Master’s Thesis

Design and Implementation of an Incentive System for Sharing Distributed Computing Resources

Dong Ho Son (손 동 호)
Department of Computer Science and Engineering
Pohang University of Science and Technology
2018
Design and Implementation of an Incentive System for Sharing Distributed Computing Resources
Design and Implementation of an Incentive System for Sharing Distributed Computing Resources

by

Dong Ho Son
Department of Computer Science and Engineering
Pohang University of Science and Technology

A thesis submitted to the faculty of the Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Computer Science and Engineering

Pohang, Korea
12. 14. 2017
Approved by
James Won-Ki Hong (Signature)
Academic advisor
Design and Implementation of an Incentive System for Sharing Distributed Computing Resources

Dong Ho Son

The undersigned have examined this thesis and hereby certify that it is worthy of acceptance for a master’s degree from POSTECH

12. 14. 2017

Committee Chair    James Won-Ki Hong    (Seal)
Member              Young-Joo Suh       (Seal)
Member              Hanjun Kim         (Seal)
ABSTRACT

Most people use computing devices such as smartphones, tablets, and PCs during the day and sleep at night. In other words, they are using computing devices with tremendous computing power for several hours a day. When the devices are not in use, they wait idle. Then people can lend their computing devices to other people when they are not using the devices. By lending computing devices and receiving a small charge, the computing device owners can make money at a time when the devices are not in use. Those who need computing devices can borrow and use them at low cost. These transactions require a rational incentive model that has a direct impact on costs and revenues. Therefore, this thesis designs, implements, and evaluates an incentive system that can support distributed computing resources sharing.
Contents

I. Introduction 1

II. Related Work 4
   2.1 File Sharing 4
   2.2 Storage Sharing 5
   2.3 Computing Sharing 5

III. Design 8
   3.1 Overview 8
   3.2 High Level Design 10
   3.3 System Architecture 12
   3.4 Sequence Diagram and Algorithms 17
   3.5 ISAC Coin Market 20
   3.6 Incentive Model 24
   3.7 Dealing with risk 30

IV. Implementation 32
   4.1 ISAC Network 32
   4.2 ISAC Agent 34
   4.3 ISAC Payment 35

V. Evaluation 36
VI. Conclusion

Summary (in Korean)

References
List of Figures

2.1 How BitTorrent works ............................................. 4

3.1 An overview of the ISAC Platform ............................... 9

3.2 High-level design of the ISAC platform .......................... 11

3.3 System architecture of ISAC Platform ........................... 15

3.4 Sequence diagram of how ISAC platform works ................. 17

3.5 ISAC coin’s markets ............................................. 21

3.6 The pricing strategies of cloud computing service providers .. 25

3.7 A pricing strategy of ISAC platform ............................. 27

3.8 Device usage over a day .......................................... 29

4.1 Graphic user interfaces of ISAC Network ........................ 33

4.2 Docker Hub where tasks are containerized and stored .......... 34

4.3 A process of passing isac coin to other person ................. 35

5.1 Average CPU utilization over time .............................. 36

5.2 Average disk usage over time .................................... 37

5.3 Average memory utilization over time ........................... 38

5.4 Average network throughput over time ........................... 38

5.5 Comparing ISAC with usage-based pricing only and ISAC with time-dependent pricing ............................. 39
5.6 Hourly cost comparison of computing resources . . . . . . . . . . . . 40
5.7 Total cost comparison of computing resources . . . . . . . . . . . . 41
I. Introduction

Today, we are working with many IoT devices [1]. We are enjoying countless funny and interesting content every moment, and developers are developing convenient services at this moment and providing them to us. Developers prefer to run on-premises servers in order to deploy and maintain their own services. However, it is true that it is costly to operate a small scare rather than a large-scale project, and it puts a lot of financial pressures on individual developers if they operate a local server. So they rent resources from cloud service providers such as Amazon Web Service [2], Microsoft Azure [3], and Google Cloud Platform [4]. Among the products set by the cloud service providers, individual developers choose the products that are most similar to the conditions they want, and pay their monthly fee to use them. It is easy to use and looks good because it costs less than operating a local server, but it also has a problem. If you use a lot of borrowed computing resources, the cost is never cheap. In fact, according to researches by Cisco, IoT devices, which were about 14.4 billion in 2014, are expected to surge to about 50 billion by 2020 [5]. The US market research firm IDC also expects the global IoT market to grow from $1.9 trillion in 2013 to $7.1 trillion in 2020 [6]. These increasing devices generate more data. Processing the increased data requires the increase in resource utilization, resulting in a cost burden. Therefore, we decided that we need a service that makes it available for us to use computing resources more cheaply. Also, there are not many operating
systems in the computing resources provided by cloud service providers. It means that if you need to work on an Android or iOS environment, you may not be able to use the service. So, we thought that the client should be able to choose from a variety of OSs and came up with the idea of using idle computing resources around us to make our services affordable. The utilization of PCs and mobile devices are very low at night and at down when people usually go to bed. In other words, these devices are idle while people are sleeping. When you do not use your devices, if you do not make them idle, lend resources to others, and get a prescribed usage fee from them, the lender will earn money in exchange for them. The borrower can use them at a much lower cost rather than borrowing expensive computing resources from cloud service providers. Therefore, it is a strategy that both lenders and borrowers can be mutually beneficial. So, we propose ISAC(Idle Storage And Computing) Platform, starting from these ideas. The ISAC Platform is a platform that lends distributed computing resources that each user has to those who need them. The ISAC platform can be expressed by the following five keywords.

