Technology Management, Economics and Policy Papers

SEOUL NATIONAL UNIVERSITY
College of Engineering
Technology Management, Economics and Policy Program

The GridEcon Platform: A Business Scenario Testbed for Commercial Cloud Services

Marcel Risch, Jörn Altmann, Li Guo, Alan Fleming, and Costas Courcoubetis

TEMEP Discussion Paper No. 2010:39

서 울 대 학 교
공 과 대 학, 기술경영경제정책 대학원과정
151-742 서울시 관악구 신림동 산 56-1

Technology Management, Economics and Policy Program
College of Engineering, Seoul National University
San 56-1, Shillim-Dong, Kwanak-Gu, Seoul 151-742 ,KOREA
Phone: ++82-2-880-9140, Fax: ++82-2-880-8389

Technology Management, Economics and Policy Program (TEMEP) is a graduate program at the Seoul National University. It includes both M.Sc. and Ph.D. programs which are integrated into the graduate program at the College of Engineering.

TEMEP Discussion Papers is intended to serve as an outlet for publishing research about theoretical, methodological and applied aspects of industrial economics, especially those related to the institute's areas of specialization, namely management of technology, information and telecommunication, health and energy industries, as well as development economics infrastructure. In particular submission of studies analyzing current technology and industry related issues and discussion of their implications and possible alternative policies are welcome. The objective is to gain insights into important policy issues and acquiring a balanced viewpoint of policymaking, technology management and economics which enable us to identify the problems in the industries accurately and to come up with optimal and effective guidelines. Another important aim with the series is to facilitate communication with external research institutes, individual researchers and policy makers.

Research disseminated by TEMEP may include views on information and telecommunication, technology management, energy, health and development economics policy, but the institute itself takes no institutional policy positions. When appropriate the policy views of the institute are disseminated in separate policy briefs. Thus, any opinions expressed in TEMEP Discussion Papers are those of the author(s) and not necessarily the institutes.

Editor: Almas Heshmati
Professor of Economics

Technology Management, Economics and Policy Program
College of Engineering
#37-306, Seoul National University
San 56-1, Shilim-dong, Kwanak-gu
Seoul 151-742 Korea

Phone: 0082-(0)2-880-5845
Fax: 0082-(0)2-886-8220
E-mail: heshmati@snu.ac.kr

The GridEcon Platform: A Business Scenario Testbed for Commercial Cloud Services

Marcel Risch[1], Jörn Altmann[1], Li Guo[2], Alan Fleming[3], Costas Courcoubetis[4]

[1] TEMEP, School of Industrial and Management Engineering
College of Engineering, Seoul National University
599 Gwanak-Ro, Gwanak-Gu, Seoul 151-742, South-Korea
marcel.risch@temep.snu.ac.kr, jorn.altmann@acm.org

[2] Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, UK
liguo@doc.ic.ac.uk

[3] Real Time Limited
Lochside House, 3 Lochside Way, Edinburgh Park, Edinburgh EH12 9DT, UK
alan.fleming@rtel.com

[4] Department of Computer Science, Athens University of Economics and Business
47A Evelpidon Str., Athens 11362, Greece
courcou@aueb.gr

January 2010

Abstract: Within this paper, we present the GridEcon Platform, a testbed for designing and evaluating economics-aware services in a commercial Cloud computing setting. The Platform is based on the idea that the exact working of such services is difficult to predict in the context of a market and, therefore, an environment for evaluating its behavior in an emulated market is needed. To identify the components of the GridEcon Platform, a number of economics-aware services and their interactions have been envisioned. The two most important components of the platform are the Marketplace and the Workflow Engine. The Workflow Engine allows the simple composition of a market environment by describing the service interactions between economics-aware services. The Marketplace allows trading goods using different market mechanisms. The capabilities of these components of the GridEcon Platform in conjunction with the economics-aware services are described in this paper in detail. The validation of an implemented market mechanism and a capacity planning service using the GridEcon Platform also demonstrated the usefulness of the GridEcon Platform.

Keywords: Cloud computing, Grid computing, marketplace for computing resources, market-based resource allocation, service-oriented architectures, utility computing, market emulation tool, market mechanism.

JEL Classification Numbers: C15, C51, C53, C61, C63, C65, D40, D44, L11, L21, L22, L86, M15, M21.

