On Efficient Routing Structures for Information Acquisition in Distributed Markets

René Brunner, Felix Freitag, Leandro Navarro
Computer Architecture Department, Polytechnic University of Catalonia
08034, Barcelona
Spain
{rbrunner, felix, leandro}@ac.upc.edu

***Abstract*—In a market, information about its specifications and the behavior of its participants is essential for sophisticated and efficient negotiation strategies. However, there is currently no completely researched system to provide and consult an overall knowledge of economic information in distributed markets. This paper presents a prototype for a Decentralized Market Information System (DMIS) which compares the lookup process based on unbalanced binary trees with B-Trees for a fast information provision. We propose an architecture for the DMIS which combines technologies of structured overlay networks and tree-based routing structures to obtain in a simple manner efficient and fast information acquisition. The architecture has been designed to meet both the economic information requirements and that of scalability and robustness of a large-scale distributed environment. Initial measurements confirm the proof-of-concept implementation of the existing prototype with varying routing structures.**

I. Introduction

In the last few years the emerging of Grid markets put the focus on market mechanisms. The distributed nature of Grid applications set a trend to use distributed markets for ressource allocation. Examples of such approaches based on peer-to-peer (P2P) are the market-based Grid platforms developed in several projects such as Grid4All [14], GridEcon [15], Tycoon [18] or Sorma [21]. These markets use auction mechanisms like the Continuous Double Auction (CDA) or the English auction. But there are also other recent bargaining approaches like Catallaxy-based Grid Markets [13].

A problem resulting from distributed markets is the gathering of information about the market, its prices, products and the participating traders. The knowledge about the market is essential for sophisticated and efficient negotiation strategies. Examples are computational approaches like the game theory, predicting the future through forecasting or using learning rules on former or actual trading information. However, there is currently no completely researched system to provide and consult an overall knowledge of economic information in distributed markets.

Bergemann's survey [4] shows that the economic aspect of information acquisition in market mechanisms such as auctions got more attention by the economic research community. Moreover, the study demonstrates the importance of the economic information disclosure for market participants. The need for this information lies in both being able to apply sophisticated economic strategies and to feed business models, which are behind these strategies.

The objective of this paper is to apply existing routing structure to distributed P2P-based aggregation systems and filter-based publish-subscribe models. These will be evaluated in regard to the properties of an Distributed Market Information System (DMIS). Obtaining efficiently economic attributes depends on their properties like frequency of update or frequency of requests. B-Trees promises to be more efficient in looking up filtered attributes, where unbalanced binary trees promise to be more efficient in subscription process.

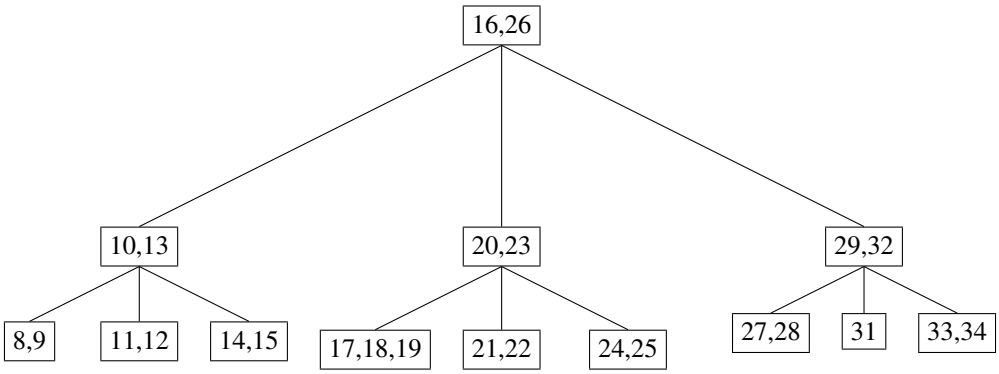
For the gathering of information our approach integrates and adapts existing technologies and models based on structured overlay networks. This is given by the integration of distributed information aggregation technologies and ensures a high scalability. Furthermore, implementing the structures of a distributed publish-subscribe model allows providing time sensitive data to the traders participating in markets.

In this paper we first present in Section II the related work by showing B-Tree implementations in distributed environments. Section III illustrates the motivation for different routing structures. An overview of the architecture shows the different layers and the API of the implemented DMIS prototype in the Section IV. In the Section V we demonstrate the proof-of-concept implementation in form of simulations. Finally, we conclude this work and give an outlook on further steps to enable efficient routing in market and general distributed information systems.

