A Resource Trusted Model Based on Game Theory in Cloud Computing

Qinghong Shang*
School of Physics and Electronic Engineering Sichuan Normal University, Chengdu, Sichuan, China
*Corresponding author e-mail: shangqinghong@qq.com

Abstract. With the development of cloud computing, task scheduling has been an important problem in cloud computing. We should consider not only the performance of resources, but also the reliability of resources and users for resource allocation and scheduling. We proposed a trusted model based on game theory to evaluate the trusted scores of cloud participants. Simulation results show cloud users and resource tend to send feedback information to agents in order to increase their scores. Also trusted score is an important factor to affect the decision of resource allocation. Results show our model improved resource utilization and load balance.

1. Introduction
With the increasing development of cloud computing, task scheduling problem as a crucial aspect of cloud computing gets a lot of attention and development. It studies mapping requests from cloud users to resources provided by cloud service providers. The biggest benefit of cloud computing is to allocate resource as needed, which will greatly improve resource utilization and reduce the cost. Many studies which focus on task completion time and task overhead are proposed to optimize these objectives. [1] used a directed acyclic graph to describe the relationships of subtasks and communication cost, which makes separate resource allocation decisions. [2] proposed an algorithm based on the meta-heuristic optimization technique, which aims to minimize the overall workflow execution cost while meeting deadline constraints. [3] proposed a priority and load balance based method to achieve minimum response time. All these algorithms do not take resource reliability into consideration. We proposed a resource trusted model based on game theory in this paper. In addition to the evaluation of CPU performance, storage performance, bandwidth performance and cost of resources, the reliability of resources should also be evaluated as a feature of resources. The resource reliability is evaluated by scores from users. Meanwhile, resource also grades users according to their behaviors. It is a two-way evaluation mechanism. User tasks are allocated to the resource with high comprehensive score including reliability score. Resource providers evaluate users’ behaviors and provide differentiated services according to users’ scores.

Simulation results show the algorithm improved resource utilization and load balance. The main contribution of this paper is as bellows:

(1) Proposing a comprehensive evaluation mechanism for resources, which not only consider the performance of resources but also focus on their reliabilities.
(2) Using game theory to establish a trusted model, which marks the trusted level of resources and cloud users. Rewarding mechanism is used to improve resource utilization according to resource reliability and user reputation.

Section 2 introduces the related work. Section 3 shows the trusted model based on game theory. Section 4 and 5 show the related performance results and conclusion.

2. Related work

When people make full use of the convenience brought by cloud computing, the security and trusted problem in cloud computing becomes one of the important problems to be solved. When organizations or individuals consider whether to use the cloud, their primary concern is the credibility of service providers. The trusted model in cloud computing has been studied by many researchers. [4] proposed an algorithm which not only consider the reliability but also cost while allocating appropriate resources to the users. [5] studied a service-level agreement to protect data security of the cloud service user and improve the service requester's satisfaction. [6] proposed a hierarchal level trust model based on artificial bee colony algorithm. It helps cloud users to choose high trust value resource by feedback from other users. [7] proposed a recommendation mechanism to choose the optimal services based on both direct trust and recommendation trust. [8] focuses on inter-clouds for establishing trust in cloud computing environment. The aim is to promote the use of inter-clouds in cloud computing environment. Reliable, secure and dynamic evaluation mechanism is needed in cloud computing. Our algorithm is a bidirectional evaluation system between cloud users and resource providers. Users prefer to choose resources with high reliability. Meanwhile, service providers identify users' reputation through their trust score and behaviors.

When there are multiple players in a system, we can use game theory to determine the most likely outcomes. The key to game theory is that one player's payoff is contingent on the strategy implemented by the other player. In recent years, there have been many researches on cloud computing based on game theory. [9] constructed a reward and punishment mechanism to improve nodes trust in cloud file sharing systems. [10] proposed a Min-Max Game approach to solve the resource allocation problems in cloud computing, which considers the time and budget constraints of every user. [11] studied the cause and effect of interdependency in a public cloud platform based on game theory. [12] is a resource allocation mechanism based on the principles of coalition formation and the uncertainty principle of game theory. It aims to minimize the overhead of allocation strategies. [13] studied the computation offloading decision making problem among multiple mobile device users based on game theory and showed the existence of Nash equilibrium. Our trusted model is based on game theory, in which participants including normal users and resource providers are motivated by providing evaluations.

