A Reasonable Solution for Dynamic Scheduling and On-demand DistribUtion of Business Resources in Cloud Resource Management Platform

Wang Tingting, Yu Cong, Qian Hengshun, Liu Hui, Fan Jiang

NARI Group co., LTD., Nanjing, Jiangsu 211100, China

Keywords: Cloud computing resource, allocation mechanism, game theory.

Abstract: The traditional cloud computing resource allocation mechanism is allocated according to the user's ordering demand, and the user pays the corresponding fee according to the usage. With the development of cloud computing business and the increasing diversification of users' demands, cloud resource providers began to try different cloud resource allocation mechanisms and take into account users' personalized needs more reasonably. In this paper, it assumes that every cloud user is rational and intelligent, and it establishes a cloud resource allocation related model. It studies the cloud computing resource allocation mechanism with the help of game theory, mechanism design and relevant theories.

1. Introduction

The key technology to realize cloud computing is virtualization technology. Cloud users can be provided with a flexible and extensible service platform through virtualization. At present, there are many cloud computing systems in the form of virtual machines to meet the computing resource needs of users such as the current cloud computing system, etc.. With the application of virtualization technology, the allocation and management of virtual resources has become one of the hot issues in current research. There are multiple participants in a cloud resource allocation mechanism. When making decisions, each participant should not only consider his own strategic behavior, but also consider the strategic behavior of other participants. Therefore, it is advisable to adopt the theory of abandonment and mechanism design as theoretical tools to analyze and study the problem of cloud resource allocation mechanism. Reasonable cloud resource allocation mechanism enables cloud resources to be allocated to cloud users with corresponding requirements, which improves the utilization rate of cloud resources and avoids idle and waste of resources[1,2].

2. Game Theory

The concept of game theory. 1) Utility. Utility theory ensures that participants' preferences are expressed in terms of payments within a certain range of utility. Utility theory is an outcome based on allocation in a way that reflects the preferences of the participants. 2) Rationality. An important assumption in the game is that the user is rational. Assuming that the objective of each participant is to maximize his or her expected utility, the participant always makes decisions in pursuit of this objective, then the participant is considered rational. 3) Intelligence. Another key concept in game theory is that players are intelligent. The idea is that every player in a game knows something about the theory of the game, and the player can make any inferences about a game in which the theory is formulated. Each participant assumes sufficient computational resources and capabilities to perform the computation and determine the optimal response strategy. Then intelligence can be understood to mean that each player is an expert in game theory. 4) Public knowledge. If a fact is common knowledge among participants, it means that every participant knows it, every participant knows it and so on. That is to say, “every participant knows every participant knows.” If one participant has private information, that is to say, the information of the participant is not public knowledge for all participants[3].

  DOI: 10.25236/mmmce.2019.109

3. The Concept of Equilibrium

1) Strong dominant strategy. Given a game $\tau =< N,(S_i),(u_i) >$, If compared to other strategies $s_i \in S_i$, the strategy $s_i^* \in S_i$ of participant i has $u_i(s_i^*,s_{-i}) > u_i(s_i,s_{-i}),\forall s_i \neq s_i^*,\forall s_{-i} \in S_{-i}$, s_i^* is called a strong dominant strategy of participant i.

2) Strong dominant strategy equilibrium. Given a game $\tau =< N,(S_i),(u_i) >$, s_i^* is a strong dominant strategy of participant i, the strategy combination $(s_1^*,s_2^*,\dots,s_n^*)$ is called a strong dominant strategy equilibrium of game τ.

3) Weak dominant strategy. Given a game $\tau =< N,(S_i),(u_i) >$, If compared to other strategies $s_i \in S_i$, the strategy $s_i^* \in S_i$ of participant i has $u_i(s_i^*,s_{-i}) \geq u_i(s_i,s_{-i}),\forall s_i \neq s_i^*,\forall s_{-i} \in S_{-i}$, and there is at least one $s_{-i} \in S_{-i}$, $u_i(s_i^*,s_{-i}) > u_i(s_i,s_{-i})$, s_i^* is called a weak dominant strategy of participant i.

4) Weak dominant strategy equilibrium. Given a game $\tau =< N,(S_i),(u_i) >$, s_i^* is a weak dominant strategy of participant i, the strategy combination $(s_1^*,s_2^*,\dots,s_n^*)$ is called a weak dominant strategy equilibrium of game τ.

