Behavioral model to obtain profit with guaranteed quality of service in cloud computing.

[1]Anjali C.Tak, [2]Prof. S.R.Ghungrad

[1]M.E Student, [2]Professor
Computer Science
Matsyodari Shikshan Sanstha's
College of Engineering and Technology, Jalna, Aurangabad.

***Abstract*—As an efficient and effective way to provide IT resources and services to customers on demand, cloud computing has become increasingly popular. From the point of view of cloud service providers, profit is one of the most important considerations and is primarily determined by the configuration of a cloud service platform under market demand. However, a single long-term leasing system is usually adopted to configure a cloud platform, which cannot guarantee service quality but results in severe resource losses. In this article, a dual resource rental system is designed first of all in which short-term leasing and long-term leasing are combined by targeting existing issues. This dual rental system can effectively guarantee the quality of service of all applications and significantly reduce waste of resources. Second, a service system is considered as a model of M / M / m + D queuing and the performance indicators that affect the benefit of our dual rental system are analyzed, for example, the average load, the ratio of Requests that need temporary servers and so on. Third, a profit maximization problem is formulated for the double lease scheme and the optimized configuration of a cloud platform is obtained by solving the profit maximization problem. Finally, a series of calculations are made to compare the benefit of our plan plan to the one-time rent plan. The results show that our system can not only guarantee the quality of service of all applications, but also obtain more profit than the latter.**

***IndexTerms*—Cloud computing, guaranteed quality of service, multi-server system, profit maximization, queuing model, service level agreement.**

I. Introduction

Prompted by the major industrial companies, cloud computing has recently raised concerns. With a growing number of cloud service providers (CSPs) providing services to cloud customers, maximizing the benefits of CSPs becomes a critical issue. Existing approaches are difficult to solve because they do not make full use of temporal price differences. This paper presents a dynamic approach to leasing virtual resources that attempts to dynamically adjust the strategy for leasing virtual resources according to the distribution of prices and the urgency of the task. Given the urgency of the task and the distribution of prices, we design a weak equilibrium operator to calculate the acceptable price for each type of virtual resource. All types of virtual resources that are at an acceptable price are inserted into a set. Then, a price prediction algorithm is presented to predict the price of virtual resources at the next price interval. Finally, we design a new rental decision algorithm to select the most cost-effective resource in the set. We implemented our approach and realized experiments on real and synthetic data sets. The results show that our approach gets the best benefit from the other five approaches.

Many plans have been designed to increase the benefit of the PUC. There are varieties of virtual resources and price models in the cloud computing environment; This makes it difficult for the PSC to choose the most cost-effective VRS. Framework for selecting the Virtual Resouce Selector is called CloudCmp. This framework uses a set of tools to predict cost and performance when the same application is deployed on virtual resources using different VRS. It is used to minimize the cost of running the workflow on the cloud computing environment. The work takes into account the costs of calculating and transmitting data. A heuristic algorithm has been proposed to solve the problem.

Regardless of temporal price differences, the aforementioned schemes are difficult to maximize profits in the dynamic pricing model of virtual resources. In the dynamic pricing model, the price of the resource is not fixed but changes dynamically and periodically according to current demand and supply. Therefore, it generally determines the current prices of virtual resources through an auction.

II. Related Work

In this section, we review recent work relevant to cloud service providers. The benefit of service providers is related to many factors such as price, market demand, system configuration, customer satisfaction and so on. Service providers naturally want to set a higher price for a higher profit margin; But doing so would reduce customer satisfaction, which could discourage demand in the future. Therefore, the choice of a reasonable pricing strategy is important for service providers. Pricing strategies are divided into two categories: static pricing and dynamic pricing. Static price means that the price of a service request is fixed and known in advance, and it does not change with the conditions. With a dynamic price, a service provider delays the pricing decision until the customer's request is revealed, so that the service provider can adjust prices accordingly [9]. Static pricing is the dominant strategy that is widely used in the real world and in research [2, 10, 11]. Ghamkhari et al. [11] adopted a flat rate pricing strategy and set a flat price for all applications, but Odlyzko [12] argued that predominant flat rate pricing encourages waste and is inconsistent with service differentiation. Another type of static pricing strategies are price based on usage. For example, the price of a service request is proportional to the service time and the task execution requirement (measured by the number of instructions to be executed) in [10] and [2], respectively. Use-based pricing reveals that resources can be used more efficiently [13, 14].

