A VCG Mechanism Based Storage Allocation Strategy in Cloud Computing Environment

Zhenqiang Mi[1], Miao Zhang[1] and Zenggang Xiong[2]
[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China
[2] School of Computer and Information Science, Hubei Engineering University, Xiaogan, Hubei, China
Email: mizq@ustb.edu.cn; {zhangmiao_lyy, jkxxzg2003}@163.com

***Abstract*—**With the significant growth in the quantity of cloud storage customers, it becomes a primary issue for cloud storage companies to properly allocate cloud resources among selfish and rational virtual machines (VMs), which compete for resources with others only to maximize their own performance regardless of the overall system equalization. In this paper, a mechanism approach is presented to cope with this problem. Distinguish to previous studies, the proposed approach not only considers storage cost and valuation of user data, but also comprehensively takes the memory of storage nodes, data center CPU and network bandwidth into consideration. Moreover, we design a guidance algorithm to achieve dominant equilibrium of the system. Through theoretical analysis and experiments, the proposed mechanism has proved its effectiveness and incentive compatibility for the overall system equalization.

***Index Terms*—**Cloud computing, storage allocation, mechanism design, VCG mechanism

I. Introduction

Cloud computing develops rapidly in recent years. It has been widely accepted as the third technology revolution following personal computers and the Internet [1], [2]. Cloud computing is computing that involves a large number of computers connected through a communication network such as the Internet, and in which dynamically scalable and often virtualized resources are pro-vided as services over the Internet.

Cloud storage is a typical application of cloud computing. In scenarios where the core of the cloud computing system is the storage and management of massive data, cloud computing system requires a lot of storage device configuration. Then, the cloud computing system turns into a cloud storage system which aims to establish an information-sharing storage environment across the Internet so as to provide clients with data storage service at any given time and place.

The storage allocation strategy is one of the most important problems in cloud storage system. As Virtual Machines (VMs) are selfish and rational, a VM will compete with others to maximize its own profit. This problem brings new challenges to cloud providers who must thwart non-cooperative behavior as well as allocate resources among selfish VMs efficiently. Many re-source allocation works have been done in the past. In [3], it is pointed that, the growing demand increases the energy consumption of data centers, which translates to high energy cost and high carbon emissions which are not environmentally sustainable, so the strategy pro-posed focus on energy saving. There also exist other storage allocation strategies such as load balancing algorithm [4] and value-based scheduling [5], whereas most of the existing work focuses on the storage cost to achieve a better storage allocation performance. But they often ignore the importance of the user's data value and synthesized performance metrics of the cloud nodes. Data value represents the importance of data in both users' and cloud providers' perspective. Different values of data should be considered when allocating the cloud storage, in order to maximize the profits while minimizing the impacts on users' quality of services (QoS). In the meantime, considering the unique characteristic of cloud storage system, sophisticated performance metrics, e.g., network bandwidth, CPU and memories, should be taken into consideration when allocating the storage to effectively balance the load of each cloud node, and finally reduce the cost of the entire system.

Interestingly, the storage allocation problem is also a specific aspect in the research area of resource allocation, in which many proposed solutions are readily applicable. Resource allocation is one of the most challenging problems in resource management. As a key issue in the research of cloud computing, resource allocation has attracted numerous efforts in both academic and industry societies [6]. Work [7] proposes a reinforcement learning based management system for dynamic allocation of servers, the proposed method tried to maximize the profit of the host data center in cloud computing. A generic request allocation and scheduling scheme was proposed in [8] to achieve desired percentile service level agreements (SLA) goals of consumers and to increase the profits to the cloud provider. Storage allocation is a particular aspect of resource allocation, especially in

Manuscript received September 29, 2014; revised December 30, 2014.

Corresponding author email: jkxxzg2003@163.com.

This work was supported by the National Science Foundation of China (Grant No. 61272432, 61202508, 61370092, 61370132 and 61472033), Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410), and Fundamental Research Funds for the Central Universities (FRF-TP-14-045A2).

doi:10.12720/jcm.9.12.962-970

cloud computing environment, where heterogeneous clusters hinder the efficiency of storage strategy. In [9], a dynamic storage allocation algorithm was proposed based on the performance of heterogeneous nodes. The proposed algorithm aims at improving the performance of Hadoop. Game-theory based resource allocation mechanisms have also received a considerable amount of attentions to solve the optimization problem of resource allocation in cloud computing environment [10]-[12]. Our work differs from the existing methods by considering the selfish behaviors of the cloud node, so that auction-based method [13] is proposed to achieve the objective of storage allocation in an interactive manner.

Motivated by the observations, in this paper, we propose a mechanism design approach for storage allocation in cloud computing platforms. The proposed method explicitly considers individual VMs' rationality so as to thwart selfish behavior and enforce truth-revealing for selfish VMs. First, we model the cloud storage system according to its massive and heterogeneous nature. Then, a Vickrey-Clarke-Groves (VCG) [14] mechanism based storage allocation strategy is pro-posed to solve the storage allocation problem on the basis of storage cost and the valuation of user data, as well as considering various aspects of performance of nodes. The proposed approach can effectively avoid the selfish behavior of cloud node and force the cloud node to truthfully bid on tasks, storage allocation is formed based on the truthful bidding according to each node's performance to maximize the profit of the cloud storage system. We provide extensive theoretical analysis and experimental studies to verify the correctness and efficiency of the proposed strategy. In [15], the authors proposed a VCG mechanism aiming at load balancing in distributed system, which is similar to our work. The difference is that we focus on system cost rather than load balancing. Our proposed VCG mechanism aims to solve the storage allocation problem on the basis of storage cost and the valuation of user data as well as considering various aspects of performance of nodes. The proposed mechanism achieves incentive compatible and enforces truth-reavling for selfish VMs.

