Towards the Development of a Decentralized Market Information System: Requirements and Architecture

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. These markets are implemented for example by Grid applications and gained importance over the last few years. This paper presents the economic information requirements and a high-level architecture overview for a Decentralized Market Information System (DMIS). The proposed system acquires economic data in a distributed environment for providing it to individual traders or other participants in a decentralized manner. First, we outline the economic information requirements which the system needs to achieve. Therefore their properties and a privacy model has to be considered. Then we propose an architecture for the system which combines technologies of distributed information aggregation system and distributed publish-subscribe models, based on a structured overlay network. 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.**

I. Introduction

The emerging Grid markets put the focus on market mechanisms in the last few years. The distributed nature of Grid applications set the trend to integrate distributed markets, which are for example based on peer-to-peer networks. Examples of such approaches based on P2P are the market-based Grid platforms developed in several projects such as Grid4All [1], GridEcon [2], Tycoon [3] or Sorma [4]. These markets use auctions like the Continuous Double Auction (CDA) or the English auction. But there are also other recent bargaining approaches like Catallaxy-based Grid Markets [5].

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 [6] 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 combine the results and solutions provided by the economic research community with solutions proposed from the research community of distributed computing in order to achieve a Decentralized Market Information System (DMIS). However, to our knowledge there is no general approach focusing on our solution, which underlines the need for new solutions or adapting existing solutions. Therefore, we introduce DMIS to fit in these demands and merge finance science and the computer science in regard to provide economic information in distributed systems.

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 ensures a high scalability. Furthermore, implementing the structures of a distributed publish-subscribe model allows providing time sensitive data to traders participating in a market.

In this paper we first present the related work in economics and computing in Section II. Section III illustrates the motivation for the development of a DMIS. The economical and technical requirements for auction-based marketplaces in Grids are analyzed in the Section IV. An overview of the architecture shows the different layers and the API of the implemented DMIS prototype in the Section V. In the Section VI we demonstrate the proof-of-concept implementation in form of simulations. Finally, we conclude this work and give an outlook on further steps to be developed within the DMIS.

II. Related Work

The current work concerning a market information system in a distributed environment can be structured in an economic part and a technical part in distributed systems. The economic work is about the investigation, acquisition and aggregation of information in markets. The technical work contains algorithms, models and systems applied within a structured overlay network.

A survey from Bergemann [6] 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 [7] or the acquisition of information [8]. Also researches about required information for economic markets are mentioned in [9]. However, the above mentioned literature focus on theoretical analysis. In our work we aim to make empirical analysis in form of simulations and in developing a prototype.

The proposed DMIS follows a similar approach like existing peer-to-peer (P2P) aggregation frameworks achieved in Astrolabe [10] based on an unstructured P2P overlay. The projects Willow [11], DASIS [12] and SOMO [13] have a single hierarchical tree architecture, that aggregate all attributes along one tree. Consequentially, the root node and the hierarchical higher nodes are a point of failure and bear an over proportional network load. Cone [14] has a multiple tree aggregation, where each attribute has its own tree but requires a total order on the attribute values. DSMIS [15] is a hierarchical multiple tree aggregation framework which is based on a Distributed Hash Table (DHT) overlay network. The application obtains a high flexibility by deciding the propagation level of the aggregation.

The DMIS needs to integrate a content-based publish-subscribe model with filtering of routing messages. This is related to recent research in P2P-based publish-subscribe models, which can be categorized into three subscription models: topic-based, content-based and type-based. Topic-based systems like SCRIBE [16] exchange information through a set of predefined subjects called topics. Although these are easy to integrate on top of a DHT, they are not flexible to changes after the creation of a topic. In contrary content-based systems like Gryphon [17], Siena [18], Meghdoot [19] and Ferry [20] are more flexible as subscriptions are related to specific information content. Therefore each combination of information items can actually be seen as a single dynamic logical channel. Type-based systems like Hermes [21] are a combination of the previous two approaches; they have the advantage of type-safety that allows the publishers to extend the types of events without forcing the subscribers to update their subscription. Ferry contains more relevant properties like aggregation guaranties, range prediction and persistence. However it is not filter-based which is expected to be an import feature for the DMIS.

