

Intelligent Continuous Double Auction method For Service Allocation in Cloud Computing

Nima Farajian

Department of Computer Engineering, Faculty of Engineering, University of Isfahan,
Isfahan, Iran
E-mail: Nimaff2000@yahoo.com

Kamran Zamanifar

Department of Computer Engineering, Faculty of Engineering, University of Isfahan,
Isfahan, Iran

Abstract:

Market-oriented approach is an effective method for resource management because of its regulation of supply and demand and is suitable for cloud environment where the computing resources, either software or hardware, are virtualized and allocated as services from providers to users. In this paper a continuous double auction method for efficient cloud service allocation is presented in which i) enables consumers to order various resources (services) for workflows and co-allocation, ii) consumers and providers make bid and request prices based on deadline and workload time and in addition providers can tradeoff between utilization time and price of bids, iii) auctioneers can intelligently find optimum matching by sharing and merging resources which result more trades. Experimental results show that proposed method is efficient in terms of successful allocation rate and resource utilization.

Keywords: Cloud computing, continuous double auction, intelligent agent, resource allocation, multiagent system.

1. Introduction

Cloud computing is new generation of computing in which resources such as computing power, storage and online applications are provided as ‘services’ over the internet [1]. Several interconnected and virtualized computers and resources could be delivered to consumers as a one or more unified computing resource based on service-level agreements (SLAs) [2]. For utilizing a service multiple resources should be allocated simultaneously or as a workflow to the service. So resource management of cloud computing consists of finding and allocating required resources of each service. Because of dynamic and elastic environment of the Cloud, market-oriented approach has high potential for Cloud resource management. It regulates supply and demand for resources; motivates the consumers to trade-off between deadline, budget, and the required level of quality of service; and provides an incentive for resource providers to participate in Cloud [3]. To fit the cloud computing environment, resource allocation mechanism should consider following conditions:

1. Combination of resource: for utilizing a service, multiple resources should be allocated for workflows or co-allocation. So the market should allow consumers to acquire multiple resources for the requested services.
2. Fair pricing: The trading price should be fair and flexible. It must only depend on supply and demand of the market and never giving advantage to sellers or buyers side. Fair market motivates consumers and providers to stay in the cloud and play their rule.
3. Efficiency: Every consumer and provider desires an efficient allocation of services. The cloud marketplace should maximize the benefit of the participants and should not waste any resource.
4. Time consideration: the speed of resource allocation is important because each consumer has time constraint for allocating its resource and any delay could lead to failure in delivering the service. So consumers and providers should relaxed their criteria and converge their prices to the acceptable price of the market rapidly.

Two categories of market based models that are used for cloud resource management are commodities market models and auction models. In the commodity market model, resource prices are specified by resource providers, and consumers are charged according to the amount of the resource which they consume. In the auction model, each provider and consumer make decision autonomously and agree privately on the selling price. Auctions are used for products that have no standard values and supply and demand of the market affect the prices [3]. Generally there are 4 kinds of auction: English auction, Dutch auction, First and Second Price Auction and Double auction. The double auction model has a high potential for cloud computing. In a double auction model, consumers submit bids and providers submit requests at any time during the trading period. If at any time there are bids and requests that match or are compatible with a price, then a trade is executed immediately [12].

Three most popular double auctions are: Preston-McAfee Double Auction Protocol (PMDA) [5], Threshold Price Double Auction Protocol (TPDA) [6], and Continuous Double Auction Protocol (CDA) Kant and Grosu [7] showed that the CDA protocol provides high resource utilization in grid environments and is better from both the resource's and the user's perspective.

In this paper, an intelligent continuous double auction method for service allocation in cloud computing is presented in which:

- i. Consumers can acquire multiple resources for workflows and co-allocation.
- ii. By considering deadline and workload time, consumers and providers adjust their prices quickly to the acceptable price of the market. So consumers can acquire more resources before their deadline.
- iii. For increasing resource utilization, providers can trade-off between price and utilization time of consumer requests and also express their willingness of sharing their resource between multiple consumers.
- iv. Auctioneers intelligently find optimum allocation for each consumer request by merging and sharing resources.

