IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Special Issue-3, May 2016 E-ISSN: 2347-2693

Profit Maximization for Cloud Services in Multiserver Environment

Raushan kashypa[1*] and Sowmya Naik P.T [2]

[1,2] *Dept. of CSE, City Engineering College, India.*

Available online at: www.ijcseonline.org

Abstract Cloud computing provides resources and services to customers in a dynamic basis. That's why it has become an effective and very efficient way for computing. Profit plays a very important role from perspective of a cloud service provider. This profit will be determined based on how a cloud service platform has been configured and it also depends on market demand. Generally a single long term renting will be used to configure a cloud platform which is not capable of quality service and it also leads to a greater resource waste.

Keywords— Cloud computing, guaranteed service quality, multiserver system, profit maximization, queuing model, service-level agreement, waiting time.

I. INTRODUCTION

Cloud computing is quickly becoming an effective and efficient way of computing resources. By centralized management of resources and services, cloud computing delivers host based services over the Internet. Cloud computing is able to provide the most cost effective and energy efficient way of computing resource management. It turns information technology into ordinary commodities and utilities by using the pay-per-use pricing model. A service provider rents resources from the infrastructure vendors, builds correct multi server machines and provides many services to the subscribers. A customer submits a request to a service provider, receives the desired result from the service provider with certain rules. These rules or agreement will be called as Service level agreement or SLA.

Owing to redundancy of computer system networks and storage system cloud may not be reliable for data, the security score is concerned. In cloud computing security is tremendously improved because of a superior technology security system, which is now easily available and affordable. Applications no longer run on desktops Personal computers but run in the clouds. This means that PC does not need the processing power or hard disk space as demanded by traditional desktops software. Powerful servers and the like are no longer required. The computing power of the cloud can be used to replace or supplement internal computing resources. Organizations no longer have to purchase computing resources to handle the capacity peaks.
Peaks are easily handled by the clouds. The payment of the most cloud computing services is based on a pay as you go model. This means that customers only pay for what they use. Distributed systems are group of networked computer, which have the same goal for their work. In parallel computing all processors may have access to shared memory to exchange information between processors. In distributed computing, each processors have its own private memory which is the distributed memory. Information is exchanged by passing messages between the processors.

A process knows its own state and it knows what state other processes were in progress. Distributed computing also refers to the use of the distributed systems to solve computational problems. In this a problem is divided into many tasks. Each of which is solved by one or many computers. Each of which communicate with each other by message passing method. Parallel and distributed methods have proved to be effective in tackling the problems with the high computational complexity in a wide range of domains. Distributed computing is the process of aggregating the power of several computing entities which are logically distributed and may even be geologically distributed to collaboratively run a single computational task in the transparent and the coherent way. Cloud computing is emerging at the convergence of three major trends of service orientation, virtualization and standardization of computing through Internet. Most cloud computing infrastructures consists of services delivered through shared data centers and appear as a single point of access for consumers computing needs.

II. Related work

In this section, we review recent works relevant to the profit of cloud service providers. Profit of service providers is related with many factors such as the price, the market demand, the system configuration, the customer satisfaction and so forth. Service providers naturally wish to set a higher price to get a higher profit margin; but doing so would decrease the customer satisfaction, which leads to a risk of discouraging demand in the future. Hence, selecting a reasonable pricing strategy is important for service providers.

The pricing strategies are divided into two categories, i.e., static pricing and dynamic pricing. Static pricing means that the price of a service request is fixed and known in advance, and it does not change with the conditions. With dynamic pricing a service provider delays the pricing decision until after the customer demand is revealed, so that the service provider can adjust prices accordingly. Static pricing is the dominant strategy which is widely used in real world and in research Dynamic pricing emerges as an attractive alternative to better cope with unpredictable customer demand.

The second factor affecting the profit of service providers is customer satisfaction which is determined by the quality of service and the charge. In order to improve the customer satisfaction level, there is a service-level agreement (SLA) between a service provider and the customers. The SLA adopts a price compensation mechanism for the customers with low service quality. The mechanism is to guarantee the service quality and the customer satisfaction so that more customers are attracted. In previous research, different SLAs are adopted.