Dynamic pricing appears as an attractive alternative to better cope with unpredictable customer demand [15]. Mac'ıas et al. [16] used a genetic algorithm to iteratively optimize price policy. Amazon EC2 [17, 18] introduced a "spot price" function, where the spot price for a virtual instance is dynamically updated to match supply and demand. However, consumers do not like prices to change, especially if they perceive that the changes are "unjust" [19, 20]. After comparison, we select the use-based pricing strategy in this document because it fully agrees with the concept of cloud computing. The second factor that affects the benefit of service providers is customer satisfaction which is determined by the quality of service and burden. To improve the level of customer satisfaction, there is a Service Level Agreement (SLA) between a service provider and customers. SLA adopts a price offset mechanism for low quality service customers. The mechanism is to guarantee the quality of service and customer satisfaction so that more customers are attracted. In previous research, different SLAs have been adopted.

Ghamkhari et al. [11] has adopted a progressive load function with two steps. If a service request is processed before its deadline, it is normally debited; But if a service request is not processed before its deadline, it is dropped and the supplier pays for it due to a penalty. In [2, 10, 21], the load decreases continuously with increasing waiting time until the load is free. In this paper we use a two-step charging function, where high quality service requests are normally loaded, otherwise, are served free of charge. Since profit is an important concern for cloud service providers, much work has been done on how to boost their profits. A large number of books have recently focused on reducing the cost of energy to increase the benefits of service providers [22, 23, 24, 25]. However, only the reduction in the cost of energy can not maximize profit. Many researchers have studied the trade-off between minimizing costs and maximizing revenues to maximize profits. Both [11] and [26] adjusted the number of servers switched periodically using different strategies and different profit maximization models were constructed to get the number of switched servers. However, this work did not take into account the cost of configuring resources.

Chiang and Ouyang [27] considered a cloud server system as a M / M / R / K queuing system where all service requests that exceed its maximum capacity are rejected. A profit maximization function is can be defined as a process to obtain optimal combination for the server size R with a queue with capacity denoted by K in such a manner that the obtained profit is thoroughly maximized. However, this strategy has other implications than the mere loss of revenues from certain services, as it also involves a loss of reputation and thus a loss of future customers [3]. In [2], Cao et al. Processed a cloud service platform as an M / M / m model and the problem of optimal multiserver configuration for profit maximization was formulated and resolved. This work is the most relevant work for us, but it adopts a unique leasing scheme to configure a multiserver system that can not adapt to the changing market demand and leads to low quality of service and Waste of resources. To overcome this weakness, another resource management strategy is used in [28, 29, 30, 31], which is the federation of clouds. Using the federation, the different providers that perform services that have complementary resource needs over time can work together to share their respective resources in order to meet the demands of each. However, providers must make an intelligent decision about the use of federation (as a contributor or consumer of resources) depending on the different conditions they might face, which is a complex issue. In this article, to overcome the aforementioned shortcomings, a dual rental system is designed to configure a cloud service platform, which can guarantee the quality of service of all requests and significantly reduce the waste of resources. In addition, a profit maximization problem is formulated and solved to obtain the optimal multiserver configuration that can produce more benefit than the optimal configuration in [2].

