LETTER

A QoS-Enabled Double Auction Protocol for the Service Grid

Zhan GAO^{†a)}, Nonmember and Siwei LUO[†], Member

SUMMARY Traditional double auction protocols only concern the price information of participants without considering their QoS requirements, which makes them unsuitable for the service grid. In this paper we first introduce QoS information into double auction to present the QoS-enabled Double Auction Protocol (QDAP). QDAP tries to assign the asks which have the least candidate bids firstly to make more participants trade and provides QoS guarantee at the same time. Simulation experiments have been performed to compare QDAP with two traditional double auction protocols and the result shows that QDAP is more suitable for the service grid.

key words: resource allocation, QoS, double auction, service grid

1. Introduction

Grid computing aims at enabling large scale resource sharing and integration seamlessly. However, due to the features of grid resources, such as geographical distribution, heterogeneity and site autonomy, it is challenging to manage grid resources. With the emergence of the grid economy, economic-based resource allocation model is proposed. The mechanisms proposed for use include auctions, commodity markets, tenders and posted price [1]. Auction-based resource allocation has attracted much attention because it requires little global information, has decentralized structure and is easy to implement. And the double auction is considered suitable for allocating grid resources because it can truly reflect the fluctuation of demand or supply and determines the price of goods dynamically. It was preferred by many projects such as Tycoon [2] and JaWS [3]. But these projects only adopted the basic auction methods instead of researching them deeply. In [4], CDA, Preston-McAfee double auction and threshold price double auction were evaluated in allocating grid computing resources. [5] expanded CDA to utility-based resource allocation. Furthermore, double auctions are also used in scheduling grid workflows [6] and resource co-allocation [7].

In recent years, the Grid technology has been developing to the “service” direction and OGSA (Open Grid Services Architecture) has been presented as the standard grid architecture. In OGSA, grid resources are abstracted as services with unified and standard interfaces. Therefore, we are now faced with how to allocate grid services efficiently. Unfortunately, existing auction protocols cannot be applied to grid service allocation directly for they do not promise

QoS (Quality of Service) guarantee, which is an important criterion for the service grid. Under those protocols, the auctioneers decide the successful traders only by the price information and neither the QoS requirements of the buyers nor the QoS provided by the sellers are considered. Thus a buyer may buy a service that cannot meet his or her QoS requirement, which is unreasonable.

To our best knowledge, no previous research has introduced QoS into double auction protocol. In this paper, we put forward QDAP (QoS-enabled double auction protocol) to allocate grid services.

2. Auction Model for the Service Grid

In the service grid, entities that can be traded are different kinds of grid services, such as storage service and computing service. The auction market is constituted by three components, Seller Agent (SA), who works on behalf of the sellers, Buyer Agent (BA), who works on behalf of the buyers and Auction Agent (AA), who manages the auction market. A buyer i will send to its BA his or her bid, which is expressed by the BA in the form of $b_i = (Q_b, p)$, where $b_i.Q_b$ and $b_i.p$ are the QoS requirements and the bidding price for per unit of the grid service he or she wants to buy respectively. $b_i.Q_b$ consists of QoS attribute ranges, which are combined by logical operators. For example, a Q_b for a grid service may have the form of $((Bandwidth \geq 512 Kbps) \wedge (Memory \geq 30 MB))$. Similarly, an ask j has the form of $a_j = (Q_a, p)$. $a_j.Q_a$ describes the QoS of the service seller j can supply and $a_j.p$ is the asking price for per unit of the service. After receiving the trading requests from SAs and BAs, AA makes use of some auction protocol to decide the buyers and sellers who can trade and the trading price.

3. QoS-Enabled Double Auction Protocol

We put forward QDAP to allocate grid services with QoS constraints, which works in the following 3 phases.

Phase 1: Bidding

- 1.1 $BA_i, i = 1, 2, \dots, m$, sends b_i to AA.
- 1.2 AA receives b_i and puts it into the bid set, $bSet$.
- 1.3 $SA_j, j = 1, 2, \dots, n$, sends a_j to AA.
- 1.4 AA receives a_j and puts it into the ask set, $aSet$.