III. Motivation

Information about markets is essential for sophisticated and efficient negotiation strategies for traders. There are several aspects that point out the need for a decentralized market information system provided by distributed markets. It might increase the usability for users trading in markets. Moreover, the need for information can be seen in central markets and adopted to distributed markets.

Existing information systems for markets prove the need for a market information system. The New York Stock Exchange, the marketplace with the highest volume, shows the existence of various stock information systems. For example Google Finance [22], Yahoo! Finance [23] or Reuters Stocks Information [24] provide a free service an information. Some institutions offer additional payable services like Stockgroup [25]. All mentioned services base on a central market, but by moving to a decentralized market the information will continue to be requested.

Using a market information system brings information into the black box market. It can be supposed that users prefer to trade in a market, when knowing more details. This might be a psychological effect, especially when the users put money into the market. Unfortunately, these are only assumptions, there is no scientific proof of that.

Another aspect is to provide an entry price for users, trying to enter the market. If the market is already trading and users don't know the current price of a good, that leads to a trade with their price limit (budget or minimum costs). In some cases this can turn to less benefit than in the case that they adapt their limit price to the current market price like shown in Figure 1. But with the information about the current price a client can enter a market and apply directly its own strategy.

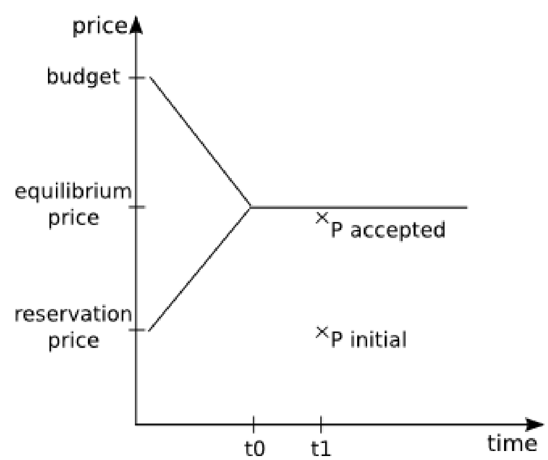

Fig. 1. Market finding its equilibriums price P_0 at t_0. New seller enters at t_1 and will get probably the lowest price.

Trader agents that solve either simple or sophisticated problems need at least some information about the market or about offers from other traders. This supposes no problem in a central organized auction because an auctioneer with an overall knowledge can transmit information to the trader agents. But in decentralized auctions or bargaining, other strategies need to be applied to ensure scalability of information dissemination. Examples of such strategies are dividing auctions in subgroups [26] or approaches like Catallaxy-based bargaining [27]. In such contexts the overall knowledge gets lost and no accurate information is available (e.g., about the price of all offers in the market).

Until now well-known and successful trading strategies are based on central auctions. One example is the ZIP strategy [28] with its derivations. If these strategies are applied in the decentralized auction mechanism they might build clusters with different equilibriums price. This can be caused by a non global view and when some groups, often trading among each other have different prices. For example when traders from Europe are trading always in the same auction and traders from Asia or US are trading always in another auction. Moreover traders preferring certain auction mechanisms might build "isolated price clusters".

There are approaches to oppose this price isolation within clusters. Ogston's approach [26] uses the exchange among the individual clusters. By selecting always the worst offers and transferring them to other clusters, they will move the extreme price. The equilibriums price of each cluster will move to one global equilibriums price in the end. However, in some cases concerning the security or type of auction mechanisms the traders wont move to other clusters.

IV. Requirements

This section outlines requirements for the DMIS. First, requirements are conducted from the demands of economic information on a market. Second, time-sensitiveness and privacy introduce new categories of requirements. Finally, the technical difficulties and requirements concerning the development of a distributed system are illustrated.

