A Novel Market Learning Algorithm for Grid Resource Allocation

Saeed Parsa[1], Amin Shokri[2]
[1]Department of Computer Engineering
University of Science and Technology, daneshjo Alley
Hengam Street, Resalat Square, Tehran
Iran
parsa@iust.ac.ir

[2]Department of Computer Engineering
University of Science and Technology, daneshjo Alley
Hengam Street, Resalat Square, Tehran
Iran
am_shokri@comp.iust.ac.ir

ABSTRACT: *The aim of this research has been to introduce a learning algorithm based on market concepts for grid resource allocation and resolve some incompetence with current mechanisms. Grid resource allocation could be considered as a double auction in which grid resource manager acts as an auctioneer and jobs and resource owners act as buyers (resource consumers) and sellers. In our novel algorithm resource allocation is based on activity history of each participant in the auction. Bids in the auctions are calculated in each round, based on proportion of former bid fluctuations and prior activity of each buyer and deadline issues of each buyer's job. As the same, asks in the auctions are calculated in each round, based on proportion of former ask fluctuations and prior activity of each seller probability of failure for each seller (resource owner) which is calculated by Poisson function. Practical evaluation results demonstrate that market equilibrium in our mechanism is improved in comparison with other mechanisms. Also essence of learning features in this novel algorithm increases competition incentive in participants which do selfishly in every round of an auction to maximize their utility.*

Keywords: Double Auctions, Market Theory, Equilibrium, Pricing mechanism, Grid Resource Allocation

Received: 18 August 2009, Revised 29 September 2009, Accepted 8 October 2009

1. Introduction

Adaptation is changing and adjusting to a new condition .In computational grid where conditions are changing many times, being adapted is a great feature[2]. Computational grid has an important rule in distributed processing. One of the major problems in grid is resource allocation problem [1, 11]. Nowadays market and game theory concepts are mapped in computer science, especially in resource allocation problems. A sustainable market-like computational grid has two characteristics: it must allow resource providers and resource consumers to make autonomous scheduling decisions and both parties of providers and consumers must have sufficient incentives to stay and play in the market [9]. Two categories of market based models that are used for grid resource management are commodities market models and auction models [3].

In commodity market model, providers specify their resource price and charge users according to the amount of resource they consume. In auction model, each provider and consumer acts independently and they agree privately on the selling price. Auctions are used for products that have no standard values and the prices are affected by supply and demand at a specific time. Auctions require little global price information, are decentralized, and easy to implement in grid setting [3].

Based on interactions between consumers and providers, auctions can be classified into four basic types: the ascending auction (English auction), the descending auction (Dutch auction), the first-price and second-price sealed auction, and the double auction. The double auction model has a high potential for grid computing [3, 7]. 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. Three most popular double auctions are: Preston-McAfee Double Auction Protocol (PMDA) [13], Threshold Price Double Auction Protocol (TPDA) [15], and Continuous Double Auction Protocol (CDA). It's corroborated that CDA protocol is better than both resource's and user's perspective providing high resource utilization in grid environments [8, 18].

In this research we are going to introduce an auction based adaptable learning algorithm for grid resource allocation. In our approach, we present a new learning mechanism for pricing in Continuous Double Auction. Also we offered a reward function to prove the winner choosing method in the auctioneer. Experimental results prove that proposed method is effective in resource utilization and make a condition for the participants to have enough incentives to be in the auction. Market equilibrium is achieved in our algorithm more idealistic than the other methods .Mainly price fluctuation is decreased perspicuously in our mechanism.

The remainder of this research is organized in the following manner. In Section 2 we investigate the related works and in Section 3 we formulate the problem. In section 4, we present our method entirely followed by experimental results in Section 5 and finally Section 6 concludes this work.

2. Related Work

Market based resource allocation has been investigated by several researches in [3, 4, 5, 6, 10, 14, 15, 16, 17]. Buyya [3] used concepts of commodity market and posted price bargaining modelling for grid resource allocation. Gjerstad et al.[14] proposed a pricing strategy for continuous double auction. This strategy is a memory based algorithm. GD traders record all asks and bids they made in the history H_m of their last m transactions. From H_m, a GD trader computes a brief function to estimate the probability of a bid or ask at a specific price being accepted [14]. Gode et al.[5] introduced a bidding strategy in CDA named Zero Intelligence strategy. In this strategy bids and asks are generated randomly in a specific interval and sent to the auctioneer. However ZI mechanism is easy to implement but its results in CDA are unpredictably desirable [17] .Reddy et al.[15] developed a sealed bid method for optimizing time and budget in a grid environment. Also Zue [17] referred a CDA bidding strategy called Fixed Mark up Strategy. In this strategy buyers and sellers change their offers with a fixed mark up rate to achieve market equilibrium.