We know that a strongly dominant strategy equilibrium is naturally a weakly dominant strategy equilibrium from the definition. There may be multiple weakly dominant strategy equilibrium in a game. But for a strongly dominant strategy equilibrium, if it exists, it must be unique[4].

4. The Relationship between Mechanism Design and Game

Mechanism design is a special game of incomplete information and the process of mechanism design is establishing game rules. For example, the seller who owns the goods sets the quantity and price of the goods when selling them. From the perspective of game, he is designing or formulating a game rule. Buyers can make strategic choices according to the provisions of the seller on the quantity and price of goods to be sold. In real life, there are many examples of mechanisms such as how to set auction rules for commodities, how to set prices for monopolistic enterprises, how to set government tax policies and so on. A typical mechanism design can be divided into three stages of incomplete information game. In the first phase, the mechanism designer designs a mechanism. The mechanism here establishes the corresponding rules such as the rules of distribution and payment, which specify a game rule. In the second phase, participants also make a choice about whether or not to accept the mechanism designed by the mechanism designer. If the participant doesn't chooses to accept, he will get his external reservation utility. If the participant chooses to accept, he will enter the third stage, and the participant will play according to the rules made by the mechanism[5].

5. Hybrid Cloud Computing Workload Decomposition Model

1) Participants set. $n = (1,2,\dots,n)$ represents the set of n users within an enterprise, each user is rational and intelligent.

2) $S_i = \{\sigma_i | \sigma_i \in [0,1]\}$ represents the policy of user I, σ_i is the percentage of the total number of jobs assigned to the public cloud by user i. $1 - \sigma_i$ is the percentage of the total number of jobs assigned to the private cloud by user i. $\vec{\sigma} = (\sigma_1,\sigma_2,\dots,\sigma_n)$ is the combination of policies for all users.

3) ω_i is the task amount of user i, it indicates how many tasks the user needs to perform.

4) u_L is the speed at which tasks are processed in the private cloud and u_P is the speed at which tasks are processed in the public cloud.

5) T_i^L is the amount of time that user i spends executing tasks on the private cloud. T_i^P is the time spent on the task on the public cloud by user i, so $T_i^P = \sigma_i\omega_i/u_P$, and the time spent on the task by user i is the greater of T_i^L and T_i^P.

6) Assuming that, (1) according to the SLA between the cloud provider and the user, the public cloud provider can provide service guarantee to the user, and the user's task submission can always get the execution speed within a certain range provided by the public cloud. Therefore, it is assumed

that the speed μ_p of public cloud processing tasks provided by the public cloud to the user is a constant value. (2) the execution time of any user i on the private cloud is related not only to its own policies, but also to the execution time submitted locally by other users. Similar to literature,

$$T_i^L = \begin{cases} \frac{(1-\sigma_i)\omega_i}{u_L} + \sum_{k\neq i}\frac{(1-\sigma_k)\omega_k}{u_L} & \sigma_i \epsilon[0,1] \\ 0 & \sigma_i = 1 \end{cases}$$

6. Experimental Analysis

The Nash equilibrium strategy of user i is $\sigma_i^* = \frac{n}{n+p_0u_L(\frac{\alpha p_0 u_L}{\beta})^{\frac{1}{\gamma-1}}}, \forall i \in N$. It can be seen that the equilibrium strategy is related to the price of unit task quantity. When some parameter settings are changed, the equilibrium strategy will also change. The specific situation is shown in the diagram 1.

