying system workload, to reduce the resource waste greatly and to maximize the profit of the cloud service providers. In DQG-RR Scheme short-term renting and longterm renting are combined, multiserver system is configured with M/M/m+D queuing model and QOS of cloud service provider is improved by balancing workload of VMs.

References

- [1] Jing Mei, Kenli Li, Aijia Ouyang, Keqin Li, "A Profit Maximization Scheme with Guaranteed Quality of

- Service in Cloud Computing,” IEEE Transactions on Computers, Vol. 1, No 2015.
- [2] J. Cao, K. Hwang, K. Li, and A. Y. Zomaya, “Optimal multiserver configuration for profit maximization in cloud computing,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1087–1096, 2013.
- [3] R.Raju, M.Kalaiarasi, S.Santhana Krishnan, “Profit Maximization in Cloud Computing,” ICICES2014-S.A.Engineering College, Chennai, Tamil Nadu, India.
- [4] Rajwinder Kaur, Pawan Luthra, ”Load Balancing in Cloud Computing,” Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC.
- [5] May Al-Roomi, Shaikha Al-Ebrahim, Sabika Buqrais and Imtiaz Ahmad, Cloud Computing Pricing Models: A Survey, International Journal of Grid and Distributed Computing, Vol.6, No.5, pp.93-106, 2013.
- [6] Dr. P. Balakumar, Deepa.V, Qos Supported SLA for Profit maximization of multi-server configuration in cloud computing, International Journal of Engineering Research Technology (IJERT), Vol. 3 Issue 3, March – 2014
- [7] Kshitija S. Sontakke, Karan S. Mashal, Survey on Optimization of Resource Cost and Service Price Scheme in Cloud Computing, International Journal of Science and Research (IJSR), Vol. 4, Issue 12, December 2015
- [8] Subasish Mohapatra, K.Smriti Rekha, Subhadarshini Mohanty, A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing,” International Journal of Computer Applications, Volume 68- No.6, April 2013
- [9] Akshada Bhujbal, Prajakta Jakate, Manasi Wagh, Madhura Pise, Prof.M.V.Marathe, Load Balancing Model in Cloud Computing, International Journal of Emerging Engineering Research and Technology Volume 3, Issue 2, February 2015.

Author Profile



Kshitija S. Sontakke pursuing ME in Computer Engineering at RMD Sinhgad School of Engineering, Savitribai Phule Pune University. She has received B.E. in Computer Science and Engineering from Brahmdevdada Mane Institute of Technology, Solapur University, Solapur (2012). Currently she is pursuing M.E. in Computer Engineering from RMD Sinhgad School of Engineering, Warje, Pune, Savitribai Phule Pune University



Prof. Karan S. Mashal Prof. Karan Mashal received M.tech in Information Technology. He is currently working as an Assistant Professor in Computer Engineering, RMD Sinhgad School of Engineering, Savitribai Phule Pune University. His area of interest is web development, operating system and smart system. He has 5 years of teaching experience.