Xhafa et al.[6] introduced a genetic algorithm for grid scheduling. In this work it's presented an extensive study on the usefulness of GAs for designing efficient grid schedulers when make span and flow time are minimized. Zue [17] also referred a memory based bidding strategy called Kaplan strategy. In this mechanism a buyer waits until bids are getting enough decreased then sends a bid lower than current bid and wins the auction. Researches affirm that in an auction in which all participants have Kaplan strategy, the utility is the worst [17] because all buyers and sellers wait and not participate in the auction. Herbert et al.[4] mentioned a memory based bidding strategy in CDA called Historical Pricing. In this strategy, next bid (ask) is equal to the average of n former bids (asks).

Yokoo et al.[10] demonstrated a robust double auction protocol and considered the possibility of a new type of cheating in the auctions. A new double auction protocol called the Threshold Price Double auction (TPD) protocol was developed which is dominant strategy incentive compatible even if participants can submit false-name bids. Song et al.[16] considered trusted grid computing and developed a security-binding scheme through site reputation assessment and trust integration across grid sites.

3. Problem Formulation

The entities in our auction are users (resource consumers) and providers (resource owners). Each user has some independent jobs for execution and recourses are willing to rent themselves to the users in case of getting some profits. Users and resources contract with each other in form of Continuous Double Auction to achieve their objectives.

We assume that there is set of n users $U=\{u_1, u_2, u_3,...,u_n\}$ and there is set of m resources $R=\{r_1, r_2, r_3,...,r_m\}$. Each user has a set of k jobs $J=\{j_1, j_2, j_3,...,j_k\}$ which must be executed by resources. Each user u_i characterized with six-tuple $\left(ID_i, \varphi_i, HL_i, b_{\max}, b_{\min}, b_i^{last}\right)$. In which φ_i is the budget of i'th user, HL_i is length of bidding history for i'th user. $b_{\max}$ is the maximum bid that i'th user can send specified by G$ per MI. $b_{\min}$ is the minimum bid that i'th user can send and expressed in G$ per MI . b_i^{last} is the last bid of i'th user specified by G$ per MI. Each job *jk* is introduced with three tuple (ID_k, l_k, d_k). In which l_k is the length of k'th job and is expressed in form of MI and d_k is the deadline of k'th job. As the same manner each resource r_j is characterized with eight-tuple $\left(ID_j, PO_j, \theta_j, HL_j, a_{\max}, a_{\min}, a_j^{last}, \lambda_j\right)$. In which PO_j is the power of j'th resource and specified by MIPS. θ_j is the overall budget of j'th resource specified by G$. HL_j is the length of asking history for j'th resource. a_{max} is the maximum ask that j'th resource can offer specified by G$ per MI. a_{min} is the minimum ask that j'th resource can offer expressed in form of G$ per MI. a_j^{last} is the last ask of j'th resource and specified by G$ per MI. λ_j is the Poisson parameter of failure probability for j'th resource in HL_j auction rounds.

In this approach we assume each resource can execute one job per round and in each round of auction just one pair of (user, resource) is the winner. Bids and asks in the agents are evaluated in form of G$/MI and compared with each other in form of G$/second in the auctioneer. Final price in each round of auction which is exchange between winners is evaluated in form of G$.

4. Our Proposed Algorithm

As mentioned in previous section consumers and providers send their offers in form of G$ per MI. Resources try to maximize their budget by accepting more jobs and on the other hand try to send their offers with higher prices. Users also try to win the auction besides make an offer with lower price and maximize their budget. Auctioneer receives offers in each round and trade is happened when conditions in (1) are confirmed.

$$\begin{cases} b^i_{\max} > a^j_{\min} \\ d^i_k - \dfrac{l_i}{PO_j} \geq 0 \end{cases} \tag{1}$$

Means maximum bid is at least equal to the lowest ask and deadline of k'th job in i'th user won't end during its execution by the j'th resource which wins the auction round. Final price is evaluated by (2) in form of G$.

$$P_{final} = \left(\left(\frac{B_{\max} + A_{\min}}{2} \right) \times \frac{l_{winner}}{P_{winner}} \right) \tag{2}$$

In which B_{max} is the maximum bid in form of G$/S and A_{min} is the minimum ask specified by G$/S in current auction round. l_{winner} is the length of the winner job expressed in MI and P_{winner} is the power of the winner resource. P_{final} is the final price in form of G$ which effects on the budget of both winners according to (3) and (4).

$$\varphi^t_{buyer} = \varphi^{t-1}_{buyer} - p^t_{final} \tag{3}$$

$$\theta^t_{seller} = \theta^{t-1}_{seller} + p^t_{final} \tag{4}$$

In which φ^t_{buyer} and φ^{t-1}_{buyer} are budget of the winner user after and before the trade .Also θ^t_{seller} and θ^{t-1}_{seller} are the budget of the winner resource after and before the trade.

