Supporting Co-allocation in an Auctioning-based Resource Allocator for Grid Systems

Chunming Chen, Muthucumaru Maheswaran, and Michel Toulouse

University of Manitoba and TR*Labs*
Winnipeg, Manitoba
Canada
E-mail: {cmchen, maheswar, toulouse}@cs.umanitoba.ca

Abstract

In this paper, we present the overall design for an auctioning based resource trading/acquiring system that can be deployed in wide-area computing systems such as Grid systems. Selecting the winning bids is one of the core issues in any system that utilizes the auctioning paradigm. We identify the unique aspects of our system that impact the winner selection process. More specifically, the necessity to acquire or trade resources as a bundle (i.e., perform co-allocation) presents a challenge to traditional bidding mechanisms. We present a new bidding mechanism called "co-bids" to address this problem. Two heuristics for winner selection with co-bids are proposed. The performance of the heuristics are examined via simulations.

1. Introduction

The rapid advancements in computer and networking technologies is enabling the Internet to be positioned as a viable communication medium for a variety of applications including business critical ones. These applications demand the network infrastructure to support a multitude of pervasive services. For example, an Internet-based news agency would depend on a content delivery service to deliver the news to all its subscribers. To enable such services, we require the network to provide *resource management* support in a wide-area to acquire, manage, and use resources on an "as needed" basis.

As part of the research work reported here, we are developing a *computational market* (CM) [Che01, ChM01] that enables decentralized resource management in wide-area networked systems. This system uses real-time auctioning as its primary mechanism for resource management. This paper examines the issues in designing bid selection heuristics for the CM. The heuristics focus on selecting the winning bids subject to the constraints that are imposed by the CM.

The CM architecture divides the wide-area network into regions called *local markets* (LMs). An LM has an auction server that manages the resource transactions within the LM. The first level of CM focuses on intra-local market resource flows and the second level of CM handles the inter-local market resource flows. In [ChM01], we assumed that a LM has "uniform" network connectivity among its resources. With this assumption, we were able to treat an LM as a set of varied resources where some resources had multiple instances. However, resources in an actual LM will have non-uniform connectivity among them. This paper relaxes the uniformity assumption within an LM.

In this paper, we develop a resource allocation model for an LM with non-uniform connectivity. We identify the issues unique to this problem and explain why we cannot handle these using traditional multi-unit combinatorial auctioning mechanisms.

Section 2 presents the overall CM system architecture and motivates the need for a fast algorithm for bid selection. Section 3 describes the proposed heuristics for selecting the winning set of bids. The simulation results are examined in Section 4. The related work is reviewed in Section 5.

2. Computation Market

As shown in Figure 1, an LM is made up of an *auction server* (AS), *local brokers* (LBs), *supplier agents* (SAs), and *consumer agents* (CAs). The SAs represent machines that provide resources while the CAs represent clients that are willing to "pay" for the resources. Users may specify the attributes of the desired resource using attribute-value pairs [RaL98]. To make the bid selection process manageable,

This research was supported by a TR*Labs* Scholarship and a Natural Sciences and Engineering Research Council of Canada Research Grant.

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 **IEEE**

we group the resources into a predefined set of hierarchical classes based on the attributes [Che01].

Three major functions are offered by an AS: (a) a virtual market for local clients; (b) trading services in the global market; and (c) selection of the winning bids. The AS earns its revenue from three sources: (a) transaction fees charged on SAs and CAs; (b) profit made by selling cheap remote resources in the local market; and (c) profit made by selling surplus local resources in the global market. The goal of the AS is to maximize the profits from the services it offers. The LBs examine the *bid posts* presented by the SAs on behalf of their resources and determine whether different bid posts can be grouped to form virtual resource clusters. The AS may decide to "unbundle" the virtual clusters created by LBs if it can increase the total revenue.

The CM uses English auctioning. In this type of auctioning, the AS only accepts bids with prices that are higher than the reserved prices. The CAs bid iteratively for an item and each subsequent bid should be higher than the current highest bid. The auction closes at a predefined time and the winning bids are determined by the AS. The period in between two auction closings is called an *auction session*. Once the AS closes an auction session, it needs to decide the winning bids. The time available for selecting the winners is constrained by the start time of the next auction session and the "available" times of the resources. When the auctions are proceeding continuously with periodic closings, as it is here, the bid selections should be performed within a small time interval.

