

Assessing a Distributed Market Infrastructure for Economics-Based Service Selection*

René Brunner¹, Isaac Chao¹, Pablo Chacin¹, Felix Freitag¹, Leandro Navarro¹,
Oscar Ardaiz², Liviu Joita³, and Omer F. Rana³

¹ Computer Architecture Department, Polytechnic University of Catalonia, Spain
{rbrunner, ichao, pchacin, felix, leandro}@ac.upc.edu

² Department of Mathematics and Informatics, Public University of Navarra, Spain
oscar.ardaiz@unavarra.es

³ School of Computer Science and the Welsh eScience Centre, Cardiff University, UK
{l.joita, o.f.rana}@cs.cardiff.ac.uk

Abstract. Service selection is an important issue for market-oriented Grid infrastructures. However, few results have been published on the use and evaluation of market models in deployed prototypes, making it difficult to assess their capabilities. In this paper we study the integration of an extended version of Zero Intelligence Plus (ZIP) agents in a middleware for economics-based selection of Grid services. The advantages of these agents compared to alternatives is their fairly simple messaging protocol and negotiation strategy. By deploying the middleware on several machines and running experiments we observed that services are proportionally assigned to competing traders as should be in a fair market. Furthermore, varying the environmental conditions we show that the agents are able to respond to the varying environmental constraints by adapting their market prices.

Keywords: Automatic Resource Allocation, ZIP Agents, Decentralized Economic Models, Service Oriented Grids.

1 Introduction

Grid Computing leverages the power of thousands of resources distributed across computers/supercomputers/clusters linked by networks (from Intranet to the Internet). Through the concept of Virtual Organizations (VOs), the Grid enables the dynamic composition of such resources into interoperable services, which multiply exponentially the VOs added value. However, given the unpredictability of the underlying platform (Internet), scalable realization of such synergies (in both physical and organizational levels) poses serious challenges to modern large scale distributed systems research. Contrarily to other distributed systems, Grids have many independent resource providers with varying access policies. In

* This work was supported in part by the Ministry of Education and Science of Spain under Contract TIN2006-5614-C03-01, and the European Union under Contract CATNETS EU IST-FP6-003769.

addition to large sizes, the diversity of policies leads to a complex allocation task that cannot be handled manually by users. Automatic and adaptive resource management is the solution to these challenges.

The primary visions for Grid computing are utility computing infrastructure and Grid services/service providers. In utility computing, a third party service provider hosts and manages the Grid solution dedicated to serving a single organization or the needs of multiple ones. Customers only pay for the used resources. Grid services/service provider's modularity enables the dynamic composition and coordination of e-services which can be exchanged or traded between Grid users or brokers, following the usage models from utility computing. These two features enable for the automatic trading of Grid Services in Service Oriented Grids (SOGs).

There has been recently an increase of interest in SOGs within the Grid community towards services that are often considered as a natural progression from component based software development, and as a mean to integrate different component development frameworks. A service in this context may be defined as a behavior that is provided by a component to be used by any other component. A service stresses interoperability and may be dynamically discovered and used [1]. Utility computing assumes service instances are created on the fly and automatically bound as applications that are configured dynamically. The service viewpoint abstracts the infrastructure level of an application. It enables the efficient usage of Grid resources and facilitates utility computing, especially when redundant services can be used to achieve fault tolerance. A SOG system is configured on-demand and flexibly, which means different components are bound to each other late in the composition process. Thus, the configuration can change dynamically as needed and without loss of correctness. Decentralized Grid Markets based on agents have been proposed as suitable coordination mechanisms for Grids and SOAs [2]. The market here is nothing more than a communication bus it is not a central entity of its own and does not participate in matching participants requirements using some optimization mechanisms. Direct agent to agent bargaining allows participants to use the negotiation strategy more suitable to its objectives and current circumstances. Local bilateral bargaining also facilitates the scalability of the system and the quick adaptation to fluctuations in resource allocation dynamics. This enables for high scalability in both physical and organizational levels. These concepts have been capitalized in [3] for SOG purposes.