3. Trusted model based on game theory

There are various threats in cloud computing. For malicious resource providers they may provide fake resources or steal user data. For dishonest users they block the services of other normal users by sending malicious requests. To solve the problem of fraud and non-cooperation between cloud users and resource providers, we proposed a trusted model based on game theory.

3.1. Evaluation Mechanism

We define three types of entities in our trusted model, including normal users, resource providers and agents. Let $U=\{u_1, u_2, \ldots, u_n\}$. $U$ is the normal user set. Let $R=\{r_1, r_2, \ldots, r_m\}$. $R$ is the resource set. Agent is a third party organization, which is responsible for evaluating the trusted score of each entity. It also provides reference for resource allocation and scheduling. A user request can be decomposed into multiple subtasks, which may be allocated to different resources.

Definition 1: $T=\{t_1, t_2, \ldots, t_k\}$ is a set of tasks from user $u_i$. The mapping matrix $E$ is defined as bellows:
For each $e_{ij} \in E$, if $e_{ij}=1$, it means task $t_i$ is allocated to resource $r_j$. $\forall t_i \in T$, if $\exists e_{ij}=1$, it means user $u_x$ has an interactivity with resource $r_j$. They can choose to evaluate each other or not. As table 1 shown, $q_1$ is the reward user get, when task is executed on a resource node successfully. If the user gives a feedback to the resource, it gets the $q_3$ payment. The parameter $q_2$ is the cost that the user report to the agent. The parameter $q_4$ is the reward resource get for providing the service. There are four situations.

1. If a user gives a positive feedback, its reward is $q_1 + \omega \cdot q_1 - q_2$ and the reward of the resource is $\omega \cdot q_2 + q_4$. It means the user is satisfied with the resource according to the completing time and cost constraints. The parameter $\omega$ is the user weight. The user with better trusted score has a higher weight to affect the evaluation value.

2. If the feedback is negative, it means the user’s task fails. This may be because the response time of the resource exceeds the user's requirements or the resource is an untrusted node. In this case, the user gets $\omega \cdot q_2 - q_4$ and the resource node gets a minus score $-\omega \cdot q_1$.

3. The task is executed on the resource successfully, but the user doesn’t give a feedback to the agent. The user gets $q_1$ and the resource node gets $q_4$.

4. In the last case, the task fails to execute on the resource and doesn’t give a feedback. The user and the resource both get 0.

Table 1. Payment matrix.

| Resource | Positive | Negative |
|----------|----------|----------|
| Evaluate | $q_1 + \omega \cdot q_1 - q_2, \omega \cdot q_2 + q_4$ | $\omega \cdot q_2 - q_4, -\omega \cdot q_1$ |
| Not evaluate | $q_1, q_4$ | $0, 0$ |

As table 1 shown, (Evaluate, Positive) is the optimal strategy. For both cloud users and resource providers, they get the highest benefits if a task is executed on the resource successfully and gives a positive feedback. We assume the probability of user evaluation is $p$ and the probability of positive behavior of resource is $q$. The reward of a user for its evaluation is defined as $Q_u$.

$$Q_u = q \cdot (q_1 + \omega \cdot q_1 - q_2) + (1-q) \cdot (\omega \cdot q_2 - q_4)$$

(2)

If a user doesn’t send evaluation information to the agent, its reward is $q \cdot q_1$. The average reward for a user is defined in formula 3.

$$\bar{Q}_u = p \cdot Q_u + (1-p) \cdot q \cdot q_1$$

(3)

As formula 2 and 3 shown, users get reward for their feedback behaviors. They tend to give agents a feedback under the incentive mechanism. Considering the timeliness of evaluation, we use an attenuation factor to remark the trusted score for both cloud users and resource providers. The resource scheduler query scores from agents when matching resources for a task. What they need is a real-time score that best reflects the current state of the resource. As formula 4 shown, $r_{score_i}$ is the real-time trusted score of resource $r_i$ when query information arrived. The parameter $rh_{score_i}$ is the historical
score of $r_t$, which is recorded by the agent. The function $f(t)$ is calculated by the timestamp. The score provides less reference if the timestamp is long time ago. Cloud participants are encouraged to provide the latest evaluation results.

\[ r_{score; t} = f(t) \cdot rh_{score; t} \]  \hspace{1cm} (4)

For resource providers they have a similar situation. They get rewards if they evaluate the behaviors of cloud users. This is a two-way incentive evaluation mechanism.

