

## Application of VCG in Replica Placement Strategy of Cloud Storage

Wang Hongxia

Computer Department, Beijing Youth Politics College, Beijing, China  
[whx4617@163.com](mailto:whx4617@163.com)

### Abstract

*With the rapid development of the cloud computing and big data, many data center network are growing more large scale, and the number of servers is increasing at an exponential rate. In this paper, VCG (Vickrey-Clarke-Groves) mechanism was introduced into replica placement. Mapping from replica placement model to VCG mechanism was established and suitable payment function was designed. Data storage strategy is built on the distributed file system. In order to ensure the reliability of data storage GFS and HDFS are further both referred to the concept of replica of data block. This paper generally introduced replica strategy, and mentioned that the storage location of data replica is a key point of the research on replica technology currently. This paper mainly applied one cloud platform simulation tool---the CloudSim to simulate the algorithm. The performance of the algorithm was analyzed in the CloudSim platform and compared it with other related algorithms. The simulation experiments were done on the replica placement strategy of the data block, which shows that replica placement strategy can also achieve the established goal.*

**Keywords:** VCG; replica placement strategy; cloud storage

### 1. Introduction

Cloud Storage is developed on the basis of Cloud Computing which is used to store and manage the mass data in network. Cloud Storage integrates a large number of heterogeneous devices to provide external storage and business access *etc.* through the application of software. Cloud Storage integrates of Grid Computing, Distributed Systems, Virtualization, Load Balancing and other traditional network technology. It has the properties of high reliability, high scalability, high efficiency and low cost *etc.* The focus researches are data security, performance optimization, resources scheduling strategy, and service payment model in cloud storage system currently.

With the rapid development of information technology and the high speed grows of global data [1], confronting PB level storage requirements, traditional file system has some difficulties in capacity and performance expansion. Cloud storage is widely used by many companies because of its advantages such as expandability, cost-effective, high fault-tolerance [2-4]. As the basis of cloud storage, distributed file system achieves promotion of system availability and fault-tolerance by introducing data replication technology. However, replica brings about a lot of problems such as bandwidth and storage consumption, unexpected data inconsistency *etc.* [5, 6]

Data storage strategy is built on the distributed file system (DFS). In order to ensure the reliability of data storage further, GFS (Google File System) and HDFS (Hadoop Distributed File System) are both referred to the concept of replica technology of data block. It is paid attention that the storage location of data replica is a key point of the research on replica technology currently.

This paper briefly introduces the storage mechanism when place a single replica, and then Multiple-replica-to-multiple-service placement strategy is drawn out. Following that,

this paper presents the construction and the implementation of the mechanisms dominant strategy equilibrium situation which purpose is to guide the mechanism to achieve dominant strategy equilibrium. VCG (Vickrey-Clarke-Groves) mechanism [7, 8] was introduced into replica placement strategy in this paper. Mapping from replica placement model to VCG mechanism was established and suitable payment function was designed. The cloud simulation platform---CloudSim was used to test some properties and analyze the algorithm performances.

## 2. Designation of Mechanism and Description of Core Algorithm

Generally, the default number of data block in cloud storage system is three, but it can also be set by users according to the practical situation  $s$  ( $0 \leq s \leq 3$  or  $s > 3$ ). This paper deems that one of the replicas is placed randomly. And then the other two replicas are placed on different racks in accordance with the principle of minimum cost of the system in the study. Based on this storage thinking this paper makes some relevant deductions of formulas and theories on the basis of data storage strategy and then provides replica placement strategy based on VCG mechanism.

Based on the mathematical mode, the unit data storing cost of data storage nodes is quantified as  $c$ , the unit data value of data object nodes is  $v$ , which are both assumed to be  $M$  unit. If data blocks are divided into several parts which are all 64M a node' s storing cost for every single replica is  $64c$  and a replica' s own value is  $64v$ . In order to seek clear and convenient statements in this paper, a single replica' s storing cost as  $c$  is considered which type is  $t(c)$ ; and its own value is  $v$  which type is  $t(v)$ .

### 2.1 Single-Replica-to-Multiple-Service Mechanism

When multiple storage nodes compete with each other for a single replica, the form of auction changes into the tradition model of a single item for a single user which is also the basic form of the initial study of VCG mechanism called the second-price auction. Its theoretical derivation is given below based on a general data storage strategy.

#### (1) Allocation function and allocation rule

Assume that there are data storage nodes in the system, for any set of quotation collection  $b = (b_1, b_2, b_3, \dots, b_n)$ , the final allocation result can be expressed as:

$$o_i(b) = \begin{cases} 1 & i \text{ is saved} \\ 0 & i \text{ is not saved} \end{cases} \quad (1)$$

Minimizing the total cost of system is still a target when replicas are stored. Storage nodes which satisfy  $b_i < v$  are selected as candidate nodes. Assume that the number of candidate nodes is  $n$ .  $n$  candidate nodes can be allocated as follows:

$$\begin{aligned} o^{VCG}(b) &= \arg \min_{o \in O} \sum_i C_i(b_i, o) \\ &= \arg \min_{o \in O} \sum_i (b_i \times o_i) \\ &= \arg \min_{o \in O} (b_1, b_2, \dots, b_n) \end{aligned} \quad (2)$$

From Formula (2) the conclusion can be drawn: the mechanism will select the minimum-cost node when a single replica was stored. The inversely proportional relationship of nodes between cost and performance shows that the placement node is considered to be the top-quality one in the system.

