# Cloud Resource Allocation using Double Auction

Vedsar Kushwaha, Ishwar Raut, Satyanath Bhat, Shweta Jain and Y. Narahari

## Abstract

From the last few years, cloud computing is becoming more and more popular. It looks that in near future, cloud service provider will compete with each other to sell the services and buyers will compete with each other to buy these services. If that happens then there is a need to define a mechanism which will decide the price that seller should charge and the price that buyer should pay. In this work we are defining a mechanism which not only defines the price charged by seller and price paid by buyers but we are also defining which buyer to get map with which seller. Our defined mechanism is incentive compatible and individually rational. We showed experimentally that it is also weekly budget balance. Our experimental results showed that using our mechanism both buyers and sellers will remain in profit if trade happened. We also showed the high mapping of buyers to sellers experimentally.

## 1 INTRODUCTION

Cloud computing is a large scale distributed computing paradigm. It is driven by economies of scale where a pool of abstracted, vitalized and dynamically scaled storage, software platforms and infrastructure services are delivered on customers demand over the Internet. Cloud service providers and customers, both are increasing every day with a very high rate. With respect to the growth and popularity in cloud computing, we envision that there comes a day when cloud service providers will compete with other cloud service providers to offer cloud computing services. Similarly thousands of users will also compete with each other to receive those services to run their complex tasks with guaranteed QoS and limited budgets. For instance, Amazon declared its price reduction in public cloud service(Amazon Web Service) on March 26,2014 effective from April 1,2014 just one day after google declared their cloud price reduction on March 25,2014 [7]. This is amazon's $42^{nd}$ price reduction from 2006 when it was launched.

Cloud service providers(CSP) publish their prices and charge from users accordingly. But, it looks that there is a need to define prices that the CSP will charge and the prices that the users should pay. Auction based mechanism, especially double sided auction will be an efficient and flexible solution for this. In this mechanism, user will decide what is the maximum price she wanted to pay. Similarly, CSP will decided at what price they will charge from users.

In this work, we are proposing a double sided auction based model. Besides seller and buyers, there is one more neutral agent, called Exchange. CSPs and Buyers put their price and resource requirement to Exchange. Then the exchange will decide which buyer will get mapped to which CSPs, at what price buyer pays and at what price seller should charge.

The rest of the paper is organized as follows. Section 2 highlights on the related work done on cloud resource allocation. Section 3 describes our proposed model where we define how buyers will interact with sellers. Section 4 describes our proposed mechanism. Section 5 describes data set taken and experiment done. Section 6 shows the result and section 7 explains conclusion and future work.

## 2 RELEVANT WORK

Managing and scheduling cloud economically is very important in cloud computing. This problem has also been researched in grid computing, and some economy resource allocation frameworks have been proposed [1]. These works can be regarded as an application and extension of some market economy theories in computing resources markets, the key point of which is to set prices of the commodities in the markets. Now there are less research works on cloud computing pricing than on Internet pricing. But some solutions to the latter can be applied to cloud computing, because the allocation of cloud computing resources is similar to that of network resources. Many network pricing policies are based on auction mechanism, which are proved efficient. In this section we will present related works on auctions.

In a real world market, there are various economic models for setting the price of services based on supply-and-demand and their value to users, including commodity market, posted prices, tender and auction models. Many research works have attempted to apply these economic models to grid computing. In the above models bidding and auction have high potential for computing resource allocation in gird or cloud environments. But there are few researches engaged to analyze CSPs bidding strategies on game theory approach, which is important problem in biding and auction. The auction model supports one-to-many or many-to-many negotiations, between a service provider (sellers) and many consumers (buyers), and reduces negotiations to a single value (i.e. price). Auctions can be conducted as open or closed depending on whether they allow back and forth offers and counter offers. The consumer may update the bid and the provider may update the offered sale price. Depending on these parameters, auctions can be classified into the following types: English Auction (first-price open cry), first-price sealed-bid auction, second-price sealed-bid auction (Vickrey auction), Dutch auction and double auction.