**Distributed:** The ISAC platform is targeted at distributed computing resources. This is a contradictory concept to existing cloud computing service providers borrowing centralized computing resources.

**Resources Rental:** Clients can borrow distributed computing resources. Conversely, providers can borrow the computing resources of their idle devices.

**Flexibility:** Clients are free to borrow various types of computing resources. Providers can freely set up computing resources rentals when they want. Usage
fees for computing resources are not always the same and vary greatly depending on the situation

**Usage Monitoring:** Clients can periodically monitor their borrowing computing resources and understand how much they are currently using. Providers can monitor computing resources they are borrowing in order to check if there are any problems with the devices.

**Payment:** Clients who rent computing resources pay for them. At this time, it is necessary to pay the required amount by using ISAC coin, the currency of the ISAC platform.

Based on the above five keywords, we have created a simple prototype ISAC Platform. We have done some experiments with this prototype, which will show the possibility to rent computing resources at a much lower price than using the computing resources of Cloud service providers.

The remainder of this paper is organized as follows. Chapter II introduces background knowledge and related research. Chapter III goes into details about the design of ISAC, the distributed computing resource sharing platform that we would like to propose. Chapter IV describes the prototype implementation of the ISAC platform. Chapter V describes the experimental results based on the implementation. Finally, chapter VI concludes this paper with future work.
II. Related Work

We have known the service of sharing files or storage for a long time, not the concept of sharing computing resources. This chapter explains concepts of file sharing, storage sharing and computing sharing, and describes typical services.

2.1 File Sharing

File sharing [7] refers to the act of providing access to digitally stored information such as computer programs, multimedia, documents, and e-books, and of distributing them. This is done in various storage, delivery, and distribution models. Peer-to-peer type file sharing is the most representative type. BitTorrent [8] [9] is the most popular protocol and program of P2p sharing(Fig. 2.1).

Figure 2.1: How BitTorrent works
BitTorrent works as follows. A file is divided into several pieces, which many people share. First of all, the user executes the torrent and tries to access the tracker specified in the torrent file. The tracker server delivers a list of peers, which have pieces of that torrent file, to users. The users request file sharing to peers who have each file fragments. Each of peers sends the user the file fragments they have, and the user eventually gets one complete file.

2.2 Storage Sharing

Storage sharing refers to the act of receiving and using a certain amount of storage. There is cloud storage \[ \text{[10]} \] in the form of easy-to-reach storage sharing. Cloud storage is a technology that makes it available for me to store my information in a storage space located somewhere on the Internet and retrieve it through various devices when needed. There is Dropbox \[ \text{[11] [12]} \] as a typical cloud storage service.

Dropbox is the most well-known cloud storage service. How Dropbox behaves is as follows. When a user pays a certain amount, he/she will receive the cloud storage for the capacity corresponding to the amount. Users can store their files in this cloud storage, and they can access files freely regardless of devices.

2.3 Computing Sharing

Computing sharing refers to the technology that is used after receiving computing resources required for computation from other users. There are centralized and distributed computing \[ \text{[13]} \], depending on who provides the computing re-
Centralized Computing Sharing: It means borrowing computing resources from only one provider. Representative services include Amazon Web Service, Microsoft Azure, and cloud computing services on Google Cloud Platform [14]. They have a large amount of computing resources and receive usage charges from users while lending their computing resources to them little by little.

Distributed Computing Sharing [15]: This means borrowing computing resources from multiple peers. It is not a form received from the center. Thus, all of the scattered peers become providers. Representative examples of distributed computing include grid computing [16] and fog computing [17].

Grid computing is the type of a parallel distributed system that is used to improve computing power by sharing the processing power of CPU and memory, the storage space of hard disk, etc. each other. The grid computing is a kind of distributed computing conceptually, but there are differences in the way it is viewed. Distributed computing focuses on the utilization of multiple clients connected in parallel by the server like multiple independent computing cores, and the client only provides its own computation and processing power. However, grid computing is focused on reorganizing them into virtual computers with unrivaled abilities, and computers that are connected in parallel can provide computing and processing capabilities to each other. An analogy is that the grid computing is M&A, and the distributed computing can be seen as a subcontractor or outsourcing.

Fog computing is a concept that distributes networks in a location physically
close to the users who exchange data. It is about not relying solely on the cloud for data processing but placing the applications close to where the data is generated so that more data can be used. Unlike the cloud computing approach, the computing responds in real time to the network endpoints and autonomously processes them. The advantage is that the end devices have independence and differentiation compared to the cloud method which monopolizes and controls all information and functions in the center.