#### (2) Payoff function and payoff rule

Those storage nodes which satisfy will be paid as follows:

$$\begin{aligned}
 p_i^{VCG}(\mathbf{b}) &= \min_{j \neq i} \sum C_j(b_j, o_j) - \sum_{j \neq i} C_j(b_j, o_j^{VCG}) \\
 &= \min_{j \neq i} \sum b_j \times o_j(\mathbf{b}_{-i}) - \sum_{j \neq i} b_j \times o_j(b) \\
 &= \min(b_1, b_2, \dots, b_{i-1}, b_{i+1}, \dots, b_n) - \min_{j \neq i}(b_1, b_2, \dots, b_n)
 \end{aligned} \tag{3}$$

From Formula (3), it can be concluded after simple derivation that  $p_i = \min(b_1, b_2, \dots, b_{i-1}, b_{i+1}, \dots, b_n)$  for  $b = \min(b_1, b_2, \dots, b_n)$ ; That is when the node  $i$ 's quotation is the minimum one among all nodes, the mechanism pays the quotation which is the minimum one among all nodes but  $i$ . Also known as Second-Price Auction Thought; If  $b \neq \min(b_1, b_2, \dots, b_n)$  which means the node  $i$ 's quotation is not the minimum one among all nodes, after a simple derivation a conclusion can be drawn that:

$$\min(b_1, b_2, \dots, b_{i-1}, b_{i+1}, \dots, b_n) = \min_{j \neq i}(b_1, b_2, \dots, b_n)$$

Therefore  $p_i = 0$ . If the node  $i$ 's quotation is not the minimum one, the minimum quotation of other nodes is the same with that of all nodes with the presence of node  $i$ . At this moment the payoff for node  $i$  is 0.

According to the above analysis, Formula (3) can be simplified as follows:

$$p_i(b) = \begin{cases} b_k & i \text{ is saved} \\ 0 & i \text{ is not saved} \end{cases} \tag{4}$$

In this formula is the minimum node's quotation among all nodes except node  $i$ ,  $b_k = \min(b_1, b_2, \dots, b_{i-1}, b_{i+1}, \dots, b_n)$ .

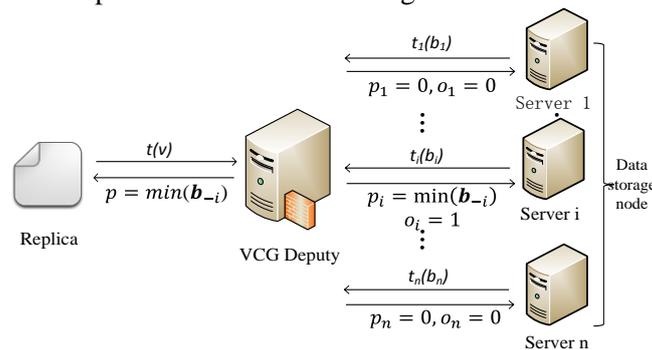
Combine the distribution function Formula (1) and the payoff function Formula (4), the profit that the data storage node obtains can be calculated:

$$\begin{aligned}
 u_i(b) &= p_i(b) - c_i \times o_i(b) \\
 &= \begin{cases} b_k - c_i & i \text{ is saved} \\ 0 & i \text{ is not saved} \end{cases}
 \end{aligned} \tag{5}$$

In the formula (5),  $b_k$  is the minimum node's quotation among all nodes except node  $i$ . Then  $b_k$  is the second lower quotation when node exists. If the node  $i$ 's quotation is true, there must be  $b_k > c_i$ . And the profit of storage replica of the node is positive; if node  $i$ 's quotation is false, then  $b_k > b_i \neq c_i$  the profit is  $u_i = (b_k - b_i) - (c_i - b_i)$ , so it cannot be guaranteed to be positive.

### (3)Simplified model of the mechanism

Through the introduction of the core algorithm of the mechanism the simplified mode of Single-replica-to-multiple-service mechanism is given as below:



**Figure 1. The simplified model of Single-Replica-to-Multiple-Service Mechanism**

Figure 1 shows a simple bidding process of mechanism. All storage nodes of cloud report their own types to the mechanism agency. The mechanism will choose the minimum nodes quotation as the replica placement node, and return the allocation result and payoff result to the storage nodes. All storage nodes of cloud will compare profits

with each other which depend on those results in order to determine which kind of quotation can make the maximum profits. After several rounds of iterations all nodes will choose the optimal quotations, and the mechanism reaches the state of equilibrium at this moment. In the state of equilibrium the mechanism will return the final allocation result and payoff result to the data storage nodes of cloud and the data object nodes of the clients.