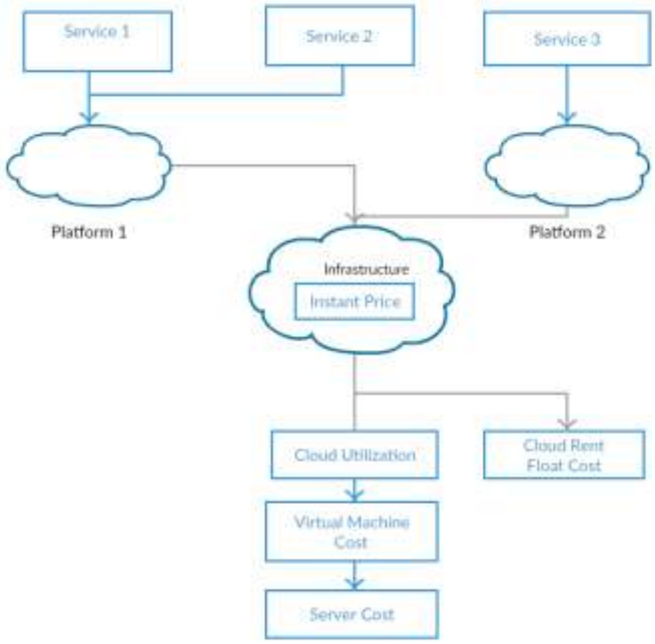
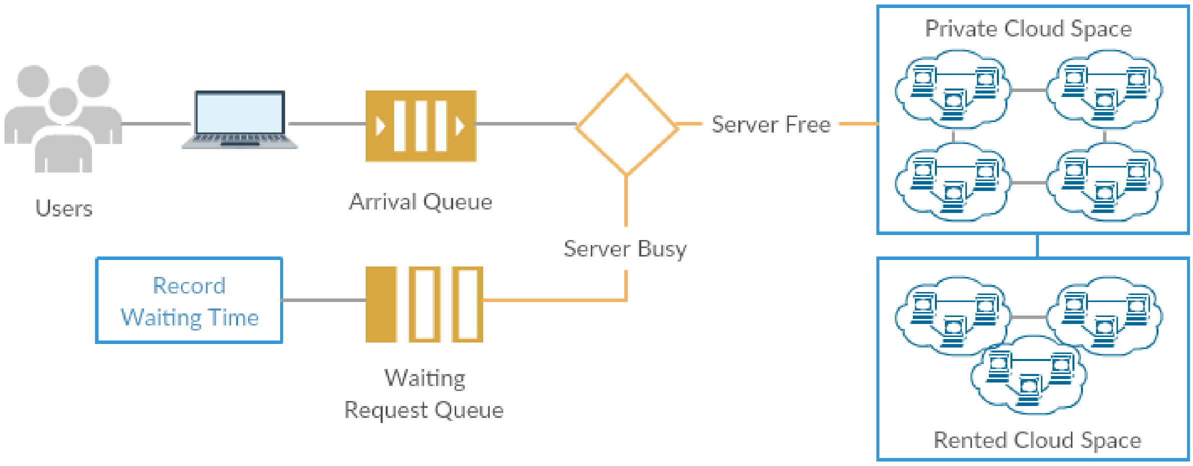


Figure 1.0 Cloud Accounting Model

III. Proposed Architecture

In this section, we first propose the Improvised Quality Guaranteed (IQG) resource leasing system that combines long-term leasing with short-term leasing. The main computing capacity is provided by servers leased long term due to their low price. Short-term leased servers provide additional capacity during peak periods. A novel double renting scheme is proposed for service providers. It combines long-term renting with short-term renting, which can not only satisfy quality-of-service requirements under the varying system workload, but also reduce the resource waste greatly. A multiserver system adopted in our paper is modeled as an M/M/m+D queuing model and the performance indicators are analyzed such as the average service charge, the ratio of requests that need short- term servers.

Assume that that a cloud service platform consists of m long term rented servers. It is known that part of requests need temporary servers to serve, so that their quality can be guaranteed. Denoted by pext(D) the steady-state probability that a request is assigned to a temporary server, or put differently, pext(D) is the long-run fraction of requests whose waiting times exceed the deadline D. pext(D) is different from FW(D). In calculating FW(D), all service requests, whether exceed the deadline, will be waiting in the queue. However, in calculating pext(D), the requests whose waiting times are equal to the deadline will be assigned to the temporary servers, which will reduce the waiting time of the following requests. In general, pext(D) is much less than FW(D).