Many studies such as file sharing, storage sharing, and computing sharing have been actively conducted, and there are many commercially available services. Particularly in computing sharing, fog computing has a high possibility of complementing the shortcomings of cloud computing. In this paper, we propose a platform which you can borrow resources based on distributed computing as above. In particular, this platform has the advantage of being able to distribute high load in the center, which is one of the disadvantages of cloud computing, and to use it relatively cheaply. In addition, it has high security as the payment method is based on a block chain. There is the possibilities of a great future to open up another market because it uses a cryptocurrency as a means of payment.
III. Design

In this chapter, we would like to describe the design of the ISAC platform. And we would like to explain briefly the concept of ISAC platform and how it works through the overview of ISAC platform and the high-level design. After that, we will describe the detailed design such as the system architecture and the sequence diagram. Finally, we will also explain the payment part of the ISAC platform. And we will depict the incentive model for computing resources and the market for ISAC coin in the payment part.

3.1 Overview

The heart of the ISAC platform is that the clients are to borrow and pay for the idle resources of the providers. The providers can benefit by renting their computing resources at times when they do not use their devices. The clients can take advantage of computing resources at an affordable cheap price instead of the cloud computing offered by cloud service providers at a high price(Fig. 3.1).
Figure 3.1: An overview of the ISAC Platform
3.2 High Level Design

Transactions in the ISAC System fall into three categories largely: Clients-Providers Matching, Task Push and Run, and Payment for Usage. (Fig. 3.2).

3.2.1 Clients-Providers Matching

Providers can set as many as they want their computing resources available to lend to clients. Providers can register the computing resource information to be borrowed on the ISAC platform. And clients that want to require computing resources ask the ISAC platform for computing resources of the desired conditions. At this time, the ISAC platform searches for clients and providers whose conditions match. And then, the ISAC platform matches the clients and providers by passing information about the providers with the computing resources of the conditions that clients are looking for to the clients.

3.2.2 Task Push and Run

When the matching between clients and providers is established after searching, the tasks of the clients are delivered to the providers. Providers use their computing resources to perform tasks on clients. The results of the completed tasks are delivered to clients.

3.2.3 Payment for Usage

Clients should pay ISAC coin for the providers as much as they use the computing resources of the providers. Details of ISAC coin will be covered in 3.3.3 ISAC Payment and ISAC coin’s market.
Figure 3.2: High-level design of the ISAC platform
3.3 System Architecture

The architecture of the ISAC System is shown in the figure below. First, ISAC System is divided into the Networking part, the Agent part, and the Payment part (Fig. 3.3).

3.3.1 ISAC Network

ISAC Network is to manage the information of clients and providers and supervise mutual matching with them. The main components of ISAC Network are the Peer DB and the Matching manager.

A. Peer DB

Peer DB is a database containing information of providers and clients. In case of providers DB, providers will store detailed information about their device information, the time band available for borrowing computing resources, and the desired revenue unit price, etc. In case of clients DB, clients store detailed information such as the type of their tasks, the type of computing resource they need, and the unit cost.

B. Matching Manager

It is the role of the matching manager to match a client that need computing resources and a provider to lend computing resources. In other words, the matching manager compares and analyzes with the cost price set by clients and the revenue unit price set by providers, and it serves to match clients and providers with matching unit prices of each other. The tug of war between the
clients who want to borrow the providers’ computing resources by minimizing the costs and the providers who want to earn more revenue soon form the market that deals with the unit price of the computing resource. Clients who want to borrow computing resources at too low cost are less likely to match a provider than other clients. Similarly, providers who want to borrow computing resources at too high revenue rate are less likely to be matched a client relative to other providers. After all, in this market, clients and providers will look for a match on the auction-based and proceed with the transaction.

3.3.2 ISAC Agent

The agent is the application nearest to peers, which means the service that peers have the most direct influence on using the ISAC Platform. The agent is divided into a client agent and a provider agent, and each consists of main components such as the Task manager, the Resource monitor, and the Contract manager.

A. Task Manager

The Task Manager basically manages and runs tasks. The client agent’s task manager delivers the tasks of the clients to the providers as a container form. And then, it plays a role to help the clients to verify the tasks after getting back the completed tasks from the providers. The Task Manager of the provider agent is responsible for executing tasks received from the client. The tasks that have been completed are sent to the client.
B. Resource Monitor

The core of the Resource Monitor is to monitor the resource and task containers being used. The Resource Manager of the provider agent monitors the computing resources (CPU, memory, storage, etc.) used in providers’ own devices respectively and reports them to the client agent. The client agent’s Resource Monitor provides a comprehensive status view of the computing resources of the providers. In addition, it can be monitored and managed containers because all tasks are moved, deployed, and run in the containerized form.