## 2.2 Multiple-Replica-to-Multiple-Service Mechanism

The large data is usually divided into several data blocks and then they are stored. The backup of each data block will produce a large number of replicas. In Section A. the storage strategy of a single replica is introduced. Based on the previous section this section will introduce how to store multiple replicas in the cloud system.

A data object node is selected and assumed that its storage data is divided into data blocks, and then  $3s$  corresponding replicas need to be placed. Assume that the system places one group of replicas randomly, and then the other two groups will be placed on different racks in accordance with the principle which can achieve the lowest cost. In order to ensure data reliability to the greatest extent, those replicas of data blocks will be placed on different nodes respectively. In this section, how to select data storage nodes as the placement points of replicas is the object of study. Assume that the data object node owns the type  $t_j(v_j, s_j, PRI_j)$ .

### (1) Allocation function and allocation rule

Assume that data replicas are equivalent when they are on the same data object nodes. In other words, the data value of each replicas is  $v$ , and any replica' s type is  $t(v)$ ; assume that there are data storage nodes on the cloud, the single-replica storage of the node  $i$  is  $c_i$ , and its type is  $t_i(c_i)$ . Assume that storage nodes satisfy the condition  $c_i < v$ .

Referring Formula (2) in Section A. it can be drawn that the allocation of candidate nodes made by mechanism is:

$$\begin{aligned} o^{VCG}(b) &= \arg \min_{o \in O} \sum_i C_i(b_i, o) \\ &= \arg \min_{o \in O} \sum_i (b_i \times o_i) \\ &= \arg \min_{o \in O} \sum_i b_i \end{aligned} \quad (6)$$

For the realization of Formula (6),  $n$  candidate nodes are arranged according to quotations from small to large. Select the nodes which are the first lowest quotations as placement nodes, and the set of replica placement is  $S$ . Therefore, Formula (6) can be simplified as:

$$o_i(b) = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases} \quad (7)$$

### (2) Payoff function and payoff rule

Referring to Formula (3), the payoff function of the mechanism can be described as follows:

$$\begin{aligned} p_i^{VCG}(b) &= \min \sum_{j \neq i} C_j(b_j, o_{-i}) - \sum_{j \neq i} C_j(b_j, o^{VCG}) \\ &= \min \sum_{j \neq i} b_j \times o_j(b_{-i}) - \sum_{j \neq i} b_j \times o_j(b) \\ &= \sum_{j \in S_{-i}} b_j - \sum_{j \in S, j \neq i} b_j \end{aligned} \quad (8)$$

In this formula,  $S$  is the set of replica placement when the node is included;  $S_{-i}$  is the set of replica placement when the node is not included.

Arrange the quotations in ascending order. For  $i \in S$ , that is  $b_i \leq b_s$ , Formula (8) can be converted to:

$$\begin{aligned} p_i^{VCG}(b) &= \sum_{j \in S'} b_j - \sum_{j \in S_{-i}} b_j \\ &= (b_1 + \dots + b_{i-1} + b_{i+1} + \dots + b_{s+1}) \\ &\quad - (b_1 + \dots + b_{i-1} + b_{i+1} + \dots + b_s) \\ &= b_{s+1} \end{aligned}$$

For  $i \notin S$ , that is  $b_i > b_s$ , the equation Formula (8) is right, and  $p_i=0$  at this moment. Therefore, the payoff function in the formula (8) can be simplified as:

$$p_i(b) = \begin{cases} b_{s+1} & i \in S \\ 0 & i \notin S \end{cases} \quad (9)$$

In this formula,  $b_{s+1}$  is the  $s+1$  node price in an ascending order.

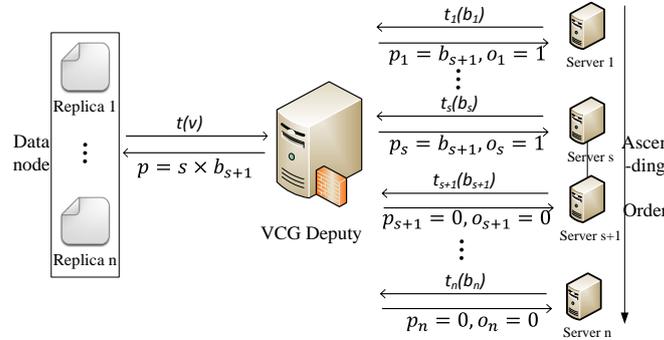
The profits that storage nodes obtain is:

$$\begin{aligned} u_i(b) &= p_i(b) - c_i \times o_i(b) \\ &= \begin{cases} b_{s+1} - c_i & i \in S \\ 0 & i \notin S \end{cases} \end{aligned} \quad (10)$$

From Formula (10) the conclusion can be drawn: If the node  $i$ 's quotation is true there must be  $b_k > c_i$ . And the profit of storage replica of the node is positive at this time; But if node  $i$ 's quotation is false then  $b_{s+1} > b_i \neq c_i$ , the profit is that  $u_i = (b_{s+1} - b_i) - (c_i - b_i)$ , and so it cannot be guaranteed to be positive; If the node itself costs too much the false quotation will lead to negative profit.