Phase 2: Preprocessing

- 2.1 Set $candidateAsk_i$ and $candidateBid_j$ to be $\emptyset$ respectively, where $1 \leq i \leq m, 1 \leq j \leq n$.

Manuscript received August 31, 2009.

Manuscript revised December 1, 2009.

[†]The authors are with the School of Computer and Information Technology, Beijing Jiaotong University, China.

a) E-mail: bjtugaozhan@gmail.com

DOI: 10.1587/transinf.E93.D.1297

- 2.2 for all the bids, b_i , in $bSet$
- 2.3 for all the asks, a_j , in $aSet$
- 2.4 If $a_j.Q_a$ can satisfy the QoS requirements of buyer i , $b_i.Q_b$, put a_j into $candidateAsk_i$ and b_i into $candidateBid_j$.

Phase 3: Completion

- 3.1 **while**($aSet \neq \emptyset$ and $bSet \neq \emptyset$)
- 3.2 Set $currentSet$ to be $\emptyset$.
- 3.3 **for** all the bids in $bSet$
- 3.4 Find bid b_i with $|candidateAsk_i| = 0$ and delete it from $bSet$.
- 3.5 **endfor**
- 3.6 **for** all the asks in $aSet$
- 3.7 Find ask a_i with $|candidateBid_i| = 0$ and delete it from $aSet$.
- 3.8 Find ask a_j who has the smallest $|candidateBid_j|$ and put it into $currentSet$.
- 3.9 **endfor**
- 3.10 Sort all the asks in $currentSet$ in the non-decreasing order of $a_j.p$ ($a_j \in currentSet$): $a_{\theta(1)}.p \leq a_{\theta(2)}.p \leq \dots \leq a_{\theta(|currentSet|)}.p$, where θ is the permutations defining the order statistics above.
- 3.11 Let a_h be one element of $currentSet$; sort all the bids, b_i , in $candidateBid_h$ in the non-increasing order of $b_i.p$ ($b_i \in candidateBid_h$): $b_{\pi(1)}.p \geq b_{\pi(2)}.p \geq \dots \geq b_{\pi(|candidateBid_h|)}.p$, where π is the permutations defining the order statistics above.
- 3.12 **if** $b_{\pi(|candidateBid_h|)}.p \geq a_{\theta(|currentSet|)}.p$
- 3.13 Set k to be the smaller one between $|currentSet|$ and $|candidateBid_h|$.
- 3.14 Determine the trading price t , $t = \frac{1}{2}(b_{\pi(k)}.p + a_{\theta(k)}.p)$.
- 3.15 goto 3.21
- 3.16 **endif**
- 3.17 **esle** find k so that $b_{\pi(k)}.p \geq a_{\theta(k)}.p$ and $b_{\pi(k+1)}.p < a_{\theta(k+1)}.p$.
- 3.18 Determine the trading price t , $t = \frac{1}{2}(b_{\pi(k+1)}.p + a_{\theta(k+1)}.p)$.
- 3.19 **endelse**
- 3.20 **if** $a_{\theta(k)}.p \leq t \leq b_{\pi(k)}.p$
- 3.21 Notify $BA_{\pi(i)}$ and $SA_{\theta(i)}$, $1 \leq i \leq k$, that they can trade at price t . Delete $b_{\pi(i)}$ and $a_{\theta(i)}$ from $bSet$ and $aSet$ respectively.
- 3.22 for $j=1$ to $|bSet|$, if $a_{\theta(i)} \in candidateAsk_j$, $i = 1, 2, \dots, k$, delete $a_{\theta(i)}$ from $candidateAsk_j$.
- 3.23 for $j=1$ to $|aSet|$, if $b_{\pi(i)} \in candidateBid_j$, $i = 1, 2, \dots, k$, delete $b_{\pi(i)}$ from $candidateBid_j$.
- 3.24 goto 3.1
- 3.25 **endif**
- 3.26 **elseif** $t < a_{\theta(k)}.p$ or $t > b_{\pi(k)}.p$
- 3.27 Notify $BA_{\pi(i)}$ and $SA_{\theta(i)}$, $1 \leq i \leq k-1$, that they can trade. Each SA gets $a_{\theta(k)}.p$ and each BA pays $b_{\pi(k)}.p$. Delete $b_{\pi(i)}$ and $a_{\theta(i)}$ from $bSet$ and $aSet$ respectively.
- 3.28 for $j=1$ to $|bSet|$, if $a_{\theta(i)} \in candidateAsk_j$, $i = 1, 2, \dots, k-1$, delete $a_{\theta(i)}$ from $candidateAsk_j$.
- 3.29 for $j=1$ to $|aSet|$, if $b_{\pi(i)} \in candidateBid_j$, $i = 1, 2, \dots, k-1$, delete $b_{\pi(i)}$ from $candidateBid_j$.