C. Contract Manager

The Contract Manager helps a client pay the fee to the providers which given their resource. The Contract Manager inquires the pricing manager of the ISAC payment about the consolidated price of the computing resource. The pricing manager initially determines the aggregate price of the computing resources used to borrow based on the unit price policy agreed upon between the client and the providers. The final price is informed to the client agent and the client agent’s contract manager sends ISAC coin to ISAC coin purse of providers. The provider agent’s Contract Manager verifies revenue and notifies providers. By paying for the transaction, the transaction is terminated and the transaction result is transferred to the peer DB of the ISAC network to update the account of the clients and the providers.
Figure 3.3: System architecture of ISAC Platform
### 3.3.3 ISAC Payment

The payment is responsible for setting and managing the providers’ computing resource prices. It also plays a role to balance the unit cost of a computing resource with the unit cost of revenue. Major components are the Pricing Manager and the Unit Price Market.

#### A. Pricing Manager

The Pricing Manager determines the total price of the computing resources that the client borrows from the providers. It calculates the final price based on the unit prices by items agreed upon in the matching phase of clients-providers. The calculated result is sent to the Contract Manager of the client agent.

#### B. Unit Price Market

The Unit Price Market is a market that balances the price of computing resource items. Like that how much CPU utilization per second, how much memory is used per second, how much storage is used, etc., the criteria information affects computing resource costs. Clients will set these prices as cheap as possible, and providers will set them as high as possible. Parties whose interests fall into each other agree on a unit price for computing resource leasing and begin trading on this.

The currency used in the ISAC Platform is called ISAC coin. ISAC coin is a token based on Ethereum and is the means by which clients pay providers. The price and value of ISAC coin are determined by the demand and supply of ISAC coin in the market. We will cover the ISAC coin market later.
3.4 Sequence Diagram and Algorithms

The high level design of the previous section is described by the sequence diagram and the transaction algorithm of the client and the provider (Fig. 3.4).

Figure 3.4: Sequence diagram of how ISAC platform works

Providers register their computing resources on the ISAC Platform. Clients request the required computing resources from the ISAC platform. ISAC Network of ISAC platform matches clients-providers with fitting their conditions through clients-providers matching process and informs clients of the information of providers. And then clients send their task image URL to providers. The
providers download the image based on the delivered URL and execute the task. After task execution ends, providers return the results to clients. Clients will pay for the final cost received from ISAC Payment.

Algorithm 1 Provider transaction algorithm

1: procedure Provider’s transaction
2:     info ← Information of computing resource
3:     top:
4:     ISAC.Register(info)
5:     loop:
6:     if ISAC.match() = True then
7:         u ← getTaskUrl()
8:         t ← pullTaskImage(u)
9:         result ← Run(t)
10:        sendResult(result)
11:        evaluate(client)
12:        return 0
13:     goto loop
Algorithm 2 Client transaction algorithm

1: **procedure** Client’s transaction

2: \( info \leftarrow \text{Information of computing resource} \)

3: \( p \leftarrow \text{null} \)

4: \( u \leftarrow \text{Task Image URL} \)

5: \( t \leftarrow \text{Task container} \)

6: **top:**

7: \( ISAC.Request(info) \)

8: **loop 1:**

9: if \( ISAC.match() = \text{True} \) then

10: \( p \leftarrow ISAC.getInfo() \)

11: \( \text{sendUrl}(p,u) \)

12: \( \text{pushTask}(t) \)

13: **loop 2:**

14: if \( \text{getResult()} = \text{True} \) then

15: \( ISAC.\text{pay()} \)

16: \( \text{evaluate}(p) \)

17: **return** 0

18: **goto** loop 2

19: **goto** loop
3.5 ISAC Coin Market

ISAC coin has two markets: The Computing Resource Market and the Cryptocurrency Market (Fig. 3.5). These two markets share the common feature that ISAC coin is financing as the central currency, but the purpose differs.

3.5.1 Computing Resource Market

Basically, the purpose of the ISAC platform is to share idle computing resources among others. In this process, the person who rents the computing resource pays a certain amount of money to the person who provided the resource. This is a kind of contract between the borrower and the supplier of computing resources. All contracts within the ISAC platform are made through ISAC Payment. And the Computing Resource Market is a place to be gathered the contracts after making a contract. The Computing Resource Market and the Unit Price Market differ in their categories. The unit price market is a market formed by matching clients and providers. It is a market that exists to balance the process of minimizing the unit cost of clients and maximizing the revenue cost of providers. However, the Computing Resource Market not only includes the unit price market, but also balances the final price of rental costs with the final price of rental revenues and grades clients and providers.

The Computing Resource Market consists of a number of contracts between clients and providers.
Figure 3.5: ISAC coin’s markets
Providers

Providers are people who provide computing resources. Providers can set the rental price to match their computing resource level. If a provider sets the unit price too high, clients trying to borrow a provider’s resource are fewer. This can hinder the maximization of providers’ revenues.

Clients

Clients are people who need computing resources. Clients can set the unit price that they are willing to pay for borrowing computing resources. If a client set the unit price of computing resource to borrow too low, the number of providers to be lending their resources will be small. This can be an obstacle to clients’ matching with providers of various conditions.

3.5.2 Cryptocurrency Market

The price of ISAC coin used in the ISAC Platform is determined by the balance of supply and demand for ISAC coin in the Cryptocurrency Market.