### (3) Simplified model of Mechanism

The simplified model of multiple-replica-to-multiple-service Mechanism is given below as follows:



**Figure 2. The Simplified Model of Multiple-Replica-to-Multiple-Service Mechanism**

Figure 2 is a simplified model which shows that multiple replicas are placed in the cloud and it shows one transaction case in the game of the mechanism. In the transaction all storage nodes will choose their optimal quotations according to the allocation results and payment results returned from the mechanism. Mechanism's final transaction will be conducted in the equilibrium situation where all nodes choose their optimal quotations. And the mechanism will return the final results back to the data storage nodes on the cloud and the data object nodes on clients. What should be paid attention to is that these replicas are all belong to the same data object node in the research, so the user pays every replica  $p_i = b_{s+1}$  and thus the user need to pay totally  $p = \sum p_i = s \times b_{s+1}$  for this replica placement.

### 3. Analysis of the Mechanism's Performance

Because the research on replica placement strategy is derived on the basis of general data storage strategy and its essence is a special simplified form of data storage strategy. So the replica placement strategy still satisfies the conclusions drawn from above. This section gives simple analysis of the multiple replicas to multiple services mechanism's performance in order to apply obtained conclusions better to the replica placement strategy.

Conclusion 1: Real quotation of the storage node is its dominant strategy and the mechanism meets the feature of incentive compatibility.

Proof: The node raises its quotation which means  $b_i > c_i$ . If  $c_i$  meets the inequality  $c_i > b_s$  and the node doesn't get a replica successfully when the node gives a real quotation, it still can't get a replica after raising the offer because of the inequality  $b_i > c_i > b_s$ , so the profits are all zero. If  $c_i$  meets the inequality  $c_i < b_s$  and the node gets a replica successfully when the node gives a real quotation as well as  $b_i$  meets the inequality  $c_i < b_i < b_s$  after raising the offer, it still can obtain a replica and the profit keeps unchanged as a result of the formula  $u_i' = u_i = b_{s+1} - c_i$ . The node gets a replica successfully when it gives a real quotation and  $c_i$  meets the inequality  $c_i < b_s$ , but it can't get a replica after raising the offer because of  $c_i < b_i < b_s$ . So the profit  $u_i'$  meets the inequality  $0 = u_i' < u_i = b_{s+1} - c_i$ , the node loses its profit which it could originally get.

The node reduces its quotation, namely that  $b_i < c_i$ . If  $c_i$  meets the inequality  $c_i < b_s$  and the node gets a replica successfully when the node gives a real quotation, it still can get a replica after reducing the offer because of the inequality  $b_i < c_i < b_s$ , so the profits keeps unchanged as a result of the formula. If  $c_i$  meets the inequality  $c_i > b_s$  and the node can't get a replica successfully when the node gives a real quotation as well as  $b_i$  meets the inequality  $c_i > b_i > b_s$  after raising the offer, it still can obtain a replica and the profits are all zero. If the node can't get a replica successfully when it gives a real quotation and  $c_i$  meets the inequality  $c_i > b_s$ , but it can get a replica after reducing because of  $c_i > b_s > b_i$ . So the profit  $u_i'$  meets the formula  $u_i' = b_s - c_i$  at this time. But because of  $c_i > b_s$ ,  $u_i'$  is negative. Therefore, the node's actual profit is negative despite its opportunity to obtain a replica.

In conclusion, the node cannot raise its profit by giving a false quotation no matter what the case it is in. Considering that all nodes are conservative, they will all choose the true quotations as their optimal quotations in the end. Therefore, the real quotation of the storage node is its dominant strategy, and the mechanism meets the incentive compatibility.

Conclusion 2: Mechanism meets ex-post individual rationality for storage nodes.

Proof: For any set of quotation collection  $b_{-i} = (b_1, b_2, \dots, b_{i-1}, b_{i+1}, \dots, b_n)$ , storage nodes all have the dominant strategy  $b_i^* = c_i$ . If the node gets a replica under the dominant equilibrium situation  $u_i$  which equals  $b_{s+1} - c_i$  is positive and the profit as well. If the node is able to get a replica  $u_i$  equals zero. So the profits of the nodes are non-negative under dominant strategy, and the mechanism meets ex-post individual rationality for each storage node.

#### 4. The Implementation of the Mechanisms Dominant Equilibrium Situation

In the section2 the conclusion have been drawn that the storage node' s real quotation is its dominant strategy through analysis. But the storage node isn' t aware of that at the beginning. So a realistic transaction step must be designed to guide the selfish and conservative storage nodes in the system to make an optimal choice. Only in that way the mechanism can be achieved. Because the mechanism meets the incentive compatibility characteristic, the dominant situation which the mechanism finally reaches also achieves the goal of the designer.

