REVENUE OF LIMIT FOR CLOUD AGENT IN CLOUD COMPUTING

Divya M[1] , Mr. M Vazralu[2], Dr.Jayapal Medida[3]

[1] PG Student, Malla Reddy College of Engineering and Technology, Hyderabad.

[2]Assoc professor, Dept of IT, Malla Reddy College of Engineering and Technology, Hyderabad.

[3]Assoc professor, Dept of CSE, Malla Reddy College of Engineering and Technology, Hyderabad.

ABSTRACT:

We attempt to plan a provider mechanism for income optimizations of each a cloud issuer and its more than one users. We think about the hassle from a sport theoretic standpoint and signify the relationship between the cloud issuer and its more than one customers as a Stackelberg game, in which the techniques of all customers are issue to that of the cloud provider. The cloud company tries to pick and provision fabulous servers and configure a appropriate request allocation method to limit power value whilst pleasing its cloud users at the equal time. We approximate its servers resolution house by means of including a controlling parameter and configure an best request allocation strategy. For every user, we format a utility characteristic which combines the internet earnings with time effectivity and attempt to maximize its price below the method of the cloud provider. We formulate the competitions amongst all customers as a generalized Nash equilibrium trouble (GNEP). We clear up the trouble by means of using version inequality (VI) principle and show that there exists a generalized Nash equilibrium answer set for the formulated GNEP. Finally, we advise an iterative algorithm (IA), which characterizes the total procedure of our proposed provider mechanism. We behavior some numerical calculations to confirm our theoretical analyses. The experimental outcomes exhibit that our IA algorithm can gain each of a cloud company and its more than one customers through configuring ideal strategies.

Keywords: Cloud computing, Generalized Nash equilibrium, Non-cooperative game theory, Profit optimization, Resource allocation, Variational inequality theory.

1. INTRODUCTION:

Cloud computing is an more and more famous paradigm of presenting subscription-oriented offerings to corporations and buyers [1]. Usually, the furnished offerings refer to Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), which are all made on hand to the regular public in a pay-as-you-go manner [2], [3]. To assist more than a few services, greater and extra cloud facilities are geared up with hundreds of computing nodes, which consequences in super power value [4]. It is pronounced that about 50% administration finances of Amazon′ s facts middle is used for powering and colling the bodily servers [5]. There are additionally researchers who have studied the value of records facilities and concluded that round 40% of the amortized fee of a statistics middle falls into energy associated classes [6]. Hence, it is vital to minimize power value for enhancing the earnings of a cloud provider. However, it can regularly be considered that there are many under-utilized servers in cloud centers, or on the contrary, cloud companies supply much less processing potential and for this reason dissatisfy their customers for bad carrier quality. Therefore, it is essential for a cloud issuer to choose fabulous servers to grant services, such that it reduces price as an awful lot as feasible whilst fulfilling its customers at the identical time. For a cloud provider, the profits (i.e., the revenue) is the carrier cost to the aggregated requests from all cloud customers [7]. When the per request cost is determined, servers choice and request allocation approach are two widespread elements that have to be taken into account. The cause in the back of lies in that each of them are no longer simply for the earnings of a cloud provider, however for the appeals to extra cloud customers in the market to use cloud carrier and hence additionally influence the profit. Specifically, if the supplied computing potential is giant adequate (i.e., many servers are under-utilized), this will end result in top notch quantity of electricity waste with massive value and accordingly reduces the income of the cloud provider. On the different hand, if the cloud issuer presents much less computing capability or improperly configures the request allocation strategy, this will lead to low carrier satisfactory (e.g, lengthy assignment response time) and as a consequence dissatisfies its cloud customers or manageable cloud customers in the market. A rational consumer will select a method to use the provider that maximizes his/her personal internet reward, i.e., the utility received with the aid of selecting the cloud carrier minus the fee [8]. In addition, the utility of a consumer is no longer solely decided via the internet earnings of his/her requests (i.e., how a lot gain the consumer can get hold of by means of ending the configured tasks), however additionally carefully associated to the urgency of the duties (i.e., how rapidly they can be finished). The identical quantity of duties are in a position to generate extra utility for a cloud consumer if they can be finished inside a shorter duration of time in the cloud middle [8]. However, thinking about from power saving and monetary reasons, it is irrational for a cloud company to grant adequate computing assets to whole all requests in a quick length of time. Therefore, more than one cloud customers have to configure the quantity of requests in extraordinary time slots. Since the requests from customers are submitted randomly, in our paper, we about symbolize the request arrivals as a Poisson method [9]. Since the fee and time effectivity of every of the cloud users are affected through the choices of others, it is herbal to analyze the behaviors of these customers as strategic video games [10]. In this paper, we attempt to graph a new carrier mechanism for income optimizations of each a cloud issuer and its a couple of users. We reflect onconsideration on the hassle from a sport theoretic standpoint and symbolize the relationship between the cloud issuer and its customers as a Stackelberg game, in which the techniques of all customers are challenge to that of the cloud provider. In our mechanism, the cloud company tries to pick excellent servers and configure a appropriate request allocation method to decrease electricity value whilst gratifying its customers at the identical time.