In this paper, we address the design and economic behavior of an economic SOG infrastructure and its evaluation based on the Grid Market Middleware (GMM), a resource allocation middleware which incorporates decentralized economic models [4]. We show an economic adaptability of the agents that incorporate a decentralized economic algorithm based on an extension of the ZIP Agent [5]. The experiments prove a seamless integration of the economic components. Furthermore, economic reaction to environmental change in the resources type or the demand is shown. The proposed economic matchmaking mechanism has the advantage to scale easily to large and dynamic environments, keeping the ability

to provide important information and shouted prices from other agents, which rest incertitude or an approximation in alternative decentralized approaches.

2 Related Work

Economy based resource allocation has received a great deal of attention in the last years. The GridBus Project [6] is a reference in SOG and utility based computing, and has proposed a great variety of market models and tools for the trading of Grid Resources. However, its strong emphasis on computational intensive Grids and the hierarchical nature of some of the proposed components, like the Grid Market Directory, diverges from the fully decentralized resource allocation mechanism proposed here. Centralized approaches exist such as [7], but scalability issues both in size and computational requirements further complicate its applicability to large size Grids. Tycoon [8] is a market-based system for managing compute resources in distributed clusters or Grids. It uses distributed auctions with users having a limited amount of credits. Users who provide resources can, in turn, spend their earnings to use resources later.

A few papers address fully decentralized market mechanisms for computational resources. In [9], a peer-to-peer (P2P) double auctioning mechanism is proposed which builds on Zero-Intelligence agents (ZIP Agents) [5]. It was shown that the results with original ZIP agents in continuous double auctions (CDAs) depend strongly on the availability of the complete set of bid and offers coming from all buyers and sellers, and the commitment to winner-to-winner allocations. But a P2P or fully decentralized trading mechanism must be free of any central authority for scalability reasons. To oppose these problems Ogston [10] propose a P2P agent auction with centralized clusters. This offers complete information about other traders in the same cluster, it assures price stability and it copes as well with scalability issues of distributed systems. Another fully decentralized approach is the one adopted with the catalactic agents [11]. In this approach bilateral negotiations are established between a set of learning agents, and the spontaneous coordination arises from both the bargaining and co-evolutionary learning processes.

However, none of these approaches provide the infrastructure for integrating explicitly the market based algorithms into service oriented Grids. Tycoon has been used mostly in a clusters environment, and GridBus is provided as complete software toolkit, not as a service. Our approach is to offer the economic algorithms as Web Services for a seamless integration in any SOG.

3 Service Oriented Grid Market Middleware

3.1 The Grid Market Middleware

The Grid Market Middleware (GMM) provides the mechanisms to register, manage, locate and negotiate for services and resources. It allows trading agents to

meet each other based on its requirements and engage in negotiations. Furthermore, the middleware offers a set of generic negotiation mechanisms, on which specialized strategies and policies can be dynamically plugged in. The GMM has a layered architecture (Figure 1), which allows a clear separation of platform specific concerns from the economic mechanisms, to cope with highly heterogeneous environments. A detailed description of both the design and implementation of the GMM architecture can be found in [4].

Applications interact with the GMM in order to obtain the Grid services required to fulfill the application tasks. The Base Platform supports the application by providing a hosting environment for the Grid services. When a client issues a request, the application determines which grid services are required to fulfill it. These grid services represent either software services (e.g., a data processing algorithm) or computational resources. The application service translates these requirements to a WS-Agreement format [12] which is submitted to the Grid Market Middleware. The middleware searches among the available service providers, which have registered their particular service specifications, like contractual conditions, policies and QoS levels. When a suitable service provider is found, the application requirements are negotiated within the middleware by agents who act in behalf of the service providers as sellers and the application as buyers. Once an agreement is reached between the trading agents, a grid service instance is created for the application. Afterwards a reference is returned to the application, which can invoke it.

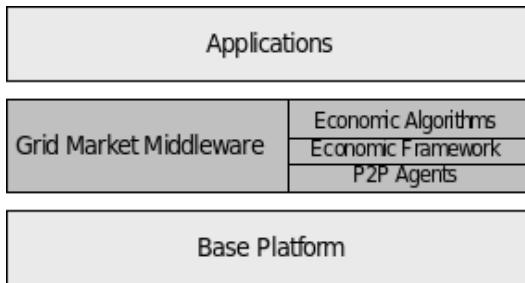


Fig. 1. Layered middleware architecture

3.2 The Extended Zero Intelligence Economic Agents

In this work we consider a simplified Grid market with only one homogeneous Data Mining service being traded. The execution time of the service can be varied during the experiments. The auction mechanism is a continuous double auction in which the agents follow a modified ZIP strategy based on [5].