During an auction session, a local market may have higher demand than supply resulting in higher prices whereas another local market may have higher supply than demand resulting in lower prices. If remote resources can be consumed by at least some CAs, then inter-market resource flows should be carried out to reduce the supply and demand disparities. Inter-market resource flows are treated in greater detail in [Che01].

The auctioning process in the CM is not a sequential auction. In a sequential auction, the items are auctioned one at a time and is simple to implement. In CM, the users are interested in buying items in multiple units of the different resources (i.e., the users require bundles of resources that are allocated in an "all or none" manner). Further, the user requests can specify different combinations of the various resources at different quantities. Auctioning processes that handle bidding with these constraints is called the *multi-unit combinatorial auctions* (MUCAs). Although the CM auctioning process is similar to a MUCA process, the bidding and bid selection criteria are different as explained below. Before examining the differences between the traditional MUCA and CM bidding scenarios, we discuss the MUCA process itself to appreciate the complexity of bid selection.

The MUCA is essentially a multiple knapsack problem. Let $R = 1, ..., N$ be the set of N classes of resources (knapsacks) to be auctioned with capacities $(c_1, c_2, ..., c_n)$. Let $B = 1, ..., K$ be the set of K bids. A bid $j \in [1..K]$ is denoted by $b(j) = (W_j, P_j)$, where $W_j = (w_{1j}, w_{2j}, ..., w_{Nj})$ represents the bidding sizes, and $P_j = (p_{1j}, p_{2j}, ..., p_{Nj})$ represents the bidding prices, for resources of class $1, 2, .., N$, respectively. The bid selection process determines a subset U of B that maximizes the revenue obtained by the auction server. The bid selection process in MUCA environments is known to be NP-hard [GoL00, LeS00, San99].

If the resources are completely uniformly interconnected to each other, then CM auctioning can be performed by a MUCA. We elaborate more on this in the next section. In a CM, we have various machines interconnected by different networks. A CA may present a resource request of different forms: (a) a CA may ask for a given number of machines without specifying any connectivity constraints. These machines may be allocated on "any" network without any concern for network-level *quality of service* (QoS). (b) a CA may request two clusters with given number of machines in each cluster and specify the capacity of the virtual network path that should connect the two clusters. The second form requires the CM to ensure that the allocated network resources for the virtual path do indeed connect the two clusters. In the next section, we identify such bids as co-bids.

3. Bid Selection Heuristics

3.1. Overview

Recently, bid selection in MUCA environments has been the focus of several studies [GoL00, LeS00]. Below we explain why bids in a CM system cannot be selected by a straightforward extension of the MUCA algorithms.

Suppose we have a CM system with machine and network resources that are connected together. Let machine resources be grouped into three clusters in total with two `type-A` and one `type-B` clusters. The machines within each (`type-A` and `type-B`) clusters are assumed to have uniform connectivity. One `type-A` cluster is connected to the `type-B` cluster using a `type-1` shared network and the other `type-A` cluster is connected via a `type-2` shared network (i.e., the inter-cluster network connectivity is non-uniform). Let the network of `type-1` have a maximum speed of 100 Mb/s and `type-2` have a maximum speed of 10 Mb/s.

Consider the scenario where a CA presents a bid that requests n_A number of `type-A` machines that are connected by a `type-1` network to n_B number of `type-B` machines. When allocating resources for this request, it is necessary to ensure that the network allocated actually connects the two