Since profit is an important concern to cloud service providers, many works have been done on how to boost their profit. A large body of works have recently focused on reducing the energy cost to increase profit of service providers and the idle server turning off strategy and *dynamic CPU clock frequency scaling* are adopted to reduce energy cost. However, only reducing energy cost cannot obtain profit maximization. Many researchers investigated the trade-off between minimizing cost and maximizing revenue to optimize profit.

Chiang and Ouyang considered a cloud server system as an *M/M/R/K* queuing system where all service requests that exceed its maximum capacity are rejected. A profit maximization function is defined to find an optimal combination of the server size R and the queue capacity K such that the profit is maximized. However, this strategy has further implications other than just losing the revenue from some services, because it also implies loss of reputation and therefore loss of future customers [3]. In [2], Cao *et al.* treated a cloud service platform as an *M/M/m model*, and the problem of optimal multiserver configuration for profit maximization was formulated and solved. This work is the most relevant work to ours, but it adopts a single renting scheme to configure a multiserver system, which cannot adapt to the varying market demand and leads to low service quality and great resource waste. To overcome this weakness, another resource management strategy is used in, which is cloud federation [7]. Using federation, different providers running services that have complementary resource requirements over time can mutually collaborate to share their respective resources in order to fulfil each one's demand . However, providers should make an intelligent decision about utilization of the federation (either as a contributor or as a consumer of resources) depending on different conditions that they might face, which is a complicated problem.

In this paper, to overcome the shortcomings mentioned above, a double renting scheme is designed to configure a cloud service platform, which can guarantee the service quality of all requests and reduce the resource waste greatly. Moreover, a profit maximization problem is formulated and solved to get the optimal multiserver configuration which can product more profit than the optimal configuration in [2].

III. THE MODELS

In this section, we first describe the three-tier cloud computing structure. Then, we introduce the related models used in this paper including a multiserver system model, a revenue model, and a cost model.

A Cloud System Model

The cloud structure (see Fig. 1) consists of three typical parties, i.e., infrastructure providers, service providers and customers. This three-tier structure is used commonly in existing literatures.

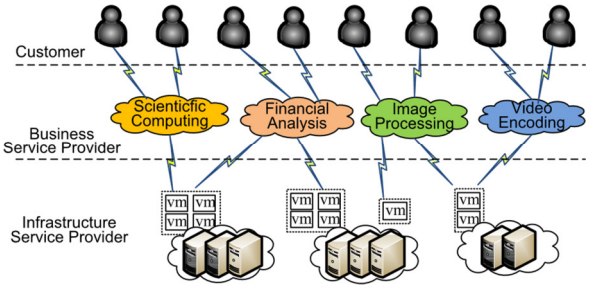

Fig. 1: The three-tier cloud structure.

In the three-tier structure, an infrastructure provider the basic hardware and software facilities. A service provider rents resources from infrastructure providers and prepares

a set of services in the form of virtual machine (VM). Infrastructure providers provide two kinds of resource renting schemes, e.g., long-term renting and short-term renting. In general, the rental price of long-term renting is much cheaper than that of short-term renting [6]. A customer submits a service request to a service provider which delivers services on demand Service providers pay infrastructure providers for renting their physical resources, and charge customers for processing their service requests, which generates cost and revenue, respectively. The profit is generated from the gap between the revenue and the cost.

A Multiserver Model

In this paper, we consider the cloud service platform as a multiserver system with a service request queue. Fig. 2 gives the schematic diagram of cloud computing.

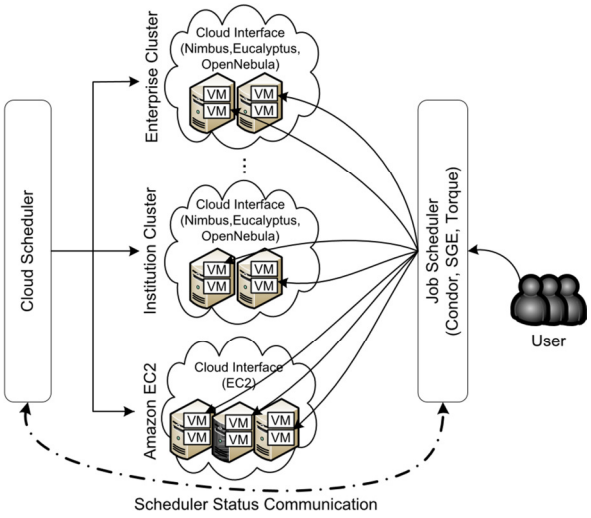

Fig. 2: The schematic diagram of cloud computing.