1 Introduction

In the past, many Grid research projects implemented models of markets for computing resources. Some examples include the Popcorn Market [1], the Spawn system [2], the Grid Architecture for Computational Economy (GRACE) [3], and Tycoon [4]. All of these projects focused very much on providing economically efficient resource allocations or on using various market mechanisms to sell computing resources. However, all these implementations remained sample prototypes. They could not be re-used by other projects because of its static implementation. For example, the market mechanism developed by Popcorn is not interoperable with the market mechanism developed by Tycoon. Furthermore, all these market implementations did not consider the market environment, i.e. the value-added services around the market mechanism.

To model a market environment comprehensively, we have developed the GridEcon Platform within the GridEcon Project [5] [6]. In particular, the GridEcon Platform allows evaluating economics-aware services in a commercial Cloud computing setting. The Platform is based on the idea that the exact working of such a service is difficult to predict in the context of a market and, therefore, an environment for evaluating its behavior and testing its interfaces in an emulated market is needed. The most important two components of the GridEcon Platform are the Marketplace and the Workflow Engine. The Workflow Engine component allows the simple composition of a market environment from different economics-aware services by describing the interaction between these services. The Marketplace allows the execution of different kinds of market mechanisms for trading computing resources.

We demonstrate the usefulness of the GridEcon Platform by implementing a market mechanism (continuous double auction) and a capacity planning service and integrating them into the GridEcon Platform. The easy-to-perform integration enabled us to conduct performance measurements on the market mechanism and the capacity planning component.

The remainder of the paper is structured as follows: In section 2, we give an overview of the state-of-the-art in Grid market research and economics-aware services for Cloud computing markets. In section 3, we describe the GridEcon Platform, in particular the Workflow Engine and the Marketplace component. Besides those, we introduce some value-added Cloud services that may be of use in a commercial computing resource environment. The implementation of the GridEcon Platform and a validation of the implementation is given in section 4. We conclude the paper in section 5.

2 State of the Art of Grid Market Research

This section gives an overview of previous Grid market projects and their approach to developing market environments. This demonstrates that previous Grid market implementations did not take into account the need for platforms, on which businesses can test and evaluate their services in a realistic market setting.

The Popcorn Market [1] [7] has identified three actors: buyers, sellers, and the market, which implements the market mechanism. By looking at the list of entities, it is already obvious that no support services has been considered during the implementation phase. While the market is flexible enough to support three different market mechanisms, the problem of implementing business services on top of the market has not been addressed. Therefore, this market is of limited use, if it comes to testing different business models of Cloud computing services as the effort for integrating additional services is quite high.

The Spawn system [2] presents the idea of value-added services but without considering the full economic potential and the requirements of these services. Since the implementation of these services and their integration into the Spawn environment will be challenging, this environment is of limited use if it comes to modeling the behavior of additional market services within Spawn.

The Grid Architecture for Computational Economy (GRACE) [3] has identified one central component: "The resource broker acts as a mediator between the consumer and Grid resources using middleware services. It is responsible for resource discovery, resource selection, binding of software, data, and hardware resources, initiating computations, adapting to the changes in Grid resources and presenting the Grid to the consumer as a single, unified resource" (p. 4, [3]). It can be clearly seen from the task description, that the Grid Resource Broker is a component with many functionalities, which do not have to be provided by the same entity. This would be the idea starting point for looking into services, which can be provided by various businesses competing for customers. However, since the Grid Resource Broker represents a monopolistic service provider, the GRACE implementation cannot be used to determine how the existence of different providers with different business models will affect the market. Furthermore, extending this architecture to provide different services would require significant effort which makes GRACE an unattractive choice for simulating economics-aware Cloud services.

The Tycoon market is a highly distributed market [4], in which agents take over a number of jobs for the consumer. These jobs include: (1) interpreting the consumer's preferences, (2) examining the state of the system, (3) make bids, and (4) verify that the acquired resources are available. The first job is very important to consumers, who have little or no Grid expertise and need help to determine the correct resource purchase. It is also a very challenging job, since consumers may not know how to express their preferences due to their lack of IT expertise. In such a case, the agent has to perform tasks such as capacity planning, portfolio management, and workflow analysis. All these tasks are highly complex and, thus, mean that the agents have to be very powerful to be able to perform them. The main weakness of this model is that, while agents could be provided by different providers, the idea of an agent market, in which consumers can buy agent services, has not been developed.