Continuous Double Auction(CDA) is one of the most common forms of market places and has emerged as the dominant financial institution for trading securities and financial instruments. In-deed, today the major exchanges, like the NASDAQ and the New York Stock Exchange (NYSE) and the major foreign exchange (FX), apply variants of the CDA institution[2]. Other significant applications are in market based control, where CDAs provide a dynamic and efficient approach to the decentralized allocation of scarce resources [3]. In the CDA model, buy orders (bids) and sell orders (offers or asks) may be submitted at anytime during the trading period. If at any time there are open asks and bids that match or are compatible in terms of prices and requirements (e.g., quantity of goods or shares), a trade is executed immediately. In this auction, orders are ranked from the highest to the lowest to generate demand and supply profiles. From the profiles, the maximum quantity exchanged can be determined by matching asks (starting with the lowest price and moving up) with demand bids (starting with the highest price and moving down). Re-searchers have developed software-based agent mechanisms to automate double auction for stock trading with or without human interaction [4].

If a CDA mechanism can be designed for cloud markets, it is also can be implemented on an e-auction platform. In cloud markets, the commodities can be purchased and de-

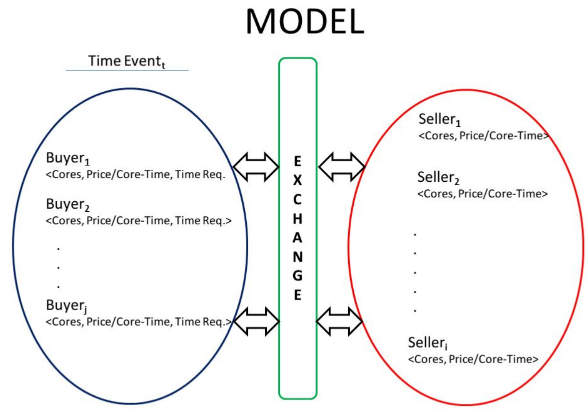


Figure 1: Model

livered over the Internet [5]. Cloud users often have a variety of application and valuation types [6]. Rather than letting resources sit idle, CSPs are also inclined to sell unused resources at a reduced price via using auctions to users. Because of the above characteristics of cloud, on-line e-auction is an effective method to implement computing resources allocation. Especially, the double auction mechanism is more flexible for it supports buyers and sellers to bid simultaneously. However, double auction mechanisms have not been applied in cloud markets up till the present moment.

# 3 MODEL

In this work, we are allocating Virtual Machines from sellers to buyers. We are considering a notion of discrete time, called time interval. At the start of each time interval, some buyers will come with Virtual Machine requirement with price they are willing to pay. These buyers will get mapped with the appropriate sellers based on a mechanism defined in section 4. There is a posted price which is calculated using number of remaining cores on sellers side. In each time interval, before beginning of trading, the exchange will calculate this posted price. The posted price value is inversely proportional to the remaining cores on sellers side. To get allocation, buyers should pay at least posted price. Note that all the resource requirement of one buyer will be fulfilled by only one seller. Also note that buyer will not get allocation if her requirement will not fit completely in any individual sellers

While selling virtual machines, the most expensive element is cores. For example, in Amazon's AWS the price of each compute instances(Virtual Machine) is defined by number of cores that compute instance has[8]. This price is directly proportional to the number of cores. In our model also, we are considering sellers and buyers to sell and purchase cores. The model 1 is defined formally as:

## 3.1 Seller

Number of seller remains fix in each time interval. Each seller will have two information:

1. Number of cores to sell
2. Price per core per time unit

In each time interval, number of sellers at the start time and number of sellers at the end time will remain equal. If some resources are allocated to buyer, then that seller will not go until the resource requirement will not gets completed. Although, the seller can remove herself from the next round of allocation.

## 3.2 Buyer

Buyers will come before the start of new time interval. Number of buyers may be different in different time interval but number of buyers remain fixed in a particular time interval. If some resources are allocated to buyer, then buyer will have to pay the complete money irrespective of whether she quits the allocation or not. Each buyer will have three information:

1. Number of cores required
2. Price per core per time
3. Time slot

Note that the above mentioned information is for one Virtual Machine as each buyer will demand for one Virtual Machine only. Buyer can also demand multiple machines. In that case we can break the demands of one buyer into multiple buyers where each new buyer will have one demand from original buyer. For example, give example here

## 3.3 Exchange

Exchange is the neutral entity. Sellers will share their information (number of cores to sell and price per core) to exchange. Similarly, buyers will also share their information(number of cores required, price per core and time slot) with exchange. Exchange will decide which buyer will be mapped with which seller. Price that seller will charge and buyer will pay. Behaviour of exchange is explained in Section 4.