The primary contributions of this paper are listed as follows.

- We symbolize the relationship between the cloud issuer and its customers as a Stackelberg game, and attempt to optimize the earnings of each a cloud issuer and its customers at the identical time.
- We formulate the competitions amongst all customers as a generalized Nash equilibrium trouble (GNEP), and show that there exists a generalized Nash equilibrium answer set for the formulated GNEP.
- We remedy the GNEP by means of using varational inequality (VI) concept and advise an iterative

algorithm (IA) to symbolize the total method of our proposed provider mechanism. Experimental effects exhibit that our IA algorithm can gain each of the cloud issuer and its more than one customers via configuring appropriate strategies.

TERMINOLOGY AND PROBLEM STATEMENT

In general, a service provider rents a certain number of servers from the infrastructure providers and builds different multi-server systems for different application domains. Each multiserver system is to execute a special type of service requests and applications. Hence, the renting cost is proportional to the number of servers in a multiserver system. The power consumption of a multiserver system is linearly proportional to the number of servers and the server utilization, and to the square of execution speed. The revenue of a service provider is related to the amount of service and the quality of service. To summarize, the profit of a service provider is mainly determined by the configuration of its service platform.

To configure a cloud service platform, a service provider usually adopts a single renting scheme. That's to say, the servers in the service system are all long-term rented. Because of the limited number of servers, some of the incoming service requests cannot be processed immediately. So they are first inserted into a queue until they can handle by any available server.

The waiting time of the service requests is too long.

Sharp increase of the renting cost or the electricity cost. Such increased cost may counterweight the gain from penalty reduction. In conclusion, the single renting scheme is not a good scheme for service providers.

PROPOSED TECHNOLOGY

Since the requests with waiting time D are all assigned to temporary servers, it is apparent that all service requests can guarantee their deadline and are charged based on the workload according to the SLA. Hence, the revenue of the service provider increases.

Increase in the quality of service requests and maximize the profit of service providers.

This scheme combines short-term renting with long-term renting, which can reduce the resource waste greatly and adapt to the dynamical demand of computing capacity.

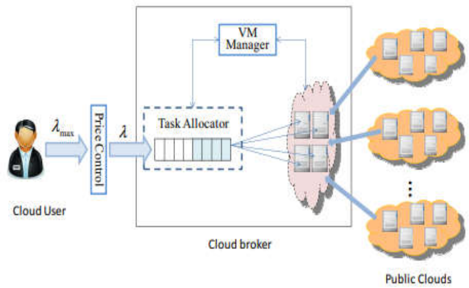

Figure 1: The M/M/n/n queue model.