Referring to the simplified model of Figure 1 and Figure 2, designed steps to achieve the mechanism are as follows.

- (1) Data agency nodes report the replica type  $t_j(v_j, s_j, PRI_j)$  to the mechanism, and then the mechanism sorts the nodes in descending order according to PRI, and puts the nodes under the same priority in descending order according to  $v$ . The mechanism deals with the data object nodes in sequence.
- (2) The data storage nodes determine their own quotation collection  $B_i = (b_{i1}, b_{i2}, \dots, b_{ik_i})$  and initialize their profit vector  $U_i = (u_{i1}, u_{i2}, \dots, u_{ik_i}) = 0$ .
- (3) The data storage nodes select quotations which meet  $b_{it} \in B_i$ , and report the types  $t_i(b_{it})$  to the mechanism. The mechanism chooses the nodes which meets the inequality  $b_{it} < v_j$  as candidate nodes and puts these nodes in ascending order according to the quotation.
- (4) The mechanism chooses the first  $s$  nodes as replica placement nodes and pays them  $p$  which equals to  $b_{s+1}$ .
- (5) Each data storage node calculates the profit of this transaction according to this formula  $u_{it} = p_{it} - c_i \times o_{it}$ , and then updates the profit vector  $U_i = (\dots, u_{it}, \dots)$ ;
- (6) Repeat steps ② ~ ⑤ until all storage nodes finish vector updating.
- (7) Each storage node selects the quotation  $b_i^*$  corresponding to  $\max(U_i)$  as the final quotation so the quotation set is  $b^* = (b_1^*, b_2^*, \dots, b_n^*)$ . Follow the steps ③ ~ to complete this transaction and then update the type  $t_i(b_i)$ .
- (8) Sort up all the data object nodes according to value and repeat steps ② ~ ⑦ until all replicas are placed.

The pseudo-codes which can achieve the mechanism's dominant strategy equilibrium situation are as below Figure 3:

---

**Input:** Bids submitted by Nodes:  $\mathbf{b} = (b_1, b_2, \dots, b_N)$   
 Replication value submitted by Users:  $\mathbf{v} = (v_1, v_2, \dots, v_M)$   
 Number of Replications submitted by Users:  $\mathbf{s} = (s_1, s_2, \dots, s_M)$

**Output:** Allocation:  $\mathbf{o} = (o_1, o_2, \dots, o_N)$   
 Pay:  $\mathbf{p} = (p_1, p_2, \dots, p_N)$

- 1: Sort the Users in decreasing order of their values ( $v_1 \geq v_2 \geq \dots \geq v_M$ );
- 2: for  $j = 1$  to  $M$  do
- 3:     for  $i = 1$  to  $N$  do
- 4:         Initiate  $U_i \leftarrow 0$ ;
- 5:         for  $t = 1$  to  $k_i$  do
- 6:              $\mathbf{b} \leftarrow (b_{it}, \mathbf{b}_{-i})$ ;
- 7:             Choose Nodes satisfying  $b_i < v_j$ ;
- 8:             Sort the chosen Nodes in increasing order of their bids ( $b_1 \leq b_2 \leq \dots \leq b_n$ );
- 9:             if  $b_{it} < b_{sj}$  then
- 10:                  $o_{it} \leftarrow 1$ ;
- 11:                  $p_{it} \leftarrow b_{sj+1}$ ;
- 12:             else
- 13:                  $o_{it} \leftarrow 0$ ;
- 14:                  $p_{it} \leftarrow 0$ ;
- 15:             end if
- 16:              $u_{it} \leftarrow p_{it} - c_i \times o_{it}$ ;
- 17:              $U_i \leftarrow (u_{it}, U_{i,-t})$ ;
- 18:         end for
- 19:         Select  $b_i^*$  ensure maximizing  $U_i$ ;
- 20:     end for
- 21:     Compute  $\mathbf{o}(\mathbf{b}^*), \mathbf{p}(\mathbf{b}^*)$  under  $\mathbf{b}^* = (b_1^*, b_2^*, \dots, b_N^*)$ ;
- 22: end for

---

**Figure 3. Dominant Equilibrium Situation Implementation Algorithm based on the Multiple Replicas to Multiple Services Mechanism**

### 5. The Experimental Simulation of Replica Placement Strategy

The experiment in this section follows the parameters setting and the experiment environment when the data storage strategy is simulated. We select eight replicas of the same user as the research objects and the size of each replica is 64M. To ensure the reliability and availability of the data, only one replica of the same user's data is placed on each storage node.

```
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID  STATUS  Data center ID  VM ID
0           SUCCESS  2               0
1           SUCCESS  3               1
2           SUCCESS  4               2
3           SUCCESS  5               3
4           SUCCESS  6               4
5           SUCCESS  7               5
6           SUCCESS  8               6
7           SUCCESS  9               7
```

```
*****
****PowerDatacenter: Datacenter_4*****
User id      Debt
17           5.12
*****
****PowerDatacenter: Datacenter_5*****
User id      Debt
17           5.76
*****
****PowerDatacenter: Datacenter_6*****
User id      Debt
17           6.4
*****
****PowerDatacenter: Datacenter_7*****
User id      Debt
17           7.68
*****
****PowerDatacenter: Datacenter_8*****
User id      Debt
*****
****PowerDatacenter: *****
id           Debt
DebtSum      35.2
```