II. Related Work

The importance of information acquisition in markets shows Bergemann's survey [4]. It discusses the retrieval and aggregation of information in mechanism like markets. Other literature put emphasis on the theoretical analysis of the influence of information to markets [17] or the acquisition of information [5]. Also researches about required information for economic markets are mentioned in [16]. However, the above mentioned literature focus on theoretical analysis.

Fig. 1. B-Tree Example



In distributed systems, only a few works concentrate on using B-Trees [11] in distributed systems. [1] proposes an alternative system using B-Tree as an efficient search algorithm. It is a database system, applied in a distributed environment. Thereby, the focus is on small-scale, as it is conceived for several servers to distribute the content. In comparison to our approach, it is not completely decentralized for large-scale like our approach.

P-Tree [12] is a structured overlay network itself while using a variation of a B-Tree. However, it does not take into account providing time-sensitive data by using a publish-subscribe model. Message aggregation to obtain an efficient information acquisition in large-scale P2P systems is not considered in this approach. Furthermore, no flexibility is provided in changing the tree-based routing structures to adapt to the characteristics of nodes in the system.

YA [8] simulates a B-Tree for the lookup of idle resources in a Grid environment. It connects the Grid resources to each other without using a Distributed Hash Table (DHT) as bottom layer. Furthermore, it uses one tree for the whole system. Our approach integrates distributed aggregation mechanisms and filter-based publish-subscribe models.

[9] proposes a B-Tree-based approach for range queries in distributed applications. In comparison to our approach it is the overlay itself and does not use a DHT, as a separation of the logic for an easier development. There is no flexibility in regard to changing properties and the adaption to different routing structures.

III. Motivation

Actually, DHTs are wide-deployed overlay structures for P2P Applications. For example the Kademlia based Kad protocol has currently over a million users. But not only file sharing applications bases on DHTs, moreover many applications with a scalable amount of users base on structured overlays. These are for example Grid Applications [14][21], publish-subscribe systems and Monitoring Systems.

Separating the overlay layer and the application layer or other routing mechanisms results in an easier development, which illustrates the study of Chawathe et al. [10]. It shows that a less effective routing has fewer disadvantages than mixing up structures. Mainly this concerns the DHT and systems like publish-subscribe or aggregation systems, which are using these services. The decoupling of structures and code provides also higher flexibility to changes and evolutions.

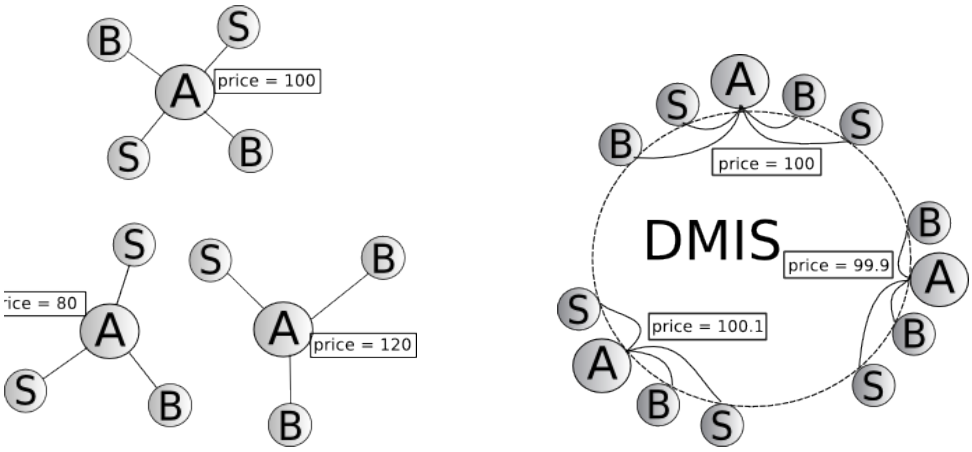
Many existing systems already incorporate this principle of the separation of logical layers. These are systems proposed for publish-subscribe models or distributed aggregation mechanisms. For example, Scribe [20] is build on top of the DHT Pastry [19], while reusing the Id and the lookup process for the rendez-vous node. This allows joining a topic and of course the underlying messaging. Other examples for such systems based on structured overlay networks are Meghdoot, systems presented by Baldoni [2], [3] and SelectCast [6].

Deployment of the DMIS within large-scale applications pursues a high scalability and robustness against failures and churn. This makes a DHT preferable, which already proved to have these properties. Providing of the information results in using aggregation mechanisms, which are an efficient concept in distributed large-scale systems. DHT-based publish-subscribe systems, which use filtering allow the provision of time-sensitive data.