The main contributions of this work are as following: 1) we model the storage allocation problem according to users' data value, as well as different types of data, e.g., text/video/picture, according to their unique properties in cloud storage system; 2) we design an efficient storage allocation strategy based on mechanism design method, which is able to maximize the overall profits by eliminating the selfish behavior of VMs; 3) we provide non-trivial experimental studies to evaluate the performances and compare it with the analysis of the proposed strategy.

The rest of this paper is organized as follows. Preliminaries and problem formulation are presented in section II. Then in section III, we discuss the VCG based storage allocation strategy consisting algorithm and truthful bidding. The performance of the algorithm is analyzed in section IV. Section V gives experimental performance evaluation. And the paper is finally concluded in Section VI.

II. Preliminaries

Before the introduction of the system models and the proposed methods, we first provide the definitions of the terms and notions in Table I for future references.

TABLE I. Notations

Notation	Description
$X(o,p)$	Direct Displaying Mechanism
c_i	unit storage cost of a storage node
$\boldsymbol{C_i}$	the set of cost of a data storage node
b_i	the set of quotation of a storage node
$\boldsymbol{b}$	the set of quotations of all participants, $\boldsymbol{b}=(b_1,b_2,...,b_n)$
$\boldsymbol{b_{-i}}$	The set of quotation of storage nodes except i, $\boldsymbol{b_{-i}}=(b_1,b_2,...,b_{i-1},b_{i+1},...,b_N)$
o_i	the distribution of a participant
$\boldsymbol{o}$	the set of distribution of all participants, $\boldsymbol{o}=(o_1,o_2,...,o_n)$
$\boldsymbol{o_{-i}}$	the set of distribution when user i is not participating, $\boldsymbol{o_{-i}}=(o_1,o_2,...,o_{i-1},o_{i+1},...,o_n)$
p_i	the payment to a storage node
$U_i(.)$	user's benefit function
$q_i(m_i,b_i,pr_i)$	the property of node i
$f_i(.)$	A private inverse function.
$t_i(c_i,m_i)$	the type of the node i
m	Data Scale
bw	Bandwidth
pr	processing rate
$r_i(m_i,bw_i,pr_i)$	data attribute
v	data value
$g(.)$	a private proportional function
ω_l	the weight of each data type

A. Introduction to VCG Mechanism

The original scheme of VCG mechanism is proposed to solve the following problem [14]: Assuming the cost of a manufacturer for producing some given goods is C. If user i gets these goods, the real benefits would be U_i. All the users need to report their benefits $B_i(B_i = U_i \ or \ B_i \neq U_i)$ to the manufacturer at first. The manufacturer, in turn, according to the users' benefits, implements goods distribution and user payments under certain policies. The designing of mechanism has to meet two goals: First, the maximum social benefit of mechanism; second, $B_i = U_i$ is the best bidding strategy for users.

The Definition of VCG Mechanism is as following: If a Direct Displaying Mechanism $X(o,p)$ satisfies the following two constrains, it belongs to VCG Mechanism. The constrains are provided in Eq. (1) and Eq. (2) as follows:

$$o^{VCG}(\boldsymbol{b}) = \arg\max_{\boldsymbol{o}\in\boldsymbol{O}} \sum_i V_i(b_i, \boldsymbol{o}) \quad (1)$$

$$p_i^{VCG}(\boldsymbol{b}) = \max \sum_{j\neq i} V_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j\neq i} V_j(b_j, \boldsymbol{o}^{VCG}) \quad (2)$$

In which, $\boldsymbol{b}$ represents the set of quotations of all participants, $\boldsymbol{b} = (b_1, b_2, ..., b_n)$; $\boldsymbol{o}$ represents the set of distribution of all participants, $\boldsymbol{o} = (o_1, o_2, ..., o_n)$; while $\boldsymbol{o}_{-i}$ represents the set of distribution when user i is not participating, $\boldsymbol{o}_{-i} = (o_1, o_2, ..., o_{i-1}, o_{i+1}, ..., o_n)$. And $V_i(b_i, \boldsymbol{o})$ represents the desired profits of the mechanism for participant i .

Equation (1) defines the allocation rule of mechanism, which selects the maximum social benefits for society. While. (2) defines the payment rule of mechanism. The first part of which refers to the maximum benefits achieved when reusing (1) after removing participant i , and the second part presents the total benefits under the current allocation after removing participant i .

The payment rule is the core of mechanism design and the essence of VCG mechanism, which is implemented on the basis of the allocation rule. From (2), we know that the definition of each participant's payment is decided by its "externality [16]" to the mechanism. The payment is required to enforce truth-revealing effectively since only under this premise the designing and implementing of mechanism is meaningful.

When the total currency in circulation is 0, the system fulfills BB (Budget Balance). At this time, the payment of buyers is just enough to make up seller's production costs. VCG Mechanism, however, has the BU [17] (Budget Unbalance) defect, i.e., $\sum_i Pay_i < C$, where $\sum_i Pay_i$ denotes the total payment of participant i . Reference [18] analyzes the BU defect of regulatory mechanism of single-user, which points out that the benefits generated when the user access to services is measured by its own, and it is not exactly the same as real payment. For example, when a user wants to process some important confidential files immediately regardless of how much the fee is, the user gets the benefits which cannot be measured by currency. Therefore, the VCG Mechanism can be used within the single-user environment, although it has BU defect. Another important property of VCG Mechanism is that VCG Mechanism is proved to be the only real implementation under the restriction of quasi-linear preferences [19].

B. Models of Storage Node

According to the heterogeneous devices within the Cloud Storage System, the performance of each node is different, which means the providing quality of service varies. For one data storage node i , we set its property as $q_i(m_i, bw_i, pr_i)$, in which m_i and bw_i refer to the memory size and the network bandwidth that node i can offer, respectively. And pr_i refers to the average CPU processing rate of node i .

We set the unit storage cost of a node i as c_i , which indicates the resources consumed while the node is providing storage service to users, i.e., the prices needed to be paid. The unit storage cost of each node is related to its own property, say, $c_i = f_i(q_i)$ in which $f_i(.)$ is a private inverse function. The explanation of inverse relationship is as following: When providing the same storage services, the better the quality of a data storage node is, the less the resources is consumed, and in this way the costs are reduced. Vise verse, if the quality of a data storage node is low, the node would find it difficult to offer the same storage services, and it would consume more resources and increase the costs.