3.2. Feedback Mechanism
As the above introduction, we proposed a bi-directional evaluation mechanism. Based on game theory, participants for both cloud users and resource providers are tend to send a feedback information in order to get rewards. The feedback mechanism is introduced in this subsection. As figure 1 shown, when a user sends request to the cloud, it may be decomposed into subtasks as a workflow according to the task complexity. Then the task scheduler dispatches each subtask to the resource according to the QoS constraints and scheduling algorithm. If the task is executed on a resource successfully, it gives a positive feedback to the agent. The trusted score of the resource increases because it provides computing services, storage services or other services. The user’s score also increases because it actively provides feedback. The other case is the task fails to execute on the resource for some reasons. For example, if the resource is overloaded and the waiting time of the task has already exceeded its maximum waiting time, the task would be re-allocated to another resource. In this situation the user gives a negative feedback to the agent. Negative feedbacks decrease the trusted score of the resource, which affects the probability to be chosen in a certain period of time. So the feedback mechanism improves the resource utilization and load balance.

On the other hand the resource nodes report users’ behaviors to affect their trusted scores. For example, malicious users download a large number of data and files illegally by agents. Malicious users also can launch a large-scale DDoS attack, which affects the normal users. The resource nodes

![Fig 1. Workflows scheduling model.](image)
refuse requests from these users if the users’ trusted scores are less than the minimal value. The bi-directional evaluation and feedback algorithm is as bellows.

**Table 2. Algorithm 1: Evaluation and feedback algorithm.**

| Input: user request                           |
| Output: task scheduling sequence and feedback information                           |
| Scheduling Process:                                      |
| 1. decompose the user request into subtasks, \( T = \{t_1, t_2, \ldots, t_n \} \)       |
| 2. for each task \( t_i \in T \)                                          |
| 3. calculate priority score according to the task requirement and relationships |
| 4. put \( t_i \) into Queue\( (s_k) \) according to the priority score and QoS constraints |
| 5. end for                                                                 |
| Monitoring process:                                           |
| 1. while true                                              |
| 2. for each task \( t_i \) in the queue of \( s_k \)       |
| 3. if trusted score of the task user < minimal value       |
| 4. deny of service and negative evaluation for \( u_j \)   |
| 5. else if waiting time of \( t_i \) > maximum waiting time |
| according to the QoS constraints                            |
| 6. notify and reschedule \( t_i \)                        |
| 7. negative evaluation for \( s_k \)                        |
| 8. else if \( t_i \) is executed on \( s_k \) successfully |
| 9. positive evaluation for \( s_k \) and \( u_j \)           |
| 10. end for                                                |
| 11. end while                                              |

### 3.3. Trust-transmitting Mechanism

As the above mentioned, cloud users and resource providers send feedback to the agent, which is responsible to monitor the trusted scores of participants within a certain range. The trusted scores stored on the agent are comprehensive evaluation of nodes. Most feedback information are indirect evaluation. If the user task executed on a resource node, the interaction generates a direct evaluation. Let \( tr_{local\_ij} \) equals the local trusted score of resource \( s_j \) stored on the cloud user \( u_i \). Let \( tr_{agent\_j} \) equals the comprehensive trusted score of resource \( s_j \) stored on the agent. We provide different strategies to consider \( tr_{local\_ij} \) and \( tr_{agent\_j} \).

In the first strategy, the local trusted score \( tr_{local\_ij} \) has a higher priority. Users are more willing to believe the evaluation of resources stored locally. If there is no local evaluation data, the \( tr_{agent\_j} \) stored on the agent is referenced for resource allocation. The weakness of this strategy is \( tr_{local\_ij} \) only reflects the historical behavior of resource \( s_j \) and it doesn't represent the current state of \( s_j \). The strategy doesn't consider the timeliness of feedback information. The advantage of this strategy is saving communication time with agents. Also it avoids the interference of false evaluation from other malicious users.

