

SCALABLE ARCHITECTURES FOR RESPONSIVE AUCTIONS

Alessandro Amoroso^{*†}, Dino Derek Hughes, Tommaso Micheletti, Fabio Panzieri

Bologna University, Italy

E-mail: {amoroso, huges, michelet, panzieri}@cs.unibo.it

In this Paper we discuss and compare the design and implementation of two distributed architectures we have developed in order to support “responsive” (i.e., timely and available) auction services in a large scale distributed context. Specifically, our architectures can support distributed auction services whose timing requirements fall within the range of a few seconds. In addition, those architectures can adapt timely to possible variations of the network conditions (e.g., congestion, load).

In order to provide the users with highly available auction services, our architectures allow one to implement those services by means of replicated auction servers. Hence, we believe that the Grid technology can provide us with an ideal infrastructure for their implementation.

1st International Workshop on Grid Technology for Financial Modeling and Simulation

3-4 February 2006

Palermo, Italy

^{*}Speaker.

[†]This work has been partially funded by the project *FIRB/“WebMinds”* of the Italian Ministry of Education, University and Research.

1. Scenario

In recent years a number of auction services have been made available over the Internet (e.g., *www.ebay.com*, *www.antiquorum.com*, *www.artnet.com*, to name a few). A common feature of these services is the considerable amount of time they require to complete the auctioning process. Typically, a user of these services can submit a bid and, only after an amount of time that can range from hours to days, that user knows whether her/his bid has been accepted. This feature is a consequence of the *asynchronous* nature of the Internet, based on a *best-effort* communication service [1, 2, 3, 4].

In addition, current Internet-based auction services rely, in general, on centralized auction server architectures. A centralized architecture cannot deal adequately with issues of service *availability* and *scalability*. Typically, such an architecture can be vulnerable to server's failures, if not equipped with sufficient redundancy; in addition, server's overloading may occur, if an arbitrary large number of users concurrently access the service. The increasing number of customers of Internet-based auction services suggests that both these issues are crucial in the design of those services. In particular, as pointed out in [5], service availability is required as a frequently unavailable service may discourage users from using it, and result in a business loss for its provider.

Owing to the above observations, in this Paper we propose and compare two architectures for supporting auction services over the Internet that are based on replicating the service across a number of auction servers distributed over this network. The first architecture, named SARA (*Synchronous Architecture for Responsive Auctions*) [9], is based on a soft real-time scheme. The second architecture, named ADA (*Asynchronous Distributed Auctions*), is based on both hierarchical lightweight communication, and *coordinator-cohort* computation [6].

Both of our architectures offer advantages to the auction service provider and to the users. Specifically, from the service provider perspective, our distributed architectures can accommodate an arbitrary number of users by balancing the work load among all the servers implementing the auction service. From the user perspective, our architectures can support a responsive auction service, provided that the clients be bound to the "most convenient" replica server (e.g., the most lightly loaded replica, the replica with the least congested path to the client).

A number of distributed architectures for supporting auction services over the Internet can be found in the literature. In [4] the authors propose a hierarchical scalable architecture, while in [5] the servers define a peer group over a wide-area network. In [1] the authors points out that *scaleability* and *responsiveness* are key features for the design of an electronic market server. The soft real-time requirements for an auction service over the Internet are addressed in several works, including [2, 7].

2. The two approaches

The entities involved in both our architectures are the *clients* and auction *servers*. Clients run on the user's computer; a client is connected to a single auction server and sends to it the user's bids. The clients receive from the connected server the current auction state, *i.e.* the *current best*

*bid*¹. The submission of a new bid is asynchronous as it depends on the user will, while the rate at which the servers send to the clients the current auction state depends on the architecture. SARA implements periodic transmission of the auction state; in contrast, ADA updates the clients at a non predictable pace.

Fig. 1–(A) and Fig. 2–(A) illustrate an example of the two different architectures. The clients are represented as rounded rectangles; the servers are represented as circles. The edges between servers and clients, and between servers, represent bidirectional communication channels. The auction service is represented as a dark dashed ellipse that includes all the servers. The architectural differences between SARA and ADA lie into the auction service, *i.e.* the way the servers are connected and communicate with each-other. The clients are essentially the same in both SARA and ADA. The following sections discuss and compare the two architectures regarding these criteria:

Responsiveness: we can express the responsiveness of the system, R , by assessing how long it takes to identify the current best bid, and to communicate it to all the clients. We set to $2sec$ the maximum acceptable latency of any single message.