For instance, the unit storage cost of a data storage node c_i , combines with m_i and forms the type of the node $t_i(c_i, m_i)$. c_i is the private information of a data storage node, meaning that except for the node itself, the information remains unknown to the system and other users. While m_i is public knowledge, that is, the available storage space of each storage node can be detected by the system as well as other participants of the mechanism.

We denote $\boldsymbol{T_i} = (t_{i1}, t_{i2}, ..., t_{ik_i})$ and $\boldsymbol{C_i}$ respectively as the set of types and the set of costs of a data storage node i . Assuming the true cost of a data storage node i is $c_i (c_i \in \boldsymbol{C_i})$, and quotation of node i to the external is $b_i (b_i \in \boldsymbol{C_i})$. So either $b_i = c_i$ or $b_i \neq c_i$.

We choose several metrics such as Data Scale (m), requiring bandwidth (bw), desired processing rate (pr) and data type as the objects of study. The first three are described as the properties of data, and data type acts as a form of data classification (such as text, picture, video and other data types). We define the attributes for the text data as $r_1(m_1, bw_1, pr_1)$, for the picture data as $r_2(m_2, bw_2, pr_2)$, and for the video data as $r_3(m_3, bw_3, pr_3)$. The total data size on the data node is represented as $m = m_1 + m_2 + m_3$.

According to the performance requirements to the data storage node, data node estimates its own data value as $v = g(r)$, in which $g(.)$ is a private proportional function. The explanation of proportional relationship is as following: If the node believes that its own data is of great significance, then it would rather save the data on higher-quality node, we consider this one has a higher data value. Since different data types require different emphasizes on the performances, function $g(.)$ varies. A crucial document, for instance, requires a large number of duplicates, while the video data demands high real-time. Assuming that on one data node the recognized value of text data is $v_1 = g_1(r_1)$, picture data is $v_2 = g_2(r_2)$, and video data is $v_3 = g_3(r_3)$. So to the whole data node, its recognized value of data is:

$$v = \sum_{l=1}^{3} (v_l \times \omega_l) \tag{3}$$

In which ω_l denotes the weight of each data type, and $0 \le \omega_l \le 1$, $\sum_{l=1}^{3} \omega_l = 1$.

The type for a data node reported to the mechanism is $t(v, m)$, in which v, m are both true values.

III. VCG BASED STORAGE ALLOCATION STRATEGY

Supposing there are N data storage nodes providing services. At one point T, there are M users issuing storage requests.

We know that when we design the mechanism of multi-user to single-service ($M:1$), the mechanism cannot activate the data storage nodes to provide services because of BU defect within the VCG mechanism, which results in its unavailability in practice. However, if the mechanism is designed for single-user to multi-service ($1:N$), it is feasible for the mechanism to be implemented since users can choose nodes which match the quality of service (QoS) requirement.

A. VCG based Storage Allocation Algorithm

The function format of VCG Algorithm may vary according to applications, but the guiding ideology remains the same. Focusing on the objectives in this work, we can convert (1) to the following expression:

$$\begin{aligned} \boldsymbol{o}^{VCG}(\boldsymbol{b}) &= \arg\max_{\boldsymbol{o} \in \boldsymbol{O}} \sum_i V_i(b_i, \boldsymbol{o}) \\ &= \arg\max_{\boldsymbol{o} \in \boldsymbol{O}} \sum_i (V - C_i(b_i, \boldsymbol{o})) \\ &= \arg\max_{\boldsymbol{o} \in \boldsymbol{O}} (v \times m - \sum_i C_i(b_i, \boldsymbol{o})) \end{aligned} \tag{4}$$

Since the total data value of a data node $V = v \times m$ in (4) is a fixed value. We can further convert (4) to:

$$\begin{aligned} \boldsymbol{o}^{VCG}(\boldsymbol{b}) &= \arg\max_{\boldsymbol{o} \in \boldsymbol{O}} (v \times m - \sum_i C_i(b_i, \boldsymbol{o})) \\ &= \arg\min_{\boldsymbol{o} \in \boldsymbol{O}} \sum_i C_i(b_i, \boldsymbol{o}) \\ &= \arg\min_{\boldsymbol{o} \in \boldsymbol{O}} \sum_i (b_i \times o_i) \end{aligned} \tag{5}$$

In which $b_i \in \boldsymbol{C_i}$; and o_i represents the allocation of storage node i that derived from the set of quotation $\boldsymbol{b}$, $o_i(\boldsymbol{b}) \in \boldsymbol{o}(\boldsymbol{b})$.

Note for (5): b_i is the cost value that a storage node i reports to the system, and it is uncertain to be the real cost c_i. Since c_i is the private information of node i, which is not known by the outside, the mechanism can only implement the allocation on the basis of its quotation b_i.

It can be seen from (5) that the goal of allocation function is to minimize the total cost of storage system. Based on this guiding ideology, the system would prefer lower quoted nodes when allocating tasks. The lower quotation of node means the better comprehensive performances. Thus, this type of allocation is to assign tasks to high-quality nodes as more as possible.

In order to overcome the BU defect, the mechanism would follow the principle of $b_i \le v$ which means the quoted cost is lower than user's estimated value while selecting the data storage nodes. Nodes that meet $b_i > v$ will be abandoned because of the high cost. Based on this principle, better performed nodes would serve more users, while the worse ones would gradually be eliminated.

We can conclude from the previous analysis, the storage tasks allocated to nodes have to meet the following conditions:

$$0 \le o_i(\boldsymbol{b}) \le m_i, \quad \sum_i o_i(\boldsymbol{b}) = m$$

TABLE II. THE ALLOCATION OF SINGLE-USER TO MULTI-SERVICE MECHANISM.