# 4 Mechanism Design

We now define our proposed mechanism for the Cloud Double Auction(CDA) for cloud resource allocation and pricing scheme as follow:

## 4.1 Input:

**Trading Point** $L_{TD}$. A trading point is the point at the start of each time interval during which users are allowed to submit her request and CSPs are allowed to submit there free resources (resulting in transactions whenever they match).

**Seller's Set** $S = \{S_1, ..., S_n\}$: Each CSP from the seller's set has some core $C = \{C_1, ..., C_n\}$. The valuation of these cores for CSP is $V_s = \{V_{s1}, ..., V_{sn}\}$ respectively where $V_{si}$ is the price that $CSP_i$ is willing to charge for her each core. The CSP are considered as the static entity for entire trading day. From the simulation point of view, we have considered the number of cores each CSP will have as a random natural number between 20 to 50. The valuation(price that CSP is willing to pay for each core per unit time) for the CSP is taken as a random real number between 1 and 10. Both these values are picked from a uniformly distributed so as to get a good mixture of varieties of CSPs with different cores and prices.

**Buyer's Set** $B = \{B_1, ..., B_m\}$: Each buyers has some core requirement $R = \{R_1, ..., R_m\}$. The valuation of each buyer for each core is $V_b = \{V_{b1}, ..., V_{bm}\}$ respectively where $V_{bi}$ is the price that $V_i$ is willing to pay for her each core request per unit time. Buyers are considered as dynamic entity. Buyers are constant only during in a particular time interval. At each time interval buyers are generated as poison process with average 5. The core requirement of buyer is uniformly distributed over the range 1 to 5 and their valuation is uniformly distributed over the range 1 to 10. Buyer will also have time slot $T = \{T_1, ..., T_m\}$ for which they want to utilize the cloud resources. Here $T_i$ is the time slot that buyer $B_i$ wants.

## 4.2 Pricing scheme and Allocation:

The posted price is price charged by the exchange to buyers. At each trading point, the exchange calculates the posted price. This posted price depends upon the number of cores remaining on sellers side and the accumulated profit of the exchange. For calculating the posted price we considered the following two equations.

$$\text{Posted Price} = \frac{\sum_{i=1}^{n} C_i \times V_{si}}{\sum_{i=1}^{n} C_i} \quad (1)$$

$$\text{Posted Price} = \frac{\sum_{i=1}^{n} C_i \times V_{si} - AP_{i-1}}{\sum_{i=1}^{n} C_i} \quad (2)$$

Where $AP_{i-1}$ is **Accumulated profit of the exchange** upto previous time interval $i-1$

According to equation 1, the number of cores from the cloud provider decreases then Posted price will increase. It is because the number of core which has less valuation go on decreasing as allocation happens and the cores with higher valuation remains.

In equation 2, it can be seen that the exchange is trying to minimize the Posted price by pouring its own profit. Thus, exchange is trying to be a neutral entity which works on minimum profit.

The Buyers are allocated to sellers only when their valuation is higher than the posted price. the price charged from the buyer by the exchange is as follow:

$$\text{Price Charged from buyer}_i = \text{Posted Price} \times \mathrm{T}_i \times \mathrm{R}_i \quad (3)$$

when buyers are being assigned to the CSP, the CSP with lowest valuation is give priority. If a CSP is unable to fulfill the requested number of core then the chance is given to CSP with next lowest valuation of core. The payment given to the CSP by the exchange is according to the second price auction.

$P_{CSP}$ is payment paid to CSP according to Second Price Auction (SPA). For example, suppose there are 3 sellers with number of core 5,6,3 respectively and price 5.5,6,6.5 respectively. Now, if 5 cores are allocated from seller 1, then price charged by seller 1 will be 6*5=30 where 6 is the price next seller is willing to charge and 5 is the number of cores seller 1 is allocating. Similarly, if sellers have number of cores 5,2,3 respectively then 6*2+6.5*3=31.5 price charged by seller 1.

Alogorithm 1 is used for Simulation of continuous double auction for cloud resources .