Scalability: we can assess the resource requirements with respect to the number of clients, N ; specifically we assess both the number of servers, S , and the number, M , of messages exchanged by the servers, *i.e.* the internal bandwidth of the auction service. According to previous measurements, we assume that a server can satisfy about 200 requests per second by means of secure communication [8]; in the following computation we assume that a server has at most 1000 clients.

2.1 Synchronous architecture

Within the SARA architecture, the auction servers cooperate as in a *periodic real-time system*, as proposed in [5]. Specifically, an auction is organized as a sequence of one or more *rounds*. Within each round the auction servers collect bids from the participants, asynchronously from each other, and evaluate the locally received best bid. Periodically, the servers synchronize in order to reach consensus on the received best bid in the current round, and to maintain consistently what we term the *shared auction state*. This shared state information includes the item on sale, the best bid received for that item, the identifier of the client that submitted that bid. Following each synchronization phase, the servers resume the bid collection process, asynchronously from each other, until the next synchronization phase. The auction terminates during a synchronization phase in which specific auction termination conditions are met. Fig. 1–(B) summarizes this scenario.

Note that, in our architecture, the servers do not need access to any kind of global timing service, for the purposes of the periodic synchronization mentioned above, as it can be shown that the servers’s local physical clocks are sufficiently accurate to maintain the required synchronization among them. Rather, a global timing service is required for initialization purposes, only. To this end, we use the *NTP* timing service currently available over Internet.

Responsiveness A local best bid is detected by a server when it receives the bid. The servers send the local best bids to the other servers at the end of the current collection phase. At the end of the synchronization phase, all the servers know the current best bid, and send it to their

¹The *current best bid* depends on the type of auction being held, (*e.g.* open cry, sealed bid, dutch, double auction). For the purpose of this discussion, we define “best bid” the bid that would win the auction if the auction terminates at the time that this bid is evaluated.

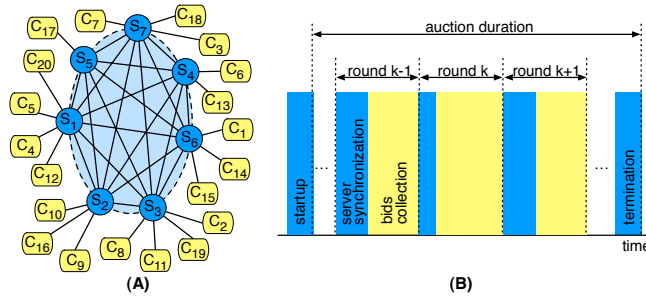


Figure 1: The architecture of SARA (A), and its time scan of an auction (B).

clients. Summarizing, in the worst case it takes $10sec$ plus the whole duration of the collection and synchronization phase to identify and diffuse the current best bid. Considering the collection phase concurrent to the synchronization phase, *i.e.* it lasts $0sec$, the duration of a synch phase depends on the network latency, and it is at least $50sec$ [9], then the responsiveness of SARA is about a minute, $R = 60sec$.

Scalability The number of servers is linearly proportional to the number of clients *i.e.* $S = N/1000$. Each synchronization phase requires S broadcast to S servers, that is $M = O(N^2)$.

2.2 Asynchronous architecture

ADA implements a *coordinator-cohort* mechanism to manage the auction. We differentiate three kinds of servers: *plain*, *cohort* and *coordinator*. From the clients point of view, the plain servers are connected to the clients, collect their bids, and sometime they send back the current auction state. From the system point of view, every plain server is connected to a cohort server, periodically sends to its cohort server the best bid that has collected, and sometimes it receives back the current auction state. In Fig. 2-(A) the plain servers are those in the lower hierarchical level inside the system. The cohort periodically collects the best bid sent by the connected plain server, calculates the best bid that have received, and communicates that bid to the leader. The coordinator periodically computes the best bid it has received hitherto, and sends it to the cohorts as the current state of the auction. In Fig. 2-(A) the cohort servers are the few in the intermediate hierarchical level inside the system, while the coordinator is the one at the top of the hierarchy. The example shown in Fig. 2 is simplified with respect to the whole design, where all the servers are allowed to be connected to the clients, to emphasize the roles of the coordinator and the cohorts. Fig. 2-(B) shows a space-time diagram, where the processes are represented as a continuous horizontal line, and the time flows from left to right. In this figure, three clients, C_{17} , C_{20} and C_8 , submit a bid to their servers. The ticks on the servers represent the scheduled time to send the local best bid, in the case of cohort servers, and to send the current best bid in the case of the coordinator.