Input: Bids submitted by Nodes: $\boldsymbol{b} = (b_1, b_2, \cdots, b_N)$
Memory submitted by Nodes: $\boldsymbol{m} = (m_1, m_2, \cdots, m_N)$
Data value submitted by User: v
Total Data size submitted by User: m
Output: Allocation: $\boldsymbol{o} = (o_1, o_2, \cdots, o_n)$

```
Choose Nodes satisfying b_i < v;
Sort the chosen Nodes in increasing order of their bids (b_1 ≤ b_2 ≤ ··· ≤ b_n);
m_s ← Σ_i m_i;
while m_s > m do
    o_n ← 0;
    m_s ← m_s − m_n;
    n ← n − 1;
end while
for i = 1 to n do
    o_i ← m_i;
end for
o_{n+1} ← m − Σ_i m_i;
```

The mechanism allocation is shown as in Table II and is implemented as following:

1) Select the nodes which meet $b_i \le v$ as candidate nodes according to the quotation of N data storage nodes. Assuming there are n candidate nodes.

2) Arrange these n nodes in an ascending order of b_i.

3) The distribution of these n nodes after ascending arrangement is:

$$o_i(\boldsymbol{b}) = \begin{cases} m_i & 0 < i < k \\ m - \sum_{j=1}^{k-1} m_j & i = k \\ 0 & k < i < n \end{cases} \tag{6}$$

In (6), the kth ranked node satisfies the conditions $\sum_{j=1}^{k-1} m_j < m$ and $\sum_{j=1}^{k} m_j \ge m$.

B. Truthful Bidding

Under the guidance of allocation function, mechanism can achieve the goal of minimum total system cost. But it is not enough just having allocation function since the quotation b_i may not be equal to its true cost c_i. If any competitor reveals false quotation, the real cost does not reach the minimum. As a result, mechanism has to adopt

an incentive method to enforce truth-revealing. It is completed by the payment function to enforce truth-revealing within VCG mechanism design. The design idea of payment function is that the payment of a storage node is not only relevant to its own quotation, but a quantification of influence the node have on the whole system. It is a comprehensive performance of this node and the quotation of other participants, which provides a function to all service providers to restrain each other.

On the basis of (5), we make payments of n storage nodes that satisfy the condition. Converting (2) to a payment function of node i :

$$p_i^{VCG}(\boldsymbol{b}) = \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j \neq i} C_j(b_j, \boldsymbol{o}^{VCG}) \quad (7)$$

In which $\boldsymbol{b}$ represents the set of quotation of all participants, $\boldsymbol{b} = (b_1, b_2, ..., b_n)$; $\boldsymbol{o}$ represents the set of allocation to all participants, $\boldsymbol{o} = (o_1, o_2, ..., o_n)$; while $\boldsymbol{o}_{-i}$ is the set of allocation when node i is not participating in the competition, $\boldsymbol{o} = (o_1, o_2, ..., o_{i-1}, o_{i+1}, ..., o_n)$.

In (7), the first part refers to the minimum cost when i is not participating in this bidding. The second part refers to the total cost of other nodes when node i participates. Note that the two parts in (7) are calculated separately under the same set of quotation $\boldsymbol{b}$. We denote the quotation set of the remaining nodes apart from the node i as $\boldsymbol{b}_{-i} = (b_1, b_2, ..., b_{i-1}, b_{i+1}, ..., b_n)$, thus $\boldsymbol{b} = (b_i, \boldsymbol{b}_{-i})$.

We know from (7) that the relationship between the quotation b_i and its profit is not displayed directly. Instead, it is shown by whether the existence of node i will have an influence on other nodes in the system. To further illustrate the construction of payment function, we provide the profit expression of node i in the bidding.

1) If $b_i = c_i$, node i reports its real cost and the profit is:

$$\begin{aligned} u_i^{VCG}(\boldsymbol{b}) &= p_i^{VCG}(\boldsymbol{b}) - C_i(c_i, \boldsymbol{o}^{VCG}) \\ &= \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j \neq i} C_j(b_j, \boldsymbol{o}^{VCG}) - C_i(b_i, \boldsymbol{o}^{VCG}) \\ &= \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j} C_j(b_j, \boldsymbol{o}^{VCG}) \end{aligned} \quad (8)$$

From (8), we know that when node i chooses truth-revealing, the payment is actually the difference between the total system cost for whether it participates the bidding or not. Since the allocation of mechanism is ordered in an ascending way, the difference cannot be negative. Equation (8) shows that the total cost of system decreases because of the participation of node i . The positive part is converted to profit which is awarded to node i .

2) If $b_i \neq c_i$, when node i chooses false-revealing, the profit is:

$$\begin{aligned} u_i^{VCG}(\boldsymbol{b}) &= p_i^{VCG}(\boldsymbol{b}) - C_i(c_i, \boldsymbol{o}^{VCG}) \\ &= \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j \neq i} C_j(b_j, \boldsymbol{o}^{VCG}) - C_i(c_i, \boldsymbol{o}^{VCG}) \\ &= \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j} C_j(b_j, \boldsymbol{o}^{VCG}) \\ &\quad - (C_i(c_i, \boldsymbol{o}^{VCG}) - C_i(b_i, \boldsymbol{o}^{VCG})) \end{aligned} \quad (9)$$

From (9) we know that when node i chooses false-revealing, the mechanism would still convert the contribution to the system as a reward, but it is not the actual profit of node i . The profit of participants under the false quotation will be analyzed in detail in the following section.

IV. Performance Analysis

Before analyzing our proposed VCG based Storage Allocation Strategy (VSAS), we need to make several assumptions of the characteristics of the nodes in Cloud.