Proposed Architecture

Cloud Computing

Cloud computing describes a type of outsourcing of IT services, similar to the way in which the electricity supply is outsourced. Users can simply use it. They do not need to worry about where electricity comes from, how it is made or transported. Each month they pay for what they consume. The idea behind cloud computing is similar: the user can simply use storage, computing power or specially designed development environments without having to worry about how they work internally. Cloud computing is usually Internet-based computing. The cloud is a metaphor for the Internet based on how the Internet is described in computer network diagrams; Which means that it is an abstraction that hides the complex infrastructure of the Internet. It is a computing style in which IT-related capabilities are provided "as a service", allowing users to access Internet technology services ("in the cloud") without knowledge or control over the technologies behind these servers.

Queuing Model

We consider the cloud service platform as a multi-server system with a service request queue. Clouds provide resources for jobs in the form of a virtual machine (VM). In addition, users send their jobs to the cloud using a queuing system like SGE, PBS or Condor. All jobs are scheduled by the task scheduler and assigned to different virtual machines centrally. Therefore, we can consider it as a service request queue. For example, Condor is a specialized workload management system for intensive computing jobs and provides a queuing mechanism, scheduling policy, priority scheme, resource monitoring and resource management. Users submit their work to Condor, and Condor puts them in a queue, chooses when and where to run them based on a policy. An M / M / m + Dqueuing model is built for our multi-server system with different system sizes. And then, an optimum benefit maximization configuration problem is formulated in which many factors are taken into account, such as market demand, workload of requests, server-level agreement, cost of renting Servers, the cost of energy consumption and, go ahead. Optimal solutions are solved for two different situations, which are optimal ideal solutions and real optimal solutions.

Business Service Providers Module

Service providers pay infrastructure providers for leasing their physical resources and charge customers for processing their service requests, which generates costs and revenues, respectively. The benefit is generated from the gap between income and cost. In this module, service providers are considered as cloud intermediaries, as they can play an important role among cloud customers and infrastructure providers, and can establish an indirect connection between cloud customers and cloud providers. infrastructure

Cloud Consumers & Customers

A client sends a service request to a service provider that provides on-demand services. The customer receives the desired result from the service provider with some level of service agreement and pays the service according to the amount of service and quality of service.

The detail of the scheme is illustrated in the algorithm 1.

```
Algorithm 1 Improvised-Quality-Guaranteed (IQG) Scheme
A System with multiple clusters m waiting to process requests R
Initialize Arrival Aq - Queue as empty
Initialize Waiting Q Wq as empty

Case: New request arrival
Step 1 Add request to Arrival Queue Aq
Access Cluster Monitor Process CMP and obtain current status of Clusters
If cluster clstr is free then
assign request from arrival queue Aq to Cluster processing queue Pqn
if cluster Clstr is busy
Step 2 Obtain request REQ waiting time i.e TTL ( Time To Live)
push request REQ into waiting queue.
Monitor clusters Clstr to seek if it gets empty
If cluster Clstr becomes empty
Step 3 obtain request REQ with minimum waiting time i.e. Time To Live
push request REQ towards processing queue of cluster
if clusters Clstr not empty && waiting time is near to expire
Step 4 Rent a temporary server cluster Clstr for request REQ and process request REQ and release the temporary server when the request is completed .
repeat for all requests until request queue is empty.
END
```