- 3.30 goto 3.1
- 3.31 **endelse**
- 3.32 **endwhile**
- 3.33 Send reject message to the SAs and BAs that do not trade.

As for a bid b_i , $candidateAsk_i$ means all the candidate asks that can meet its QoS requirement. As for an ask a_j , $candidateBid_j$ means all the candidate bids whose requirements are below the QoS it can provide. In phase 3, we always try to auction the asks with the least candidate bids firstly, which constitute $currentSet$. Because the QoS they can provide is relatively low, if some of their candidate bids fail to trade these buyers still have the opportunity to trade with other sellers providing a higher QoS, which may result in more successful traders. As for a certain $currentSet$ and its $candidateBid$, the double auction mechanism developed by McAfee [8] is adopted. In line 3.11, we only consider the candidate bids of one element in $currentSet$, for it is easy to prove that if two asks have the same number of candidate bids, their candidate bids are identical. If the condition in line 3.26 holds, $SA_{\theta(i)}$ gets $a_{\theta(k)}.p$ and $BA_{\pi(i)}$ pays $b_{\pi(k)}.p$, $i = 1, 2, \dots, k-1$. There is a surplus of $(k-1)(b_{\pi(k)}.p - a_{\theta(k)}.p)$ for trading per unit of service. We assume here that this surplus is kept by the market manager. Note that if $|currentSet| = 1$, the auction becomes the first-price sealed-bid auction; as for the $candidateBid$ of a $currentSet$, if $|candidateBid| = 1$, the auction degrades to be the reverse auction.

4. Experiments and Analysis

To evaluate our protocol, simulation experiments have been conducted. QoS may relate to availability, reliability, security and capacity of services. Here we only consider the service capacity and assume bandwidth to be the sole factor. We assume there are 2000 bids and asks respectively in the market and the bandwidths supplied by the servers or required by the users are uniformly distributed between 0 (exclusive) and 1024 (Kbps). For simplicity, we use 5 QoS levels to represent different ranges of bandwidth, though in QDAP it is not necessary to divide the services into different QoS levels. Let the ranges of 0–64 (Kbps), 64–128 (Kbps), 128–256 (Kbps), 256–512 (Kbps) and 512–1024 (Kbps) be level 1, level 2, level 3, level 4 and level 5 respectively and a buyer requires that the QoS level of the service should not be lower than the QoS level he or her requires. Let the QoS level required or supplied by an auction participant be L , ($1 \leq L \leq 5$). The bidding or asking price is generated in the following three ways. Case I: The price is uniformly distributed between $10(L-1)$ (exclusive) and 50 (inclusive). This case represents a reasonable market because generally speaking, the service with a higher QoS will also demand a higher price. Case II: The price is uniformly distributed between 0 (exclusive) and 50 (inclusive). In this case the participant has no information about the market and the bidding or asking price is generated arbitrarily. Case III: The price

is uniformly distributed between $10(5 - L)$ (exclusive) and 50 (inclusive). This case represents an unreasonable market where the service with a higher QoS demands a lower price.

We regard the number of successful sellers as the utilization of the grid services and assume each buyer bids for one unit of the service. Then we apply the Preston-McAfee Double Auction (PMDA), CDA and QDAP to the generated asks and bids respectively to figure out the service utilization and the corresponding budget spent.