---

**Algorithm 1:** SIMULATION of continuous double auction for cloud reourses

---

**Input**: $S = \{S_1, ..., S_n\}$: **Seller set** of n CSP.
$C = \{C_1, ..., C_n\}$ number of cores each CSP has. $V_s = \{V_{s1}, ..., V_{sn}\}$ Per-core valuation of each CSP

$Acumulated\_profit\_exchange \leftarrow 0$
$Acumulated\_profit\_buyers \leftarrow 0$
$Acumulated\_profit\_CSP \leftarrow 0$
**for** $t \leftarrow 1$ ***to*** $L_{TD}$ **do**
    Calculate the $Posted\_price$
    Generate m buyers using poisson process. $B = \{B_1, ..., B_m\}$ is buyer set . $R = \{R_1, ..., R_m\}$ core requirement of each buyer. $V_b = \{V_{b1}, ..., V_{bm}\}$ valuation of each buyer. $T = \{T_1, ..., T_m\}$ time for which cores are required.
    Find $Accepted\_buyer =$ buyers whose valuation is higher than posted price
    **for** $Buyer$ ***in*** $Accepted\_buyer$ **do**
        Select a lowest asking CSP whose number of Cores $C >$ Core requirement of $Buyer$
        Match the $Buyer$ to CSP
        $P_{CSP} \leftarrow$Payment paid to CSP according to Second Price Auction
        Charge Buyer according to $Postedprice$
        $Acumulated\_profit\_exchange \leftarrow Acumulated\_profit\_exchange + (Posted\_price * R_{buyer} * T_{Buyer} - P_{CSP}$
        $Acumulated\_profit\_buyers \leftarrow Acumulated\_profit\_buyers + (V_{Buyer} - Posted\_price) * R_{buyer} * T_{Buyer}$
        $Acumulated\_profit\_CSP \leftarrow Acumulated\_profit\_CSP + (P_{CSP} - C_{CSP} * R_{Buyer}) * T_{buyer}$

---

### 4.3 Incentive Compatibility

In the proposed mechanism, buyers are incentive compatible because buyer are charged only at posted price. If their valuation is less than posted price the resources will not be allocated at all and buyers will not be charged. In this mechanism Sellers are also incentive compatible because sellers are paid using second price auction mechanism. If seller will charge their resources with price other than the actual valuation then seller may run into losses or buyers are not allocated to that Seller and thus no profit at all becuase second price auction is defined for sellers charged price.

### 4.4 Individual Rationality

In this propsed mechanism, buyers are individually rational. It is because if resources are allocated to the buyer, then it will be charged according to the posted price which is less than their actual valuation. Sellers are individually rational as they are charged according to second price auction mechanism.

## 5 Data Set and Experiments

The scenario of buyers, sellers, exchange, auction, etc will run continuously. Different resources will get allocated and de-allocated in each time interval. Since, it is very difficult to do analysis on continuously running model. We took first 1000 time interval for our analysis.

We fixed number of sellers to 10. Number of cores that each seller will have for selling will be a natural number between 20 to 50. The number is taken at random using uniform distribution between the given range. Similarly, number of cores that each buyer will have is random natural number between range 1 to 5. Price per core for sellers and buyers are taken as random natural number between 1 to 10. At any time interval, number of buyers are coming using Poisson distribution with mean 5. The time slot required by buyers to run there are request is taken at random between 1 and 25. Here 25 is calculated as total time interval/40. This time slot is again taken at random from uniform distribution.

We run this simulation for two different experiment sets. In first experiment, the posted price is calculated using equation 1. While in second experiment the posted price is calculated using equation 2.

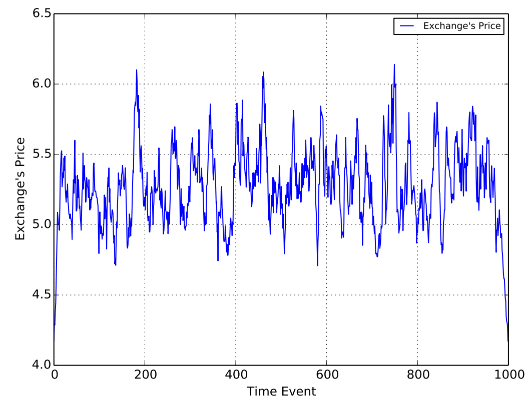


Figure 2: Posted Price using Equation 1

## 6 Results

After completing simulations for 1000 time interval, we analyzed the posted price for each time interval. We also looked into how many buyers in total gets allocated to sellers. Finally we did analysis on profit by sellers, buyers and exchange.