4.1 Our Proposed Pricing Mechanism

$2 \times n$ In our novel pricing strategy, each user evaluates next bid based on its last bid, deadline time of its current job which will be executed and an activity history of itself which is a 2×n matrix. This matrix includes n former bids and the result of the auction for each bid (0 means losing the round and 1 means winning the round). Each resource is acts same as users and sends next ask based on its last ask, its failure probability (which is computed by a Poisson function) and its activity history. In our strategy, in each round bid value is determined by (5).

$$b^t_i = \begin{cases} b^{t-1}_i + \left[k^{win}_i . \left(b_{\max} - b_{\min} \right) \right] & if\ (last\_Success = 1) \\ b^{t-1}_i + \left[k^{lose}_i . \left(b_{\max} - b_{\min} \right) \right] & iif\ (last\_Success = 0) \end{cases}$$

$$bFinal^t_i = \left[\frac{\left(\frac{1}{d^i_k} \right)^x}{\sum \left(\frac{1}{d} \right)^x} \right] . \begin{cases} b_{\min} & if\ b^t_i < b_{\min} \\ b_{\max} & if\ b^t_i > b_{\max} \\ b^t_i & otherwise \end{cases} \tag{5}$$

In which b_i^{t-1} is the same as b_i^{last} and b_i^t is the preliminary bid value for the current round. d_k^i is the deadline of k'th job in i'th user. It's analyzed that when the deadline of a user's job is getting close to end, user should increase its bid value. In the same manner when the deadline is far enough to be finished, user should decrease its bid value. $bFinal_i^t$ is the final current bid value which is sent to the auctioneer. It is considered that bid values must be always in the interval $[b_{\min}, b_{\max}]$. k_i^{win} and k_i^{lose} are learning parameters and determined in (6) and (7).

$$k_i^{win} = -\left(1-\left(\frac{HL_i - n_i + 1}{HL_i + 1}\right)^2\right)^x \tag{6}$$

$$k_i^{lose} = +\left(1-\left(\frac{n_i + 1}{HL_i + 1}\right)^2\right)^x \tag{7}$$

In which n_i is the number of wining rounds in HL_i auction rounds. HL_i is the number of columns in activity history matrix of i'th user .It is derived form (5), (6) and (7) that when a user won in the last auction round and has a good activity history (means wins more out of n rounds) the next bid value will be decreased more than the condition which it has a bad activity history. As the same when a user lost in the last auction round and has a bad activity history (means loses more out of n rounds) the next bid value will be increased more than the condition which it has a good activity history. x is changing size of learning parameters effect in (5) .Resource request values are determined by (8).

$$a_j^t = \begin{cases} a_j^{t-1} + \left[h_j^{win}.\left[\left(a_{\max} - a_{\min}\right)\cdot\left(\frac{PO_{\min}}{PO_j}\right)\right]\right] & if\ (last\_Success = 1) \\ a_j^{t-1} + \left[h_j^{lose}.\left[\left(a_{\max} - a_{\min}\right)\cdot\left(\frac{PO_{\min}}{PO_j}\right)\right]\right] & if\ (last\_Success = 0) \end{cases} \tag{8}$$

$$aFinal_j^t = \left(1-\left(P_{fail}\right)^2\right)^x \cdot \begin{cases} a_{\min} & if\ a_j^t < a_{\min} \\ a_{\max} & if\ a_j^t > a_{\max} \\ a_j^t & otherwise \end{cases}$$

In which a_j^{t-1} is acts like a_j^{lost} and a_j^t is the preliminary current ask value and $aFinal_j^t$ is the final current ask that is sent to the auctioneer. PO_{min} is the minimum power of all resources and PO_j is the power of j'th resource. As the same in users, request values in resources must be always in the interval $[a_{\min}, a_{\max}]$. h_j^{win} and h_j^{lose} are learning parameters in (8) and determined in (9) and (10).

$$h_j^{win} = +\left(1-\left(\frac{HL_j - n_j + 1}{HL_j + 1}\right)^2\right)^x \tag{9}$$

$$h_j^{lose} = -\left(1-\left(\frac{n_j + 1}{HL_j + 1}\right)^2\right)^x \tag{10}$$