Beyond the Grid Markets described so far, some additional research comes from the gridbus [8] project. This has been the only project that has introduced a number of services for the user, namely a Workflow Broker [9], the Nimrod-G Broker [10], and the Gridbus Data Broker [11]. Furthermore, gridbus has developed the GridSim [12] and CloudSim [13] simulation environments. However, these environments are aimed at analyzing the effects of different resource allocation algorithms as well as the effects of resource failures. The effect of additional services cannot be easily modeled in these environments. Neumann et al described the technological and economical challenges, which must be overcome when developing a Grid market, in [19].

In conclusion, existing Grid Market research has not considered the idea of allowing an evaluation of new services added to an existing market or an evaluation of a new design of a market. In particular, a component that helps composing these services has also not been proposed.

3 The GridEcon Platform

The GridEcon Platform, the architecture of which is shown in Figure 1, consists of three tiers: the service provider tier, the marketplace tier, and the economics-aware services tier. Each of these tiers is built using a service-oriented architecture (SOA). The interaction between the tiers and between the different value-added services is definable through the Workflow Engine. In order to achieve that, the Workflow Engine provides common interfaces to services.

The marketplace tier encompasses the main trading location of this environment. It is flexible enough to allow trading of computing resources to be based on any kind of market mechanism such as auctions, bargaining, or posted prices.

The service provider tier contains all providers of services, which are traded in the marketplace. No distinction will be made of the type of services. This means that the GridEcon Platform could be used to sell access to data, Software (Software-as-a-Service), and computing resources. However, for simplicity, we focus on computing resources for the remainder of the paper exclusively.

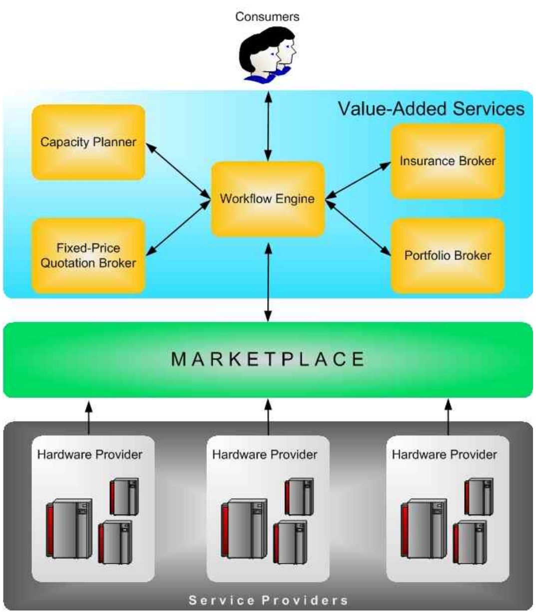

Figure 1. The GridEcon Marketplace Environment.

The economics-aware service tier is the place where different providers can offer their value-added services. Although the GridEcon project has considered a number of services, which could be of use in such an environment, in a production-level market, there may be demand for additional services.

In order to illustrate the workings of the GridEcon Platform in detail, in the following subsections, we will focus, in particular, on the Workflow Engine, the Marketplace component, and the Capacity Planning Service as an example for a value-added service.

3.2 Workflow Engine Component

The Workflow Engine is a component designed to facilitate the coordination and communication between the different components and services of the GridEcon Platform. With respect to the GridEcon architecture, the Workflow Engine sits between all other GridEcon Platform components and value-added services. It hides the underlying system complexity and the interactions within the system from end-users and, thus, makes the GridEcon Platform easier to work with. The entire system architecture for the Workflow Engine is shown in Figure 2.

The Workflow Engine is a specification-driven system, i.e. the composition of the components and value-added services is defined by feeding a specification (i.e. workflow) into the Workflow Engine. This idea was borrowed from the multi-agent domain. This sort of system design significantly reduces the effort required for developing a market environment by reducing the development life cycle. The development of a different market environment is made simple by feeding the new workflow into the Workflow Engine. Because of this design, the Workflow Engine can also manage different market mechanisms. In other words, the GridEcon Platform is independent of any specific market mechanism.