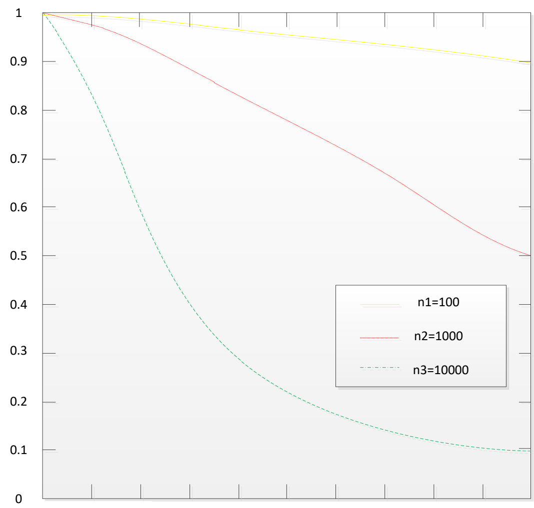

Fig. 1 The equilibrium strategy relates to price and number of users

There are three curves in the figure 1. The horizontal coordinate is the price per unit of task quantity, which is taken from 0 to 10. Other parameters have been set. When n=100, corresponding to the bottom curve, as the pricing of public cloud providers increases, the proportion of users' workload assigned to the public cloud gradually decreases. When other parameters set in the middle of the curve is the same case, the user of the total n increased 10 times, with the change of the price, the user submits to equilibrium strategies on public clouds and front obviously improved a lot, that is to say, when a user increases, local resources are difficult to meet the needs of users, users will put more tasks submitted to the public cloud. The top curve increases n by 10 times on the basis of the parameters of the middle curve. We can see that the user's balancing strategy is to divide almost all tasks into the public cloud, which also reflects the necessity and superiority of the public cloud in handling multi-user and multi-task.

Figure 2 and figure 3 are the curves of equalization strategy under parameters $\alpha:\beta = 10:1$ and $\alpha:\beta = 1:10$ respectively, indicating the degree of influence of time and expense on user's utility value. And the change of equalization strategy under different proportions of α and β with the change of unit task price p_0. As can be seen in figure 2, because the cost reduces the user's utility more quickly, as the price increases, the user does not will divide more work into the public cloud to execute. While figure 3 shows that the user's balancing strategy changes slowly as the price increases, that is to say, the cost reduces the user's utility at a slower rate.

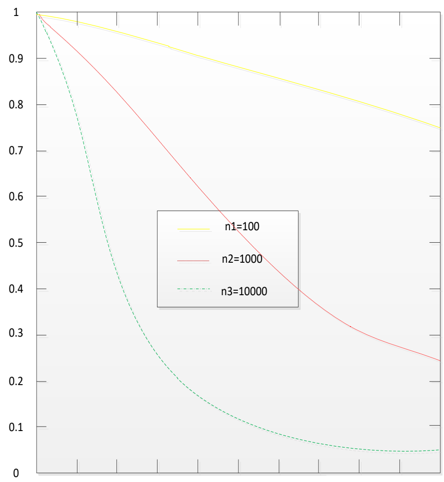

Fig. 2 $\alpha : \beta = 10 : 1$

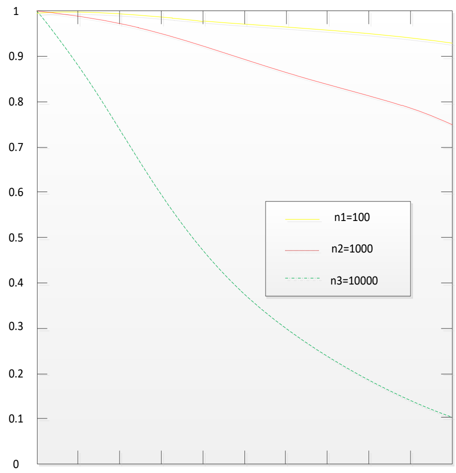

Fig. 3 $\alpha : \beta = 1 : 10$

7. Summary

This paper makes theoretical analysis and experimental simulation of the model of cloud computing resource allocation mechanism, and draws some good conclusions. However, users of cloud resources need to consider many factors such as security, fairness, trust, etc. in real life, and the model in this paper is constructed without considering too many factors. The mechanism design considers the influence of various factors on cloud users, which is a deeper research and an important research trend in the future.

References

[1] Upadhyaya P, Balazinska M, Suciu D. How to price shared optimizations in the cloud, Vol. 5 (2012) No.6, p.562-573.

[2] Calbeiros R N, Ranjan R, Beloglazov A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Vol. 41 (2011) No.1, p.23-50.

[3] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing, Vol. 53 (2010) No.4, p.50-58.

[4] Mell P, Grance T. The NIST definition of cloud computing, Vol. 53 (2009) No.6, p.50.

[5] Patel P, Ranabahu A H, Sheth A P. Service level agreement in cloud computing, 2009.