We call this method intelligent continuous double auction method (ICDA) and the results illustrate that the proposed method is efficient in resource utilization and successful allocation rate. The rest of the paper is organized as follow. Section 2 describes some relevant works. The design details of this model and its mechanism are presented in Section 3. The evaluation and experimental results are shown in Section 4 and conclusion is provided in Section 5.

2. Related work

Most of the studies and researches on management of resources and services in cloud computing is derived from methods and techniques which have been used in grid computing. The reason for this is high similarity between cloud and grid computing and also earlier emergence of grid computing.

Buyya et al. [3] used economic based concepts including commodity market, posted price modeling, contract net models, bargaining modeling for grid resource allocation. Schnizler et al. [8] introduced a double-sided combinational auction to allocate grid resources. Tan et al. [9] proposed a stable continues double auction, based on the more conventional CDA. It reduces the unnecessarily volatile behavior of the CDA, while maintaining other beneficial features. Amar et al. [10] illustrated a comprehensive grid market model including a futures market and a centralized/decentralized spot market. Izakian et al. [11, 12] introduced a continuous double auction method in which resources are considered as provider agents and users as consumer agents. In each time unit, each provider agent determines its requested value based on its workload and each consumer agent determines its bid value based on the remaining time for bidding and the remaining resources for bidding. In [13] authors present a reverse VCG auction for cloud computing in which consumers hold auctions for required resources and providers submit their offers and winner determination is based on lowest offer. Fujiwara et al. [14] Presented combinational auction model for cloud computing and design spot and future auction market for provider to put their resource in it. Importantly, this method use combinational auction which auctioneer puts multiple different

types of resources in auction and each consumer can bid for combination of them. Winners determination is based on maximum profit. Teymouri [15] proposed a new UCDA method based on CDA method in which bids are updated by auctioneer itself. Also providers determine the resource price based on their workload and users determine their bids based on jobs deadlines.

Sim [16] presents a market-driven negotiation mechanism for Grid resource management which includes (i) a market-driven strategy and (ii) a relaxed-criteria negotiation protocol. In market-driven strategy, agents consider opportunity and time factors to determine the value of prices. Sim [17] presented a resource management model for cloud computing based on market-driven agent which introduced in [16]. In this model there are provider, consumer and broker agents. Through a 4-stage resource discovery process (selection, evaluation, filtering, and recommendation), Broker agents match consumers' requests to suitable Resources and then consumer and provider agents negotiate for mutually acceptable resource time slots.

3. Market mechanism and problem definition

3.1. Cloud computing market model

We assume the cloud computing environment includes resource providers, consumers and an auction market. Consumers refer to cloud to use and consume its required services. Each service is composed of different resources which are provided by providers who sell them to make profit. The auction market consists of multiple auctioneers which each of them holds auction for one resource type. For each entity in cloud market there is one agent. Consumer agent works on behalf of the consumer and acquires required resources of the requested service by interaction with auctioneers, provider agent which works on behalf of the provider and sells its resource to consumers through related auctioneer and auctioneer agent which receives bids and requests from all consumers and providers and executes trade for all matching bids and requests.

We assume that the resources allocation satisfies the following conditions:

- The quantity of a resource can be measured in arbitrary units (e.g. 60 units of resource A).
- A resource can be divided into an arbitrary fraction (e.g. a resource of 60 units is divided into 20 units for consumer 1 and 40 units for consumer 2).
- A resource request of a service can be divided into sub-requests and acquired from multiple providers (e.g. a resource request of 40 units utilized as 10 units from provider 1 and 30 units from provider 2).

Figure 1 shows a cloud computing environment with the proposed mechanism.

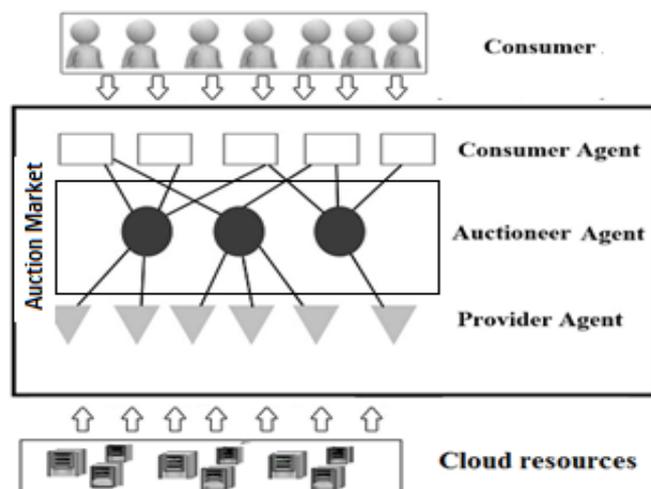


Figure 1. Resource allocation schema in proposed method

3.2. Problem formulation

A resource from a provider (R_i) includes the following information:

- RP_{min} : the minimum price per timeslot at which the provider wishes to sell the resource (Reserve price)
- RP_{max} : the maximum price per timeslot at which the provider wishes to sell the resource (Maximum price)
- R_{typ} : the kind of resource
- R_q : the amount (quantity) of the resource
- R_{wr} : the time at which resource could be allocated to new request (workload time)

Each service request is composed of some resource requests (C_i) which each of them includes the following information:

- CP_{max} : the maximum price per timeslot at which the Consumer allocates to the requested resource
- C_{typ} : the kind of requested resource
- C_q : the amount (quantity) of the requested resource
- C_{ut} : the amount of time which consumer needs to utilize the resource
- C_{td} : deadline time for consumer which consumer must finish utilizing the requested resource before deadline

3.3. Continuous double auction method

In the continuous double auction method at each time unit, consumers and providers submit bids and requests to the related auctioneer. An auctioneer maintains a list of the current bids and requests and executes trade for matching bids and requests.

In this method, consumers, providers and auctioneer make decision autonomously which is described as follows:

3.4. Consumer agents

Each consumer is looking for utilizing its requested service with minimum price before its deadline. To utilize a service all required resources should be allocated before deadline and otherwise service failed to utilize and consumer must pay penalty to providers for all other resources which is allocated to it. So Consumer should adjust its bid price rapidly to the acceptable price of the market. Since consumers are generally sensitive to deadline in acquiring requested service, it is intuitive to consider deadline time when formulating the bid price. Consumer agent time dependent bid price formula is determined in (1).

$$Bid(t, t) = \left[\frac{R_{min}}{CP_{max}} + \left(1 - \frac{R_{min}}{CP_{max}} \right) \left(1 - \frac{tre}{Ctd} \right)^\rho \right] \times CP_{max} \quad (1)$$

R_{min} is minimum reserve price between all providers and tre is time interval between t (current time) and td . $0 < \rho < \infty$ is the concession making strategy. Three classes of strategies are specified as follows: Conservative ($\rho > 1$) which the consumer maintains a low bid value until current time gets close to td , Linear ($\rho = 1$), and Conciliatory ($0 < \rho < 1$) which the consumer starts with a bid value close to mps . Fig. 2 depicts the different convexity degrees of the curves with $\rho = 0.2, 1, 5$

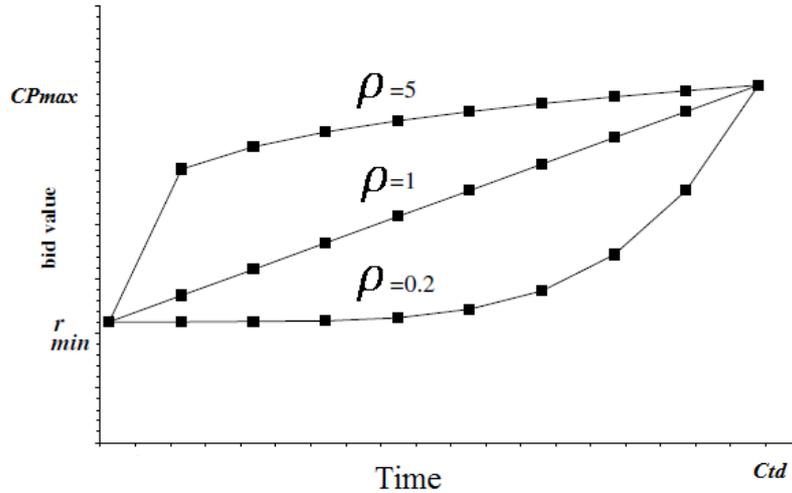


Figure 2. bid value for time factor

Consumer agent in each time unit calculates its bid prices for each requested resource and submits them to related auctioneer.