A. Scenario

Trading in distributed or semi-distributed online markets represented by an auction, opens the possibility for trading clusters. These clusters are disconnected and without global information. This can be a consequence of using different auctions like English Auction and Continuous Double Auction, geographical preferences, certain preferences of traders for certain products or providers, privacy and trust constrains or political aspects. These can lead to a separation of traders or in building clusters of traders with similar preferences and leads to a decoupling of an unique equilibriums price.

Figure 2(a) shows a scenario of decoupled auctioneers in a decentralized environment. Coordinated by auctioneers, sellers and buyers are trading on different marketplaces. Among both auctions which could be different type of auctions, there exists no implicit information exchange. Moving traders among auctions could help to equilibrate the price among the auctions.



(a) Independent markets scenario

(b) Market with global information scenario

Fig. 2. Market Scenario

But a trader needs information about other auctions to be motivated for moving. Introducing the DMIS to the scenario of electronic markets enables explicitly indirect information exchange among all participants (see Fig. 2(b)). This way, traders would be able to obtain information from other auctioneers and indirectly from other traders. This information allows the traders to adapt their price or to move to other auctions corresponding to their strategy.

Global information provision to individual traders in near real-time is very important in those markets. Latency in the information acquisition leads to an inaccurate information about the price. For example a client might receive information about a previous average price, which is not anymore valid. But the price would be used for simple and sophisticated trading strategies. The result could be an uncompetitive offer, where no match can be found or where the trader gets less benefit than necessary.

B. Fast Information Acquisition

An example for an application, which needs to incorporate fast and efficient routings on top of a DHT is the Distributed Market Information System (DMIS) [7]. It provides traders and auctions within a market with economic information. The provision of these information has to reach close to real-time as this information builds the basis for economic pricing strategies.

Benchmarks have demonstrated that B-Trees support highly available, scalable, distributed transaction processing applications [11]. Figure 1 shows an example of a B-Tree with maximal 3 keys per tree node. The advantage of the B-Tree is the reduced height in comparison to binary trees. Moreover, inserting and deleting of keys, which can be values or the Id of the P2P nodes, produces also lower costs as the tree nodes can be filled without restructuring the B-Tree.

To our knowledge these system do not profit from the efficiency of B-Trees. Many existing databases are using B-Trees [11], which is supposed to be the most efficient algorithms for querying data. This is especially important in regard to the time constraints of the DMIS, as B-Trees are faster for lookups. This tree algorithm can also be adapted to existing systems build on top of DHTs. This are mainly filter-based publish-subscribe systems and distributed aggregation systems.

Table I compares the characteristics of an unbalanced binary tree, a balanced binary tree, and a B-Tree. It shows the costs for a node lookup, for a insertion, for splitting, for merging and the maximum height of a tree. For a lookup for a certain node, the B-Tree is the most effective comparing the hops in filter-based query. But also for a lookup for all nodes, which is needed for aggregation messages, it is the fastest of the presented tree algorithms.

IV. System Architecture and Implementation

We follow an approach to keep a clear separation and decoupling of the DHT layer and advanced routing structures. Thereby an easier development is achieved, which is confirmed by the study [10] and by other projects following such a separation (compare Section III). Moreover, the separation provides the flexibility to adapt the structure to more efficient routing, e.g. through changing the DHT type.

An example for an architecture using flexible tree-based routing structures shows Figure 3, which is used by the DMIS [7]. The bottom layer uses a DHT as communication channel. Therefore it uses the function of route, send, receive, put and get from the DHT overlay. Furthermore, the unique 128-bit hash Id is also built by the DHT application. It uses a similar approach like Scribe [20].

The method `route` is used for the subscription process, which can be depending on the application, topic-based, content-based or type-based. This process executes a lookup process, by using the DHT behavior to locate a rendez-vous

TABLE I
COST OVERVIEW OF TREES (D = KEYS PER NODE)

Tree Type	*Insert/Delete*	*Split/Merge*	*Query*	*Maximum Height*
Unbalanced binary tree	O (n)	-	O (n)	n
Balanced Binary Tree	O ($2 * \log$ n)	O (log n)	O (log n)	log n
B-Tree [11]	O ($2 * \log_d(\frac{n+1}{2})$)	O ($\log_d(\frac{n+1}{2})$)	O ($\log_d(\frac{n+1}{2})$)	$\log_d(\frac{n+1}{2})$

node, which is responsible for a topic or content. To ensure robustness against failures and churn, the rendez-vous node has to be replicated, which is supported by most standard DHTs.