Assumption 1: *Each node is autonomous, that is, each node is independent, and there is no central node controlling others.*

Assumption 2: *Nodes are conservative. If a node detects that the profit achieved when false-revealing is the same as that of truth-revealing, it will choose to offer the real cost.*

Assumption 3: *Nodes are selfish. Each node tries to maximize its own benefit regardless of other nodes and the overall system.*

Lemma 4.1: *The raising of quotation of storage nodes cannot make their profits increase*

Proof: selecting one node as the study node, then any one set of quotation of storage nodes apart from i can be denoted as $\boldsymbol{b}_{-i} = (b_1, b_2, ..., b_{i-1}, b_{i+1}, ..., b_N)$. Assuming that node i is located at the ith in an ascending order while truth-revealing, and the profit of which is u_i , then there are k nodes get the tasks. When the storage node raises its quotation, it will be located at the $i^{'\text{th}}$ in an ascending order, the profit becomes $u_i^{'}$, and there are $k^{'}$ nodes get the tasks.

If the node cannot get the tasks while truth-revealing as $c_i > b_k$. After raising the quotation, the node definitely cannot get the tasks as well, since $b_i > c_i > b_k$, at this moment $u_i = u_i^{'} = 0$. If the node gets the tasks while truth-revealing as $c_i < b_k$, it cannot get the tasks after raising its quotation for $b_i > b_{k^{'}} > b_k > c_i$, thus the profit change from $u_i > 0$ to $u_i^{'} = 0$. Finally, if the node gets the tasks while truth-revealing as $c_i < b_k$, it can still get the tasks after its quotation raises, in this case, there are two situations:

1) $i < i^{'} < k = k^{'}$, the node can get the tasks in both two quotations and $o_i^{'} = o_i = m_i$, it will make no influence on other nodes' allocation, i.e., $\boldsymbol{o}^{'} = \boldsymbol{o}$, the profit of false-revealing is:

$$\begin{aligned} u_i^{'} &= p_i^{'} - C_i(c_i, \boldsymbol{o}^{'}) \\ &= \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j \neq i^{'}} C_j(b_j, \boldsymbol{o}^{'}) - c_i \times o_i^{'} \\ &= \min \sum_{j \neq i} C_j(b_j, \boldsymbol{o}_{-i}) - \sum_{j \neq i} C_j(b_j, \boldsymbol{o}) - c_i \times m_i \end{aligned} \quad (10)$$

From (10), we can know that $u_i = u_i^{'}$ .Thus the profit of node remains the same although the quotation increases.

2) $i < k \le k' = i'$, the node gets the storage tasks $o_i = m_i$ when truth-revealing, and it ranks the k'^{th} after raising the quotation and gets the tasks $o_i' < m_i$, which would have influence on the allocation of the remaining nodes. The difference between the profits achieved when false-revealing and truth-revealing of nodes is:

$$\begin{aligned} u_i' - u_i &= (\min \sum_{j \ne i} C_j(b_j, \boldsymbol{o_{-i}}) - \sum_{j \ne i'} C_j(b_j, \boldsymbol{o}') - c_i \times o_i') \\ &\quad - (\min \sum_{j \ne i} C_j(b_j, \boldsymbol{o_{-i}}) - \sum_{j \ne i} C_j(b_j, \boldsymbol{o}) - c_i \times o_i) \\ &= (\sum_{j \ne i} C_j(b_j, \boldsymbol{o}) + c_i \times o_i) - (\sum_{j \ne i'} C_j(b_j, \boldsymbol{o}') + c_i \times o_i') \\ &= (b_k \times o_k - \sum_{j=k}^{k'-1} b_j \times o_j') + (c_i \times o_i - c_i \times o_i') \\ &< b_k \times (o_k - \sum_{j=k}^{k'-1} o_j') + c_i \times (o_i - o_i') \\ &= (b_k - c_i) \times (o_i' - o_i) \end{aligned} \tag{11}$$

As to the first part of (11), when the node reveals the true cost, we can know that $i < k$, so $c_i < b_k$, i.e., $b_k - c_i > 0$. The relationship between the distribution obtained when false-revealing and that of truth-revealing is $o_i' < o_i = m_i$, thus $o_i' - o_i < 0$. We can reach that $u_i' - u_i < 0$. Therefore, the profit of node decreases while the quotation raises.

To sum up, the profit will not increase after raising the quotation whatever situation is when the node chooses truth-revealing.

Lemma 4.2: *Storage node cannot augment its profit by decreasing its quotation*

Proof: selecting one node i as the study node, then any one set of quotation of storage nodes apart from i can be denoted as $\boldsymbol{b_{-i}} = (b_1, b_2, ..., b_{i-1}, b_{i+1}, ..., b_N)$. Assuming that node i is located at the ith in an ascending order while truth-revealing, and the profit of which is u_i, then there are k nodes get the tasks. When the storage node decreases its quotation, it will be located at the i'^{th} in an ascending order, the profit becomes u_i', and there are k' nodes get the tasks.

If the node gets the tasks for $c_i \le b_k$ while truth-revealing, after decreasing its quotation, the situation becomes $b_i < c_i \le b_k$, then $o_i' = o_i$. From lemma 4.1 we know that $u_i' = u_i$. If the node reveals its real cost, i.e., $c_i = b_k$, after decreasing the quotation there will be $b_i < c_i = b_k$, then:

$$\begin{aligned} u_i' &= \min \sum_{j \ne i} C_j(b_j, \boldsymbol{o_{-i}}) - \sum_{j \ne i'} C_j(b_j, \boldsymbol{o}') - c_i \times o_i' \\ &< \min \sum_{j \ne i} C_j(b_j, \boldsymbol{o_{-i}}) - (\sum_{j \ne i', j \ne k'} C_j(b_j, \boldsymbol{o}) + \sum_{j=k'+1}^{k} C_j(b_j, \boldsymbol{o})) \\ &= \min \sum_{j \ne i} C_j(b_j, \boldsymbol{o_{-i}}) - \sum_{j} C_j(b_j, \boldsymbol{o}) \end{aligned} \tag{12}$$

From (12) we know that $u_i' < u_i$, the profit decreases after lowering the quotation.