A. Economic Requirements

Our analysis of the economic market parameters are based on different work and resources of the economic field [6] [29] [22] [9] and suggests us to classify the parameters in the following categories. According to the level of aggregation we divide them into *single* and *aggregated* parameters. Depending on their complexity they are divided into the four categories: *basic*, *composed*, *complex* and *comments*. *Basic* parameters are simple values like the price, identified by its currency or the quantity measured by units. *Composed* values are constructed by two or more basic values. *Complex* values are more sophisticated economic measurements in the sense that they need to be computed from several composed values. Finally, the *comments* are user generated information like reviews. Table I summarizes an extract of required parameters classified into the proposed categories.

B. Temporal Information Occurrence

The time-related occurrence of the information is an important aspect for the technical implementation. The information might be made persistent for later requests or the information system has to inform the traders immediately after the occurring of a market event. We propose to differentiate further the economic information identified in the Table I in *time-sensitive*, *current* and *historical* information:

- **Time-sensitive information** is pushed from the market to interested traders. This has to *subscribe* a notification service which sends a message after an occurring event. A new event is for example the entering of a new product into the market or a price fall of over 20%.
- **Current information** describes the actual state of the market and should be pulled by the traders with a *query*. For example requests for the minimum price of a product or for the average volume of traded products can acquire actual information.
- **Historical information** is mainly *archived* data. Price charts for a certain product about the last 6 month or statistics about the behavior of participants like the preferred trading time of the agents belong to this category.

C. Information Privacy

Confidentiality of information about traders plays an important role in a DMIS. The system has to ensure that only authorized user data is published. However, some user's information like the price of sold products and its last offers must or should be public to ensure an efficient trading. In [9] two kind of information privacy are introduced; *public* and *private*. But there is also a discussion about the need and cost of information [7], which suggest us to split up the category *private* into a third category, *personal* (Figure 2).

- **Public market information** is essential for the trading and is visible and accessible for all agents in the market. Examples are `offers`, `shouts` and `agreements`.
- **Private market information** has a restricted access for participants. Values like `reputation` or `free capacity` can only be accessed by agents which bring a counter value like sharing their private data.
- **Personal data** like the `budget`, `pay-off` or a detailed `portfolio` is only accessible by the trader itself because they are only comprised in their own strategy and other traders could abuse of that information.

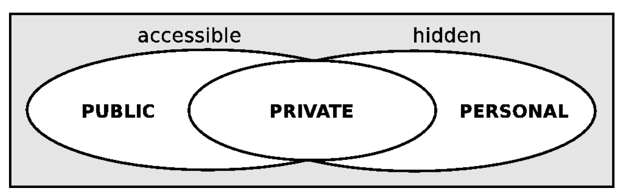

Fig. 2. Privacy model for economic information in markets.

D. Technical Requirements

This section shows technical constrains concerning the time sensitivity of the DMIS. The technical organization has a trade-off between higher installation costs (subscription setup process and the maintaining of the subscribers) or information providing (executing a query and receiving information; disseminating an event to subscribers).

It is obvious that the information providing is more time sensitive than the setup time. An execution to provide information needs to pass several nodes and calculate an aggregated value. If there were hundreds of nodes to pass, it would take

TABLE I
ECONOMIC MARKET PARAMETERS CLASSIFIED DEPENDING ON THEIR COMPLEXITY AND AGGREGATION.

Category	*Single*	*Aggregated*
Basic	price, quantity, reputation, quantity, distance, budget, shout, offer, free capacity	maximum, minimum, average
Composed	volume, variation of values, trader payoff	top/flop, standard variation
Complex	Price-Earnings Ratio, identical seller/buyer	ROI, Pareto efficiency
Comments	Expert reviews, advertisements, recommendation of alternatives	

several minutes depending on its network properties. This could cause that acceptance of a client for the waiting time is exceeded. To solve this problem we need to introduce more setup messages which organizes the nodes in a manner that the execution messages are reduced.

We believe that it is impossible or at least unrealistic to create a large-scale deployed market information system with several thousands of users, which guarantees hundred percent exact information. This is already impossible as querying and providing information needs at least several seconds. In the meantime the actual value of the information can already be changed.

Moreover, providing exact information needs a higher cost in messages. Distributed systems need to be robust against failures caused by the churn of users or in this case traders. A high robustness is generally reached by replicating messages. For example using multiple trees instead of normal binary trees for the aggregation can cause such an overhead on messages. However this results increase the robustness against failures in the system [30].