The price of the on-demand VMs provided by the cloud broker be β per unit of time. The price affects the revenue of a cloud broker from two aspects. First, the price has a direct impact on revenue. Under a given demand, a higher price conducts a higher revenue. Second, the price affects the revenue indirectly. The explanations are given as follows. The cloud broker rents reserved instances from cloud providers with a discount compared with the on demand instances and outsources them as on-demand VMs in a lower price than the same VMs provided by cloud providers. The low price is the core competitive advantage of the cloud broker, and its objective customers are those customers whose service requests are submitted occasionally and the execution time is uncertain or short. This portion of customers are inclined to rent on-demand VMs rather than reserved VMs, but they also want to enjoy the discount that the cloud providers provide for long-term customers. The cloud broker can provide customers the needed resources at a lower price. Since the main advantage for the cloud broker to attract customers is its lower price compared with public clouds, the price certainly will affect the request arrival rate, thus affecting revenue, corresponding. Hence, proper pricing is an important issue for the cloud broker. To obtain profit, the VM sales price of the cloud broker should be greater than its cost price obviously; that is, the rental price that the cloud broker rents reserved instances from cloud providers. Meanwhile, the VM sales price should be lower than the on-demand price of cloud providers to attract customers. That is because customers are inclined to select the services of public clouds when the VM sales price of the cloud broker is same as public clouds. To sum up, the VM sales price of the broker, denoted as β, should be between the range of $[\beta_{re}, \beta_{od}]$.

Algorithm 1 Finding the optimal price

Input: λmax, t, n, β_{re}, β_{od}, p_{re}, and p_{od};

Output: optimal price opt β of resources and optimal profit opt pro;

1: opt $\beta = -\infty$, opt _pro $= -\infty$;

2: $\beta_{start} \leftarrow$ the minimal price satisfying $\rho < 1$;

3: $\beta_{end} \leftarrow \beta_{od}$;

4: calculate Der_{start} and Der_{end} ;

if $Der_{start} \times Der_{end} > 0$ then

opt $\_\beta = \beta_{start}$;

calculate opt_{pro};

exit;

end if

while $Der_{start} - Der_{end} >$ error do

$\beta_{middle} = (\beta_{start} + \beta_{start})/2$;

calculate Der_{middle}

if $Der_{start} \times Der_{middle} > 0$ then

$\beta_{start} \leftarrow \beta_{middle}$;

else

$\beta_{end} \leftarrow \beta_{middle}$;

end if

end while

opt $\beta = (\beta start + \beta end)/2$;

Profit in one unit of time as a function of n and λ_{max}. Therefore, for each combination of n and λ_{max}, we find the optimal price for a cloud broker and the corresponding maximal profit it can obtain. The parameters are set to be same as . From the figures, we can see that under a given λ_{max}, the optimal price is decreasing with the increase in system size. This is explained as follows. It is obvious that more VMs lead to more cost. To utilize the resources sufficiently and improve the revenue, the VM price is lowered to attract more customers, which is so-called small profits but quick turnover (SPQT) strategy. However, the optimal profit is not monotone increasing with the increasing system size. When the system size reaches a certain point, the extra cost conduct by increasing VMs further starts to exceed the increased revenue by adopting the SPQT strategy. Hence, the total profit increases at the early stage and then decreases. Moreover, the figures show that the optimal price and the optimal profit are all related with the λ_{max}. Under a given system size, a greater λ_{max} will lead to a higher optimal price and more profit.

In Alg. 1, the partial derivative is calculated based on the estimation value of PL first, and then the extremal solutions are solved using the bisection search method. Hence, the solutions obtained by **Alg. 1** have a certain of error with the precise solutions. To verify the precision of the solutions, we compare the optimal solutions obtained by our method with that obtained by a brute force search method. The comparison results are given. In the comparison, the System size n is set from 50 to 450 in step of 50, λ_{max} is set as 100, and other parameters are same. From the results, we can see that the error is less than 2% when the n is greater than 200. When the n is smaller than 200, with the decrease of n, the error becomes greater. That is because the error between the estimation value and precision value of PL is very large when n is small.

Algorithm 2 Iterative Algorithm (IA)

Input: ε, μ, a, b, r, τ,M

Output: S, pS .