In an actual cloud computing platform such as Amazon EC2, IBM blue cloud, and private clouds, there are many work nodes managed by the cloud managers such as Eucalyptus, Open Nebula, and Nimbus. The clouds provide resources for jobs in the form of virtual machine (VM). In addition, the users submit their jobs to the cloud in which a job queuing system such as SGE, PBS, or Condor is used. All jobs are scheduled by the job scheduler and assigned to different VMs in a centralized way. Hence, we can consider it as a service request queue. For example, Condor is a specialized workload management system for compute intensive jobs and it provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their jobs to Condor, and Condor places them into a queue, chooses when and where to run they based upon a policy Hence, it is reasonable to abstract a cloud service platform as a multiserver model with a service request queue, and the model is widely adopted in existing literature.

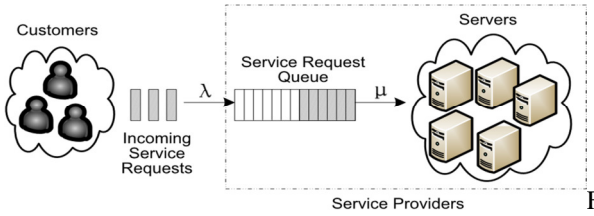

Fig. 3: The multiserver system model, where service requests are first placed in a queue before they are processed by any servers.

Which is rented from an infrastructure provider. Assume that the multiserver system consists of m long-term rented identical servers, and it can be scaled up by temporarily renting short-term servers from infrastructure providers. The servers in the system have identical execution speed s (Unit: billion instructions per second). In this paper, a multiserver system excluding the short-term servers is modelled as an M/M/m queuing system as follows (see Fig. 3). There is a Poisson stream of service requests with arrival rate λ, i.e., the interatrial times are independent and identically distributed exponential random variables with mean 1/λ. A multiserver system maintains a queue with infinite capacity. When the incoming service requests cannot be processed immediately after they arrive, they are firstly placed in the queue until they can be handled by any available server. The first-come-first-served (FCFS) queuing discipline is adopted. The task execution requirements (measured by the number of instructions) are independent and identically distributed exponential random variables r with mean r (Unit: billion instructions).

Revenue Modelling

This model will be determined by considering pricing strategy and SLA rules. In revenue model usage based pricing scheme will be adopted, as cloud service provider provide services to customers and charge them on demand basis. SLA is nothing but a negotiation between providers and customers related with service quality and pricing. Because of limited servers, all service requests may not be handled immediately after entering into job queue. They must wait if servers are not available. So waiting time of each service request must be limited to a range. This range will be determined by SLA. This is important to ensure quality of service.

SLA is widely used in all types of business these days. It also adopts a price compensation mechanism to guarantee

service quality and customer satisfaction. If a request is not serviced within its limited time or say within deadline then service provider has to service this request free as penalty. In some cases the normal charge will be reduced if request is not serviced under time range. Extra servers may increase cost of electricity but it will improve number of customers. Task is related to the amount of a service and the service level agreement. We define the service charge function for a service request with execution requirement r and waiting time W.

Cost Modeling

The cost of a service provider consists of two major parts, i.e., the rental cost of physical resources and the utility cost of energy consumption. Many existing research such as [11, 43, 44] only consider the power consumption cost. As a major difference between their models and ours, the resource rental cost is considered in this paper as well, since it is a major part which affects the profit of service providers. A similar cost model is adopted in [2]. The resources can be rented in two ways, long-term renting and short-term renting, and the rental price of long-term renting is much cheaper than that of short-term renting. This is reasonable and common in the real life. In this paper, we assume that the long-term rental price of one server for unit of time is β (Unit: cents per second) and the short-term rental price of one server for unit of time is γ (Unit: cents per second), where $\beta < \gamma$.

The cost of energy consumption is determined by the electricity price and the amount of energy consumption. In this paper, we adopt the following dynamic power model, which is adopted in the literature.