The proposed IQG program adopts the traditional spinning discipline of FCFS. For each service request entering the system, the system records its waiting time. Applications are assigned and executed on long-term leased servers in the order of arrival times. Once the request timeout reaches D, a temporary server is leased from infrastructure providers to process the request. We consider the new service model as a M / M / m + D queuing model. The M / M / m + D model is a special model of M / M / m queuing with Customers. In an M / M / m + D model, requests are impatient and have a maximum tolerable waiting time. If the waiting time exceeds the tolerable waiting time, they lose patience and leave the system. In our program, impatient requests do not leave the system but are assigned to servers leased temporarily. Because requests with timeout D are all assigned to temporary servers, it is clear that all service requests can guarantee their due date and are billed according to the SLA workload. As a result, the service provider's revenue increases. However, the cost also increases due to temporarily leased servers. In addition, the amount of cost spent on leasing temporary servers is determined by the long-term leased multi-server IT capacity. Given that income has been maximized by using our plan, minimizing cost is the key issue for profit maximization. Then, the trade-off between long-term rental cost and short-term leasing cost is considered, and an optimal problem is formulated in the following way to obtain an optimal long-run configuration so that profit is maximized.

Time Complexity of IQG

The time complexity of the IQLB algorithm is O (nm), where n is the number of nodes in the cluster. M is the number of jobs in the application and the value Of n and m are larger than 2

Prove to take O (1) to compute the response time of the task on the node. The time complexity of determining whether an internal node is overloaded is O (n) because there are n nodes in the cluster.

Steps 4 and 4 take O (1), so the complexity of time to balance the I / O resource of the disk is O (2 + 2n)

Similarly, the complexity of time to balance memory and CPU resources is both O (2 + 2n) due to the m work in parallel applications. The time complexity of the IOLB algorithm is O (2 + 2n) O (m) = O (2 (1 + n) m) The values of n and m in most cases are larger than 2, so the time complexity becomes O (nm).

IV. Conclusion

Maximizing the benefit of service providers, this article proposed a new dual-rate rent guarantee (DQG) system for service providers. This system combines short-term lease with long-term lease, which can significantly reduce the waste of resources and adapt to the dynamic demand for IT capacity. An M / M / m + Dqueuing model is built for our multi-server system with variable system size. Then, an optimum benefit optimization configuration problem is formulated which takes into account many factors, such as market demand, workload of requests, server level agreement, server rental cost , The cost of energy consumption. The optimum solutions are solved for two different situations, which are ideal optimal solutions and optimal solutions. In addition, a series of calculations are made to compare the benefits of the DQG lease system with the Single Unsecured Lease Scheme (SQU). The results show that our scheme is superior to the SQU scheme in terms of quality of service and profit.

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.

[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,"

[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.

[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,".

[7] J. Mei, K. Li, J. Hu, S. Yin, and E. H.-M. Sha, "Energyaware preemptive scheduling algorithm for sporadic tasks on dvs platform," MICROPROCESS MICROSY., vol. 37, no. 1, pp. 99–112, 2013.

[8] P. de Langen and B. Juurlink, "Leakage-aware multiprocessor scheduling," J. Signal Process. Sys., vol. 57, no. 1, pp. 73–88, 2009.

[9] G. P. Cachon and P. Feldman, "Dynamic versus static pricing in the presence of strategic consumers," Tech. Rep., 2010.

[10] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou, "Profitdriven scheduling for cloud services with data access awareness," J. Parallel Distr. Com., vol. 72, no. 4, pp. 591– 602, 2012.

[11] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: a profit maximization approach," IEEE Trans. Smart Grid, vol. 4, no. 2, pp. 1017–1025, 2013.

[12] A. Odlyzko, "Should flat-rate internet pricing continue," IT Professional, vol. 2, no. 5, pp. 48–51, 2000.

[13] G. Kesidis, A. Das, and G. de Veciana, "On flat-rate and usage-based pricing for tiered commodity internet services," in 42nd Annual Conf. Information Sciences and Systems. IEEE, 2008, pp. 304–308.

[14] S. Shakkottai, R. Srikant, A. Ozdaglar, and D. Acemoglu, "The price of simplicity," IEEE J. Selected Areas in Communications, vol. 26, no. 7, pp. 1269–1276, 2008.

[15] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," IEEE Trans. Cloud Computing, vol. 1, no. 2, pp. 158–171, July 2013.