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

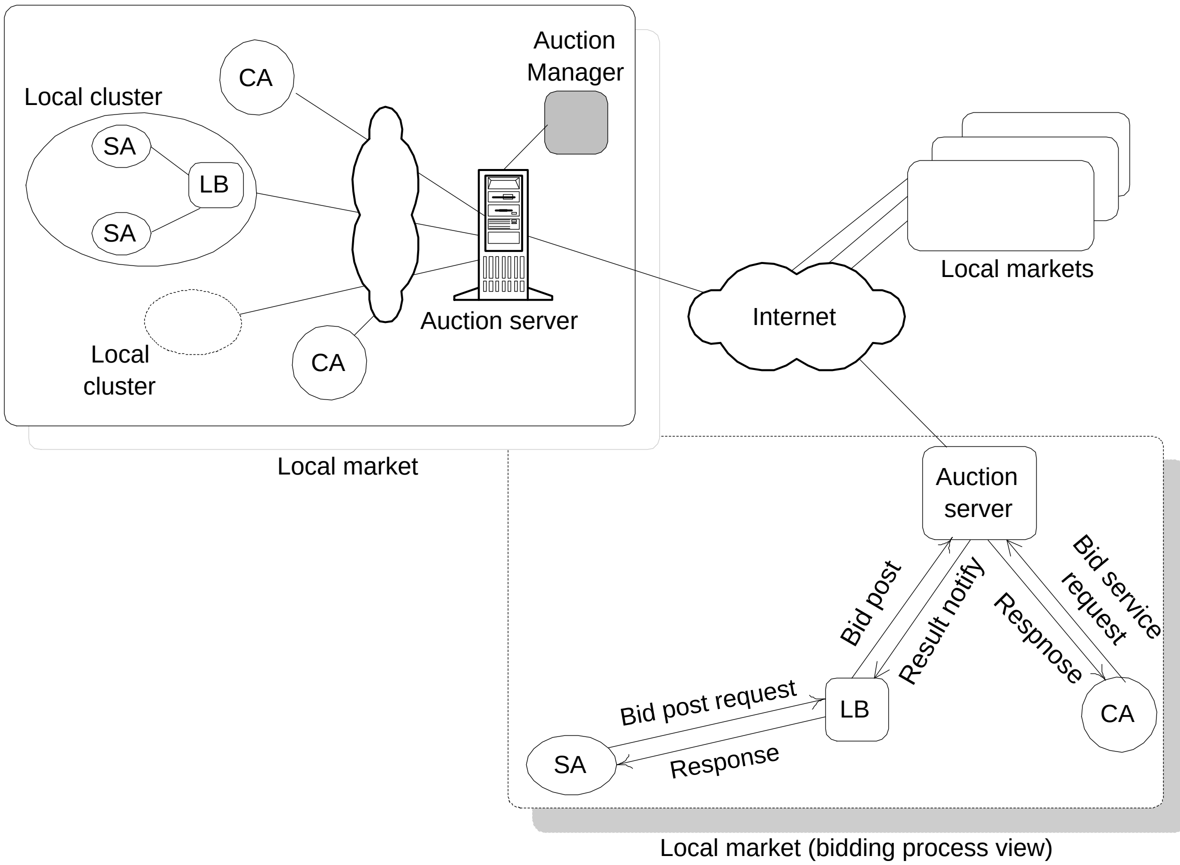

Figure 1. Two-level architecture for a computation market.

pools of machines. If the MUCA algorithms are used in a straightforward manner, the network allocated may connect the "other" `type-A` cluster. This will render the resource allocation unusable. One way of handling this in MUCA is via resource-side renaming. In the above example, we may rename one of the `type-A` clusters as `type-C`. Resource renaming in a system such as the CM is undesirable because resource naming to begin with may be based on the characteristics of the resources. Therefore, resource renaming may hinder the resource discovery process.

This paper presents a "Co-bid" based approach that allows similar resources to be aggregated at the resource side and uses explicit specifications at the bid level to enforce the requirements that the resources should satisfy. In the subsequent subsections, we examine the heuristics that may be used in selecting the winning bids under this scenario.

A bidding agent bids on resources from a particular LB using LB specific bids, from several pre-defined LBs using co-bids, and uses any-bid if it does not have any locational preference. When an any-bid requests multiple resources, there is no guarantee that all allocated resources are co-located. Therefore, any-bids should close at relatively cheaper prices. By using different types of bids, we provide a flexible auctioning mechanism that supports different applications such as multimedia, parallel computing, and data-intensive computing.