(a) The Distribution of the Cloud Task to the Nodes and the Virtual Machine

(b) The Cost of each Node and the Total Cost

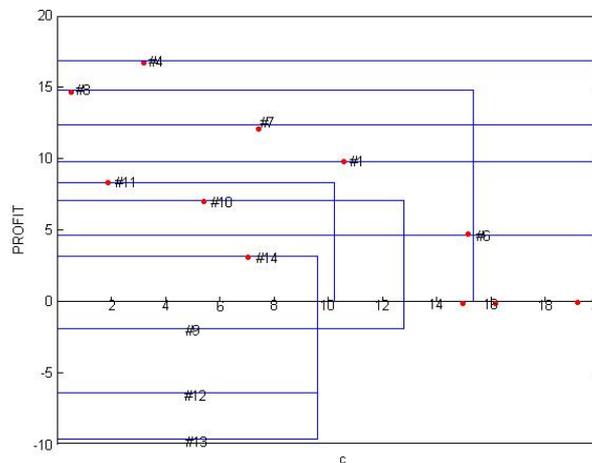
**Figure 4. Simulation Results of Multiple-Replica-to-Multiple-User Mechanism**

Figure 4 shows a part of output results of the simulation on replica placement strategy. The output results of the CloudSim platform divides into two parts: one is the allocation status of cloud tasks and the other is the cost of each node and the total system cost. From the outputs of these two parts, researchers can know the position where the replica was placed in the system, and make the cost consumed by each node and the total system cost for this reason clear.

**Table 1. The Corresponding Profits of all Nodes in Different Price Ranges based on the Multi-Replica Mechanism**

BID NODE	[0,9.6)	[9.6,10.24)	[10.24,12.8)	[12.8,15.36)	[15.36,20]
#0	15.52	15.52	15.52	15.52	15.52
#1	9.76	9.76	9.76	9.76	9.76
#2	14.88	14.88	14.88	14.88	14.88
#3	7.2	7.2	7.2	7.2	7.2
#4	16.8	16.8	16.8	16.8	16.8
#5	10.4	10.4	10.4	10.4	10.4
#6	4.64	4.64	4.64	4.64	4.64
#7	12.32	12.32	12.32	12.32	12.32
#8	14.72	14.72	14.72	14.72	0
#9	-1.92	-1.92	-1.92	0	0
#10	7.04	7.04	7.04	0	0
#11	8.32	8.32	0	0	0
#12	-6.4	0	0	0	0
#13	-9.6	0	0	0	0
#14	3.2	0	0	0	0

Table 1 records the corresponding profits of all nodes in different price ranges. In order to ensure clarity and facilitate the record experimental data, we deal with all nodes one by one in sequence during this experiment until they select their optimal quotations.

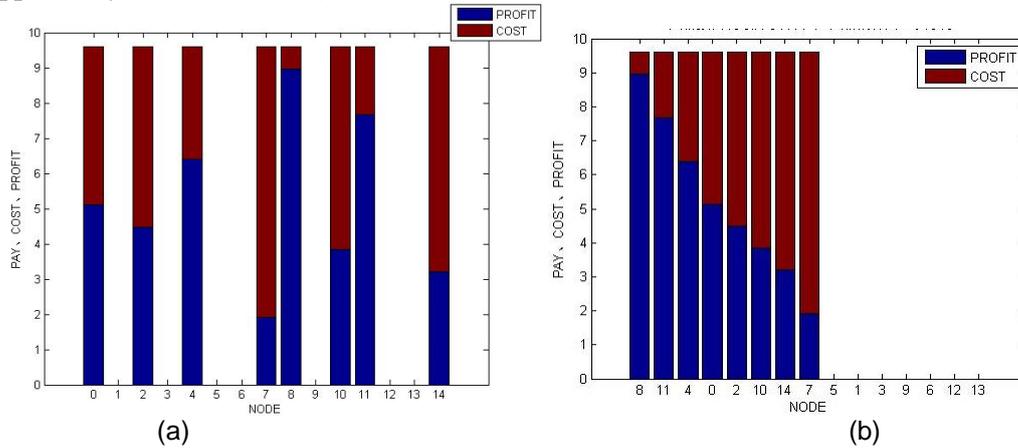


**Figure 5. The Corresponding Profits of all Nodes under Different Quotations based on the Multi-Replica Mechanism**

All nodes are the initial quotations before the auction starts in the experiment. After the researched node selecting its optimal quotation, the node's quotation stops changing. So the quotations of the researched node before the optimal quotations are all the best quotations and quotations behind the optimal one are all initial offers.