In the context of the GMM, the buyer agents are called ComplexServices (CSs) and the seller agents BasicServices (BSs). CSs aggregate BSs from the market. As BSs and CSs get involved in trading, the price will evolve by the offer and the demand, with dependence on the limited CS budget and the limited resources

which can be sold by the BS. Once the BS has sold its resource to a CS, it cannot accept more bids from other CSs CFPs until the moment when the client of the awarded CS ends the execution of the sold Data Mining service in the resource.

For the realization of the decentralized continuous double auction we divide the traders in subgroups, called *bidding clusters* (see Figure 2) which are trading independently. This allows to cope with the scalability of large networks and. Moreover this approach enables the agents to be well-informed of shouts from other agents, which in decentralized auctions is a general problem [9]. To avoid that groups are only trading isolated, agents have to join and leave the clusters. The selection of individual agents to move to another cluster depends on their trading success. This method allows reaching one global equilibrium price P_0 for all clusters situated the distributed market place. As the feasibility of a global P_0 is already shown in [10], we will concentrate our prototype analysis focusing in one bidding cluster.

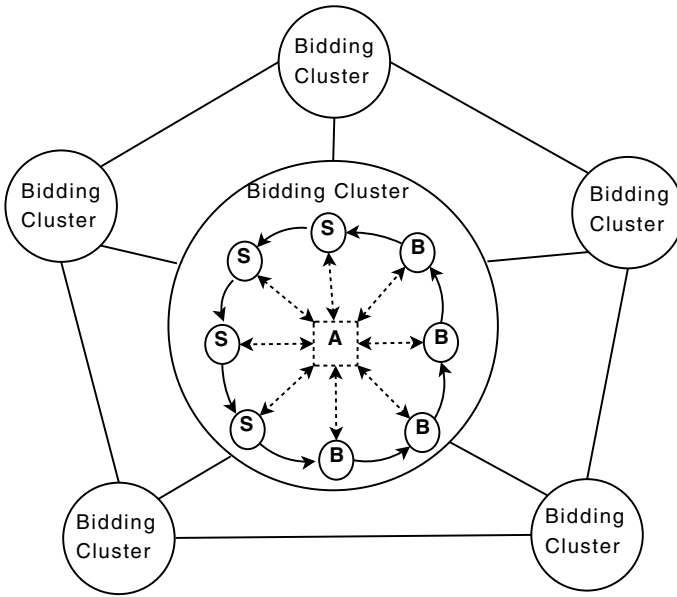


Fig. 2. Bidding clusters containing sellers S, buyers B and an auctioneer A

Each cluster deploys an own central continuous double auction. The agents are coordinated in a synchronous manner and are acting in bidding rounds. Therefore a delegated auctioneer controls the matching of the bids and offers; the highest bid corresponds to the lowest offer. No matching of a trade will be executed, if no offer exists lower than the highest bid.

3.3 Interface with Application

In a SOG infrastructure, the GMM is exposed to be accessed by applications through a convenient access point, a Web Service which can be deployed in any

application server and integrated as a service in an existent SOG. Figure 3 describes the main steps in the interaction through the access point. When a client issues a request, the application determines which Grid services are required to fulfill it. These Grid services represent either software services (e.g., a data processing algorithm) or computational resources. The application translates these requirements into a standardized WS-Agreement [12]. The application invokes the access point and passes the corresponding WS-Agreement request. This is in turn parsed and processed at the access point, which instantiates the GMM with the required economic agents to fulfill the client request.