3.2. Co-bids First Approach

The basic idea of the *co-bids first approach* (CFA) is to allocate co-bids first because they have more restrictions than other bids. The overall algorithm has the following five steps: (i) Allocate as many co-bids as available resources permit. (ii) Allocate LB specific bids for the remaining resources. (iii) Examine the bids accepted by a LB starting from the bid with the lowest rate (the LB itself is chosen arbitrarily). Select the bid with the lowest rate and undo the "other" bids that are part of the co-bid. Perform more bid allocation (from the unselected pool) to use the resources released by the above undo operation. If the resulting revenue is more than the revenue before the undo, then the undo is retained. Otherwise, the bids removed by the undo are reinstated. (iv) For resources that remain after the completion

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

of (iii), allocate any-bids. (v) Perform (iv) until there are no resources or bids.

3.3. No Preference Approach

As the name implies the *no preference approach* (NPA) does not differentiate among the different types of bids. In fact, the NPA does not consider the "other" bids when allocating co-bids. Therefore, co-bids can receive "inconsistent" allocation. The NPA algorithm does the initial "inconsistent" allocation of bids and then fixes the inconsistency by examining the co-bids that were already allocated. The overall NPA algorithm has the following steps: (i) Allocate as many bids as available resources permit. (ii) Examine the co-bids for consistency. As in CFA, we assume an arbitrary order for the LBs. From the set of co-bids that are allocated (note some of these co-bids may be partially allocated), choose a co-bid X. Let N be the set of bids that should be allocated to complete X and M be the set of bids that are already allocated. Evaluate the total revenue say p when the set M is deallocated and additional bids (not from set N) are allocated. Evaluate the total revenue say q when the set N is allocated by deallocating some bids that do not belong to co-bids that have already been checked for consistency. If $p > q$, then the co-bid is removed from the set. Otherwise, the co-bid is fixed by inserting the set of N bids. (iii) Perform (ii) until all co-bids are checked for consistency.

4. Simulation Results and Discussion

Simulations were performed to compare CFA and NPA. Different metrics were used to examine the performance difference. The metrics include: (a) bid accept ratio that is defined as the ratio of the number of winning bids over the total offered bids, (b) utilization that is defined as the ratio of number of allocated items over the total number of items auctioned, and (c) item accept ratio that is defined as the number of allocated items over the total number of items bidded. Unless specified otherwise we use the following parameters in the simulations discussed below: 20 LBs, 200-300 bids per LB that were randomly generated (see [Che01] for bid generation process), percent of co-bids in the population was varied from 5% to 20% in increments of 5%.

Figure 2 compares the performance of CFA and NPA using the above three metrics. For both approaches, the bid accept ratio and item accept ratio are stable when the percentage of co-bids changes from 5% to 20% with CFP performing slightly better than NPA. The utilization decreases with increasing percentage of co-bids. This is caused by the increase in the number of resource allocation constraints. The utilization is better with CFA than with NPA as the percentage of co-bids increases, indicating that considering co-bids first helps the allocation of more resources.

Figure 3 shows the performance variation with the number of bids per LB. The bid accept ratio and the item accept ratio decrease with the increase of the number of bids. Notice that the two approaches have almost identical curves for the item accept ratio, while the CFA yields a slightly better bid accept ratio when the number of bids increases. The better bid accept ratio suggests that by evaluating the co-bids first, we could potentially include more bids in the solution. However, the item accept ratios are the same, thus suggesting that smaller bids (bids asking for fewer items) have better chance to win with CFA. The utilization is also better for CFA for the same reasons.

The performance variation with the number of LBs is shown in Figure 4. In this comparison, 20% co-bids is used and number of LBs are varied form 5 to 20. The results are consistent with those shown in Figure 3. The *average number of LBs* (avgLBs) over co-bid is a measure used to indicate the avgLBs that a co-bid spans. The larger the avgLBs/co-bid the more constraints for resource allocation, hence, the decrease in utilization. This is demonstrated in the Figure 5 for 10% co-bids. It can be observed that the utilization of CFA decreases only slightly with the increase of AvgLBs/co-bids while the utilization of NPA decreases steeply with the increase of AvgLBs/co-bids, i.e., from 98% to 82% for a variation of AvgLBs/co-bids from 2.3 to 8.5. The increase of AvgLBs/co-bids has insignificant impact on the bid accept and item accept ratios.