4.1 Experiment Procedures

The experiment steps are as follows:

1. Generate 2000 bids and asks respectively in the three ways as we mentioned before.
2. Use our auction protocol to decide the service utilization and the budget spent.
3. Use the Preston-McAfee Double Auction (PMDA) to decide the service utilization and the budget spent without considering the QoS constraint. Then figure out the minimum and maximum service utilization under the QoS constraint.
4. Use CDA protocol to decide the service utilization and budget spent without considering the QoS constraint. Then figure out the service utilization under the QoS constraint.
5. Repeat step 1 to 4 ten times, calculate the average results and make comparison among the three auction protocols.

In step 3 we figure out the maximum and minimum service utilizations under the QoS constraints using the following method: Among all the successful bids and asks determined by PMDA, as for each bid find the ask whose QoS level is higher by the least amount and treat the two as the actual successful pair under the QoS constraint. Repeat this process until no more pair can be found. The number of successful pair is the maximum utilization. Then we find the pairs with each ask's QoS level higher than that of the corresponding bid by the most amount. And we treat the number of all the found pairs as the minimum utilization. In step 4, we assume the time sequence of all the bids and asks' appearing in the auction market is equivalent to the sequence in which they are generated. Traditional CDA protocol matches the earliest bid with the earliest ask whose asking price is lower. But under the QoS constraint, as for a matched pair only when the QoS level of the ask is higher than that of the bid, the two can actually trade.

4.2 Experiment Results

We have made a comparison among the service utilizations under the three protocols. The result is shown in Fig. 1. We can see that in Case I when the QoS constraint is not considered, PMDA and CDA will result in a larger utilization compared with QDAP. But when QoS constraint is involved, the utilizations of both PMDA and

Fig. 1 The service utilizations under different protocols. (PMDA_QoS_min and PMDA_QoS_max mean the maximum and minimum service utilizations of PMDA under the QoS constraint respectively and CDA_QoS is the service utilization of CDA under the QoS constraint.)

CDA decrease a lot. As for CDA it drops by 17.4% while as for PMDA it drops by a minimum of 61.9% (refer to PMDA_QoS_min) and a maximum of 51.3% (refer to PMDA_QoS_max). PMDA_QoS_max is much less than the utilization of QDAP but CDA_QoS still results in the largest utilization. PMDA and CDA are not suitable for the service grid for they cannot promise QoS guarantee. 17.4% successful traders will actually fail to trade under CDA due to the QoS constraint while a maximum of 61.9% under PMDA. QDAP takes into account both the service' price and QoS level thus will not cause such problem. In case II and case III the service utilizations of PMDA and CDA differ little from that of case I but the service utilization of QDAP, PMDA_QoS_min, PMDA_QoS_max and CDA_QoS have increased. Especially in case III, they increase by as much as 13.3%, 84.9%, 72.3% and 13.8% respectively. We can see that though the service utilization of QDAP is still larger than PMDA_QoS_max, this advantage decreases as the mark becomes unreasonable. This is because in an unreasonable market if a service's price is lower than a user's bidding price, there is a great likelihood that the former's QoS level is higher than the QoS level the latter required.

We have also observed the budget spent under the three protocols in case I and case II. Here we only consider the deal which meets both the price and QoS requirement. In case I according to PMDA all the successful traders trade at a uniform price (38.8 in our experiment) no matter what the QoS level is. This is unfair for those sellers who provide services of higher quality. According to QDAP, services with different QoS levels may be traded at different prices. Specifically in our experiment the budget spent in the increasing order of the QoS level are 26.1, 29.8, 35.8, 40.0 and 45.2 respectively. We can see that the service with a higher QoS level is traded at a higher price, which is reasonable in the real market. In CDA, transactions happen between every two successful participants and each service may be traded at a different price. The budget spent is shown in Table 1. Though CDA will result in the maximum service utilization (refer to Fig. 1), it is at the expense of unnecessarily high price volatility, i.e. the trading price fluctuates too much. In

Table 1 The budget spent of CDA.