From Figure 5 we can see that each node corresponds to different profits in different price ranges. The red dots in the figure represent the true quotations of the nodes. We can

see that all nodes are under the maximum profits at the true quotation. And the profits at fake quotations are not more than the ones at the real offers. The profits at the true quotation of the nodes # 9, # 12, # 13 in the graph are all zero that is because they can't successfully obtain replicas. They can get a storage chance when they report a lower price, but it can be seen from the figure that their profits are negative at this moment so the relevant nodes will not give a fake quotation in order to fight for the storage opportunity.



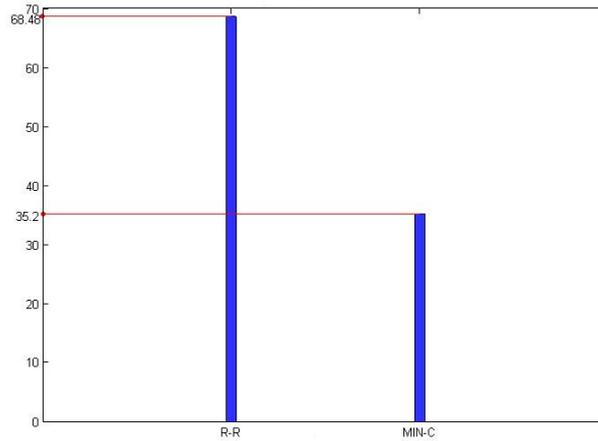
**Figure 6. The Cost and the Profit of each Node at the True Quotation based on Multiple-Replica Mechanism**

Figure 6(a) shows the payments, the costs and the profits of all nodes. Figure 6(b) shows the payment, the cost and the profit of each node after putting the nodes in ascending order according to the cost.

From Figure 6(a) we can see that the profits of the nodes which successfully place the replicas are all greater than zero and the nodes which are failed to do that can't get the payment so their costs and profits are all zero. As what the Figure 6(b) shows after putting the nodes in ascending order according to the cost, we can see that the system pays the same for all the nodes which place replicas. The difference is that the cost of each node is not the same, so different profits are obtained. The lower the node's cost is the higher its profit is.

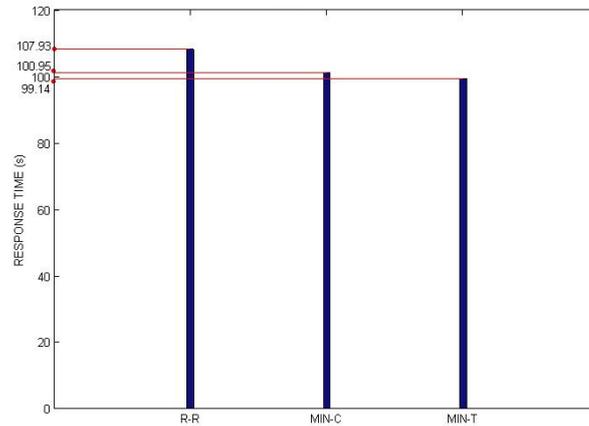
## 6. The Performance of the Algorithm Compared with Others

This section compares the built-in algorithm of the CloudSim platform (Round Robin, RR), the shortest response time algorithm (MIN-TIME) with the algorithm proposed in this paper based on the replica placement strategy. Because this section is familiar to some people so the some parts are omitted. Only some necessary explanation about the results of the experiment is given out.



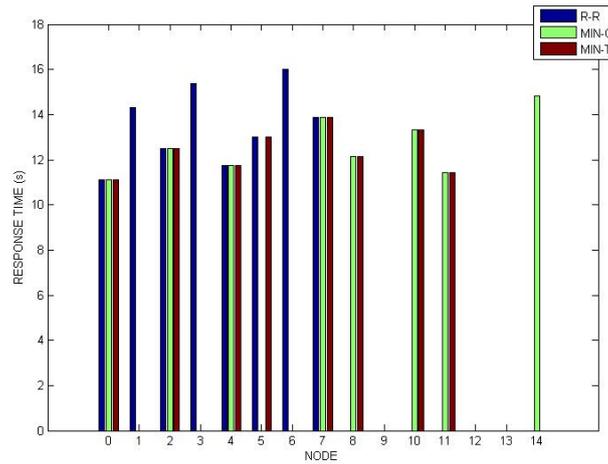
**Figure 7. The Total Cost under Different Algorithms based on Multi-Replicas Mechanism**

We can see from Figure 7 that compared with the built-in R-R algorithm of the CloudSim platform this algorithm helps the total cost which the system consumes falls by almost half.