Figure 6 shows the same comparison as Figure 5 with percent of co-bids at 20%. In Figure 5, the utilization for CFA was slightly decreasing with increasing AvgLBs/co-bids, which is not the case when the percentage of co-bids increases to 20%. The utilization for the CFA decreases from 98% to 80% when the AvgLBs/co-bids increases from 2.3 to 8.4. Without exception, the utilization for the NPA too decreases with increasing AvgLBs/co-bids and is 2% worse than that of the CFA.

5. Related Work

Due to space limitations, we provide a very brief survey of the existing literature here. An extensive survey of the literature can be found in [Che01].

Combinatorial auction multi-unit search (CAMUS) [LeS00] introduces a branch-and-bound technique for selecting the bids. Their search procedure also uses some heuristics ideas similar to the ones used in this paper. Another work that also uses a branch-and-bound technique to solve the multi-unit combinatorial auction problem is presented in [GoL00]. Their experiments show that the branch-and-bound techniques require both an upper bound for the

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

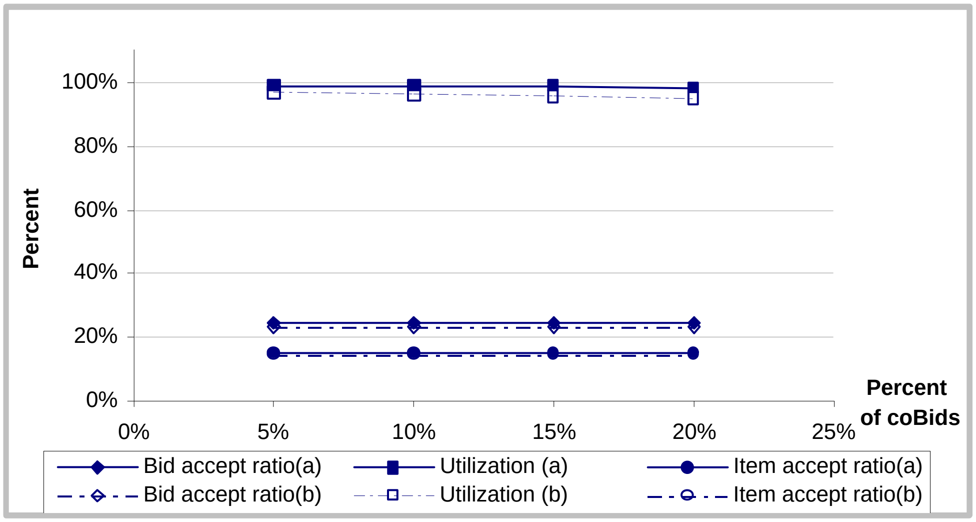

Figure 2. Variation of bid accept ratio, utilization, and item accept ratio with percent of co-bids for (a) CFA and (b) NPA for 20 LBs and [200-300] bids per LB.

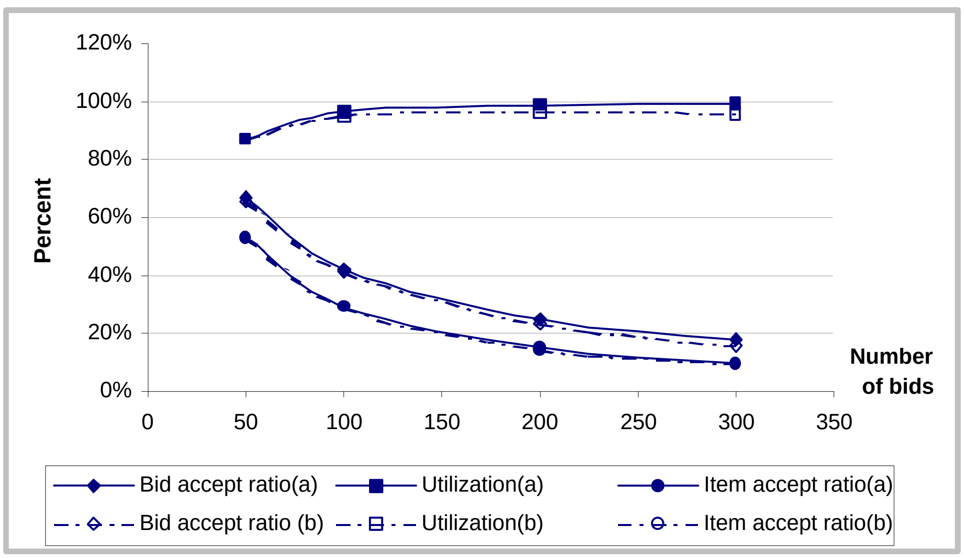

Figure 3. Variation of bid accept ratio, utilization, and item accept ratio with number of bids for (a) CFA and (b) NPA for 20 LBs and percent of co-bids of 10%.