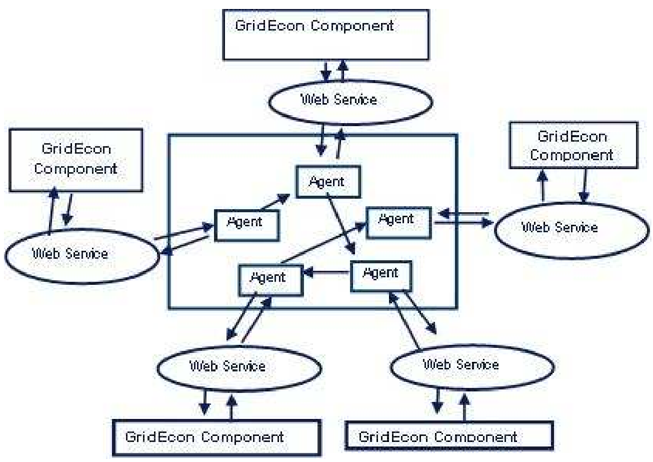

Figure 2. Architecture of the Workflow Engine

The Workflow Engine consists of a group of agents, where each agent conceptually represents a GridEcon component or value-added Service (e.g. Marketplace component, Capacity Planning service). Each GridEcon components is made public through Web services. The agents of the Workflow Engine communicate with each other to handle the interaction between the components. The interaction between agents follows a pre-defined interaction protocol, namely the current workflow. The language that is used to describe a workflow is called Lightweight Coordination Calculus (LCC) [14].

In detail, the Workflow Engine offers interfaces for users to submit asks and bids. It also offers various callback interfaces, which can be used by the economic-aware service to communicate with each other. These callback features of the Workflow Engine ensure that all communication can be performed asynchronously.

3.3 Marketplace Component

The Marketplace component is responsible for managing the trading of resources. Since computing resources can be traded based on different market mechanisms, the Marketplace can be implemented with a number of different market mechanisms. For validating the Marketplace implementation, a continuous double auction (CDA) has been implemented as the market mechanism [6]. This market mechanism has been chosen, since it is similar to market mechanisms for other non-storable goods, such as electricity [15]. With respect to the Marketplace Component itself, this market mechanism has been chosen arbitrarily.

The underlying principle for this market mechanism is that of a standard futures market: All parties announce the maximum price they are willing to buy for (i.e. bid price, for the purchasers) and the minimum price they are willing to sell for (i.e. ask price, for the sellers) in a specified time interval. These prices are recorded and put in a directory. Bids are, in fact, standing (or limit) orders and will not be immediately fulfilled unless there is a previously posted compatible reciprocal ask. If the bid price of a buy order exceeds the ask price of a sell order, the orders are immediately executed. If no compatible reciprocal bids are available, the bids remain in the directory until they are matched in the future or expire. The market mechanism takes into account both vertical and horizontal atomicity of bids and asks. Vertical atomicity in bids means that resources have to be provided by a single provider, however, the bidder is willing to switch resources during the usage period. Horizontal atomicity means that the resources are available for the entire usage duration by one provider, thereby minimizing switching costs. However, not all resources have to be offered by the same provider. For asks, vertical atomicity requires that all resources have to be sold to one user at a time. However, the resources can be sold to different users at different times. The horizontal atomicity requirement for asks means that the resources have to be sold for the entire availability time. This differentiation of atomicity gives buyers and sellers of computing resources the opportunity to specify whether they require all resources in one place, the resources for the entire duration, or both.

The architecture of the Marketplace Component is shown in Figure 3. Since the CDA mechanism was implemented in the GridEcon Platform for testing purposes, this market mechanism has been highlighted. However, other market mechanism, such as described in [20], could also be implemented.

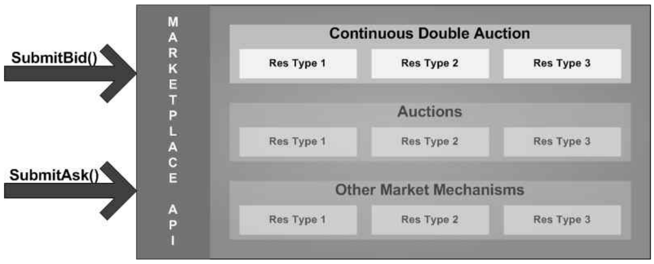

Figure 3. The Marketplace Component Architecture.

As Figure 3 indicates, the architecture of the Marketplace Component also allows trading of different types of computing resource types (e.g. Res Type 1, Res Type 2). However, the interface of the Marketplace component remains unchanged, independent of the market mechanism and the resource type.

3.4 Value-Added Services

The GridEcon project has identified a number of services, which may be of interest to resource consumers. All of these services have been examined so that the GridEcon Platform could potentially integrate any implementation of those.