The second strategy is we use both \( tr_{local\_ij} \) and \( tr_{agent\_j} \) to evaluate the resource \( s_j \). The parameter \( \varepsilon \) is defined as an adjustable factor to balance the effect of \( tr_{local\_ij} \) and \( tr_{agent\_j} \). The comprehensive score of \( s_j \) is calculated in formula 5, which contains these two scores from local and the agent. The advantage of this method is improving the accuracy of evaluation for the current state of the resource node.

\[
rscore_j = f(t) \cdot \varepsilon \cdot tr_{local\_ij} + (1-\varepsilon) \cdot tr_{agent\_j}
\] (5)
The value of $\varepsilon_1$ is adjustable and it depends on the different trust degree between local and the agent. If there are a lot of direct interactions between users and resource nodes, we can increase the influence of the local trusted score $tr_{local,j}$ by increasing the value of $\varepsilon_1$. It helps to reduce the impact of erroneous evaluation information. On the contrary, if the number of interaction is small, the local trust score may be accidental and risky. The user prefers to trust the comprehensive evaluation from the agent.

Similarly, we define $tul_{local,i}$ equals the local trusted score of user $u_i$ stored on the resource $s_j$. Let $tu_{agent,i}$ equals the comprehensive trusted score of user $u_i$ stored on the agent. The parameter $\varepsilon_2$ is the adjustable factor. The comprehensive score of $u_i$ is calculated in formula 6.

$$uscore_i = f(t) \cdot (\varepsilon_2 \cdot tul_{local,j} + (1-\varepsilon_2) \cdot tu_{agent,i})$$

Based on the above principles, we modify the monitoring process in algorithm 1 as bellows. We defined a parameter $\lambda$, which is the minimal value of interaction times between $u_i$ and $s_j$. If the interaction times is more than $\lambda$, we can use the local trusted score $tul_{local,j}$ to evaluate the user’s credit. Conversely, both $tul_{local,j}$ and $tu_{agent,i}$ are used to compute $uscore_i$.

| Table 3. Algorithm 2: Evaluation algorithm by trust-transmitting. |
|---------------------------------------------------------------|
| Input: user request | Output: task scheduling sequence and feedback information |
| Monitoring process: | |
| 1. while true | |
| 2. for each task $t_i$ in the queue of $s_k$ | |
| 3. if number of interactions between $s_k$ and $u_i < \lambda$ | |
| 4. $uscore_i = f(t) \cdot (\varepsilon_2 \cdot tul_{local,j} + (1-\varepsilon_2) \cdot tu_{agent,i})$ | |
| 5. else $uscore_i = tul_{local,j}$ | |
| 6. if $uscore_i <$ minimal value | |
| 7. deny of service and negative evaluation for $u_i$ | |
| 8. else if waiting time of $t_i >$ maximum waiting time according to the QoS constraints | |
| 9. notify and reschedule $t_i$ | |
| 10. negative evaluation for $s_k$ | |
| 11. else if $t_i$ is executed on $s_k$ successfully | |
| 12. positive evaluation for $s_k$ and $u_i$ | |
| 13. end for | |
| 14. end while | |

4. Experiment and analysis
Our algorithm aims to improve the resource utilization and load balance. Also it can evaluate the trusted degree of cloud users and resource providers. It provides an incentive mechanism to make users and nodes more willing to make evaluation based on game theory. We use CloudSim [14] to simulate our model and algorithm. We generate 100 resource nodes with random initial trusted value. Based on the value they are divided into three groups as {high trusted, low trusted, untrusted}. The number of tasks is from 0 to 500. As figure 2 shown, at the beginning the average score of high trusted group increases as the number of task increases. But when the nodes are overloaded, they cannot start the task within the given time. The average score decreases because of the negative feedbacks. The low trusted group has the similar situation. For untrusted group the change of average score is not
obvious, because few of them are chosen to provide service. As task number and execution time increase, the score is affected by the attenuation factor in formula 4 and increases slowly.

![Average scores for three groups.](image1)

**Fig 2.** Average scores for three groups.

In figure 3, we record the one-time success rate for task scheduling. The number of tasks is from 100 to 1000. As the results shown, the average success rate decreases as the number of task increases. But the overall rate of proposed algorithm based on trusted model is better than the random selection. A resource node with high trusted score means it has better performance and reliability. Choosing resource nodes with high trust score can improve the success rate of resource selection.

![Average success rate.](image2)

**Fig 3.** Average success rate.

In figure 4, we simulate the changes of nodes numbers for different groups as the increasing of task numbers. The number of high trusted nodes increased because nodes get positive feedback and their trusted scores increase. When task number is more than 700, the number of high trusted nodes
decrease since some node are overload and get negative feedbacks. However, overall, the total number of high trusted nodes and low trusted nodes increases and the number of untrusted nodes decreases. It reflects nodes prefer to provide reliable service in order to increase their trusted scores. Meanwhile, the nodes with high trusted scores have priority to be chosen by user tasks. For untrusted nodes they can be reallocate to the other two groups if they provide reliable service successfully.