value of best allocation and a good criterion to decide which bids are to be tried first. They suggest making use of average price per unit or an average price per unit related criteria in a branch-and-bound algorithm, which is quite similar to the use of rate to rank the bids in our bid selection algorithm.

A search algorithm for optimally selecting a subset of winning bids is presented in [San99]. The search algorithm consists of four preprocessing steps and the main search. A special bid tree is used in the main search. The bid tree is a binary tree with the bids inserted at the leaves. The algorithm also uses an iterative deepening A* search strategy to speedup the main search. Similar to their algorithm, CM also uses heuristics to solve the winner determination problem in combinatorial auctions. However, our problem is more general because we can have multiple units for a particular item and have co-bidding requirements.

Other computational economy based approaches in distributed computing can be found in [AmA98, StA96,

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

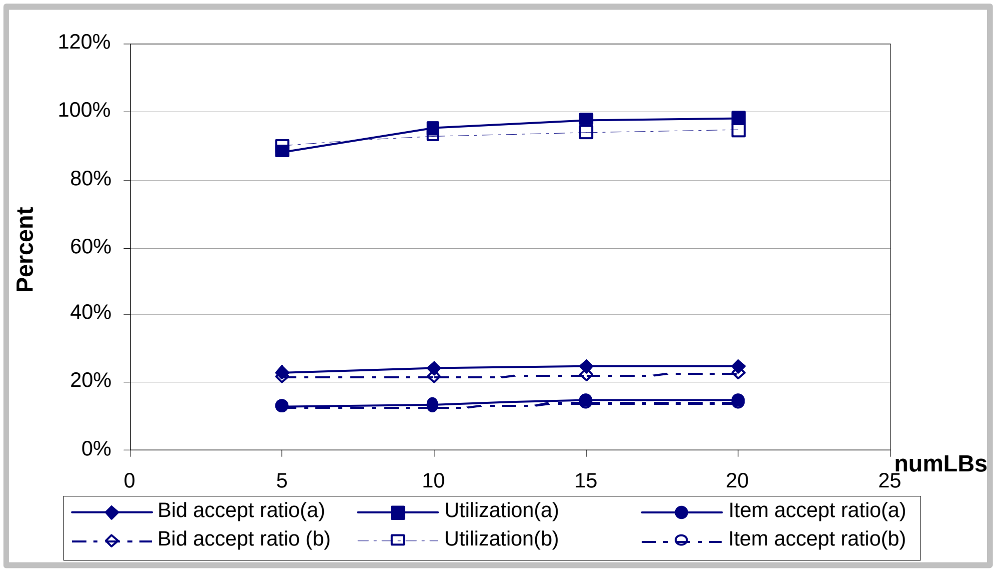

Figure 4. Variation of bid accept ratio, utilization, and item accept ratio with number of LBs for (a) CFA and (b) NPA for bids per LB of [200-300] and percentage of co-bids of 20%.