The first service is the Capacity Planning Service, which helps users to plan their computing resource capacity. Capacity planning is a difficult task in a commercial Cloud computing setting. The service has to analyze the existing infrastructure, the deployed applications, and the resource requirements before suggesting a course of action. An action can be the purchase of Cloud resources, the purchase of in-house resources, or a combination of both. Furthermore, the Capacity Planning Service can suggest that some resources can be sold on the Marketplace if they are idle [16]. More detailed information about the Capacity Planning Service can be found in [17] [18].

The Fixed-Price Quotation Broker allows selling fixed price contracts even if the underlying market price for the resource is variable. This allows consumers to plan the Cloud computing costs with certainty. At the same time, this price will usually be higher than the actual price, since the Fixed-Price Quotation Broker is expected to generate profits for the provider. At the time when the contract is due, the Fixed-Price Quotation Broker places bids on the marketplace to buy the resources in order to fulfill its contractual obligations.

The Portfolio Broker addresses the fear of Cloud users that their Cloud resources might fail. Since it is very difficult for Cloud users to determine which resources have a high risk of failure, they can use the Portfolio Broker to determine the failure risk of each resource in their resource portfolio. Based on this analysis and the user’s risk tolerance level, the Portfolio Broker can then suggest a method for reaching the risk tolerance level of the user. This method may require the diversification of resources to different vendors or the purchase of spare resources.

The Insurance Broker provides insurances against a number of events that consumers fear. The GridEcon project has identified the following events which would be harmful to the Cloud user: (1) Non-delivery of the resources, (2) delivery of incorrect resources or (3) interruption of the resource availability. Using the Insurance Broker, consumers can protect themselves against the financial effects of these events.

In return, the Insurance Broker will charge a premium which is proportional to the cost of the adverse event and the probability that this event will occur.

Since all these services can be used with a number of market mechanisms, we have analyzed which services are useful with which market mechanism. The result of this analysis is shown in the following table: a √√ shows that the component is very useful; a single √ shows that the component is moderately useful, and an × shows that the component is not useful with respect to the market mechanism.

Table 1. Service Comparison with Respect to the Market Mechanism.

	Market Mechanism			
Service	Posted Price	Bargaining	Dutch Auction	Double Auction
Capacity Planning Service	√√	√√	√√	√√
Fixed-Price Quotation Broker	×	×	√√	√√
Portfolio Broker	√√	√√	√	√
Insurance Broker	√√	√√	√	√

The Capacity Planning Service is very useful regardless of the market mechanism, since its task is to help the consumer to decide which resources should be purchased and where to purchase them. Since this problem has to be solved regardless of the market mechanism, this component is useful in any utility computing market.

The Fixed-Price Quotation Broker requires demand-based pricing, which is given when Dutch and double auctions are used. While posted prices can also be adjusted to reflect the demand, this occurs less frequently and, therefore, the Fixed-Price Quotation Broker will be less useful in such a market environment. For the same reason, it is also less useful in a bargaining market.

The Portfolio Broker is useful in a non-anonymous market mechanism, such as the posted-price and bargaining markets. This is due to the fact that in such a market it can calculate the resource reliability not only for each resource type but also for each resource type of each provider. In an anonymized market, it can only calculate the resource reliability for each resource type across all providers.

As the Portfolio Broker, the Insurance Broker is also more useful if the market mechanism is not anonymous, i.e. if providers can be uniquely identified. In this case, the Insurance Broker can determine more closely which provider is trustworthy.

4. Implementation and Validation

All GridEcon Platform components and value-added services described in the previous sections can be orchestrated by the user through the Workflow Engine. Therefore, a typical GridEcon-Platform-based market environment always relies heavily on this central component. Figure 4 shows a sequence diagram illustrating the interaction of the Marketplace, Capacity Planning Service, and the Workflow Engine.

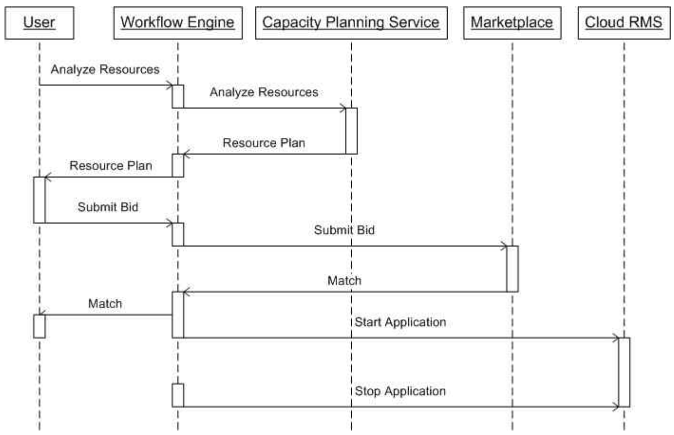

Figure 4. Interaction of Services and Components Orchestrated by the Workflow Engine.