Demand

Investors and clients are playing a role in buying ISAC coins in the Cryptocurrency Market. The price of ISAC coin in the Cryptocurrency Market will rise because of the buying of ISAC coin by these investors and clients.

1) Investors: ISAC coin is used as a sort of currency using for trading inside the ISAC platform, but is considered an investment target outside the system. Investors should make sure that the usefulness, marketability, and prospects of
the ISAC system meet their own standards, and buy ISAC coin if they consider it worth investing in.

2) Clients: Investors’ ISAC coin buyout is not the only buyer. Clients wishing to use real ISAC coin can also buy ISAC coin. They are purchasers of ISAC coin for use in the computing resource market within the ISAC system as clients, rather than investor’s position.

Supply

Investors and providers are playing a role in selling ISAC coins in the Cryptocurrency Market. Because of their sale, ISAC coins are released to the market and the price of ISAC coin is stabilized.

1) Investors: Investors who have bought ISAC coin at a low price the other day will sell it to obtain the price margin if the price of ISAC coin goes up. Alternatively, the investors may be sold even if the ISAC platform would be worthless due to failure, technical, commercial, or regulatory reasons.

2) Providers: Providers that provide idle computing resources to clients get a certain amount of ISAC coin from borrowed clients. Providers can sell acquired ISAC coins to the Cryptocurrency Market. The ISAC coin of the virtual currency can be redeemed by converting it into cash such as USD, KRW, or EUR.
3.6 Incentive Model

In order to differentiate from existing cloud computing service providers, the ISAC platform offers a rate policy based on a flexible and low-cost incentive model. It is a system that charges only as used basically. Furthermore, providers and clients can make the rate for rental to be cheaper or more expensive through auction as long as they do not exceed the rate guideline. Lastly, a discount is added to the time zone. Ultimately, the cost of using the ISAC platform is much cheaper than traditional cloud computing services.

3.6.1 Usage-Based Pricing

Cloud computing service providers are leasing computing resources in their own data centers to users in an area-specific, context-sensitive manner. In case of the location of the cloud, it allows you to use the cloud in the US, Asia, and Europe. They are renting and pricing with different numbers of CPUs, memory sizes, disk capacities and so on to match the computing resources that users want. The following table summarizes the cloud computing fee systems for Amazon [2], Google [4], and Microsoft [3] among providers of cloud computing services(Fig. 3.6).
Figure 3.6: The pricing strategies of cloud computing service providers

<table>
<thead>
<tr>
<th>Amazon Web Service</th>
<th>Google Cloud Platform</th>
<th>Microsoft Azure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPUs</strong></td>
<td><strong>Memory</strong></td>
<td><strong>Price(USD)/hour</strong></td>
</tr>
<tr>
<td>1</td>
<td>2GB</td>
<td>$0.0230</td>
</tr>
<tr>
<td>2</td>
<td>4GB</td>
<td>$0.0464</td>
</tr>
<tr>
<td>4</td>
<td>16GB</td>
<td>$0.1856</td>
</tr>
<tr>
<td>8</td>
<td>32GB</td>
<td>$0.3712</td>
</tr>
</tbody>
</table>

| **CPUs** | **Memory** | **Price(USD)/hour** | **Network** | **Monthly Usage** | **Price(USD)/GB** | **Disk** | **Price(USD)/GB** |
| 1     | 3.75GB   | $0.0535             | Ingress     | Always           | $0.000             | Standard Storage | $0.0400 |
| 2     | 7.5GB    | $0.1070             | Egress      | 0-1TB            | $0.120             | SSD Storage     | $0.1870 |
| 4     | 15GB     | $0.2140             | Egress      | 1-10TB           | $0.110             |                 |         |
| 8     | 30GB     | $0.4280             | Egress      | 10+TB            | $0.080             |                 |         |

| **CPUs** | **Memory** | **Price(USD)/hour** | **Network** | **Monthly Usage** | **Price(USD)/GB** | **Disk** | **Price(USD)/GB** |
| 1     | 2GB      | $0.0650             | Ingress     | Always           | $0.000             | Standard Storage | $0.0500 |
| 2     | 4GB      | $0.1360             | Egress      | 0-5GB            | $0.000             | SSD Storage     | $0.1540 |
| 4     | 8GB      | $0.2860             | Egress      | 5GB-10TB         | $0.087             |                 |         |
| 8     | 16GB     | $0.6000             | Egress      | 10-40TB          | $0.083             |                 |         |
The above picture is relatively cheap as a charge system when using computing for a general purpose, but it has a disadvantage that it is used only in a predetermined condition. Let’s take a brief example as a user using the Amazon Web Service. If a user needs a dual-core CPU and 8GB of memory, the user has no choice but to select a plan to be provided a quad-core CPU and 16GB of memory. This is because there is no service that meets the user’s requirements. You have to pay even four times more expensive fee. Let’s take another example. Let suppose there are two users such as A and B. User A and B are running their own tasks through both borrowing dual-core CPUs and 4GB of memory. However, user A is a very simple task, so CPU utilization is not high. User B, on the other hand, is running very heavy tasks and CPU utilization is very high. Both users are basically paying the same cost of $0.0464 per hour. This can be considered as an inefficient fee system for user A. Therefore, the ISAC platform adopted a policy that charges each usage through measuring the exact usage of each user (Fig. 3.7).