3.5. Provider agent

Each provider has two objectives: (i) make maximum profit for selling its resource (ii) increase utilization time of its resource. Provider agent in each time unit determines its request value based on workload time. In addition provider determines sharing and trade-off factors which are used to increase resource utilization.

Workload time: Provider agent is sensitive to idle time of its resource and when the time is near the finishing time of current workload, provider should relaxes its request price to allocate its resource sooner. Provider considers workload time for calculate request price as in (2).

$$Req(t, t) = \left[\frac{RPmin}{RPmax} + \left(1 - \frac{RPmin}{RPmax} \right) \left(\frac{tst}{Rwr} \right)^\tau \right] \times RPmax \quad (2)$$

tst is time interval between t and Rst . Based on (2) each provider offers its reserve price when its resource is idle and after allocation makes its price to $RPmax$ and with elapsing time decreases its price. Also τ is the concession making strategy like ρ and σ .

Trade-off factors: Provider prefers to allocate its resource to consumer with longer utilization time because the longer request increases resource utilization for provider and also give more time to provider to sell its resource with better situation in the next trade. So provider agent should consider the tradeoff between the benefit of allocating its resource to a suboptimal (slightly cheaper) consumer with longer utilized time and the benefit of allocating to an optimal consumer (more expensive) with shorter utilized time. Tradeoff factor expresses the amount of willingness of the provider for allocating its resource to the suboptimal request. So provider decreases its request price per each utilization timeslot of consumer bid as trade-off value which is calculated as in (3).

$$Rtr(t, t) = \left(1 - \frac{tst}{Rwr} \right) \times 0.01 \quad (3)$$

Sharing factor: Usually provider allocates full capacity of its resource to one consumer but in scenario which probability of finding consumer for full quantity (capacity) of the resource is low and also workload of the resource is close to finishing, provider can share its resource capacity to multiple consumer requests. Resource sharing increase resource utilization of the provider but makes some overhead for provider because when the resource is fragmented, provider should

calculate request value for each segment individually. $P(i,t)$ express the probability of finding full quantity request for provider i at time t and is calculated as in (4).

$$P(i,t) = \left[\left(\frac{tst}{Rst} \right) + \left(\frac{Rq}{Qavg + Rq} \right) + \left(\frac{Pavg}{RPmin + RPmax} \right) \right] \times \frac{1}{3} \quad (4)$$

$Qavg$ is the average quantity between all bids and $Pavg$ is the average bid price between all bidders which is calculated and submitted by auctioneer. For expressing resource sharing willingness of the provider, we use $Rsh(i,t) \in \{0,1\}$ (sharing factor) which denotes whether provider i wants to fragment its resource at time t and is determined in (5).

$$\begin{cases} Rsh(i,t) = 1, & \text{if } P(i,t) < \theta \\ Rsh(i,t) = 0, & \text{if } P(i,t) \geq \theta \end{cases} \quad (5)$$

$0 \leq \theta \leq 1$ is sharing constant. Higher (lower) value of θ cause more (less) willingness of provider for resource fragmentation. Provider agent finally submits its request price and sharing/tradeoff factors to related auctioneer in each time unit.

3.6. Auctioneer Agent

In each time unit consumers and providers submit their bid and request values to auctioneer and auctioneer sorts bids in highest order and request in lowest order and then for each bid which is higher than lowest request, auctioneer finds matching resources, executes trade and then remove the bid from the list. To find matching resources for each bid auctioneer should find optimum (minimum) price for that bid from requests list. If the minimum price is lower than the bid value then the trade occurs.

For calculating minimum price for each bid we formulate the minimum price problem into a linear integer programming (LIP). Here we introduce $V_{i,j}$ which is the value of request (j) for bid (i) and is calculated as in formula (6) and also $0 \leq Q_{j,i} \leq Rq_j$ denotes the amount of resource j allocated to consumer i .

$$V_{i,j} = \begin{cases} Req_j - (Rtr_j \times Cut_i) & , \text{if } (Cq_i \geq Rq_j) \text{ or } (Cq_i < Rq_j \text{ and } Rsh_j = 1) \\ \infty & , \text{if } (Cq_i < Rq_j) \text{ and } (Rsh_j = 0) \end{cases} \quad (6)$$

We solve minimum price problem as follows:

Minimize

$$O_i = \sum_{j=1}^M V_{i,j} \times Q_j \quad (7)$$

s.t.