QoS level	min	max	average	standard deviation	case
1	1.35	44.27	22.81	181.81	I
2	11.17	45.19	28.18	119.01	I
3	20.61	45.14	32.87	59.75	I
4	31.11	47.68	39.40	29.75	I
5	40.38	49.86	45.12	11.49	I
1	1.02	49.86	22.35	197.32	II
2	0.87	49.28	20.62	176.46	II
3	0.37	48.72	33.76	212.38	II
4	1.09	48.86	24.17	169.71	II
5	1.64	49.47	28.01	204.86	II

Table 1, when the QoS level of services is 1 the maximum budget is 31 times more than the minimum and the standard deviation reaches 181.81. With the QoS level going higher, both the deviation and the distance between the minimum and maximum budget decrease because the varying range of both the bidding price and asking price becomes smaller. When the QoS level is 5, though the price's varying range is as narrow as 25%, the maximum budget is still 1.23 times as many as the minimum. And the standard deviation (11.49) is relatively high compared with PMDA and QDAP (0). This unsteadiness of trading price will result in lower market efficiency and cause dissatisfaction among bidders [9]. In case II the standard deviation of the budget spent for services of the same QoS level under CDA becomes much larger than that in case I due to the increase of the trading price's varying range. However, this will not happen under QDAP and PMDA, where services of the same QoS level will be traded at the same price.

5. Conclusion

Double auctions have been suggested to allocate grid resources by a lot of research and several auction protocols have also been proposed. However, these protocols decide the successful traders only by their bidding prices and asking prices, without caring whether or not the bidders' QoS requirements can be satisfied, and thus they cannot be ap-

plied to the service grid. In this paper we first introduce QoS information into double auction to present QDAP, which can promise QoS guarantee for the bidders. In order to make more participants trade, QDAP tries to assign the asks which have the least candidate bids first. Simulation experiments have also been performed to compare QDAP with two tradition double auction protocols, and the result shows that PMDA and CDA will match the wrong trading pairs due to the QoS constraint, which will be avoided by QDAP. QDAP also outperforms the other two protocols in either price volatility or service utilization, and thus it is more suitable for the service grid.

References

- [1] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in grid computing," *Concurrency Computat.: Pract. Exper.*, no.14, pp.1507–1542, 2002.
- [2] K. Lai, L. Rasmusson, E. Adar, S. Sorkin, L. Zhang, and B.A. Huberman, "Tycoon: An implementation of a distributed market-based resource allocation system," Tech. Rep., HP Labs, USA, Dec. 2004.
- [3] S. Lalis and A. Karipidis, "Jaws: An open market-based framework for distributed computing over the internet," *Proc. 1st IEEE/ACM International Workshop on Grid Computing*, Bangalore, India, 2000.
- [4] U. Kant and D. Grosu, "Double auction protocols for resource allocation in grids," *Proc. International Conference on Information Technology: Coding and Computing*, 2005.
- [5] C. Satayapiwat, R. Egawa, H. Takizawa, and H. Kobayashi, "A utility-based double auction mechanism for efficient grid resource allocation," *Proc. International Symposium on Parallel and Distributed Processing with Applications*, pp.252–260, 2008.
- [6] M. Wiecek, S. Podlipnig, R. Prodan, and T. Fahringer, "Applying double auctions for scheduling of scientific workflows on the grid," *Proc. International Conference on High Performance Computing, Networking, Storage and Analysis*, 2008.
- [7] Y. Liu and H. He, "Multi-unit combinatorial auction based grid resource co-allocation approach," *Proc. 3rd International Conference on Semantics, Knowledge, and Grid*, 2007.
- [8] R.P. McAfee, "A dominant strategy double auction," *J. Economic Theory*, vol.56, no.2, pp.434–450, 1992.
- [9] Z. Tan, *Market-based Grid Resource Allocation Using a Stable Continuous Double Auction*, Ph.D. thesis, The University of Manchester, Manchester, UK, 2007.