Rates for each resource are only guidelines in a large category and it can vary any kind of adjustments from -30% to +30%, depending on the needs of the clients and providers. The exact rates are set finally by the clients after the auction is done in the clients-providers matching process.
3.6.2 Auction-Based Pricing

The incentive model of the ISAC platform is flexible and aims to set the charges by dealing with the situation of clients and providers. Therefore, the price of computing resources is not fixed, but can vary depending on the agreement between clients and providers. The matching manager matches clients and providers after looking for a fitted condition comparing cost unit price and profit unit price in order to match clients and providers. Auction-based pricing is conducted under the supervision of the matching manager. If clients want to borrow computing resources as much as possible at the lowest cost, but to be able to match trusted providers named in the whitelist, clients have to bear the burden of increasing costs by causing it, regardless that the conditions the trusted providers want that are not popular and the more difficult to find. Providers want to execute more tasks in their resource to maximize revenue. In order to
execute more tasks, matching with clients must be given priority and more often than others, and in order to do so, both parties must bear the burden of lowering the unit cost in the matching process. In this way, clients and providers will auction each cost price and profit unit price for each other’s interests. As a result of the auction, computing resource charges are settled on a new, agreed-upon rate system rather than a fixed rate. This results in satisfying both clients and providers.

3.6.3 Time-Dependent Pricing

People do not use IT devices such as PCs and smart phones all day long. There are times when usage is high, while there are times when usage is low. Simply, for example, PCs and smartphones are heavily used during daylight hours when people are active. However, relatively IT devices are less used in late at night or at dawn most people are sleeping. Using IT devices and treating many tasks of clients can sometimes cause problems with providers’ devices. On the other hand, it is relatively safe and efficient for clients’ tasks to be executed when providers are not using IT devices. Therefore, the ISAC Platform divides the time of day when IT devices usage increases during the day and the time when it does not. According to a survey conducted by the Financial Times [18], it is reported that the usage of smartphones and tablets is likely to increase mainly during the office-going hour (usually 6-9 o’clock)(Fig. 3.8). The usage of PCs is increasing during the day from 9:00 to 18:00, and the usage of smartphones and tablets increases from the quitting time of office to midnight. On the other hand, it was found that the usage rate for all IT devices decreased sharply from
midnight to 6 am.

Figure 3.8: Device usage over a day

The ISAC Platform has set up a golden time zone to suit people’s living patterns. Providers should set the time zone in which their device is idle most freely, and this time zone is called the golden time zone. In the golden time zone, clients’ task execution costs are reduced by 20%. Therefore, many clients will want their tasks to run in a cheaper golden time zone, and this can be expected to result in an effect that does not affect the providers’ daily lives, in parallel consistent with the idle state time of the IT devices of the providers.
3.7 Dealing with risk

In some cases, there may be tasks that overload the computing resources of providers in ISAC Network. This is beyond the original intent of the ISAC platform and is a very important issue that can affect the durability of the ISAC platform. Therefore, there is a problem of filtering tasks having a malicious purpose and screening out such users. The ISAC platform will address the above issues as follows: Task Verification, and Whitelist & Blacklist.

Task Verification

We agree that the software has some bugs, but it is necessary to make a difference at that level. So we think that a separate verification group should examine the clients’ tasks at the software level. And the people who participated in the task verification obtain a certain amount of compensation from the client as the task’s owner. It’s dependent on only clients’ choice whether their tasks are to be verified or not. However, because most providers want to be run only validated tasks on their computing resources, so doing task verification will help you to meet various providers when by meeting the requirement of the market.

Whitelist & Blacklist

This list means to rank users who participated in the ISAC Network. Clients may be logged if the whitelist pushes the tasks fitted for their intended purpose to the providers, or if they do not cause problems with the providers’ computing resources. Providers can also be recorded on the whitelist. Providers will provide computing resources at a set time according to their agreed upon agreement with
the clients and have a qualification to be recorded in the whitelist if they do not affect the tasks because there is no computing resource problem during the work period.

Blacklist records users who either do not use the ISAC System in the correct way or that adversely affect other peers. The following cases are included in this blacklist such as the below: Clients who have affected a negative impact on providers by executing tasks with malicious code on the providers’ computing resources, clients that cause problems with providers’ computing environment with unreasonable tasks execution, providers who do not keep their work period, and providers that provide other types of resources than agreed computing resources, and so on.

The peers specified in the blacklist are subject to some restrictions on the future participation in the ISAC Network. On the other hand, the peers specified in the whitelist get reliability on the contrary to this. For example, a client specified in a blacklist should be borne to charge a higher cost, and providers will be borne a lower profit penalty. In addition, in the clients-providers matching process, the peers specified in the blacklist are forced to pull out to a lower priority. On the other hand, the clients specified in the whitelist are able to borrow computing resources at a lower cost, and are also paired preferentially with respect to clients-providers matching. In case of the trustworthy providers, even if they set a high-profit rate, will preferentially match clients and so many clients will be preferred.
IV. Implementation

This chapter describes the implementation of a simple prototype of the ISAC platform. It explains the three most important features of the ISAC platform which are Network, Agent and Payment.