1: **Initialization:** The cloud provider approximates its solution space, i.e., Q (ε) L ← Calculate Q (ε) L (ε, c, μ, E,M). Set πS ← 0.

2: for (each server subset S ∈ Q ˜ (ε) L) do

3: Set Sc ← N , and Sl ← ∅.

4: for (each time slot h ∈ H) do

5: for (each server j ∈ S˜) do

6: Set p h j = μj/ (∑ j∈S˜ μj) .

7: end for

8: end for

9: while (Sc ≠ Sl) do

10: Set Sl ← Sc, and λ ← Calculate λ(ε, S, pS , τ).

11: for (each time slot h ∈ H) do

12: Set p h S˜ ← Calculate pS˜ h (ε, μ, λ h Σ, S) .

13: end for

14: for (each user i ∈ Sc) do

15: if (Ui (λ (k) i ,λ (k) Σ) < vi) then

16: Set λi ← 0, and Sc ← Sc − {i}.

17: end if

18: end for

19: end while

20: Set πS˜ ← c ∑ i∈N ∑ h∈H λ h i − ET (S˜) .

21: if (πS˜ > πS) then

22: Set πS ← πS˜, S ← S˜, and pS ← pS˜.

23: end if

24: end for

25: return S, pS .

CONCLUSION

We focal point on the income maximization hassle of cloud brokers. A cloud broking is an middleman entity between cloud carrier carriers and customers, which buys reserved situations from cloud carriers for lengthy intervals of time and outsources them as on-demand VMs for a decrease charge and finegrained BTU with recognize to what the cloud provider companies cost for the identical VMs. Due to the decrease provider fee and the finer-grained BTU in contrast with the public clouds, the cloud dealer can retailer an awful lot value for customers. This paper tries to information cloud brokers on how to configure the digital aid platform and how to fee their carrier such that they can gain the maximal profit. To resolve this problem, the digital aid platform is modeled as an M/M/n/n queue model, and a income maximization trouble is constructed in which many profit-affecting elements are analyzed primarily based on the queuing theory, as properly as the relationship between them. The most efficient options are solved combining the partial by-product and bisection method. Lastly, a sequence of calculations are performed to analyze the altering style of income and the ratio of person fee savings.

REFERENCES:

[1] A. Prasad and S. Rao, “A mechanism design approach to resource procurement in cloud computing,” Computers, IEEE Transactions on, vol. 63, no. 1, pp. 17–30, Jan 2014.

[2] R. Pal and P. Hui, “Economic models for cloud service markets: Pricing and capacity planning,” Theoretical Computer Science, vol. 496, no. 0, pp. 113 – 124, 2013.

3] P. D. Kaur and I. Chana, “A resource elasticity framework for qos-aware execution of cloud applications,” Future Generation Computer Systems, vol. 37, no. 0, pp. 14 – 25, 2014.

[4] L. Duan, D. Zhan, and J. Hohnerlein, “Optimizing cloud data center energy efficiency via dynamic prediction of cpu idle intervals,” in 2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015, pp. 985–988.

[5] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, “Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds,” IEEE Transactions on Services Computing, 2015, doi: 10.1109/TSC.2015.2466545.

[6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: research problems in data center networks,” ACM SIGCOMM computer communication review, vol. 39, no. 1, pp. 68– 73, 2008.

[7] J. Cao, K. Hwang, K. Li, and A. Zomaya, “Optimal multiserver configuration for profit maximization in cloud computing,” Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 6, pp. 1087–1096, June 2013.

[8] Y. Feng, B. Li, and B. Li, "Price competition in an oligopoly market with multiple iaas cloud providers," Computers, IEEE Transactions on, vol. 63, no. 1, pp. 59–73, Jan 2014.

[9] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," Computers, IEEE Transactions on, vol. 63, no. 1, pp. 45–58, Jan 2014.

[10] S. Jrgensen and G. Zaccour, "A survey of game-theoretic models of cooperative advertising," European Journal of Operational Research, vol. 237, no. 1, pp. 1 – 14, 2014.