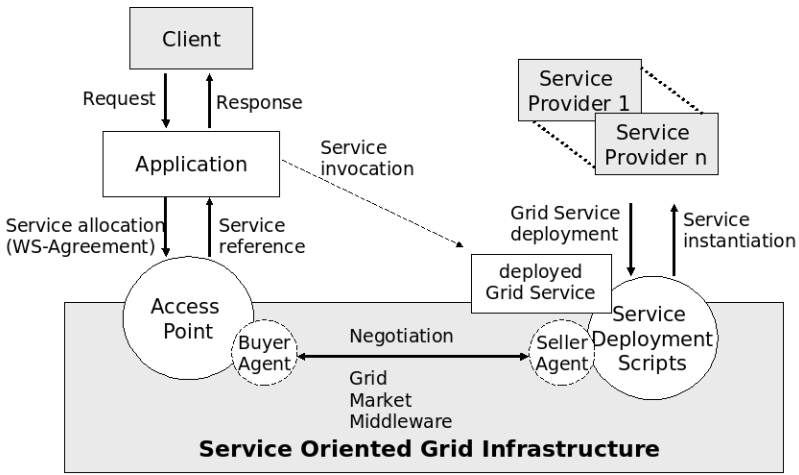


Fig. 3. Service Oriented Grid (SOG) infrastructure

The GMM searches among the available service providers, which have registered their particular service specifications, like contractual conditions, policies and QoS levels. When a suitable service provider is found, the application requirements are negotiated within the middleware by agents who act in behalf of the service providers as sellers and the application as buyers. Once an agreement is reached among the trading agents, a Grid service instance is created for the application and a reference is returned to the application/client, which can invoke it.

The server-side infrastructure is deployed by a set of scripts which allow for the bootstrapping of BSs in available resources. The scripts perform the automatic deployment and configuration of the BSs, which are then ready to be contacted by CSs. Services offered by BSs for clients executions are also deployed and exposed in Apache Tomcat application servers. Complemented by the access point, this comprises a complete infrastructure for economic-based SOGs.

4 Prototype Application

4.1 Data Mining Grid Services Application

Different types of applications can be constructed and benefit from using the GMM in the Grid, such as enabling the creation of VOs for planning, scheduling, and coordination phases within specific projects or businesses. The ability of a free-market economy to adjudicate and satisfy the needs of VOs, in terms of services and resources, represent an important feature that markets can provide. Such VOs could require a large amount of resources which can be obtained from computing systems connected over simple communication infrastructures such as the Internet.

As a proof of concept of the system model, we provide an application of the GMM with extended ZIP agents to an existing decentralized free-market prototype, the Catalactic Data Mining application [13]. The basic problem addressed by the data mining process is one of mapping low-level data (which are typically too voluminous to understand) into other forms that might be more compact (for example, a short report), more abstract (for example, a descriptive approximation or model of the process that generated the data), or more useful (for example, a predictive model for estimating the value of future cases). At the core of the process is the application of specific data-mining methods for pattern discovery and extraction. This process is often structured into a discovery pipeline/workflow, involving access, integration and analysis of data from disparate sources, and to use data patterns and models generated through intermediate stages. Selection and conversion of datasets as well as the execution of the data-mining algorithm itself are the typical required steps. In the Catalactic Data Mining services prototype, two Data Mining Services encapsulating data conversion and algorithm execution are combined in a workflow achieving a solution to the overall problem. For simplicity we restrict to the deployment of the core Data-Mining Service, and we consider the pre-processing step as given by the application.

Consider a scenario where a client issues sequential requests for Data Mining services. The CSs try to map the incoming workflows to an available set of services. The BSs, try to sell their services to the CSs which are instantiated after successful negotiation upon the client request. Figure 4 shows a scenario with two service types in the service market.

4.2 Deployment and Experimental Setup

The implemented bidding algorithm is based on extended ZIP agents, which is shown in Algorithm 1 for the seller and respectively in Algorithm 2. This allows reaching the equilibrium price P_0 , at which the maximum resources will be exchanged, with simple agents. Therefore they have to know the minimum price of the shouted offers, by sellers S_{min} and the maximum price of the shouted bids by buyers B_{max} . These two values build the basis for the agent's bidding algorithm to calculate its new price $P_{(t+1)}$. The algorithm bases on a momentum γ considering the weight of previous price changes and on the learning rate β describing the rapidity of adaption to the current target price.