In which nj is the number of wining rounds in HLj auction rounds and HLj is the number of columns in activity history matrix of j'th resource which is a 3×n matrix. This matrix includes n former asks and the result of the auction for each ask. (0 means losing the round and 1 means winning the round). Also this matrix contains the number of failures in n auction rounds for j'th resource (0 means normal state and 1 means failure state). It's deduced form (8), (9) and (10) that when a resource won in the last auction round and has a good activity history the next ask value will be increased more than the condition which it has a bad activity history. As the same when a resource lost in the last auction round and has a bad activity history, the next ask value will be decreased more than the condition which it has a good activity history. Also the failure probability P_{fail} with parameter λ_j which is calculated in Poisson function is effects the tolerance of ask value in each auction round. In a condition where a resource has more failures, the ask value is more decreased. P_{fail} is calculated based on Poisson function with parameter λ_j in (11).

$$p(X=i)=\frac{e^{\lambda_j}\lambda_j^{\,i}}{i!}$$
$$i=\sum_{k=1}^{HL_j} ActivityHistory[3][k] \tag{11}$$

In which i is the number of failure states in HL_j auction rounds of j'th resource.

4.2 Auctioneer in Proposed Algorithm

In our approach Auctioneer's role in a double auction is to match offers, update the budget of buyers and sellers, evaluate reward function which will be introduced later and generally manage the resource allocations. We introduced a novel reward function for the auctioneer which helps it in a condition when more than one user (resource) offer maximum bid (ask) in equal. By this reward function, auctioneer chooses a user (resource) as the winner which has the best reward function output. In other words i'th user and j'th resource are the winners when one of the conditions in (12) is verified.

$$\begin{aligned}
&1)\ \forall p\neq i, q\neq j:\ bFinal_i^t - aFinal_j^t > bFinal_p^t - aFinal_q^t\\
&OR\\
&2)\ \forall p\neq i, q\neq j:\\
&\quad bFinal_i^t - aFinal_j^t = bFinal_p^t - aFinal_q^t\\
&\quad AND\quad Both\ are\ \Delta_{max}\\
&\quad AND\ \ reward_{ij}^t > reward_{pq}^t
\end{aligned} \tag{12}$$

Auctioneer in the other mechanisms chooses the winner randomly in a condition when there are more than one equal maximum bids (ask). Our novel reward function causes best winner selection by the auctioneer in each auction round. This reward function updates a reward matrix in each round of the auction which its rows are user's ID and columns are resource's ID. Proposed reward function in the auctioneer is determined by (13).

$$reward_{ij}^t = reward_{ij}^{t-1} + \alpha . rewrad_{ij}^{t-1}$$

$$\alpha=\left[\frac{\left(\frac{1}{dist_{ij}}\right)^m}{\sum\left(\frac{1}{dist_{ij}}\right)^m}\right].\left(\frac{var_i^{t-1}-var_i^t}{var_i^t}+\frac{var_j^{t-1}-var_j^t}{var_j^t}\right) \tag{13}$$

In which $reward_{ij}^{t-1}$ is the last reward value for i'th user and j'th resource. $reward_{ij}^t$ is the current reward value. Coefficient α is the reinforcement parameter for this reward function and consists of two coefficients itself. $\left[\frac{\left(\frac{1}{dist_{ij}}\right)^m}{\sum\left(\frac{1}{dist_{ij}}\right)^m}\right]$ is the proportion of communication cost which is considered in our work as shortest path between resource consumers and resource owners. Whatever a distance is longer this coefficient is got decreased hence the final reward value is scaled down. $var_i^{t-1} - var_i^t$ is the difference between variance of HL_i last bids of i'th user except its current bid value and variance of its HL_i last bids including its current bid value. As the same for resources, $var_j^{t-1} - var_j^t$ is the difference between variance of HL_j last asks of j'th resource except its current ask value and variance of its HL_j last asks including its current ask value. These two proportions can be positive or negative in each auction round.

In each round, winners are selected by the auctioneer based on (12) and (13). Final price which has to be exchanged between winners is evaluated by (2) in form of G$. Auction rounds are continuously accomplished until all jobs are executed by resources.

5. Practical Evaluation Results

In order to study the efficiency of our algorithm we simulated our algorithm using gridsim a putative java based simulation toolkit [12]. GridSim allows modeling and simulation of entities in parallel and distributed computing systems such as users, applications, resources, and resource brokers/schedulers for design and evaluation of scheduling algorithms. In fact there is no benchmark in the field of market based grid scheduling hence parameter values in our simulated system are set based on different options of the simulator besides they are based on the other related work's experimental evaluations.