$$O_i = \sum_{j=1}^M V_{i,j} \times Q_j \quad (8)$$

$$Cq_i - \sum_{j=1}^M Q_{i,j} = 0 \quad (9)$$

$$Ctd_i - Rwr_j - Cut_i \geq 0, \quad 1 \leq j \leq M$$

M is number of requests in auctioneer list. After calculating O_i , if $O_i > CPmax_i \times Cut_i$, then the trade occurs at the following price:

$$price = \frac{(CPmax_i \times Cut_i) + O_i}{2} \quad (10)$$

4. Experimental Results

In order to study the efficiency of the method presented in this paper, we developed a computational Cloud simulator using the java agent development framework (JADE) [18]. JADE is a middleware for developing multi agent systems and applications according to FIPA standards for intelligent agents. In our simulated system, consumers and providers are two kinds of agent and also for each resource type there is one auctioneer agent. Resource type and quantity are shown in Table 1.

Table 1. Resource Specification

CPU Speed (GHz) : 1.5 ~ 3.5
Memory(Mb): {512,1024,2048}
Disk Capacity(Mb) : 50 ~ 1000

In our experiments we use the system load concept i.e. the ratio of aggregated length and capacity of resource requests submitted to the cloud to the aggregated capacity that the cloud is capable of providing in the simulation period. The system load can be obtained using Eq. (11).

$$\rho_{Resource1} = \frac{\sum_{j=1}^m (Cq_j \times Cut_j)}{T_{total} \times \sum_{i=1}^m Rq_i} \quad (11)$$

In the simulated system there are 100 consumer agents and each consumer agent requests 1 to 3 resources for its required service. Number of provider agent depends on system load and each provider agent provides one resource. For a fair comparison each provider set reserved price to 2\$ and maximum price to 9\$. The budget of each consumer is random integer between 3\$ and 8\$. Consumer agent utilization time of a resource (Cut) is random integer number within range [2000, 5000] and deadline is from 2 to 5 times of Cut.

To Evaluate efficiency of this method we first consider the effect of tradeoff, merging and sharing factors in success rate and resource utilization (which is summarized in TABLE II) and then consider success rate, average resource utilization in the purposed method and compared it with random continuous double auction (RCDA), continuous double auction method in [12] (CDA) and continuous double auction method in [15] (UCDA). In all simulation we call the presented method (ICDA). In the RCDA asks and bids are determined randomly between Rmin and CPmax for the consumers and the RPmin and RPmax for providers.

Fig (3) shows the effect of merging and sharing resources by auctioneer in term of success rate of allocation. As shown in it, these factors improve successful allocation rate especially in higher system load.

Table 2. success rate and resource utilization

Success rate		$Succ = \frac{R_{matched}}{R_{total}}$
$R_{matched}$	Number of requests that are matched to resource(s)	
R_{total}	Total number of requests from consumers	

Resource Utilization		$RU = \frac{T_{allocate}}{T_{total}}$
$T_{allocate}$	Total number of time slots allocated	
T_{total}	Time window	

In Figure 4 we consider tradeoff and sharing factors in providers. The result illustrates that by using these factors providers improve resource utilization. Higher resource utilization motivates more providers to participate in the cloud and also enables the cloud market to handle more consumers which influences market efficiency.

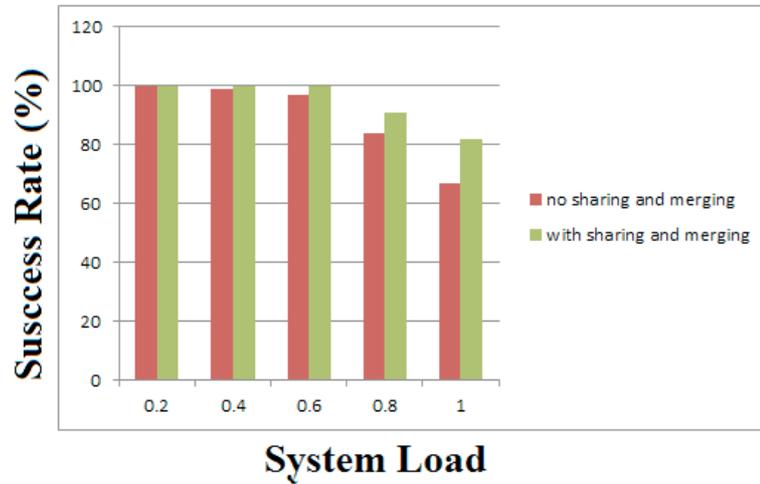


Figure 3. Sharing and merging effect on successful allocation

Figure 5 illustrates comparison of successful allocation rate between ICDA and other methods. In the proposed method, intelligent allocation and also time consideration enable consumers to acquire more resources before the deadline and as a result, the number of successful allocation is higher than other methods.