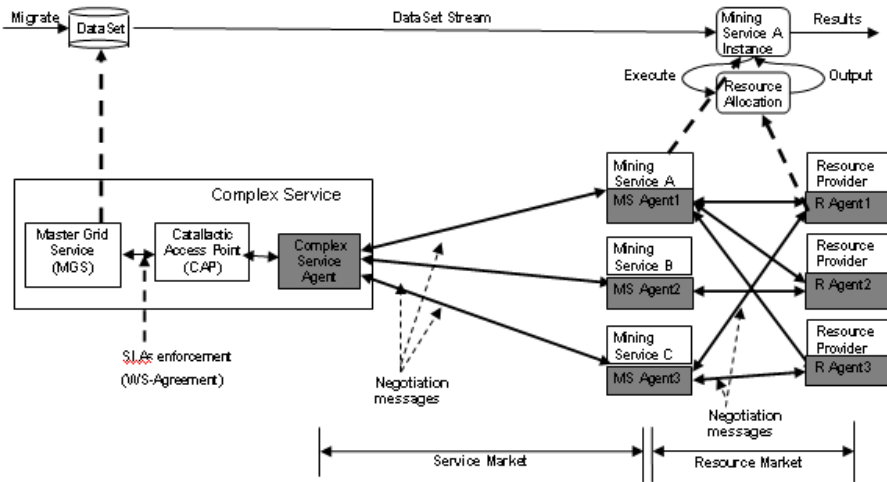


Fig. 4. Prototype Application using the SOG infrastructure

Algorithm 1. Bidding algorithm of the BS (seller)

Input: $random_1 > 0$ and < 0.2 ;
Input: $random_2 > 0$ and < 0.2 and not $random_1$;
if $S_{min} > B_{max}$ **then**
 | $P_{target} = S_{min} - (random_1 * P_{(t)} + random_2)$;
else
 | $P_{target} = B_{max} + (random_1 * P_{(t)} + random_2)$;
endif
 $priceChange = \gamma * priceChange + (1-\gamma) * \beta * (P_{target} - P_{(t)})$;
 $P_{(t+1)} = maximum(P_{(t)} + priceChange, P_{min})$;

We setup controlled experiments by deploying several instances of the GMM in a Linux server farm. Each machine has a 2 CPU Intel Xeon at 2.80 GH and 2 GB of memory. The nodes in the farm are connected by an internal Ethernet network at 100 Mps. The topology is a mesh: all interconnected. CFPs are transmitted via groupcast to all the nodes in the destination groups (in our scenario CFPs are groupcasted from CSs to BSs).

We deploy the GMM in 8 nodes. Four nodes host a BS each and the Data Mining Web Service and other four nodes host the CSs, access points and clients. The Web Services are exposed in Tomcat servers. Access for execution of these Web Services on the resource node is what is traded between BSs and CSs. The experiments consist in launching 4 clients concurrently, which use each one of the CS as broker. Each client makes requests to the CS and leaves the market after a successful trade. It will re-enter a proceeding round with the probability of $\frac{1}{3}$. Whenever a CS wins a bid with a BS, it invokes the Data Mining Service in the selected node, and the resource in the corresponding node gets locked for

Algorithm 2. Bidding algorithm of the CS (buyer)

```

Input:  $random_1 > 0$  and  $< 0.2$ ;
Input:  $random_2 > 0$  and  $< 0.2$  and not  $random_1$ ;
if  $S_{min} > B_{max}$  then
  |  $P_{target} = B_{max} + (random_1 * P_{(t)} + random_2)$ ;
else
  |  $P_{target} = S_{min} - (random_1 * P_{(t)} + random_2)$ ;
endif
priceChange =  $\gamma * priceChange + (1-\gamma) * \beta * (P_{target} - P_{(t)})$ ;
 $P_{(t+1)} = \text{minimum}(P_{(t)} + priceChange, \text{budget})$ ;

```

the duration of the service execution. We measure the selling prices of the BSs and observe the proportion of successful CFPs issued by the CSs.

5 Experiments and Evaluation

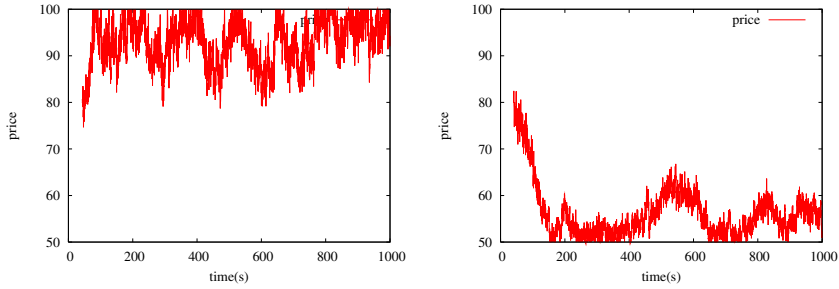
The goal of the experiments is to show the performance of the GMM as an automated economic-aware resource management tool by means of the Data Mining Grid prototype application. The extended ZIP agents are expected to show an effective and fair trading, which can be measured with the price and the allocation rate of each agent. Varying the technical parameters of the environment, we expect price adaptation of the agents in the marketplace.