If the node has not get the tasks due to $c_i > b_k$ while truth-revealing, and if $b_k < b_i < c_i$ after decreasing its quotation, then the node cannot get the tasks as well, and $u_i' = u_i = 0$. But if $b_i \le b_k < c_i$ after false-revealing, the node can get the tasks. Then according to (12) we know that $u_i' < u_i = 0$, the profit would be negative after decreasing the quotation.

To sum up, the profit will not increase after decreasing the quotation whatever situation is when the node chooses truth-revealing.

Theorem 4.1: *Truth-revealing of storage node is the dominant strategy, and the VCG mechanism meets the incentive compatibility characteristic.*

Proof: Combining lemma 4.1 and 4.2, we know that: $\forall i \in N$, $\forall \boldsymbol{b_{-i}} = (b_1, b_2, ..., b_{i-1}, b_{i+1}, ..., b_N)$, $\exists b_i^*$ makes $u_i(b_i^*, \boldsymbol{b_{-i}}) \ge u_i(b_i, \boldsymbol{b_{-i}})$, thus the dominate strategy of the node exists, which is its truth-revealing for $b_i^* = c_i$. Since $u_i(b_i^*, \boldsymbol{b_{-i}})$ is not strictly larger than $u_i(b_i, \boldsymbol{b_{-i}})$, the dominant strategy is weak here. But each node is conservative, when it finds that false-revealing cannot increase its profit, the node would eventually choose truth-revealing to avoid unnecessary risk. Thus the mechanism meets incentive compatibility.

Theorem 4.2: *The mechanism, for storage nodes, meets ex post individual rationality.*

Proof: Theorem 4.1 points out, to all storage nodes, the truth-revealing is dominant strategy. Therefore the quotation of each node, after the gaming, fulfills $b_i^* = c_i$. According to (8), the profit of node is $u_i^{VCG}(\boldsymbol{b}) = \min \sum_{j \ne i} C_j(b_j, \boldsymbol{o_{-i}}) - \sum_j C_j(b_j, \boldsymbol{o}^{VCG})$.

Apparently, the total cost of node i not participating the bidding is bigger than or equal to that of it does, then u_i is not negative. Thus, the mechanism, for all storage nodes, meets ex post individual rationality.

V. Experimental Evaluation

A. Parameters and Settings

According to section II which mentions the mathematical modeling of cloud storage system, the average storage cost of data storage node is:

$$c_i = c_{0i} + (\kappa_m \frac{\xi}{m_i} + \kappa_b \frac{\beta}{bw_i} + \kappa_u \frac{\gamma}{pr_i}) \tag{13}$$

In (13), c_{0i} represents the initial cost of data storage node. $\kappa_m, \kappa_b, \kappa_u$ indicate the weight of memory, bandwidth and processing rate respectively. In our study, we take a balanced consideration of all the factors, i.e., $\kappa_m = \kappa_b = \kappa_u = 1/3$ for the purpose of generalization, then (13) can be transformed to the following expression:

$$c_i = c_{0i} + \frac{1}{3}(\frac{\xi}{m_i} + \frac{\beta}{bw_i} + \frac{\gamma}{pr_i}) \quad (14)$$

In Eq. (14), ξ, β, γ respectively indicate the constant parameters of quantified storage costs, the specific value of the parameters are set as in Table III to indicate that the bandwidth is more influential in our designed cloud storage systems.

For data nodes, we assume the storage data to be video data. We set $\omega_1 = \omega_2 = 0, \omega_3 = 1$, and the value of unit data in the data object node is

$$v_j = v_{0j} + (\frac{\lambda_m \delta m_j + \lambda_b \eta bw_j + \lambda_u \tau pr_j}{m_j}) \quad (15)$$

In (15), $\lambda_m, \lambda_b, \lambda_u$ respectively present the weight of data size, requiring bandwidth and desired processing rate. Since we assume the data in data nodes to be video data, a higher transmission rate is required, so we set $\lambda_m = 0.2, \lambda_b = 0.5, \lambda_u = 0.3$ .The priority is the same level. So (15) can be converted to

$$v_j = v_{0j} + (\frac{0.2\delta m_j + 0.5\eta bw_j + 0.3\tau pr_j}{m_j}) \quad (16)$$

In (16), δ, η, τ are the constant parameters of quantified data values. For the quantified values of data, size of the data is the most important characteristic, which is different from the cloud node's perspective. Therefore, as shown in Table III, the value of δ is relatively larger than the rests.

TABLE III. EXPERIMENT PARAMETERS

Data Storing Nodes		Data Nodes	
c_{0i}	[0.001，0.05]	v_{0j}	[0.01，0.1]
ξ	5	δ	0.5
β	10	η	0.04
γ	3	τ	0.3

In the experiments, we select 15 data storage nodes. The initial cost is in the interval [0.001, 0.05]. The number of virtual machines is 40. The bandwidth is generated uniformly with the step length of 70 in [200, 3000] bit, and the processing speed is generated uniformly with the step length of 20 in [200, 1000] Million Instructions per Second (MIPS). We set the number of cloud tasks as 500 and the length of cloud task as 4000Million Instructions (MI). The share strategies of tasks to virtual machines and virtual machines to hosts are both time-sharing.

B. Results and Discussions

Assuming that we select node #8 as the study object, we can derive the payments and receipts when node #8 reports its real cost, raises its quotation, or lowers its quotation respectively.

As shown in Table IV, when node #8 does not participate in the bidding, it would not affect the total costs of the remaining nodes, thus the value of the first column in the table remains the same. Instead, while node #8 participates in the bidding, it may affect the total costs of the remaining nodes, thus the value of second column varies. Note: costs in the fourth column and profit in the fifth column are private information, which is only known by the node itself.