Another aspect is to provide the information within a certain time constrain. Querying thousands of participant requires some time, even if the system uses optimized aggregation and filtering algorithms. However, there is the possibility to provide previously the information, which might be requested. Again this causes a high amount on messages, which might be unnecessarily sent by anticipating possible demands of traders. Therefore one possibility is to provide information to high requested information. Also the system has to adapt dynamically to these changes of *read-dominated* (average price for a common product) or *write-dominated* (attributes which change often its value but is rarely requested) attributes [31].

Finally, it can be said that it will not be efficient to search a total guarantee of exactitude. Furthermore, it seems to be obvious to search a trade of between high guarantees and realistic amount of messages. Therefore, the aid of prediction related on former data might reduce the message overhead and increase at the same time the exactitude of information.

V. ARCHITECTURE

The technical challenge for a decentralized market information system is to meet the economical requirements in combination with the technical requirements of distributed systems. The economical side needs the disclosure of aggregated and individual data. Moreover some information has to be accomplished in a high time-sensitivity. But on the other side the technical realization has to cope with a high churn in distributed systems and to scale in regard to traders and products.

A. Layers

The proposed DMIS architecture consists of the four layers Market Information System (MIS), DMIS, Routing and DHT, which is described in the Figure 3. Each layer of the architecture has to cope with different technical or economical requirements. The MIS layer can provide a communication among a virtual organization (VO) which can allow a separation in other overlays. The Routing layer provides the routing functionalities which used by the DMIS to provide its API. The DHT layer is the main component to cope with the scalability and robustness of the system.

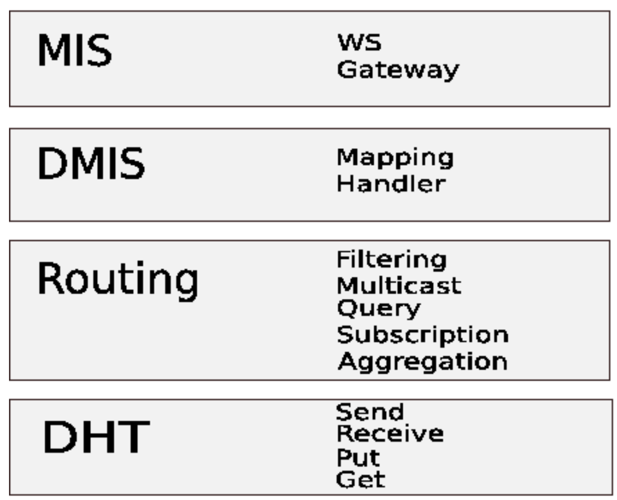

Fig. 3. Architecture Layers Building the DMIS.

The **DHT layer** operates as the communication layer and thereby it builds the basis for the DMIS system. A Kademlia-based DHT [32] is implemented in the actual prototype. But it will be designed to be flexible to switch between different types of DHTs. Some DHT solutions are already implementing publish-subscribe or multicast functionalities. Therefore a trade-off between reusing the existing models and developing new algorithms has to be made. To provide a higher flexibility and the need of unique structures force the DMIS routing structures to base on send, receive, put and get.

The core functionalities of the system are processed in the **Routing layer**. The current prototype provides the main components of the following functionalities:

- **Filtering:** is the filter based routing. Messages will only be forwarded to nodes which are interested in these. For example an event with a price of 4 is only sent to nodes interested in events with a price higher than 3. This mechanism is already implemented by content-based publish-subscribe systems.
- **Multicast:** sending messages to only a subgroup of nodes. The general multicast is easier to handle when each client knows all members of the same topic, but it is less scalable. Therefore the multicast will be changed to an algorithm where a new node takes a certain place in the tree and knows only the direct parent(s) and children.
- **Query:** This function enables to execute a query for a *read-dominated* value within the marketplace. Therefore it follows either an epidemic structure, binary-tree structure or multi-tree structure [30]. The current prototype implements a query in a binary-tree in an ordered subset of clients. An objective is to implement multiple-trees, which have a higher robustness but also increase the amount of messages.
- **Subscription:** This is the process to join a certain topic or content, and accordingly to obtain interested information. Actually the prototype follows a subscription to a certain topic, but we are looking to change to a content-based subscription to complete fully the requirements of the DMIS.
- **Aggregation:** The gathering of information need an aggregation to decrease the amount of messages and to provide scalability. As some aggregations are simple (maximum, minimum), exists more sophisticated aggregations like the calculation of an average price. Other queries are even more complex, when a combination of more parameters is requested (select price where storage $>$ 100 GB and memory $>$ 3 GHz). This concerns especially the filtering algorithms.