5.1 Idealized Experiments with Idle Resources

The experiments are sensitive to a competitive use of other processes, because this might cause an increase of the Data Mining WS execution times. Therefore we make first experiments with idle resource, which warranties stability of Data-Mining Services execution times.

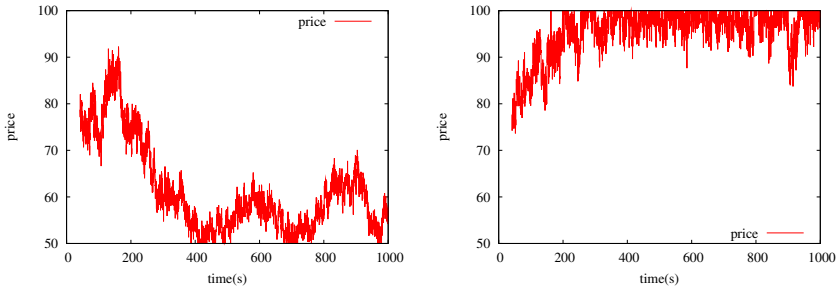
Varying the execution times of the Data-Mining Services determines the generated offer in the market. This has an impact on the equilibriums price when the demand rate is constant, which is illustrated in Figure 5. The offer is decreased in Figure 5(a) which leads to an increasing price. In Figure 5(b) the execution time is decreased to 100ms which leads to more offers and though to decreasing prices.

Besides the effect of changing the offer, also the variation of the demand for the resources needs to be proved. Therefore we change the probability that a CS reenter the market (issuing a new demand) after a successful trade. The comparison between Figure 6(a) and Figure 6(b) shows clearly the significance of the demand. In Figure 6(a) the demand rate (probability of re-entering the market) is $\frac{1}{6}$, which keeps the amount of the CS low and decreases the price. Figure 6(b) shows that the price increases when the CS re-enter the market after every successful trade (probability of 1).



(a) resource execution time of 3000 ms (b) resource execution time of 100 ms

Fig. 5. Price evolution with varying offer with and a constant demand rate of $\frac{1}{2}$



(a) demand rate $\frac{1}{6}$ (b) demand rate of 1

Fig. 6. Price evolution with varying demand rate with and a constant executionTime of 1000 ms

5.2 Adaptation to Different Constrains

The experiment in this section illustrates the adaptation of the prototype for a changing set-up environment. In this section the execution the time of the Data-Mining Services is varied to obtain real scenarios where processed input data-sets sizes might differ. To simulate such cases, the execution time of the resources will vary during the running time of the experiment. It changes iteratively, every 200 seconds the executions time from high (3000 ms) to very low (100 ms).

After a stabilization phase of about 450 seconds (phase 0), the experiment in Figure 7, shows price adaptation to varying market constrains in form of task loads (the WS execution times). From a short resource execution time (like 100 milliseconds) results that the market contains many offers. Consequently the prices of the product decreases. Contrarily, decrementing the offer by setting the execution time to 3000 milliseconds leads to an increasing price.

5.3 Process Competition

Increasing the realism of the environment, we consider an experiment were the nodes in the cluster run with other competing processes which influence the