Such rendez-vous nodes have the advantage to allow more flexibility within the routing processes. These are the bottleneck for requests and the central point of knowledge for new information. This makes them designated for controlling and adapting messaging for read-dominated or write-dominated attributes. Especially in aggregation and filtering structures the efficiency changes by deploying different routing structures [22].

After a subscription process the routing among the aggregation and filtering process is executed by the `send` message. Avoiding time intensive hops in the DHT, sending the messages direct to the child or parent node in the aggregation tree saves time and network load. Also the unsubscription can be done with the send message.

The Routing layer is responsible to provide different tree structures for an efficient routing. Depending of the attribute properties (e.g. *read-dominated* / *write-dominated*), different routing structures are preferable. For example, if there are more subscriptions to a topic or a content as queries follows that routing structures with less subscription costs and higher query costs is more efficient for the whole system. The Routing layer differentiates for example between unbalanced and balanced trees or binary tree and B-Tree. The methods `getChilds` and `getParent` are defined here. Introducing this layer enables a higher flexibility in changing the routing structures.

The Filter and Aggregation layer deploys the strategies of an optimal filtering which avoids sending messages to irrelevant nodes. For example, it parses the routing tree by sending a request message to only the nodes where price < 2. A method provided by this layer is `sentToNext`, where it defines automatically with the routing layer the next nodes to send. More complex mechanisms such as Bloom filters can be deployed in this layer.

The Application layer is the interface for applications such as the Distributed Market Information Systems (DMIS). But also other information management systems based on structured overlay networks can be deployed. In our experiments we provided the DMIS for the testing of this middleware structure.

V. PRELIMINARY MEASUREMENTS

The experiment we perform has the objective to find the worst time needed for retrieving information from the system.

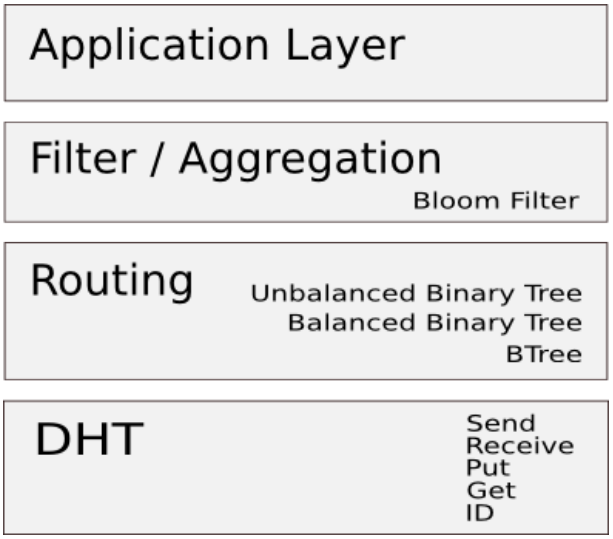


Fig. 3. Architecture Layers

For this purpose, we instantiate the prototype with a different number of nodes in the range of 10, 50 to 500 in steps of 50. The initialization process assigns each DMIS instance with a predefined `price` (100 *money units* $\pm$ 20 *money units*) and a random `storage` capacity (20 *storage units* $\pm$ *storage units*) representing one attribute of a traded product. Afterwards all nodes are bootstrapping to the overlay network. A subscription to a topic allows to publish and to receive new information, but primarily the necessity is to allow access to the information of other nodes for a query.

We measure the number of hops needed to retrieve an aggregated value as an indicator for the required time. In a real deployed large-scale system often the critical time constrain is sending of messages from one peer to another peer. Especially in Grid environments the round-trip time (RTT) can vary between low and high bandwidth networks. Grid applications follow a democratic approach which integrates connections with a lower bandwidth such as individual peers with analog connections or mobile devices. In contrary Grid applications for large enterprise or intranet applications suppose to provide a high bandwidth connection. Therefore we count the number of hops to allow the adaption of our measurements with the average RTT of the focused network type.

First a user sends a join message for an information topic such as `Price` or `CPU` to the DMIS. Afterwards we tested the retrieving of information with different user. Each instance

sends a request messages to all nodes following the defined tree structure. The aggregation process is performed on the inner nodes, where the average, minimum and maximum price is calculated for the node's subtree. Additionally the nodes are counted and a summary of the offered prices in created.