IV. A QUALITY-GUARANTEED SCHEME

The traditional single resource renting scheme cannot guarantee the quality of all requests but wastes a great amount of resources due to the uncertainty of system workload. To overcome the weakness, we propose a double renting scheme as follows, which not only can guarantee the quality of service completely but also can reduce the resource waste greatly.

Proposed system

In this section, we first propose the Double-Quality Guaranteed (DQG) resource renting scheme which combines long-term renting with short-term renting. The main computing capacity is provided by the long-term rented servers due to their low price. The short-term rented servers provide the extra capacity in peak period. The detail of the scheme is shown in Algorithm 1.

The proposed DQG scheme adopts the traditional FCFS queueing discipline. For each service request entering the system, the system records its waiting time. The requests are assigned and executed on the long-term rented servers in the order of arrival times. Once the waiting time of a request reaches D, a temporary server is rented from infrastructure.

Algorithm 1 Double-Quality-Guaranteed (DQG) Scheme

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.**

Performance comparison

Using resource renting scheme, temporary servers are rented for all request whose waiting time are equal to deadline, which in turn can guarantee that all request are served with high service quality. Hence proposed scheme is superior to traditional resource renting scheme in terms of service quality. Next a series of calculation has been done to compare profit of proposed scheme and other existing scheme. In order to distinguish the proposed scheme and the compared scheme, the proposed system has been renamed as Double quality guaranteed renting scheme and compared scheme is renamed as single renting scheme.

V. CONCLUSIONS

Proposed system has adopted a novel scheme of Double-Quality-Guaranteed for service providers in order to maximize profit of service providers and obviously enhancing quality of service for customers. This methodology has gathered both short term and long term in only one scheme. It has capability of adapting dynamic demand and reducing resource waste greatly. Earlier an M/M/m queueing model was being used which was not guaranteeing quality of service. But proposed system came up with idea of M/M/m+D queueing model. An M/m/m+D queueing model is used in a multiserver environment which have systems with scalable load.

After adopting this model, an optimal configuration problem is formulated for profit maximization. For formulating this problem many factors will be taken in account such as demand in market, request workload, server level agreement, rental cost of servers, energy consumption cost, and so on. Optimal solutions will be given for two scenarios, one for ideal optimal solution and other for actual optimal solution. With these two solutions, a series of calculation will be performed. Calculations will compare profit obtained by DQG renting scheme and SQG renting scheme. Finally, results show that proposed system has a better profit and quality of service.

REFERENCES

[1] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and Cloud Computing*. Elsevier/Morgan Kaufmann, 2012.

[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] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences*, vol. 28, 2009.

[4] 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 Gener. Comp. Sy.*, vol. 25, no. 6, pp. 599– 616, 2009.

[5] P. Mell and T. Grance, "The NIST definition of cloud computing. national institute of standards and technology," *Information Technology Laboratory*, vol. 15, p. 2009, 2009.

[6] J. Chen, C. Wang, B. B. Zhou, L. Sun, Y. C. Lee, and A. Y. Zomaya, "Tradeoffs between profit and customer satisfaction for service provisioning in the cloud," in *Proc. 20th Int'l Symp. High Performance Distributed Computing*. ACM, 2011, pp. 229–238.

[7] C T Lin "Comparative based analysis of scheduling algorithm of resource management in cloud computing environment" vol 1, issue 1, July 2013, IJCSE issue.

AUTHORS PROFILE

Raushan Kashypa received B.Tech.in computer science and engineering from RNSIT college affiliated to VTU, Belgavi, Bangalore, Karnataka in 2012. He is currently doing M. Tech. in computer science and engineering from City engineering college, Bangalore, Karnataka during 2014-2016.

Sowmya Naik P.T. received her B.Tech. in computer science from City engineering college, affiliated to VTU, Bekgavi, Bangalore Karnataka in 2007. She received her M.Tech. in computer science and engineering from AMC college, affiliated to VTU, Belgavi in 2012. She is a member of ISTE (Indian Society for Technical Education) and also a member of AMIE (Associate Member of Institution of Engineers). She is currently doing research in wireless sensor network. She is associate professor in department of computer science and engineering, city college Bangalore.