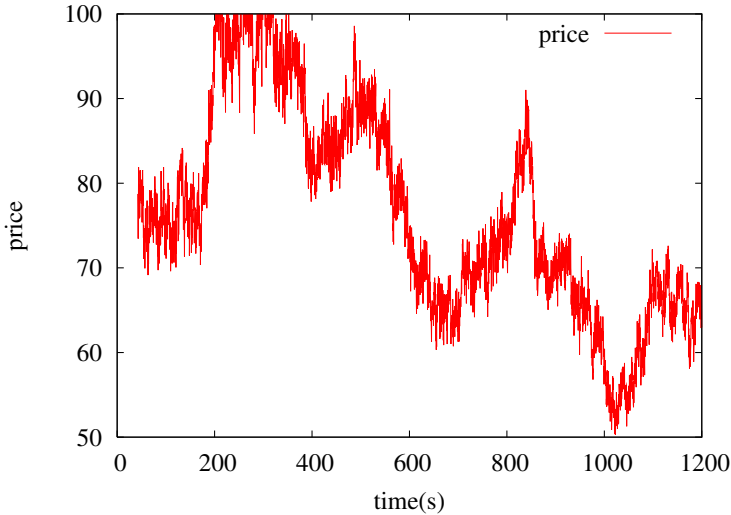


Fig. 7. Varying task load (WS execution time) dynamically. **phase 0** ($t = 0$ s - 450 s): *stabilization*; **phase 1** ($t = 450$ s - 650 s): $W_{SexecTime} = 100$ ms; **phase 2** ($t = 650$ s - 850 s): $W_{SexecTime} = 3000$ ms; **phase 3** ($t = 850$ s - 1050 s): $W_{SexecTime} = 100$ ms; **phase 4** ($t = 1050$ s - 1200 s): $W_{SexecTime} = 3000$ ms.

resource performance. This has an impact on offered resources which should be considered by the agents. We show in Figure 8 how agents effectively react to the process competition by adapting prices.

The allocation rate in Figure 9 shows the distribution of over 4000 matched trades. A nearly equal distribution of the resources to the CS can be seen as well as the nearly equal distribution of the bought BS resources can be seen. Even in a real application with uncontrolled process competition an almost fair allocation is obtained.

5.4 Evaluation

The results of the three experiments demonstrate how a simple decentralized economic algorithm based on ZIP can be plugged into the GMM infrastructure in order to allocate resources to client in service oriented applications, by achieving automatic and fair trading of resources between Grid clients and Grid service providers, mediated respectively by the CS and BS agents.

Furthermore the results show that the agents react to changes in the economic environment. The accepted price reflects the variations in demand (through demand rate) and offer (through varying execution time of the services, which results in varying resource availability). It can be seen that the price increases when the demand also increases (Figure 6) and that correspondingly the price increases when offer decreases (Figure 5), as a result of more time consumption by services. Nevertheless the distribution of allocations between buyers and sellers remain

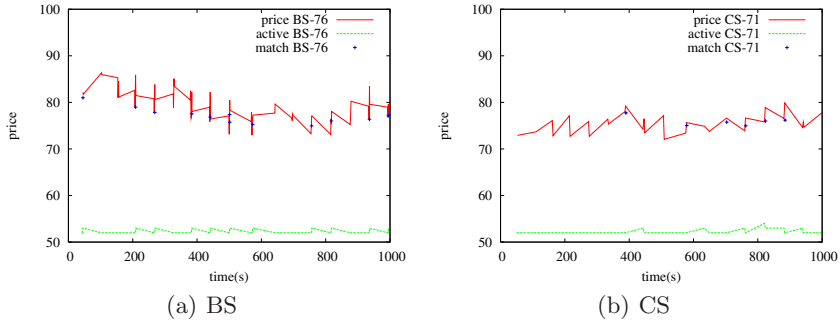


Fig. 8. Prices with competing process

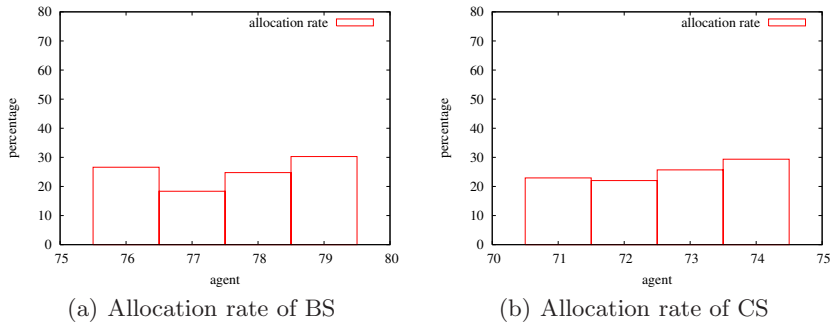


Fig. 9. Allocation rates in a experiment of competing process

proportioned (Figure 9), as expected in fair markets. Consequently it follows that the prices will increase in case of large-scale failures or delays. Moreover, this automatic price correction behavior is able to react to dynamic conditions in underlying Grid resources (Figure 7).