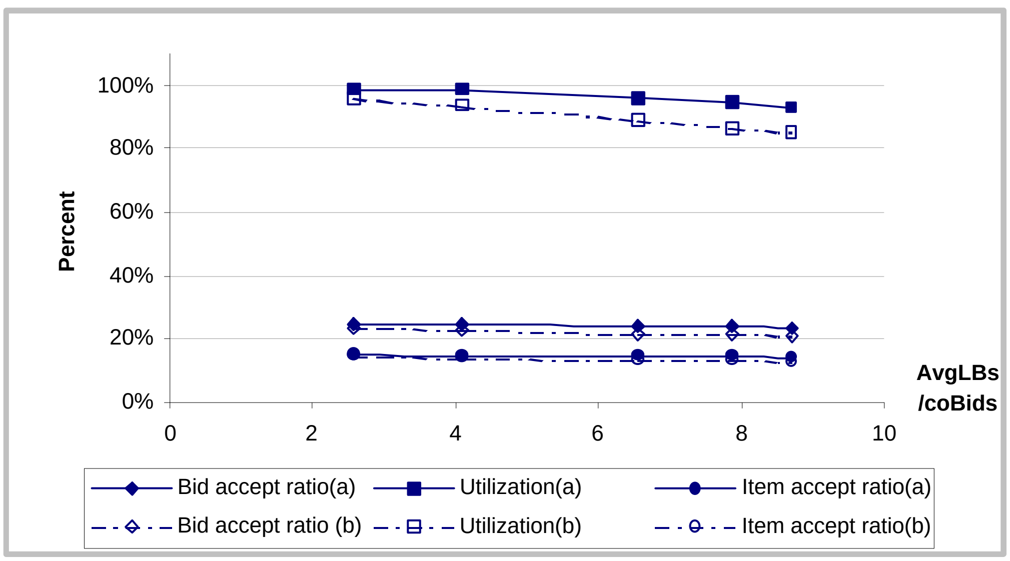

Figure 5. Variation of bid accept ratio, utilization, and item accept ratio with avgLBs/co-bids for 20 LBs and [200-300] bids per LB and percent of co-bids of 10%.

WaH92, WeM98, WeW98].

6. Conclusions

This study focuses on the development of an auctioning based "on demand" resource acquisition and/or trading system for wide-area network computing systems called the *Computation Market* (CM). This paper, in particular, focuses on designing bid selection heuristics that can be used by the auctioning system that is part of CM. We consider a real-time auctioning system that periodically closes the bidding process and chooses the winners. To address the scalability, site autonomy, heterogeneity, and extensibility we organize the CM as an interconnection of local markets.

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

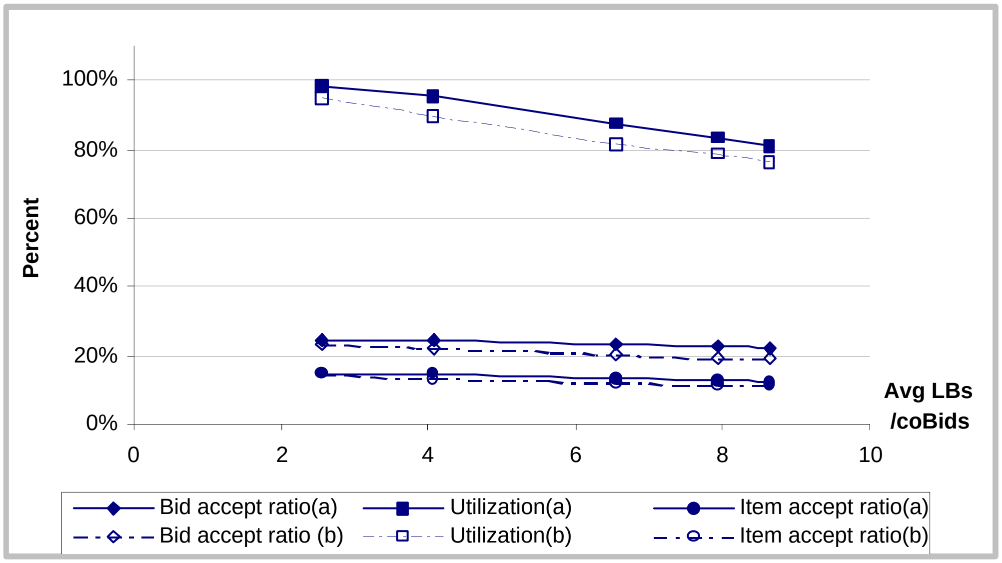

Figure 6. Variation of bid accept ratio, utilization, and item accept ratio with avgLBs/co-bids for 20 LBs and [200-300] bids per LB and percent of co-bids of 20%.

A local market spans a limited network vicinity.