In our simulation each job has a random length between [10000,15000] in form of MI. Our resource power is set as a random value between [10,20] in form of MIPS. In our simulation, number of users, number of resources, number of jobs in each user, budget value and history length of each user and resource are given as the input parameters to the simulator. Also for the auctioneer, the reward matrix is initialized with a positive large value and distance matrix is initialized with random integer values between [1, 10].

In our first study we compare the number of auction rounds in twelve different experimental evaluation conditions. In each condition, number of users, number of jobs and number of resources are different. As it's illustrated in figure 1 our mechanism has the minimum auction rounds in comparison with the other mechanisms. The longest auction round belongs to Fixed Mark up mechanism. It is derived from this experiment that our mechanism is doing resource allocation faster than the other mechanisms.

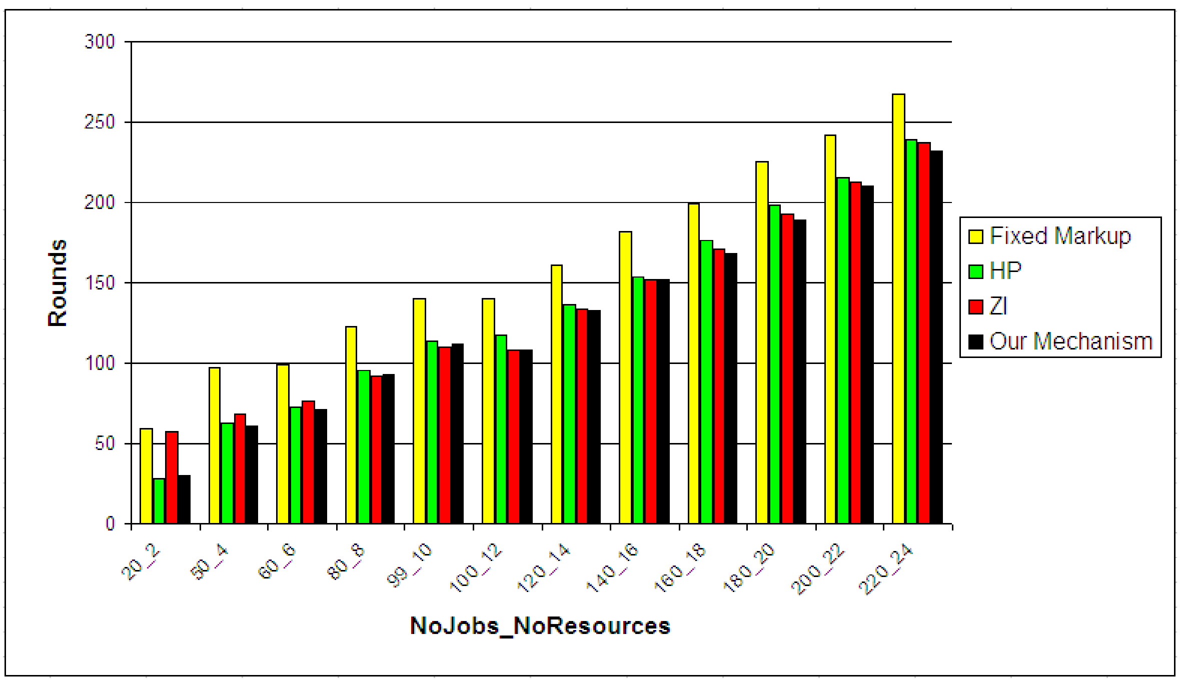

Figure 1. Number of auction rounds

For the second and third evaluation, we study the price fluctuation in each mechanism and as it is shown in figure 2 and 3 that our mechanism has the minimum fluctuation in comparison with the other mechanisms. It is demonstrated in figure 2 and 3 that our algorithm finally reaches to the market equilibrium interval more acceptably. In this interval the price is not voided and both the buyers and sellers have the maximum utility. ZI mechanism has the worst price fluctuation because of its specific random behaviour in bidding and asking in every auction round.