### 6.1 Experiment 1

We found following observation using equation 1 for posted price:
Figure 2 show the graph of posted price with respect to time interval. It can be observe that, in initial time interval the posted price is lower. After some time interval, due to consumption of resources by buyers, the posted price increases. The posted price keeps fluctuating between 4.5 units to 6 units because of continuous allocation and deal location of resources between sellers and buyers. At the end, the posted price again reduces. This is because each user has some time slot for request. At the end of simulation, buyers are not getting their time slots. This can also be evident from Figure 3 where slot became flat at the end time interval. Therefore, rate of allocation reduces which in turn increase the number of free resources on seller side. Hence, posted price reduces.

Figure 3 show the graph of total number of buyers arrived and number of buyers allocated to sellers. At any time interval $T$, the graphs show total number of buyers arrived from time interval 1 to $T$ and total number of buyers gets allocation from time interval 1 to $T$. Some buyers do not get allocation because of three reasons:(1)willingness to pay less price (2)resource requirement is very high and

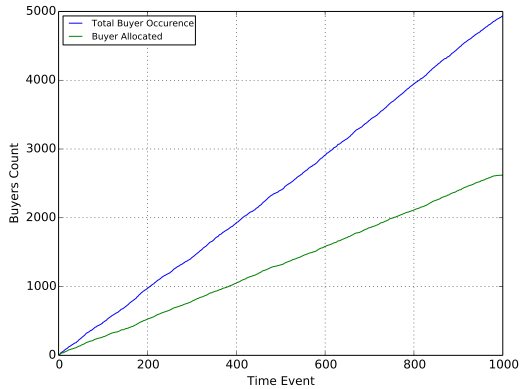


Figure 3: Buyers Allocation using Equation 1

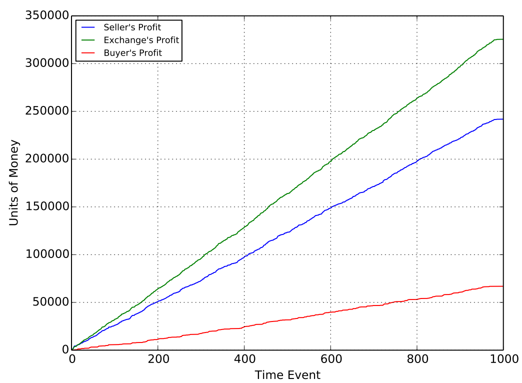


Figure 4: Profit by Agents using Equation 1

(3) time slot to run the request is very high.

Figure 4 show the graph of profit earned by each entity. At any time interval $T$, the graph show the total profit of all seller, all buyers and exchange from time interval 1 to $T$. The profit of seller is higher because number of sellers are set to 10 for each time interval, but number of buyers are arriving with possion distribution with average of 5. So total profit of sellers are higher than total profit of buyers. Profit of exchange is higher because of the posted price. This was understood after experiment 2 explained in section 6.2.

## 6.2 Experiment 2

We found following observation using equation 2 for posted price:
Figure 5 show the graph of posted price with respect to time interval. With respecet to posted price of Figure 2, it can be observe that instead of going up, it goes down from starting time interval. It goes down because of the subtraction of exchages profit from previous time interval price. The posted price keeps fluctuating between 2.5 units to 4 units which is lower than the previous posted price in figure 2. At the end, the posted price again reduces because of the same reason of not getting the time slot.

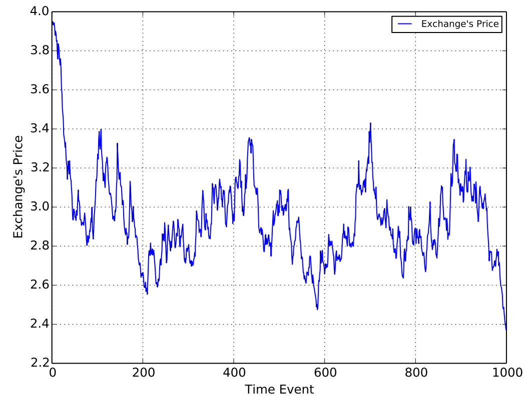


Figure 5: Posted Price after using Equation 2

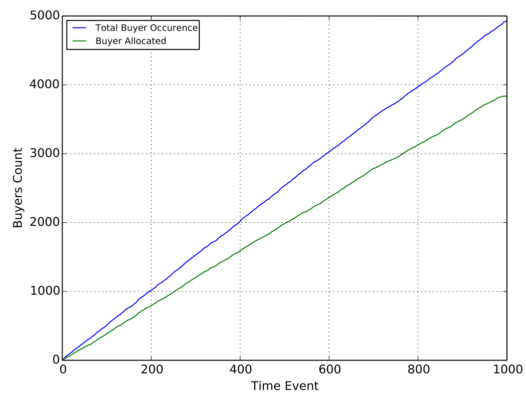


Figure 6: Buyers Allocation after using Equation 2

Description of Figure 6 is same as the description of Figure 3. The difference in this graph is the higher allocation of buyers to sellers. This is because the posted price is now reduce. So more buyers can get allocation to sellers.