Fig 4. Nodes numbers for three groups.

5. Conclusion
The security and trusted problem in cloud computing becomes one of the most important problem to be solved. We proposed a bi-directional evaluation and feedback algorithm based on game theory in this paper. Both normal cloud users and resource providers are evaluated by their trusted scores. The resource reliability is evaluated by scores from users. Meanwhile, resource also grades users according to their behaviors. They are tend to send feedback information to agents in order to get rewards based on game theory. Task scheduler prefers to allocated task to the resources with higher trusted scores. Simulation results show task scheduling based on trusted model improves the resource utilization and load balance.

Acknowledgment
This work was supported by the Fund Project of Sichuan Provincial Department of Education (No. 15ZB0036).

References
[1] Dzmitry Kliazovich; Johnatan E. Pecero; Andrei Tchernykh; Pascal Bouvry; Samee U. Khan; Albert Y. Zomaya. CA-DAG: Communication-Aware Directed Acyclic Graphs for Modeling Cloud Computing Applications. 2013 IEEE Sixth International Conference on Cloud Computing, Santa Clara, CA, USA, 2013.

[2] Maria Alejandra Rodriguez; Rajkumar Buyya. Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds. IEEE Transactions on Cloud Computing, Volume: 2, Issue: 2, 2014.

[3] Anagha Yadav; S. B. Rathod. Priority based task scheduling by mapping conflict-free resources and Optimized workload utilization in cloud computing. 2016 International Conference on Computing Communication Control and automation, Pune, India, 2016.

[4] A B M Bodrul Alam; Mohammad Zulkernine; Anwar Haque. A Reliability-Based Resource Allocation Approach for Cloud Computing. 2017 IEEE 7th International Symposium on
Cloud and Service Computing, Kanazawa, Japan, 2017.

[5] Yubiao Wang; Junhao Wen; Wei Zhou ; Fengji Luo. A Novel Dynamic Cloud Service Trust Evaluation Model in Cloud Computing. 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering, New York, NY, USA, 2018.

[6] Shweta Kaushik; Charu Gandhi. Multi-level Trust Agreement in Cloud Environment. 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, India, 2019.

[7] Dehua Kong; Yuqing Zhai. Trust Based Recommendation System in Service-oriented Cloud Computing. 2012 International Conference on Cloud and Service Computing, Shanghai, China, 2012.

[8] Mukesh Kant Tripathi; Vivek Kumar Sehgal. Establishing trust in cloud computing security with the help of inter-clouds. 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 2014.

[9] Wei Jyun Wong; Wei Tsong Lee ; Hsin Wen Wei. Base on Game Theory Model to Improve Trust Access Control in Cloud File-Sharing System. 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 2014.

[10] K G Srinivasa; S Srinidhi ; K Sharath Kumar ; Vignesh Shenvi ; U Shashank Kaushik ; Kushagra Mishra. Game theoretic resource allocation in cloud computing. The Fifth International Conference on the Applications of Digital Information and Web Technologies, Bangalore, India, 2014.

[11] Charles A. Kamhoua; Luke Kwiat ; Kevin A. Kwiat ; Joon S. Park ; Ming Zhao ; Manuel Rodriguez. Game Theoretic Modeling of Security and Interdependency in a Public Cloud. 2014 IEEE 7th International Conference on Cloud Computing, Anchorage, AK, USA, 2014.

[12] Parvathy S. Pillai; Shrisha Rao. Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory. IEEE Systems Journal, Volume: 10, Issue: 2 , 2016, 637-648.

[13] Yujiong Liu; Shangguang Wang ; Fangchun Yang. Poster Abstract: A Multi-user Computation Offloading Algorithm Based on Game Theory in Mobile Cloud Computing. 2016 IEEE/ACM Symposium on Edge Computing (SEC), Washington, DC, USA, 2016.

[14] Ekta Rani; Harpreet Kaur. Study on fundamental usage of CloudSim simulator and algorithms of resource allocation in cloud computing. 2017 8th International Conference on Computing, Communication and Networking Technologies, Delhi, India, 2017.