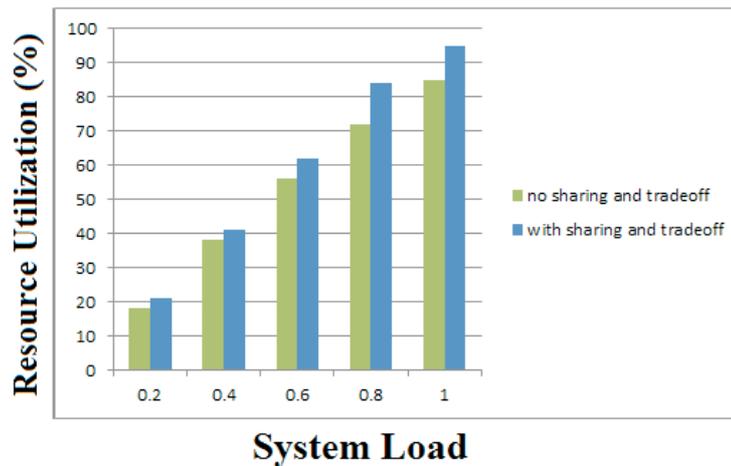


Figure 4. Sharing and tradeoff factor effect on resource utilization

Figure 6 shows resource utilization in different system loads and as shown in it, in ICDA resource utilization is more efficient than other methods especially in higher system load which is due to tradeoff and sharing factors.

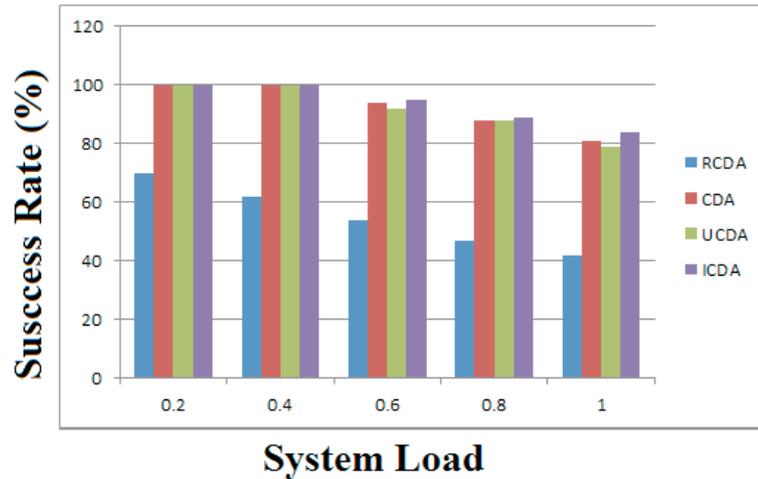


Figure 5. Comparison on successful allocation

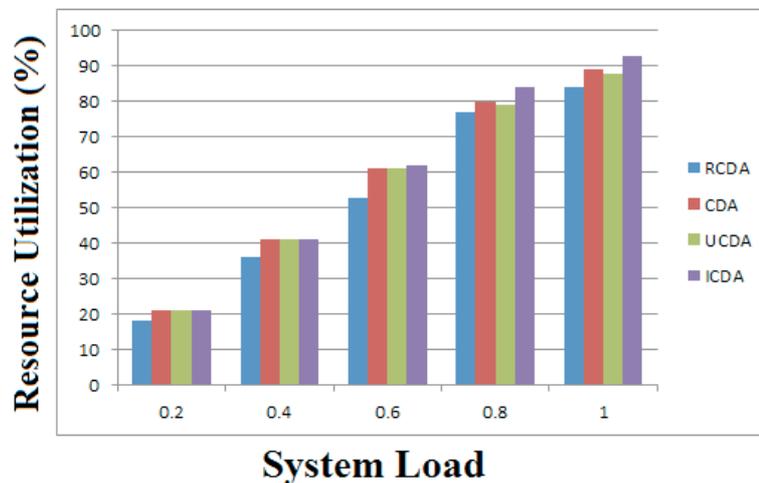


Figure 6. Comparison on resource utilization.