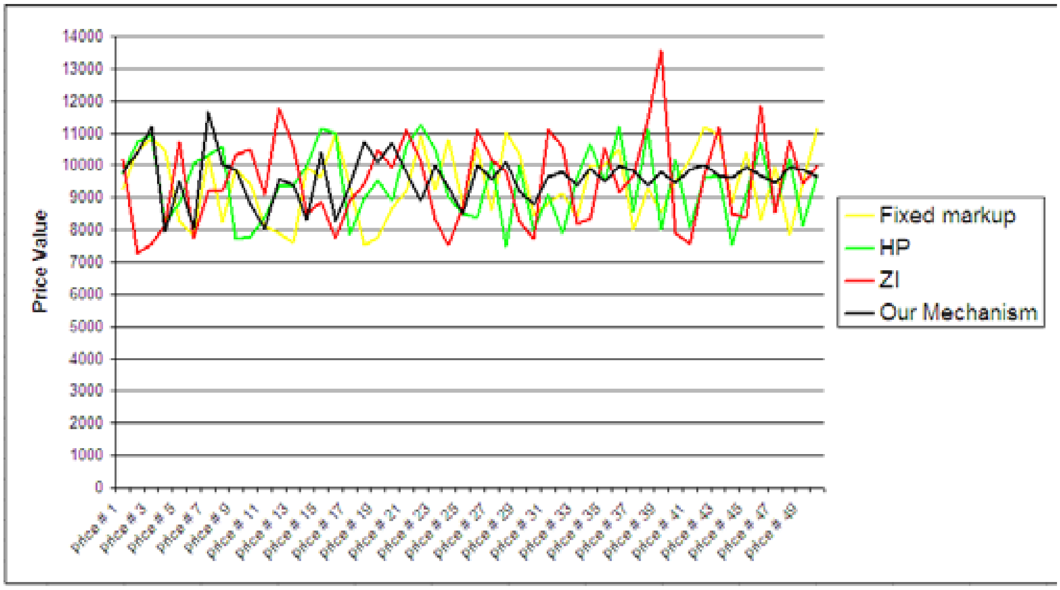

Figure 2. Price fluctuation of our mechanism in comparison with other mechanisms in an specific experimental condition (50 jobs in 5 users and 4 resources)

As it is demonstrated in table 1, the proportion of user and resource utility for these four mechanisms are studied as the next experiment. As it is clear, Fixed Mark up mechanism has the best user utility and the worst resource utility. HP mechanism has the best resource utility and the worst user utility. It is inferred from this evaluation that ZI and our mechanism are almost has the same utility for users and resources and both are admissible.

The last experimental evaluation we studied is incentive proportion of resources in participating in auctions. As it is shown in figure 4, results confirm that proposed mechanism is more acceptable than HP and Fixed Mark up mechanism. ZI has the closest result to our mechanism. Our proposed mechanism is generally makes incentives in resources to participate in the auction more than the other mechanisms.

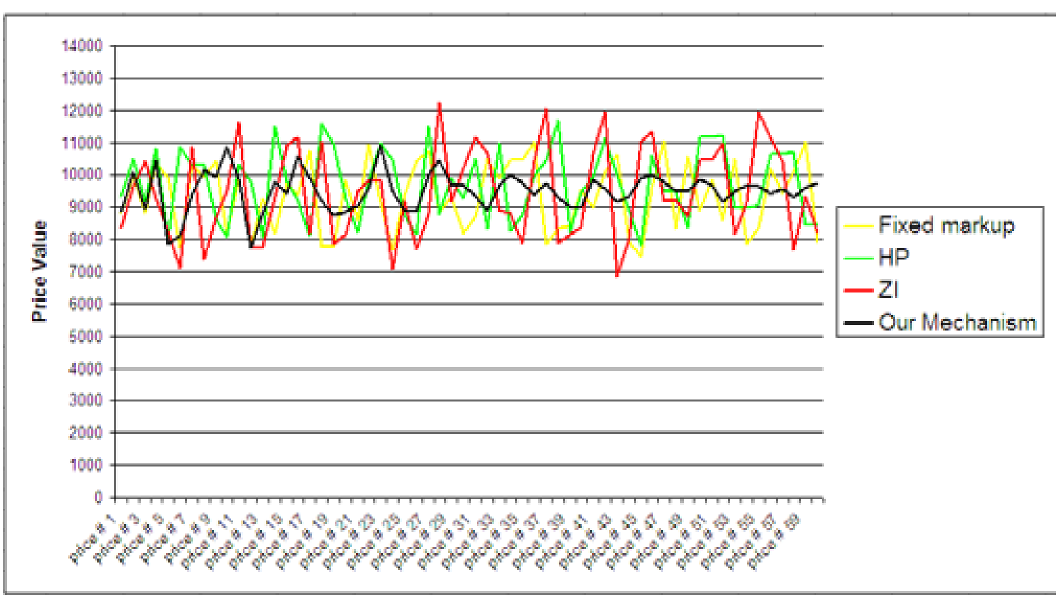

Figure 3. Price fluctuation of our mechanism in comparison with other mechanisms in an specific experimental condition (80 jobs in 8 users and 8 resources)