Responsiveness In the worst case it takes $12sec$ plus the duration of a cohort cycle and a coordinator cycle, *i.e.* the pace at which they compute the local best and the current best bid. The duration of these cycles depends on the computational speed of the cohort and coordinator, and can be the order of magnitude of seconds. Summarizing, the responsiveness of ADA is $R = 30sec$. This duration can be further reduced, and it is a fraction of the minimum value provided by SARA. Fig. 2-(B) shows an example of worst case responsiveness for the best bid submitted by client C_7 .

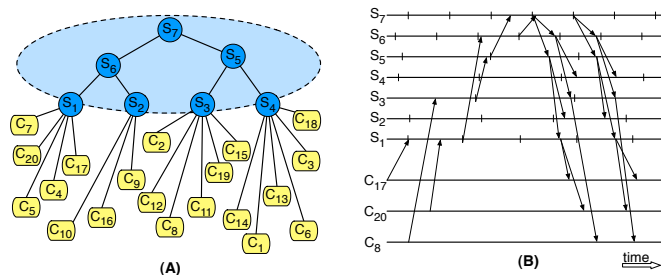


Figure 2: The architecture of ADA (A), and a significant space–time diagram (B).

Scalability The number of plain servers is $S = N/1000$, the minimum number of cohort servers is at least two order of magnitude smaller, and the coordinator is a single server, then $S = O(N)$. The messages exchanged by the servers are double of the communication channels inside the auction service, because any link is traversed by two messages at most, then $M = O(N)$.

3. Conclusions

We have implemented a working prototype of both SARA and ADA over the Internet. An experimental implementation over a GRID of the proposed architectures is currently under development in our Department. This implementation should be transparent to the clients, that would not notice any difference with respect to the previous implementation.

References

- [1] C. Wrigley, *Design criteria for electronic market servers*, *EM-Electronic Markets*, vol. 7, no. 4, pp. 12–16, 1997.
- [2] C. Peng, J. Pulido, K. Lin, and D. Blough, *The design of an internet-based real time auction system*, in proceedings of *1st IEEE Workshop on Dependable and Real-Time E-Commerce Systems*, Denver (CO), June 1998.
- [3] D. Ingham, F. Panzieri, and K. Shrivastava, *Constructing dependable web services*, *IEEE Internet computing*, vol. 4, no. 1, pp. 25–33, Jan.-Feb. 2000.
- [4] P. Ezhichevan and G. Morgan, *A dependable distributed auction system: Architecture and an implementation framework*, in proceedings of *5th International Symposium on Autonomous Decentralized Systems*, Dallas (TX), March 2001.
- [5] F. Panzieri and S. Shrivastava, *On the provision of replicated internet auctions services*, in proceedings of *18th IEEE Int. Symp. on Reliable Distributed Systems*, Lousanne (CH), Jan. 1999, pp. 390–95.
- [6] K.P. Birman, *Building Secure and Reliable Network Applications*, ch.15, Manning, 1996.
- [7] M. Wellman and P. Wurman, *Real time issues for internet auctions*, in proceedings of *First IEEE Work. on Dependable and Real-Time E-Commerce Systems(DARE-98)*, Denver (CO), June 1998.
- [8] A. Amoroso and M. Nanni, *Securing client–server communications of an internet auction service*, in proceedings of *Euromedia 2004*, Hasselt (Belgium), April 2004, pp. 102–107
- [9] A. Amoroso and F. Panzieri, *A responsive architecture for distributed auction services over the internet*, in proceedings of *WSDAAL 2001*, Como (IT), Sept. 2001, pp. 10–12