We want to characterize the performance of queries using different types of tree routing structures. Therefore we are sending queries to all nodes in the system following a tree structure. A value is aggregated for calculating the sum, average, count, maximum or minimum of the node value on the ways back to the root. Sending messages to all nodes allows implicitly to analyze the worst case scenario for finding a leaf node in terms of number of hops and time.

Figure V shows the number of hops beginning at the rendezvous node until returning back to it. Within balanced trees this value is the average for reaching each node. However, in unbalanced nodes the distribution is randomized through the hashed Id of the Pastry node. Thus different constellations of trees can exist and the presented value is an average value of the worst case in regard to the number of hops. The comparison shows that the number of hops needed to reach the leaf node is lesser in a B-Tree than in an unbalanced binary tree.

The need for fewer messages for information retrieval is very important within a market information systems. Concerning the results, a B-Tree is preferable for the DMIS in regard to the querying process, as fewer node hops imply a faster response. Considering various types of network connection can result in higher network cost (RTT), while fewer hops in querying can produce more maintenance messages.

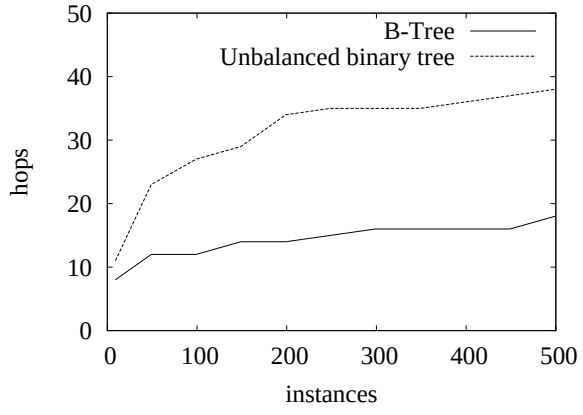


Fig. 4. Maximum number of hops to the leaf nodes.

The experiments show a well-functioning of the prototype and its flexibility in changing easily different format of routings. This allows an adaption to varying characteristics of attributes, which allows to increase the performance of the information system.

VI. Conclusion

We showed the integration of existing routing structures into the field of structured overlays. This integration combines in an easy way two successfully applied mechanisms. One is using B-Tree-based query structures, which are efficiently applied in most RDBS. The other is the separation of the overlay network from other parts of the middleware mechanisms such as publish-subscribe.

We have built a prototype of a DMIS to execute simulations for the comparison between routing algorithms based on different tree structures. The results show that the routing with a B-Tree requires fewer hops until receiving the result. This is especially important in markets as the information needs to be acquired near to real-time. The cost of maintenance in this context is considered of secondary importance.

However, depending on the characteristic of the attributes, maintenance gains on importance in regard to the time for querying. Therefore the prototype provides the flexibility to change the tree-based routing structure, which is needed for filtering and information aggregation. This allows to define a structure with a lower query time and higher maintenance costs or a structure with a higher query time and lower maintenance costs before deploying the application.

VII. Future Work

We plan the further evaluation and analysis of different routing structures. This is in terms of joining (subscription) algorithms and maintenance processes such as split and merge caused by failures and churn in a distributed environment. This includes also taking advantage of piggyback messages for the subscription process. Finding a trade-off between subscription costs and query costs and time will be evaluated. The results expect to give new insights also to other kind of P2P applications.

The following step is to use the preliminary results in combination with ongoing investigation to optimize the routing by using flexible changes of routings. We expect that depending on the kind of attribute, e.g. *read-dominated* or *write-dominated*, different routing algorithms are preferable. To increase the performance of the information system we will enable a high flexibility to change different aggregation and filtering mechanisms during the application runtime.

Another work will be about the Quality of Service (QoS) and how the failure and churn takes influence in the proposed routing mechanisms. A high level of guarantees results in increasing messages to ensure robustness against failures and to provide exact and accurate information. Therefore a trade-off between sufficient guaranties and sufficient performance for the user and traders will be shown.

VIII. Acknowledgments

This work was supported in part by the European Union under Contract Grid4All EU IST-FP6-034567, under Contract

SORMA EU IST-FP6-034286, and the Ministry of Education and Science of Spain under Contract TIN2007-68050-C03-01.

REFERENCES

[1] Marcos K. Aguilera, Arif Merchant, Mehul Shah, Alistair Veitch, and Christos Karamanolis. Sinfonia: a new paradigm for building scalable distributed systems. In *SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 159–174, New York, NY, USA, 2007. ACM.