TABLE IV. UNITS FOR MAGNETIC PROPERTIES

NODE#8	$\sum_{i\neq 8} c_{-i}$	$\sum_{i\neq 8} c_i$	PAY	COST	PROFIT
TRUE	1751.04	1049.6	701.44	40.96	660.48
HIGH	1751.04	1111.04	640	35.84	604.16
LOW	1751.04	1049.6	701.44	40.96	660.48

Furthermore, while node #8 participates in the bidding and reveals false quotation, its profit would not be higher than that of truth-revealing. So, for conservative node, it would choose truth-revealing as the optimal quotation eventually. We can achieve the result shown in Fig. 1 when further analyzing the profit of all quotation interval of node #8.

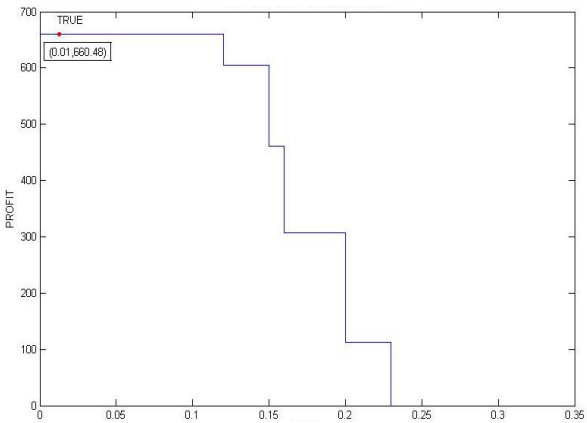

Fig. 1. The profit of node #8 in the different quotation

Within Fig. 1, the red dots indicate the real cost of node #8, we can conclude that the profit is optimal when truth-revealing. With the quotation of node increasingly deviate from the real cost, its profit gradually deceases to 0.

On the basis of the previous detailed analysis of node #8, we implement the simulation of all nodes under the condition of real quotation and spurious quotation. To make the results representative, we divide the false-revealing into two situation, that is increasing and decreasing quotation of nodes. We choose the quotation randomly in both situations.

In Fig. 2, to nodes #0, #2, #4, #8, #10, #11, #14, when they choose truth-revealing, they would get tasks and make some profits. When they decrease their quotations, they can definitely get the tasks, but at this time the profits of nodes would not increase. And when they increase their quotations, the experimental result shows the profits would decrease, even to 0 since they cannot get the tasks.

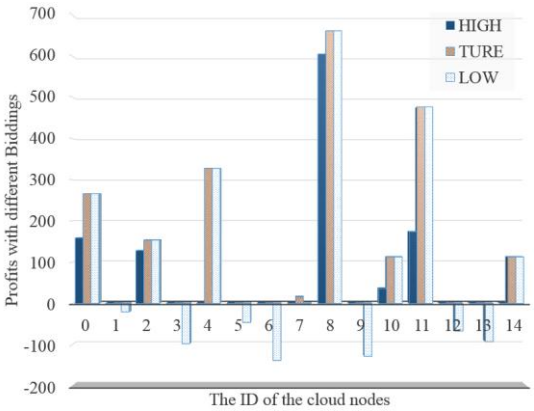

Fig. 2. The corresponding profit of all nodes under different quotations

To nodes #1, #3, #5, #6, #9, #12, #13, when they choose truth-revealing, they cannot get the tasks, the profit of which is 0. After increasing their quotation, they definitely cannot get the tasks as well. However, if these nodes intentionally decrease the price to deceive users, they can probably get the storage tasks, but the profit can be negative since its own storage cost is too high.

To sum up, all nodes, under the guidance algorithm of dominant equilibrium situation, would give up the false quotation and choose to reveal the real cost

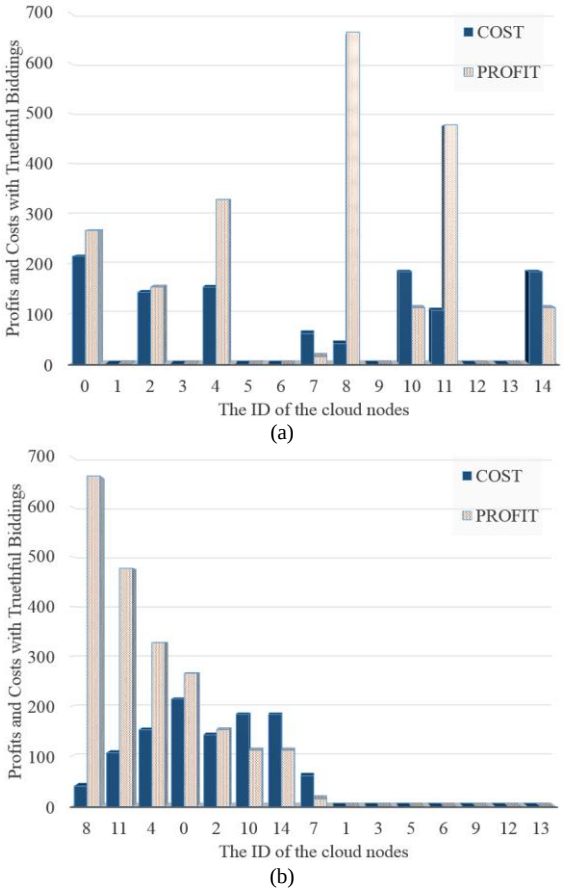

Fig. 3. The corresponding cost and profit of all nodes when truth-revealing: (a) Ordered by node ID, (b) Ordered by profits

From Fig. 2 we can conclude that all nodes would achieve their optimal profit while choosing truth-revealing and the profit would not increase when they reveal the false cost. Thus to conservative storage nodes, truth-revealing is the dominant strategy. The experimental result conforms to the theoretic proof displayed in section IV.

The corresponding profits of all nodes are calculated when truth-revealing, the results as shown in Fig. 3. The payment of the storage allocation equals to the summation of the corresponding cost and profit. As is shown in Fig. 3 (b), the node with less cost would achieve a higher payment from mechanism since its quality is better. These nodes would have lower cost while storing the data, thus the profit is higher. On the contrary, the nodes with high cost would achieve lower payment from mechanism, they get a relatively lower profit which can even be 0. Furthermore, for all nodes participating in the bidding, when they choose truth-revealing, the profit is non- negative, which proves that the mechanism is ex post individual rationality to storage nodes. The experimental conclusion conforms to the theoretic proof in section IV. In addition, it illustrates that there are enough incentives to all storage nodes, and nodes voluntarily participate this mechanism under the driven of non-negative interests.