5. Conclusion

In Cloud Computing resource allocation is based on interaction and agreement between cloud consumers and providers. According to highly dynamic environment and also different policies of each provider and consumer, we need a method in which each participant can make decision autonomously based on market condition and its interest. In this paper a continuous double auction model is presented in which consumer and provider make bid and request with workload and deadline time consideration and also auctioneer can merge or share resources for executing more trades. Experimental results illustrate that proposed method is efficient and intensive for both consumer and provider.

References

- [1] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A Break in the Clouds: Towards a Cloud Definition, ACM SIGCOMM Computer Communication Review 50 Volume 39, Number 1 (2009)
- [2] R. Buyya et al. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems, vol. 25, no. 6, pp.599–616 (2009)
- [3] R. Buyya, D. Abramson, J. Giddy, H. Stockinger. Economic models for resource management and scheduling in Grid computing, The Journal of Concurrency and Computation 14 (13_15) 1507_1542 (2002)

- [4] K. M. Sim. A Survey of Bargaining Models for Grid Resource Allocation. *ACM SIGECOM: E-commerce Exchanges*, Vol. 5, No. 5, January, pp. 22-32 (2006)
- [5] R.P. McAfee, A dominant strategy double auction, *Journal of Economic Theory* 56 434450 (1992)
- [6] M. Yokoo, Y. Sakurai, S. Matsubara. Robust double auction protocol against false-name bids, in: *Proc. of the 21st IEEE International Conference on Distributed Computing Systems*, pp. 137145 (2001)
- [7] U. Kant, D. Grosu. Auction-based resource allocation protocols in Grids, in: *16th International Conference on Parallel and Distributed Computing and Systems*, pp. 2027(2004)
- [8] B. Schnizler, D. Neumann, D. Veit, and D. Weinhardt. Trading grid services - a multi-attribute combinatorial approach, *European Journal of Operational Research*, vol. 187, no. 3, pp. 943-961 (2008)
- [9] Z. Tan and J. R. Gurd. Market-based grid resource allocation using a stable continuous double auction, *Proc. 8th IEEE/ACM Int. Conf. on Grid Computing (Grid 2007)*, pp. 283-290 (2007)
- [10] L. Amar, J. Stosser, and E. Levy. Harnessing migrations in a marketbased grid OS, *Proc. 9th IEEE/ACM Int. Conf. on Grid Computing (Grid 2008)* , pp. 85-94 (2008)
- [11] Izakian, H., Ladani, B.T., Zamanifar, K., Abraham, A., Snasel, V. A Continuous Double Auction Method for Resource Allocation in Computational Grids, *IEEE Symposium on Computational Intelligence in Scheduling, 2009*, pp. 29-35 (2009)
- [12] H. Izakian, A. Abraham, B. Tork Ladani. An auction method for resource allocation in computational grids . *Future Generation Computer Systems* vol 26 ,pp 228–235 (2011)
- [13] Xiaohong Wu at la. Cloud computing resource allocation mechanism research based on reverse auction. *Energy Procedia* vol 13, pp 736–741 (2011)
- [14] I. Fujiwara. Applying Double-sided Combinational Auctions to Resource Allocation in Cloud Computing. *IEEE 10th Annual International Symposium on Applications and the Internet* (2010)
- [15] S. Teymouri, A. M. Rahmani. A Continues Double Auction Method for Resource Allocation in Economic Grids, *International Journal of Computer Application* vol 43, pp 7-12 (2012) .
- [16] K. M. Sim. From Market-driven e-Negotiation Agents to Market-driven G-Negotiation Agents. *Proc. of the IEEE Int. Conf. on e-Technology, e-Commerce and e-Services*, pp. 408-413, Hong Kong (2005)
- [17] K. M. Sim. Agent-based Cloud Commerce. *Proc. IEEE Int. Conf. on Industrial Eng. and Eng. Management, Hong Kong*, pp. 717-721(2009)
- [18] JADE, Java Agent Development Framework, (2004). <http://jade.csel.it>.