[2] R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. Tucci Piergiovanni. Tera: Topic-based event routing for peer-to-peer architectures. In *In Proceedings of the 1th International Conference on Distributed Event-Based Systems (DEBS07)*. ACM, 6 2007.

[3] Roberto Baldoni, Carlo Marchetti, Antonino Virgillito, and Roman Vitenberg. Content-based publish-subscribe over structured overlay networks. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 437–446, Washington, DC, USA, 2005. IEEE Computer Society.

[4] D. Bergemann and J. Valimaki (2006). Information in mechanism design. In Whitney Newey Richard Blundell and Torsten Persson, editors, *Proceedings of the 9th World Congress of the Econometric Society*, volume Cambridge University Press 2007 of *Chapter 5*, pages 186– 221, 2007.

[5] Dirk Bergemann and Juuso Valimaki. Information acquisition and efficient mechanism design. *Econometrica*, 70(3):1007–1033, May 2002. available at http://ideas.repec.org/a/ecm/emetrp/v70y2002i3p1007-1033.html.

[6] Adrian Bozdog, Robbert van Renesse, and Dan Dumitriu. Selectcast: a scalable and self-repairing multicast overlay routing facility. In *SSRS '03: Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems*, pages 33–42, New York, NY, USA, 2003. ACM Press.

[7] René Brunner, Felix Freitag, and Leandro Navarro. Towards the development of a decentralized market information system: Requirements and architecture. In *Parallel and Distributed Computing in Finance (PDCoF'08).Proceedings of the 22nd IPDPS, Miami, FL, USA*, 2008.

[8] Javier Celaya and Unai Arronategui. Ya: Fast and scalable discovery of idle cpus in a p2p network. In *GRID*, volume 7th IEEE/ACM International Conference on Grid Computing (GRID 2006), September 28-29, 2006, Barcelona, Spain, Proceedings, pages 49–55, 2006.

[9] Badrish Chandramouli, Junyi Xie, and Jun Yang. On the database/network interface in large-scale publish/subscribe systems. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 587–598, New York, NY, USA, 2006. ACM.

[10] Yatin Chawathe, Sriram Ramabhadran, Sylvia Ratnasamy, Anthony LaMarca, Scott Shenker, and Joseph Hellerstein. A case study in building layered dht applications. *SIGCOMM Comput. Commun. Rev.*, 35(4):97–108, 2005.

[11] Douglas Comer. Ubiquitous b-tree. *ACM Comput. Surv.*, 11(2):121–137, 1979.

[12] Adina Crainiceanu, Prakash Linga, Johannes Gehrke, and Jayavel Shanmugasundaram. Querying peer-to-peer networks using p-trees. In *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases*, pages 25–30, New York, NY, USA, 2004. ACM.

[13] Torsten Eymann, Michael Reinicke, Werner Streitberger, Omer Rana, Liviu Joita, Dirk Neumann, Björn Schnizler, Daniel Veit, Oscar Ardaiz, Pablo Chacin, Isaac Chao, Felix Freitag, Leandro Navarro, Michele Catalano, Mauro Gallegati, Gianfranco Giulioni, Ruben Carvajal Schiaffino, and Floriano Zini. Catallaxy-based grid markets. *Multiagent Grid Syst.*, 1(4):297–307, 2005.

[14] Grid4All. http://grid4all.elibel.tm.fr/, 2007.

[15] GridEcon. http://www.gridecon.eu/, 2007.

[16] Jens Grossklags and Carsten Schmidt. Interaction of human and artificial agents on double auction markets - simulations and laboratory experiments. In *IAT '03: Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, page 400, Washington, DC, USA, 2003. IEEE Computer Society.

[17] Matthew O. Jackson. Efficiency and information aggregation in auctions with costly information. *Review of Economic Design*, vol. 8, no. 2:121–141, 2003.

[18] Kevin Lai, Bernardo A. Huberman, and Leslie Fine. Tycoon: A distributed market-based resource allocation system. Technical report, HP:arXiv:cs.DC/0404013, 2004.

[19] Antony Rowstron and Peter Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.

[20] Antony I. T. Rowstron, Anne-Marie Kermarrec, Miguel Castro, and Peter Druschel. Scribe: The design of a large-scale event notification infrastructure. In *NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication*, pages 30–43, London, UK, 2001. Springer-Verlag.

[21] SORMA. Sorma project. http://www.iw.uni-karlsruhe.de/sormang/, 2007.

[22] Praveen Yalagandula and Michael Dahlin. Shruti: A self-tuning hierarchical aggregation system. In *SASO*, pages 141–150, 2007.