4.1 ISAC Network

It is the role of the ISAC Network to request the necessary computing resources that the client needs, to register the computing resources to be provided by the providers or to match the peers. We implemented the platform so that the client and the providers use web-based applications to use ISAC platform. We have implemented the system in which the clients register the desired computing resource information and send the information to the Peer DB of the ISAC Network when the cost is set (Fig. 4.1). We implemented the system in which the providers register the computing resources to lend, to set the revenue unit price, and send this information to the Peer DB of the ISAC Network when the profit is set. We implemented the system in which they can monitor who is connected after matching.
4.2 ISAC Agent

It is the role of the ISAC Agent to enable the client to register the task and to inform the providers of the task url to download. It also includes returning the provider’s task results to the client as part of its role. We used Docker container to register three different tasks. A Docker is an open source container software platform that packages applications in ”containers”, allowing them to run between systems running Linux. Containerized tasks are stored in a container registry called Docker Hub. The containers stored in the Docker Hub have their own image urls. Anyone who knows these urls can download and run the saved tasks. On the same principle, this url was passed to the providers, and the providers could download and run the client’s tasks using the Docker on their own devices.

![Figure 4.2: Docker Hub where tasks are containerized and stored](image)

Figure 4.2: Docker Hub where tasks are containerized and stored
4.3 ISAC Payment

The role of ISAC Payment is to set and inform the total charge. The rates are finally determined based on the utilization of CPU, memory, storage and network throughput. Rates units are shown in Fig. 3.7. Ethereum is an open-source, public, blockchain-based distributed computing platform featuring smart contract (scripting) functionality [19]. ISAC coin is replaced with ethereum in the prototype. Because ethereum is a platform, it is easy to generate tokens using it. In the prototype, ISAC coin is a kind of ethereum token. Implementing it as independent cryptocurrency will be included in our future work. Fig. 4.3 is a console screen that shows the creation of an account and the payment using ethereum.

```
Welcome to the Geth JavaScript console!
instance: Geth/jayBlockChain/v1.8.0-unittest-434d86e02/linux-amd64/go/1.7.4
coinbase: 0x7698fbc713f9595485e6133905bb705e40eef4ed
at block: 0 (Thu, 01 Jan 1970 00:00:00 KST)
datadir: /home/go-ethereum/build/bin/privatetcp
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> eth.accounts
[0x6455e3c947c501a5c6ac4c555d33907e3a3565af08647f
db3e3e3]
> personal.newAccount("client3")
0xa36455e3c947c501a5c6ac4c555d33907e3a3565af08647f
db3e3e3
> eth.accounts
[0x6455e3c947c501a5c6ac4c555d33907e3a3565af08647f
db3e3e3, 0x6f71db3da0c8c6e94d4233f99ac703ae8, 0x7e694c2cc351546a86355fa08647f
db3e3e3]
> eth.sendTransaction({from: '0x6f71db3da0c8c6e94d4233f99ac703ae8', to: '0xa36455e3c947c501a5c6ac4c555d33907e3a3565af08647f', value: web3.toWei(1, "ether")})
```

Figure 4.3: A process of passing isac coin to other person
V. Evaluation

We have based our experimentation into the most simplistic scenario. Let’s suppose that a client is willing to rent computing resources in order to perform to meet his needs. This specific task was performed on a Linux machine, for 24 hours and the total usage time was monitored. Fig. 5.1 is a graph that measured the average CPU usage by specific time period.

![Average CPU utilization over time](image)

Figure 5.1: Average CPU utilization over time

Fig. 5.2 makes reference to the average disk usage. For this one task, it utilized up to hundreds of KBs of disk space.
This, Fig. 5.3 makes reference to the average used memory sorted by time. For the present task, 40% 50% out of the 4GB of memory was used.

Fig. 5.4 shows the throughput of data processed through its network. Many computing cloud services do not charge for ingress packets; therefore, we have only measured the egress ones. With this data measured we proceeded to calculate the total price. We intended to compare the results from Amazon Web Service [2], Google Cloud Platform [4], and Microsoft Azure [3] when the same task was performed. And since ISAC provides a method that is more flexible in calculating prices, we compared the minimum and maximum rates when using the ISAC platform.
Figure 5.3: Average memory utilization over time

Figure 5.4: Average network throughput over time
Figure 5.5: Comparing ISAC with usage-based pricing only and ISAC with time-dependent pricing

We calculated the cost per an hour based on the measured CPU utilization, memory utilization, disk utilization, and network throughput. We then calculated the cost of applying time-dependent pricing. The night time zone (from 23:00 to 06:00) was discounted for 8 hours(Fig.5.5).
As we could see in Fig. 5.6, using the ISAC platform for computer resource renting shows the lowest cost. Since the ISAC platform is based on a variable rate policy, to see the changes in price, we divided the maximum rate with the minimum one. The lowest prices was offered by the Cloud computing service amazon providers. But, actual usage showed that it was the ISAC platform offering the lowest rates. The latter was even after the rates were compared with ISAC platform’s maximum rates. And during the golden time on ISAC there are even bigger discounts being offered, therefore it was outstandingly low.