The **DMIS layer** provides and coordinates the core functionalities for the trader or the client. Therefore it provides several handlers like the SubscriptionHandler, QueryHandler or RequestHandler for messages returning or entering the trader. The trader can invoke methods provided by the API like subscribe, query or publish. But also non-functional methods are provided like bootstrap net to be called.

Additional functionalities are provided by the **MIS layer**. This can be necessary for the integration into different projects. It will provide one or more gateway nodes connected to the DMIS. For example interconnecting via Web Services to traders in the same VO can fulfill this service. The MIS layer is not concerning the functional behaviors of the DMIS it represents an adapter to the environment. For example it could bes the Web Service adapter for the DMIS.

B. API

This section illustrates the API and its main methods which can be called by a market participant. Furthermore there are methods for the setup and maintaining like the bootstrapping for the structured overlay. Several handlers need to be implemented to handle the incoming notifications. Events encapsulate the market information to be transferred to interested traders or markets. Patterns defines the value in which the participant is interested and it defines also its constrains.

- **public boolean query (QueryHandler handler, Pattern pattern, long timeout) throws DMISException;** calling this method executes a request for a value like the price in the DMIS. The result will be sent to the notify method of the assigned QueryHandler. The query follows a pattern which filters the search for aggregated values (minimum, maximum or average) and to narrow the search to a constrain (price $<$ 100). The timeout defines the maximum duration of a query. A DMISException will be raised when the maximum duration is passed. The method returns false if the query already exists.
- **public boolean subscribe (SubscriptionHandler handler, Pattern pattern) throws DMISException;** this method subscribes a client to a certain topic. The pattern describes in which events the client is interested in. There is also the possibility that both possibilities are valid and the client can choose between them. If an event happens in the DMIS, which conforms the filter or topic, then the notify message of the assigned SubscriptionHandler will be called. If there is already an existing subscription with the same filter/topic, then it returns false.
- **public boolean unsubscribe (SubscriptionHandler handler, Pattern pattern) throws DMISException;** identified by the handler and patter, the trader or participant will be unsubscribed from the content subscription. It returns true if it is successfully removed and false when there is no subscription for that filter/topic.
- **public void publish (Event event) throws DMISException;** if an event happens within a client, then it has to inform the DMIS. Therefore it calls the publish methods which transfers the assigned Event to the DMIS and that way to interested subscribers.

VI. Preliminary Measurements

We have implemented a prototype in Java on which simulations are to proof the proposed concept. Furthermore the measurements show the scalability in terms of the number of messages in regard to the number of participants. The process of retrieving data from all clients is investigated in detail. For scalability reasons the values are aggregated at each node and sent back to the node which started the request. In the measured process one participant send a request to nodes subscribed to certain topic. The request is, in this example, the existing `maximum price` from the participants. The returned values are aggregated at each node and transmitted to a parent node.

We run two experiments on one machine with changing the number of traders between 50, 100, 150 and 200. In these experiments all sent messages where count, also messages are for the setting up and the initial subscription to topics. Figure 4 shows that the amount of messages increases like expected with the number of requested nodes. This figure shows two different type of experiments. In the first curve, all traders have subscribed to the topic `price` and in the second only each second trader has subscribed. When all participants subscribed the messages increase by 40 when the traders are increased from 50 to 100. But increasing the amount of traders from 100 to 200 increases the messages only by about 50 messages.