	Fixed Markup		HP		ZI		Our Mechanism	
	resource utility	user utility	resource utility	user utility	resource utility	user utility	resource utility	user utility
4_5_2	91.1505	3.88025	71.151	1.22475	91.69825	3.497	95.3125	6.93405
5_10_4	94.004	5.9688	94.035	6.1442	95.285	4.7138	95.4364	5.3638
6_10_6	93.19694544	10.73733333	97.62238229	2.415833333	95.89185674	4.1455	95.92513744	5.469633333
8_10_8	93.31	6.55325	97.269625	2.728375	95.086	4.91	95.9311021	5.01905746
9_11_10	93.22588889	11.36022222	97.21177778	2.775222222	95.79211111	4.204777778	95.93365787	5.603898889
10_10_12	93.2497	13.3416	97.3475	2.6515	95.9565	3.9692	95.13833	5.5581
12_10_14	88.65408333	6.7285	98.62666667	1.387	86.9545	4.7095	95.81318	5.032833333
14_10_16	93.16221429	9.054214286	98.27364286	1.725214286	95.16142857	4.837214286	95.06107143	7.412852857
16_10_18	93.322875	6.7325	97.3479375	2.633625	95.5995625	4.3928125	95.0214	6.2117875
18_10_20	93.05811111	6.974555556	97.37916667	2.711722222	95.43355556	4.965888889	95.00905556	6.036944444
20_10_22	93.1750661	7.0393382	97.6710382	1.9340748	95.12288671	4.40711	95.01633	6.1289491
22_10_24	92.86234	7.001659	97.58301	2.1926719	95.2301639	4.3856222	95.819137	6.09621033

Table 1. User and resource utility for each mechanism, evaluated by 12 different experimental conditions which is shown in the left most column of the table (organized with number of users_ number of jobs in each user and number of resources)

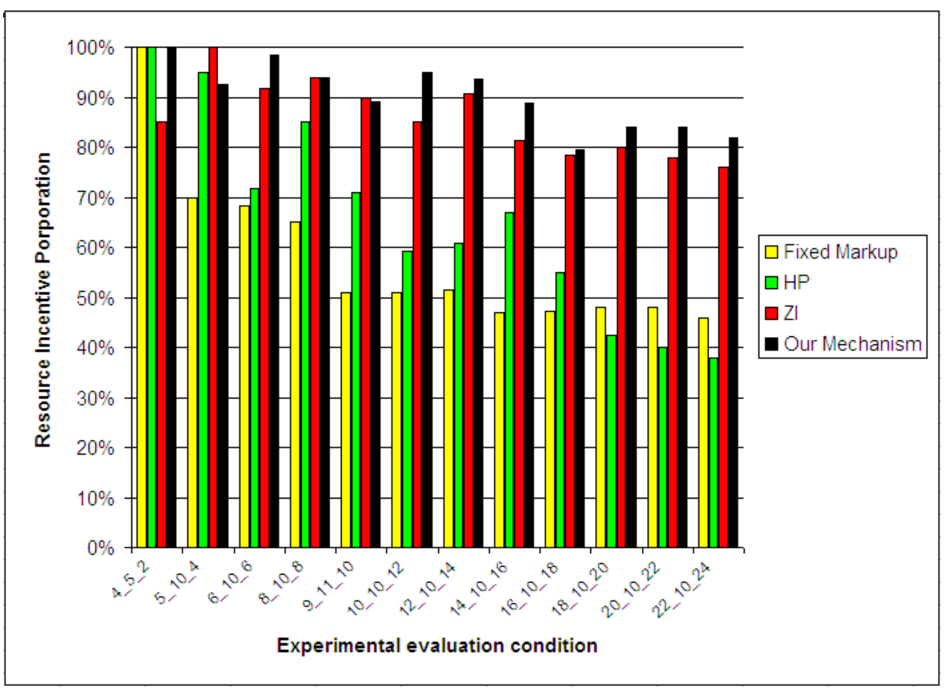

Figure 4. Incentive proportion of resources in participating in the auction rounds

5. Conclusion

Market based grid resource allocation algorithms must make a condition for participants to have enough incentives to be in the auction rounds and play in grid. In this approach we illustrated a novel learning algorithm for resource allocation in grid, based on double auctions. In our proposed algorithm we introduced two pricing mechanism for buyers and sellers which is based on learning concepts. Each user tries to make next bid based on its last bid value, history of its prior bidding, result of each auction round by these former bids and finally by considering the deadline time of its jobs. As the same each resource owner tries to make next ask based on its last ask value, history of its prior asking, result of each auction round by these former asks and at last by mentioning the probability of failure which is calculated by a Poisson function. A novel reward function is introduced for the auctioneer that is used to choose the winner in each round more precisely and accurately. In this function, cost communication and fluctuation of offers for each user and resource is considered. In each round a reward matrix is updated by this reward function. Experimental evaluation results certify that number of auction rounds and more importantly, price fluctuation in our proposed algorithm is clearly more decreased than the other mechanisms. Also market equilibrium is reached in a specific appropriate interval. It is also demonstrated that utility of both resources and users in our mechanism are acceptable.