Although auctioning has been studied in several contexts, only recently researchers have started examining the multi-unit combinatorial auctioning problem [GoL00, LeS00]. The combinatorial auctioning problem is NP-complete even for the single unit case [San99] and it is even harder to solve for the multi-unit case. Because we are examining the real-time variant of the problem, we need to solve the winner determination problem as soon as possible (i.e., winner determination should be performed within the time constraints of an auction session). Motivated by this need for a "speedy" solution, we present two fast heuristics in this paper. The simulations show that in general selecting the co-bids that have more constraints first can lead to better overall performance.

References

[AmA98] Y. Amir, B. Awerbuch, and R. S. Borgstrom, "The Java market: Transforming the Internet into a metacomputer," Department of Computer Science, Johns Hopkins University, 1998.

[ChM01] C. Chen, M. Maheswaran, and M. Toulouse, "On bid selection heuristics for real-time auctioning for wide-area network resource management," *2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2001)*, July 2001.

[Che01] C. Chen, *Computation Market Design and Experiments*, Master's thesis, Department of Computer Science, University of Manitoba, under preparation.

[GoL00] R. Gonen and D.J. Lehmann, "Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics," *ACM Conference on Electronic Commerce*, 2000, pp. 13–20.

[LeS00] K. Leyton-Brown, Y. Shoham, and M. Tennenholtz, "An algorithm for multi-unit combinatorial auctions," *17th National Conference on Artificial Intelligence*, 2000.

[RaL98] R. Raman and M. Livny, "Matchmaking: Distributed resource management for high throughput computing," *7th IEEE Int'l Symposium on High Performance Distributed Computing*, July 1998.

[San99] T. Sandholm, "An algorithm for optimal winner determination in combinatorial auctions," *International Joint Conference on Artificial Intelligence (IJCAI)*, 1999, pp. 542–547.

[StA96] M. Stonebraker, P. M. Aoki, A.Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu, "Mariposa: A wide-area distributed database system," *VLDB Journal*, Vol. 5, No. 1, Jan. 1996, pp. 48–63.

[WaH92] C. A. Waldspueger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta, "Spawn:

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 IEEE

A distributed computational economy," *IEEE Transactions On Software Engineering*, Vol. 18, No. 2, Feb. 1992.

[WeM98] M. P. Wellman, J. Mackie-Mason, and S. Jamin, "Market-based adaptive architectures for information survivability," http://www.darpa.mil/ito/psum1998, 1998.

[WeW98] M. P. Wellman and P. Wurman, "Real time issues for internet auctions," *1st IEEE Workshop on Dependable and Real-Time E-Commerce Systems (DARE-98)*, June 1998.

Biography

Chunming Chen is a graduate student in the Computer Science Department at the University of Manitoba, Canada and a research assistant at TRLabs, Canada. She received her BSc in the Department of Electrical and Computer Engineering from HuaZhong University of Science and Technology, China in 1991. Chunming's research interests lie in the area of online auctioning, e-commerce, and architecting and prototyping complex distributed computing systems.

Muthucumaru Maheswaran is an assistant professor in the Department of Computer Science at the University of Manitoba, Canada and a Scientist at TR*Labs*, Winnipeg, Canada. In 1990, he received a BSc degree in electrical and electronic engineering from the University of Peradeniya, Sri Lanka. He received an MSEE degree in 1994 and a PhD degree in 1998, both from the School of Electrical and Computer Engineering at Purdue University, USA. He held a Fulbright scholarship during his tenure as an MSEE student at Purdue University. His research interests include active networking, content delivery and routing, distributed heterogeneous computing, Grid systems, resource management systems, and web caching. He has authored or coauthored about 40 technical papers in these and related areas. He is a member of the IEEE.

Michel Toulouse received a MSc in computer science from University of Montreal in 1989 and a PhD in computer science and engineering from Ecole Polytechnique, University of Montreal in 1996. He is currently an assistant professor in the Department of Computer Science at the University of Manitoba, Canada. His research interests include search algorithms, parallel and distributed algorithms, multilevel algorithms with applications in Physical Design and non-standard computing paradigms (emergent computation, molecular computation, quantum computing, genetic algorithms). He is an Associate Editor of the Journal of Heuristic and co-author of a book on Parallel Metaheuristics.

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 **IEEE**