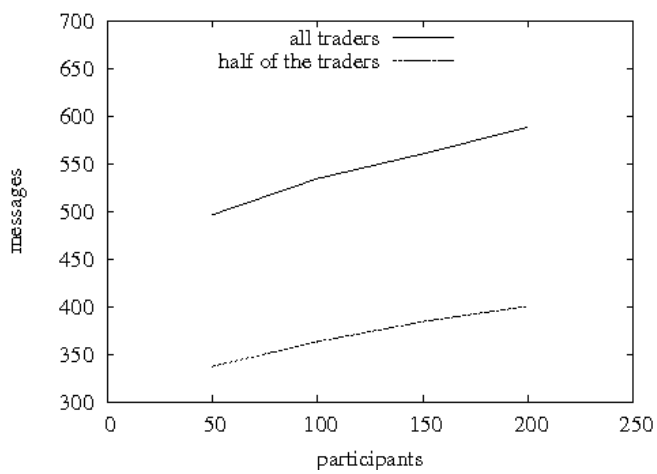

Fig. 4. Measurement of scalability with all participant subscribed to be queried and the half of them subscribed.

VII. Conclusion

After presenting the motivation for building a DMIS, we identified and analyzed the economic information requirements for markets in regard to technical aspects. The focus was thereby on existing Grid market projects. Therefore we presented and categorized economic data in relation to their level of aggregation and complexity. Then the different access possibilities are outlined depending on the importance of the time-based occurrence and information privacy.

Secondly, we sketched a high-level architecture overview for the DMIS. The designed architecture of the prototype provides the possibility to be integrated into well-known Grid markets developed by other projects. The prototype aims to fulfill both the technical and the economic requirements by providing service components based on structured overlay networks.

Finally we prove our proposed concept in form of simulations. A from of scalability was shown through a measurement of all sent messages. We found out that the sent messages increase stable when the amount of participants is increased. However, these measurements are primarily to proof our concept.

VIII. Future Work

The future research within building the DMIS focuses on further analysis of the impact of information to auctions. The investigation will analyze the benefit of traders fed with external information and the benefit of traders only knowing their own information like their past trading prices of their products.

In regard to the technical realization, there are possibilities to improve the performance. For example an advanced filter-based routing can disseminate more efficiently the messages to the subscribers. But difficulties in combination with complex queries has to be considered as these interfere with the filtered routing.

Another work will be about the Quality of Service (QoS). A high level on guarantees results in more 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 has to be outlined.

IX. 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] Grid4All, 2007. [Online]. Available: http://grid4all.elibel.tm.fr/

[2] GridEcon, 2007. [Online]. Available: http://www.gridecon.eu/

[3] K. Lai, B. A. Huberman, and L. Fine, “Tycoon: A distributed market-based resource allocation system,” HP:arXiv:cs.DC/0404013, Tech. Rep., 2004. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0404013

[4] SORMA, “Sorma project,” 2007. [Online]. Available: http://www.iw.uni-karlsruhe.de/sormang/

[5] T. Eymann, M. Reinicke, W. Streitberger, O. Rana, L. Joita, D. Neumann, B. Schnizler, D. Veit, O. Ardaiz, P. Chacin, I. Chao, F. Freitag, L. Navarro, M. Catalano, M. Gallegati, G. Giulioni, R. C. Schiaffino, and F. Zini, “Catallaxy-based grid markets,” *Multiagent Grid Syst.*, vol. 1, no. 4, pp. 297–307, 2005.

[6] D. Bergemann and J. V. (2006), “Information in mechanism design,” in *Proceedings of the 9th World Congress of the Econometric Society*, ser. Chapter 5, W. N. Richard Blundell and T. Persson, Eds., vol. Cambridge University Press 2007, 2007, pp. 186– 221.

[7] M. O. Jackson, “Efficiency and information aggregation in auctions with costly information,” *Review of Economic Design*, vol. vol. 8, no. 2, pp. 121–141, 2003.

[8] D. Bergemann and J. Valimaki, “Information acquisition and efficient mechanism design,” *Econometrica*, vol. 70, no. 3, pp. 1007–1033, May 2002, available at http://ideas.repec.org/a/ecm/emetrp/v70y2002i3p1007-1033.html.