It can be clearly seen in Figure 4 that the user only interacts with the Workflow Engine, even if different services are to be used. This approach hides the complexity of using the Cloud by giving the user a single point of entry into the Cloud.

The following section describes the implementation of the Workflow Engine, validating the workings of the GridEcon Platform. By describing the simple integration of the market mechanism and the capacity planning service into a GridEcon-Platform-based emulated market environment, we validate the usefulness of the GridEcon Platform.

4.1 Implementation of the Workflow Engine

The Workflow Engine is deployed as a Java Web Service using JAX-WS and is made available to end-users through the GridEcon Web Interface. The workflow specifications are created in XML and, therefore, can be edited by any XML editor.

The system was evaluated with respect to three parameters: system performance, system reliability, and accessibility. To test these criteria, we simultaneously submitted 5000 workflow requests for the same task 1000 times. The tests were carried out on a Linux server with two 3.2GHz CPUs and 2GB memory. We used an Apache Tomcat 6.0 server running Java JRE1.6.0_02 to deploy the Workflow Engine. The results of the performance tests are shown in Figure 5.

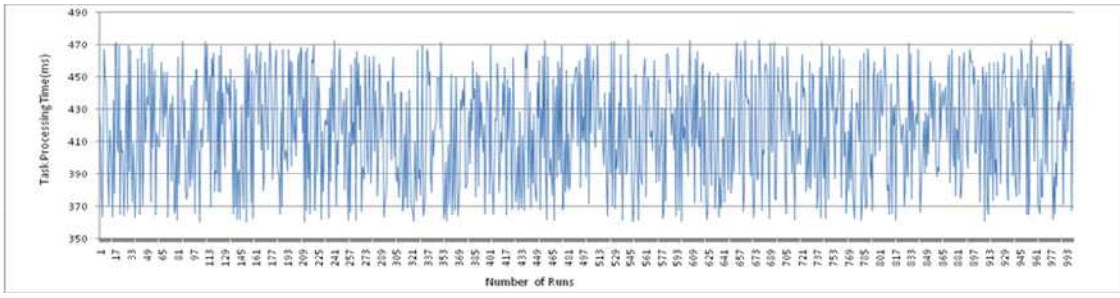

Figure 5. System Performance of the Workflow Engine.

The best response time that we obtained on the above-mentioned configuration was 360.0864ms; the worst response time obtained was 473.3914ms. The average time for the 1000 tests was 417.1616ms. Since the GridEcon Platform did not fail during these tests, we derived that the Workflow Engine and the entire system, the GridEcon Platform, is reliable and accessible.

4.2 Implementation of the Marketplace

The Marketplace was implemented using Java Web Services technologies and has been deployed in the GridEcon Platform implementation as an axis2 AAR file. A major goal in the design of the Marketplace component was to provide a flexible, pluggable architecture, allowing easy configuration and expedite future development. In order to provide this extensibility and adaptability, the Spring Framework was utilized. The Spring Framework allows the central matching mechanism to be configured via dependency injection and without requiring any further compilation or deployment steps. Dependency injection allows a single market mechanism to be re-deployed with differing configuration and allows different market mechanisms to be developed and configured into the existing Marketplace component. This far, the only market mechanism implemented and tested is the continuous double auction mechanism as described in section 3.3 [5].

As one core responsibility of the Marketplace component is to initiate the start and stop of the resulting allocations (i.e. matched offers), a scheduler mechanism (through Open Symphony Quartz) was implemented. When an allocation is produced through the matching process, a scheduled event is created in Quartz that will initiate the startup of the resource at the scheduled time.

The tests of the Marketplace component took the matching accuracy of the implemented CDA into account, as well as the time taken to find a match for a given bid and ask. The matching accuracy reached 100%, which was the design goal, since the matching algorithm of the marketplace is vital to the functioning of the market. Once this goal had been achieved, the performance of the matching algorithm with respect to the matching time was tested.