Figure 7 show the graph of profit earned by each entity. The description of this Figure is same as Figure 7. The difference here is the profit by exchange. Exchange is getting very less profit. The reason is that what ever profit exchange is getting in previous round is used to reduce the posted price of current event. Hence, more buyers will get allocated and profit of exchange remains very low through

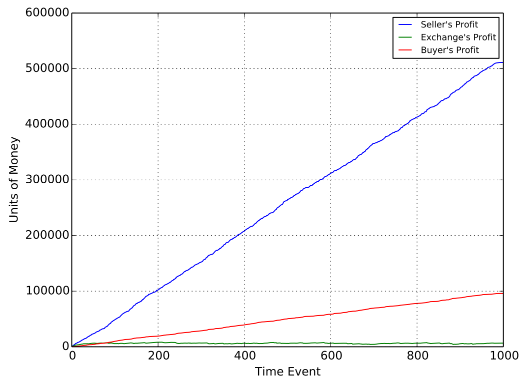


Figure 7: Profit by Agents after using Equation 2

out the time interval. This is the reason why we called exchange neutral agent.

## 7 Conclusions and Future Work

In this work we tried to design a mechanism which not only defines the price that CSP(seller) would charge and buyers would pay, but also demonstrate which buyer to map with which seller and how to give priority between sellers. Our experimental analysis shows that this mechanism design distribute the request of buyer to seller in such a way so that all three agents will get profit and remains in happy happy situation.

Our whole project, including experiments and results, is publicly available on Git [9]. The project is highly documented and completely written in python. We ensure that even a naive person with very basic knowledge of programming can understand the code with the help of documentation and comments written with the code modules.

In future work, we would like someone to prove that the mechanism proposed in this paper is weekly budget balance. We showed it experimentally. Buyers with price and core required also comes with time slot. We are allocating this request greedily to respective seller. It will be even better if someone do efficient resource allocation using techniques like interval scheduling or bin packing.

## 8 Acknowledgement

We would like to express our special thanks of gratitude to our course teacher Prof. Y. Narahari who gave us the golden opportunity to do this wonderful project, which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to him. We would also like to thank our project mentors Satyanath Bhat and Shweta Jain who are the backbone of your progress and ideas. Finally, last but not the least we would also like to express out thanks to Karrotu Santosh Srinivas and Ganesh Ghalme who helped us in clearing our lots of doubt regarding double auction and resource efficient allocation.

We are making this project not only for marks but to also increase our knowledge.

THANKS AGAIN TO ALL WHO HELPED US.

## References

[1] Buyya, Rajkumar and Abramson, David and Giddy, Jonathan and Stockinger, Heinz. *Economic models for resource management and scheduling in grid computing*, Concurrency and computation: practice and experience

[2] Vytelingum, Perukrishnen, Dave Cliff, and Nicholas R. Jennings. *Strategic bidding in continuous double auctions.*, Artificial Intelligence 172.14 (2008): 1700-1729.

[3] Clearwater, Scott H. *Market-based control: A paradigm for distributed resource allocation.*, World Scientific Publishing Co., Inc., 1996.

[4] Das, Rajarshi, et al. *Agent-human interactions in the continuous double auction.*, International Joint Conference on Artificial Intelligence. Vol. 17. No. 1. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

[5] Abhishek, Vineet, Ian A. Kash, and Peter Key. *Fixed and market pricing for cloud services.*, arXiv preprint arXiv:1201.5621 (2012).

[6] Zhang, Hong, et al. *A framework for truthful online auctions in cloud computing with heterogeneous user demands.*, INFOCOM, 2013 Proceedings IEEE. IEEE, 2013.

[7] *http://www.informationweek.com/cloud/cloud-storage/google-cloud-cuts-prices-again-adds-services/d/d-id/1317198*

[8] *http://aws.amazon.com/ec2/pricing/*

[9] *https://github.com/vedsarkushwaha/GameTheoryMiniProject.git*