VI. Conclusions and Future Works

In this paper, a mechanism design approach is proposed for solving the storage allocation problems in cloud computing platforms. The proposed method explicitly considers individual VMs' rationality so as to thwart selfish behavior and enforce truth-revealing for selfish VMs. First, we model the cloud storage system according to its massive and heterogeneous nature. Then, a VCG mechanism based storage allocation strategy is proposed to solve the storage allocation problem on the basis of storage cost and the valuation of user data, as well as considering various aspects of performance of nodes. The proposed approach can effectively avoid the selfish behavior of cloud node and force the cloud node to truthfully bid on tasks, storage allocation is formed based on the truthful bidding according to each node's performances to maximize the profit of the cloud storage system. We provide extensive theoretical analysis and experimental studies to verify the correctness and efficiency of the proposed strategy.

The mechanism cannot activate the data storage nodes to provide services because of BU defect within the VCG mechanism when dealing with multi-user to single-service (M: 1) problem, which results in its unavailability in practice. In our future work, we will investigated this particular problem and proposed solutions.

Acknowledgment

This work was supported by the National Science Foundation of China (Grant No. 61272432, 61202508, 61370092, 61370132 and 61472033), Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410), and

Fundamental Research Funds for the Central Universities (FRF-TP-14-045A2).

REFERENCES

[1] L. M. Vaquero and L. Rodero-Merino, J. Caceres, *et al*., "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50-55, 2008.

[2] I. Foster, Y. Zhao, I. Raicu, *et al*., "Cloud computing and grid computing 360-degree compared," *Grid Computing Environments Workshop*, pp. 1-10, 2008.

[3] S. K. Garg, C. S. Yeo, A. Anandasivam, *et al*., "Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers," *Journal of Parallel and Distributed Computin*g, vol. 71, no. 6, pp. 732-749, 2011.

[4] H. X. Mao, "Research of cloud storage and improved distributed strategy of data storage," *Applied Mechanics and Materials*, vol. 49, no. 6, pp. 1812-1816, 2014.

[5] N. Jain, I. Menache, J. S. Naor, *et al*., "A truthful mechanism for value-based scheduling in cloud computing," in *Algorithmic Game Theory*, Springer Berlin Heidelberg, 2011, pp. 178-189.

[6] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107-1117, 2013

[7] G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani, "A hybrid reinforcement learning approach to autonomic resource allocation," in *Proc. IEEE International Conference on Autonomic Computing*, Dublin, Ireland, June 2006, pp. 65-73.

[8] K. Boloor, R. Chirkova, Y. Viniotis, and T. Salo, "Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud," in *Proc. 2nd IEEE International Conference on Cloud Computing Technology and Science*, Indianapolis, Ind, USA, November 2010, pp. 464-472.

[9] W. H. Lin, Z. M. Lei, J. Liu, *et al*., "An efficient algorithm for dynamic storage allocation in cloud computing environment," *International Journal of Applied Mathematics and Statistics*, vol. 48, no. 18, pp. 254-262, 2013.

[10] A. Danilo, P. Barbara, and P. Mauro, "A game theoretic formulation of the service provisioning problem in cloud systems," in *Proc. 20th International Conference on World Wide Web*, 2011, pp. 177-186.

[11] D. Niyato, A. V. Vasilakos, and Z. Kun, "Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach," in *Proc. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2011, pp. 215-224.

[12] V. Krishna, *Auction Theory*, Academic press, 2009, pp. 145-167.

[13] Y. Vorobeychik, D. M. Reeves, and M. P. Wellman, "Constrained automated mechanism design for infinite games of incomplete information," *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 2, pp. 313-351, September 2012.

[14] T. Groves and M. Loeb, "Incentives and public inputs," *Journal of Public Economics*, vol. 4, no. 3, pp. 211-226, 1975.

[15] D. Grosu and A. T. Chronopoulos, "A truthful mechanism for fair load balancing in distributed systems," in *Proc. Second IEEE International Symposium on Network Computing and Applications*, 2003, pp. 289-296.

[16] A. Dimakis, R. Jain, and J. Walrand, "Mechanisms for efficient allocation in divisible capacity networks," in *Proc. 45th IEEE Conf. on Decision and Control*, Piscataway, NJ: IEEE Press, 2006, pp. 1264-1269.

[17] R. K. Dash, N. R. Jennings, and D. C. Parkes, "Computational-mechanism design: A call to arms," *Intelligent Systems*, IEEE, vol. 18, no. 6. pp. 40-47, 2003

[18] M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2009, pp. 56-78.

[19] J. Green and J. J. Laffont, "Characterization of satisfactory mechanisms for the revelation of preferences for public goods," *Econometrica: Journal of the Econometric Society*, pp. 427-438, 1977.

Zhenqiang Mi received the B.S. degree in Automation and the Ph.D. degree in Communication and Information Systems from University of Science and Tech-nology Beijing, China, in 2006 and 2011, respectively. Currently, he is an assistant professor in School of Computer and Communication Engineering, University of Science and Technology Beijing. His technical interests include cloud computing, mobile networks and game theory. He has published more than 20 technical papers in refereed conference proceedings and journals.

Miao Zhang received the B.S. degree in Communication Engineering from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, in 2013. Currently, she is Ph.D. candidate in School of Computer and Communication Engineering, University of Science and Technology Beijing. Her technical interests include cloud computing and service composition.

Zenggang Xiong received the the MA degree from Hubei University, China, in 2005, and the PhD degree in computer science from Beijing University of Science and Technology, China, in 2009. He is now a professor in Hubei Engineering University. His research interests are in the areas of peer-to-peer computing, Cloud computing, distributed systems and big data.