An excerpt of the results can be seen in Table 2. The table shows the time for handling a single bid submission and the average times for servicing 50 requests under nominal load and heavy load. Nominal load corresponds to the expected average system load. The load was generated by having 30 providers submitting 300 asks and

10 bidders submitting 100 bids over a time period of 1000 trading days. The system's performance was measured on a laptop with AMD Turion TL50 processor with 1 GB DDR2 running Windows XP. The heavy load case represents a situation in which the system has to handle increased load. In this case, 75 providers submitted 750 asks and 75 bidders submitted 750 bids over a time period of 1000 trading days.

Table 2. Marketplace Performance Test Results.

Task	Single Transaction	Nominal Load	Heavy Load
Submit Bid - No Match	45ms	46ms	88ms
Submit Bid – Match	43ms	45ms	76ms

It can be clearly seen in Table 2 that the matching process is quite fast, even under high system load. Besides validating the overall workings of the GridEcon Platform, the results also show that the Marketplace Component and, therefore, the GridEcon Platform is useful for validating market mechanisms. A market mechanism can simply be plugged into the Marketplace Component. It is simply a module within the Marketplace component.

4.3 Implementation of the Capacity Planning Service

To test the viability of a Capacity Planning Service and to show the usefulness of the GridEcon Platform, we developed a market simulation with the help of the GridEcon Platform. This emulation was aimed at showing that even basic prediction algorithms are valuable for capacity planning in Cloud Computing Markets.

The Capacity Planning Service is expected to function within the previously described continuous double auction (section 3.3). The matching algorithm of the market applied horizontal and vertical bid atomicity, meaning that the buyer of a Cloud resource obtained all resources from a single provider. At the end of each day, each trader determines its expected demand for the next day. If the expected demand is larger than the number of in-house resources, the trader will bid for resources for the next day on the Cloud market. Should the number of required resources be lower than the number of in-house resources, the trader would attempt to sell the excess resources for the next day on the Cloud market. The traded resources are made available in the morning of the following day. Therefore, each market participant can change his role between being a buyer and being a seller.

The GridEcon-Platform-based simulation consisted of 10000 traders, which traded resources for 500 trading days. A detailed overview of the simulation parameters is given in Table 3.

Table 3. Simulation Parameters Overview.

Parameters	Value
Number of traders	10,000
Number of in-house resources (per trader)	20-40
Market mechanism	CDA
Number of simulated trading days	500
Offer expiration time	1 day
Demand distribution	Normal
Demand distribution mean	30
Demand distribution variance	30

A trader's resource supply is calculated as the sum of existing in-house resources and purchased Cloud resources. Three of the 10000 traders were given identical demand for resources. In order to work with a realistic demand pattern, we used the results of network traffic measurements to the Web servers of the International University in Germany during a period from October 2007 to December 2008. The demand over this time period is shown in Figure 6 below.

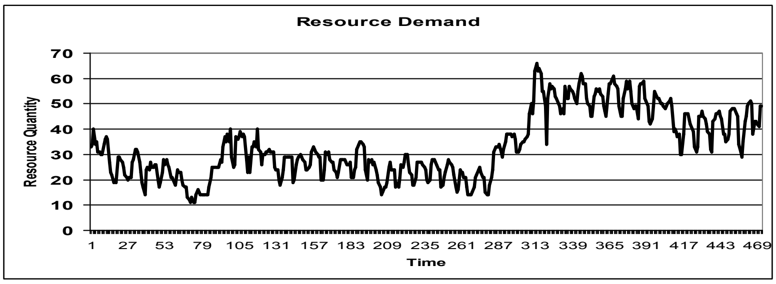

Figure 6. Network Traffic Demand from Web Server Measurements.

Each of these three traders used a specific prediction algorithm. The first trader used a basic prediction algorithm: It used its current demand level to determine the resource amount to be purchased or sold for the next day. The second trader used the past thirty days as a basis for a linear regression to predict demand. The third trader performed a Fourier analysis. The outcomes of the simulations are illustrated in Figure 7. A negative value shows that the trader has fewer resources than required, while a positive value indicates that the trader has more resources than it requires.

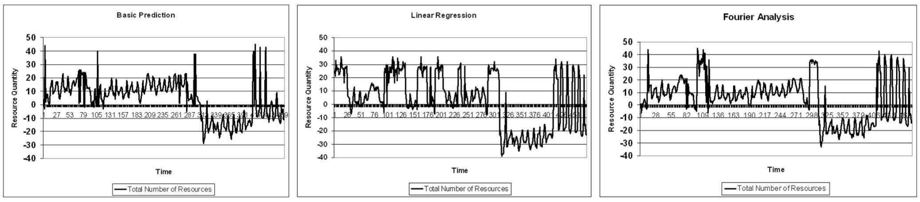

Figure 7. Resource Demand Prediction Results.