Refrences

[1] Attanasio, A., Ghiani, G., Grandinetti, L., Guerriero, F. (2006). Auction algorithms for decentralized parallel machine scheduling, Parallel Computing 32. 701–709.

[2] Pourebrahimi, B., Bertels, Koen. (2008). Adaptation to Dynamic Resource Availability in Ad-hoc Grids through a Learning Mechanism, *In:* 11'th IEEE international conference on computational science and technology, (2008)

[3] Buyya, R. (2002). Economic-based Distributed Resource Management and Scheduling for Grid Computing. PhD Thesis: Monash University, Melbourne, Australia.

[4] Herbert, D. , Turton, P. (2006). Simple Strategies of Agents in an Evolving Auction Model.

[5] Gode, D.K, Sunder , S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rational. Political Economy 101(1) 119-137.

[6] Xhafa, F., Carretero, J., Abraham, A. (2007). Genetic Algorithm Based Schedulers for Grid Computing Systems, Innovative Computing, Information and Control 3 (5) 1053-1071.

[7] Friedman, J. Rust. (1993). The Double Auction Market: Institutions, Theories, and Evidence". Addison-Wesley.

[8] Gomoluch, J. Schroeder, M. "Market-based resource allocation for grid computing: A model and simulation", In Proc. of the 1st International Workshop on Middleware for Grid Computing, (2003) pages 211–218, June .

[9] Xiao, L., Zhu, Y. M., Ni, L., Xu, Z. (2008). Incentive-Based Scheduling for Market-Like Computational Grids, IEEE Transactions on parallel and distributed computing 19 (7) 903-913

[10] Yokoo, M., Sakurai, Y., Matsubara, S (2001). Robust double auction protocol against false-name bids, *In:* Proc. of the 21'st IEEE International Conference on Distributed Computing Systems, p. 137–145.

[11] Huang, P., Peng, H., Lin, Piyuan., Li , X. (2008). Macroeconomics Based Grid Resource Allocation, Future Generation Computer Systems 24. 694–700.

[12] Buyya, R., Murshed, M(2002). GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing, Concurrency Computat: Pract. Exper.14. 1175–1220

[13] McAfee, R.P (1992). A dominant strategy double auction", Economic Theory, 56:434–450, 1992.

[14] Gjerstad, S., Dickhaut , J. (1998). Price Formation in Double Auction Games and Economic Behavior. 22: 1-29.

[15] Reddy, S. R., Gupta, A (2006). Auction Based Resource Allocation in Grids, Springer-Verlag Berlin Heidelberg .

[16] Song, S., Hwang, K., Kwok , Y.Kwong (2005). Trusted Grid Computing with Security Binding and Trust Integration. Grid Computing.

[17] Zue, Tan (2007). Market based grid resource allocation using a stable continouus double auction. Phd thesis Manchester University.

[18] Kant, U., Grosu, D (2007). Auction-Based Resource Allocation Protocols in Grids, *In:* 16th International Conference on Parallel and Distributed perspective and its simulation, Future Generation Computer Systems 23. 837–845.

Author Biographies

Saeed Parsa: Dr. Saeed Parsa received his graduate degree in computer science form University of Sharif and received his post graduate and doctoral degrees form University of Salford, Manchester United kingdom. He has been Faculty of Computer Engineering, Iran University of Science and Technology, Since October 1993. Presently he is Associate Professor and head of software group in Iran University of Science and Technology. His interests include Compiler Optimization & Programming Languages, Reverse Engineering & Re factoring, Grid and Distributed Computing, Formal Methods, Algorithms (Heuristic & Deterministic), Software Engineering.

Amin Shokri: Amin Shokri received his post graduate degree form Iran University of Science and Technology. He has had about 3 years of experience in teaching and research. Presently he is writing a book about automata and languages concepts. He was in charge of Batchular of Science global examination in field of computer science last year. His interests include Game Theory, Market concepts, Algorithm Design, Scheduling, Grid computing.