6 Conclusions

We have shown a complete infrastructure for economic-based SOGs and we have demonstrated its application in a Data Mining SOG prototype. The proposed infrastructure provides both the scripts for automatic bootstrapping of traded Grid services and the agents selling the services at the Grid service provider side, as well as the Web Service access point for the seamless usage of the SOG infrastructure by clients. The economic agents (employed an extension of ZIP agents) are able to operate in a decentralized environment, automatically evolving trading prices with varying offer and demand rates. This has important advantages in large scale Grids over computational costly centralized solutions, mainly scalability and feasibility in open and decentralized systems.

The experimental results in Section 5 show that agent-based trading of resources at stable prices can be achieved using the GMM. Moreover the allocation of traded resources is well-balanced among the seller agents as well as among the buyer agents. Our analysis demonstrates that the agents are stable against economic changes in their environment. The agents overcome dynamics of the system by conserving the expected offer and demand reactions in fair market.

Future work comprises the inclusion of more complex workflows and its integration/evaluation in the architecture, increasing the size of the test bed, and test the infrastructure with additional prototype applications. The need for more decoupling of individual agent behavior configuration from middleware will lead eventually to the design of an independent Economic Agents Framework plugable as a service in the current infrastructure.

References

1. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: Grid services for distributed system integration. *Computer* 35(6), 37–46 (2002)
2. Eymann, T., Reinicke, M., Streitberger, W., Rana, O., Joita, L., Neumann, D., Schnizler, B., Veit, D., Ardaiz, O., Chacin, P., Chao, I., Freitag, F., Navarro, L., Catalano, M., Gallegati, M., Giulioni, G., Schiaffino, R.C., Zini, F.: *Catallaxy-based grid markets*, vol. 1, pp. 297–307. IOS Press, Amsterdam, The Netherlands (2005)
3. Eymann, T., Reinicke, M., Freitag, F., Navarro, L., Ardaiz, O., Artigas, P.: A hayekian self-organization approach to service allocation in computing systems. *Advanced Engineering Informatics* 19(3), 223–233 (2005)
4. Ardaiz, O., Chacin, P., Chao, I., Freitag, F., Navarro, L.: An architecture for incorporating decentralized economic models in application layer networks. *Multiagent Grid Syst.* 1(4), 287–295 (2005)
5. Preist, C., van Tol, M.: Adaptive agents in a persistent shout double auction. In: *ICE 1998. Proceedings of the first international conference on Information and computation economies*, pp. 11–18. ACM Press, New York, NY, USA (1998)
6. Buyya, R., Abramson, D., Venugopal, S.: The grid economy. *Proceedings of the IEEE* 93, 698–714 (2005)
7. Schnizler, B., Neumann, D., Veit, D., Weinhardt, C.: Trading grid services - a multi-attribute combinatorial approach. *European Journal of Operational Research* (in Press, 2006)
8. Lai, K., Huberman, B.A., Fine, L.: Tycoon: A distributed market-based resource allocation system. Tech. rep. HP:arXiv:cs.DC/0404013 (2004)
9. Despotovic, Z., Usunier, J.-C., Aberer, K.: Towards peer-to-peer double auctioning. In: *HICSS 2004. Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 9*, p. 90289.1. IEEE Computer Society Press, Washington, DC, USA (2004)
10. Ogston, E., Vassiliadis, S.: A peer-to-peer agent auction. In: *AAMAS 2002. Proceedings of the first international joint conference on Autonomous agents and multi-agent systems*, pp. 151–159. ACM Press, New York (2003)

11. Eymann, T., Padovan, B., Schoder, D.: The catallaxy as a new paradigm for the design of information systems. In: Proceedings of the 16th IFIP World Computer Congress, Conference on Intelligent Information Processing (2000)
12. WS-Agreement, Web services agreement specification (2007), <https://forge.gridforum.org/sf/projects/graap-wg>
13. Joita, L., Rana, O.F., Freitag, F., Chao, I., Chacin, P., Navarro, L., Ardaiz, O.: A catallactic market for data mining services. *Future Gener. Comput. Syst.* 23(1), 146–153 (2007)