Finally, we can conclude from Fig. 5.7 that the ISAC platform showed the lowest rates among all cloud service providers.
Figure 5.7: Total cost comparison of computing resources
VI. Conclusion

In this thesis, we suggested ISAC platform, which is a platform that shares distributed computing resources. ISAC Network is a function which put matching clients and providers ahead of everything else. Through this process we designed it so both clients and providers can decide cost and profit autonomously. We expect that it will play a great role to stabilize value of ISAC coin that is used internally. ISAC Agent plays a role to deliver tasks to providers and receive their results. We designed it to distribute tasks by applying Docker container technology. ISAC Payment plays a role to decide final price of computing resource which is used. Incentive models of ISAC platform are usage-based model, time-dependent model, and auction-based model. Pay-as-they-use can be a huge advantage. We designed it so people can rent computing resource for discounted price in late night time when other people are sleeping. So it has an advantage to decentralize burden of device. Ultimately, we calculated and compared expected price of device when it was run. Among providers of cloud server, AWS of Amazon provided their computing resource for the cheapest price. However, when we used ISAC platform, computing resource was provided even for cheaper price. Originally, people who rent their computing resource were not supposed to make a profit. However, profit was made when they started sharing their device resources. Since borrowers can borrow for cheaper price and all can be winners from this strategy, ISAC platform has a possibility of development in the future.
too.

As for the future work, we consider a road-map of more extended ISAC platform. First, we need to realize auctions for earning and cost price by analyzing them in detail. Then we need to study methods to manage containers more efficiently when many tasks are given. Finally, we need to increase market possibility of ISAC coin by developing it from Ethereum based code money to independent code money. Ultimately, the more detailed research and development for commercializing ISAC coin will be our next goal.
요 약 문

기존 클라우드 서비스 제공 업체가 제공하는 컴퓨팅 리소스에는 사용 요금 정책에 몇 가지 문제점이 있다. 우선, 기존에는 중앙집중형 컴퓨팅 시스템이었기 때문에 중앙에 많은 부담이 된다. 이것은 전체 시스템의 위험성을 높이는 결과를 초래할 수 있다. 또한, 동일한 서비스를 사용하더라도 리소스 사용률에 따라 요금이 청구되어야 하지만, 현재 클라우드 서비스 제공업체들은 동일한 서비스에는 동일한 요금체계를 적용하고 있다. 뿐만 아니라, 사용 시간이 증가하고 자원 사용이 늘어나므로 비용이 증가하게 된다. 그러나 ISAC는 기본적으로 분산 컴퓨팅 환경을 기반으로 한다. 따라서 부하 문제를 분산화시킬 수 있다. 또한, 컴퓨팅 리소스가 필요한 사람들에게 저렴하고 저렴한 컴퓨팅 리소스를 제공할 수 있다. ISAC는 기존 클라우드 서비스 제공 업체의 요금 정책보다 혼선 유연한 가격 정책을 가지고 있다. 따라서 사용자는 사용한만큼만 지불할 수 있으며 컴퓨팅 리소스를 혼선 저렴한 비용으로 임대 할 수 있습니다. 컴퓨팅 리소스를 빌려주는 사용자도 일정량의 이익을 얻을 수 있다. 마지막으로 ISAC은 암호화폐 ISAC coin을 이용하여 값을 지불하는 플랫폼이기 때문에 ISAC coin을 이용한 가상화폐의 잠재적 성장 가능성을 가지고 있다.
References


[9] Johan Pouwelse, Pawel Garbacki, Dick Epema, and Henk Sips. The bit-
torrent p2p file-sharing system: Measurements and analysis. In IPTPS,

[10] Jiyi Wu, Lingdi Ping, Xiaoping Ge, Ya Wang, and Jianqing Fu. Cloud stor-
age as the infrastructure of cloud computing. In Intelligent Computing and
Cognitive Informatics (ICICCI), 2010 International Conference on, pages


[12] Idilio Drago, Marco Mellia, Maurizio M Munafo, Anna Sperotto, Ramin
Sadre, and Aiko Pras. Inside dropbox: understanding personal cloud storage
services. In Proceedings of the 2012 ACM conference on Internet measure-

tional considerations and management options. ACM Computing Surveys

Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion
Stoica, et al. A view of cloud computing. Communications of the ACM,

[15] Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing
in practice: the condor experience. Concurrency and computation: practice


Acknowledgements

교수님, 감사드립니다.
Curriculum Vitae

Name : Dong Ho Son

Education

2010 – 2016 Department of Computer Science and Engineering, Hanyang University (B.S.)

2016 – 2018 Department of Computer Science and Engineering, Pohang University of Science and Technology (M.S.)

Affiliation

1. Distributed Processing & Network Management Lab., Department of Computer Science and Engineering, Pohang University of Science and Technology