**Figure 8. System Response Time under Different Algorithms based on Multi-Replicas Mechanism**

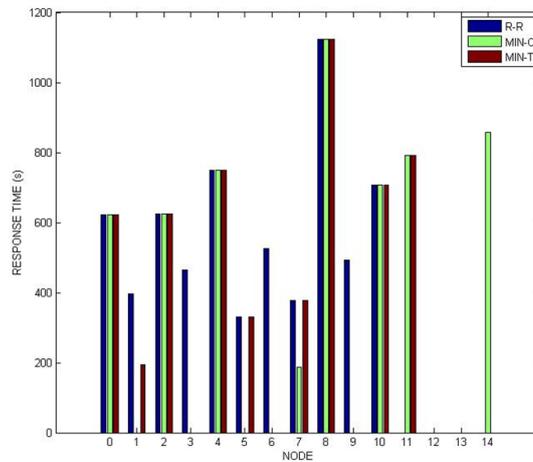
What can be concluded from Figure 8 is that the system response time of the algorithm proposed in this paper is shorter than the R-R algorithm too. But it is a little longer than the shortest response time algorithm. This is because this algorithm aims to minimize the total system cost rather than the shortest response time in the initial design.



**Figure 9. The Response Time of each Node under Different Algorithms based on Multi-Replicas Mechanism**

From Figure 9 we can see that the response time of each node based on the R-R algorithm is obviously too long, while the response time based on this algorithm and MIN-TIME algorithm is basically the same.

The object that Figure 9 studied is multiple replicas of a single user, so each storage node only owns one replica. When the object of the study expands to multiple replicas of multiple users, there are replicas of multiple users placed on each storage node, in which case the response time of each node in the cloud system can approximate to Figure 10.



**Figure 10. The Response Time of each Node under Different Algorithms based on Single-User Mechanism**

### Summary

For further consideration of the reliability and the availability of storing data, this paper further gives data block replica placement strategy. At the first place, the paper briefly introduces the storage mechanism when place a single replica, and then Multiple-replica-to-multiple-service placement strategy is drawn out. Following that, this paper presents the construction and the implementation of the mechanism's dominant strategy equilibrium situation which purpose is to guide the mechanism to achieve dominant strategy equilibrium. At last, it makes the performance analysis on replica placement

mechanism which proves that the mechanism maximizes the benefits and meets the designer's objectives at the same time. Namely it meets the incentive compatibility.

This paper mainly applies the CloudSim software, one cloud platform simulation tool, to simulate the algorithm. At the first place, the paper briefly introduces the development background of the CloudSim and gives a concise description of its architecture and core classes. Furthermore, it also expounds the environment configuration and the implementation of the program flow of the CloudSim in the practical application. Before the algorithm simulation, the paper sets the experimental parameters, and then uses the CloudSim platform to analyze the performance of the algorithm and compares it with other related algorithms. At last, it does simulation experiments on the replica placement strategy of the data block, which shows that replica placement strategy can also achieve the established goal.

## Acknowledgements

This work is supported by Beijing Higher Education Young Elite Teacher Project (No. YETP1729); Beijing Education Commission Social Science Project (No. SM201411626006); and Education & Teaching Project of Beijing Youth Politics College (No. ZY201501); and Beijing Federation of Social Science Circles of Young Talent Project (No. 2015SKL027)

## References

- [1] R N Calheiros, R Ranjan, A Beloglazov, *et al.* "CloudSim: a toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms[J]". *Software: Practice and Experience*, vol. 41, no. 1, (2011), pp. 23-50.
- [2] R N Calheiros, R Ranjan, C A F De Rose, *et al.* "Cloudsim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services[J]". *arXiv preprint arXiv:0903.2525*, (2009).
- [3] W H Bell, D G, C. Millar A P, *et al.* "Optorsim: A grid simulator for studying dynamic data replication strategies[J]". *International Journal of High Performance Computing Applications*, vol. 17, no. 4, (2003), pp. 403-416.
- [4] B D Lee, J B. Weissman "Dynamic Replica Management in the Service Grid[C]//High Performance Distributed Computing", 2001. *Proceedings. 10th IEEE International Symposium on. IEEE*, (2001), pp. 433-434.
- [5] L Anderegg, S Eidenbenz. "Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents[C]".//*Proceedings of the 9th annual international conference on Mobile computing and networking. ACM*, (2003), pp. 245-259.
- [6] R L Grossman. "The case for Cloud Computing[J]". *IT professional*, vol. 11, no. 2, (2009), pp. 23-27.
- [7] D Brodsky, M J Feeley, N C. Hutchinson, "Topology Sensitive Replica Selection[C]//Reliable Distributed Systems, 2006. SRDS'06. 25th IEEE Symposium on. IEEE, (2006), pp. 18-28.
- [8] Hershberger, S. Suri, "Vickrey prices and shortest paths: What is an edge worth?[C]". //*Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on. IEEE*, (2001), pp. 252-259.

## Author



**Wang HongXia.** Wang HongXia(1979-) was born in XinJiang, China. She received her Ph.D degree from the University of Science and Technology, Beijing, China. She obtained her B.Sc degree in computer science in 2002 and M.Sc degree in computer application in 2005 from Liaoning Shihua University, Fushun, China. Now, she is a vice-professor at the Beijing Youth Politics College. Her current research interests include information management, image processing, multimedia technology and network communication.