[9] J. Grossklags and C. 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*. Washington, DC, USA: IEEE Computer Society, 2003, p. 400.

[10] R. Van Renesse, K. P. Birman, and W. Vogels, “Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining,” *ACM Trans. Comput. Syst.*, vol. 21, no. 2, pp. 164–206, May 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=762485

[11] R. van Renesse and A. Bozdog, “Willow: Dht, aggregation, and publish/subscribe in one protocol.” in *IPTPS*, ser. Lecture Notes in Computer Science, G. M. Voelker and S. Shenker, Eds., vol. 3279. Springer, 2004, pp. 173–183. [Online]. Available: http://dblp.uni-trier.de/db/conf/iptps/iptps2004.html

[12] K. Albrecht, R. Arnold, M. Gähwiler, and R. Wattenhofer, "Aggregating information in peer-to-peer systems for improved join and leave." in *Peer-to-Peer Computing*, 2004, pp. 227–234.

[13] Z. Zhang, S. M. Shi, and J. Zhu, "Somo: Self-organized metadata overlay for resource management in p2p dht," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA, USA, 2003.

[14] R. Bhagwan, P. Mahadevan, G. Varghese, and G. M. Voelker, "Cone: A distributed heap approach to resource selection," Tech. Rep., 2004.

[15] P. Yalagandula and M. Dahlin, "A scalable distributed information management system," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 379–390, 2004.

[16] A. I. T. Rowstron, A.-M. Kermarrec, M. Castro, and P. 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*. London, UK: Springer-Verlag, 2001, pp. 30–43.

[17] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman, "An efficient multicast protocol for content-based publish-subscribe systems," *icdcs*, vol. 00, p. 0262, 1999.

[18] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Trans. Comput. Syst.*, vol. 19, no. 3, pp. 332–383, 2001, siena.

[19] A. Gupta, O. D. Sahin, D. Agrawal, and A. E. Abbadi, "Meghdoot: content-based publish/subscribe over p2p networks," in *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., 2004, pp. 254–273.

[20] Y. Zhu and Y. Hu, "Ferry: An architecture for content-based publish/subscribe services on p2p networks," in *ICPP '05: Proceedings of the 2005 International Conference on Parallel Processing*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 427–434.

[21] P. R. Pietzuch and J. Bacon, "Peer-to-peer overlay broker networks in an event-based middleware," in *DEBS '03: Proceedings of the 2nd international workshop on Distributed event-based systems*. New York, NY, USA: ACM Press, 2003, pp. 1–8.

[22] Google, "Google finance," 2007. [Online]. Available: www.finance.google.com

[23] Yahoo, "Yahoo! finance," 2007. [Online]. Available: http://finance.yahoo.com/

[24] Reuters, "Stocks information," 2007. [Online]. Available: http://www.investor.reuters.com/

[25] Stockgroup, "Stockgroup," 2007. [Online]. Available: http://www.stockgroup.com/

[26] E. Ogston and S. Vassiliadis, "A peer-to-peer agent auction," in *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM Press, 2002, pp. 151–159.

[27] CATNETS, "Annex i description of work. ist-fp6-003769," 2004. [Online]. Available: http://www.iw.uni-karlsruhe.de/catnets/

[28] C. Preist and M. van Tol, "Adaptive agents in a persistent shout double auction," in *ICE '98: Proceedings of the first international conference on Information and computation economies*. New York, NY, USA: ACM Press, 1998, pp. 11–18.

[29] M. Bichler and A. Segev, "A brokerage framework for internet commerce," *Distrib. Parallel Databases*, vol. 7, no. 2, pp. 133–148, 1999.

[30] M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani, "Estimating aggregates on a peer-to-peer network," Stanford University, Tech. Rep., 2003.

[31] P. Yalagandula and M. Dahlin, "Shruti: A self-tuning hierarchical aggregation system." in *SASO*, 2007, pp. 141–150.

[32] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer-Verlag, 2002, pp. 53–65.