It can be clearly seen in Figure 7 that, initially, all traders had sufficient resources at their disposal. However, after about 300 days, all were unprepared for a sudden demand spike. This result is expected, since all prediction methods are based on past measurements. Another factor that contributed to this result is that not necessarily all resource purchases are matched. As in any market, if there is no matching supplier, then the resource demand cannot be met. This is caused by a difference in prices or the horizontal and vertical bid atomicity that we applied. If this requirement were to be removed, the market would have much greater liquidity and the traders would have more resources at their disposal.

With the help of the Capacity Planning Service, we also demonstrated how a value-added service can easily be in integrated into a GridEcon-Platform-based market environment. The GridEcon Platform allowed integrating a prototype of the capacity planning service and conducting performance measurements on the value-added Capacity Planning Service. This highlights the usefulness of the GridEcon Platform for validating new economics-aware, value-added services.

5. Conclusion and Future Work

In this paper, we have analyzed the existing approaches to utility computing markets, mainly the markets of Popcorn, Spawn, GRACE, and Tycoon. However, due to their lack of flexibility, these prototypes cannot be used to evaluate new market designs or the performance of economics-aware services in a market environment. This limitation has been recognized by the GridEcon project, which created the GridEcon Platform, a testbed for evaluating new designs of computing resource markets and for evaluating value-added services in a commercial Cloud Computing setting.

As part of the GridEcon platform, two central components have been created: the Workflow Engine, which enables the composition of a market environment in a simple way, and the Marketplace component, which allows the execution of different market mechanisms. The Workflow Engine can be used by users of the GridEcon Platform to orchestrate different market scenarios needed for their analysis. The Marketplace component allows testing and validating different market mechanisms.

The GridEcon Platform, in particular, the Workflow Engine, has been tested. It performed its tasks as expected and proved to be vital for developing additional value-added services. The usefulness of the GridEcon Platform has been demonstrated by integrating a market mechanism (continuous double auction) into the GridEcon Marketplace Component and by integrating a capacity planning service into a GridEcon-Platform-based emulated market environment. The GridEcon-Platform-based market environment allowed validating the market mechanism and the capacity planning service by conducting performance measurements.

6. Acknowledgement

This work has been funded by the European Commission within the framework of the GridEcon project. GridEcon (contract no. 033634) is a STREP project under the Framework Program FP6 in the area of IST. The authors would also like to thank the members of the GridEcon consortium for their contributions. Special acknowledgement is due to Kostas Giannakakis, Alon Lahav, and Sergios Soursos.

6. References

1. Regev, O., Nisan, N.: The POPCORN market - An Online Market for Computational Resources. In: Proceedings of the First international Conference on Information and Computation Economies. ICE 1998. ACM. New York, USA (1998).

2. Waldspurger, C.A., Hogg, T., Huberman, B.A., Kephart, J.O., Stornetta, W.S.: Spawn: A Distributed Computational Economy. IEEE Transactions on Software Engineering, vol. 18, no. 2, pp. 103-117. (February 1992).

3. Buyya, R., Abramson, D., Giddy, J.: An Economy Grid Architecture for Service-Oriented Grid Computing. 10th IEEE International Heterogeneous Computing Workshop. HCW 2001. IEEE Computer Society Press, San Francisco, USA (April 2001).

4. Lai, K., Rasmusson, L., Adar, E., Zhang, L., and Huberman, B.A.: Tycoon: An Implementation of a Distributed, Market-Based Resource Allocation System. Multiagent and Grid Systems, vol. 1, no. 3, pp. 169-182 (August 2005).

5. Altmann, J., Courcoubetis, C., Stamoulis, G.D., Dramitinos, M., Rayna, T., Risch, M., Bannink, C.: GridEcon - A Market Place for Computing Resources," In: Proceedings of the 5th Workshop on Grid Economics and Business Models, Gecon 2008, Springer LNCS, Las Palmas, Spain (August 2008).

6. Courcoubetis, C., Dramitinos, M., Rayna, T., Soursos, S., Stamoulis, G.D.: Market Mechanisms for Trading Grid Resources", In: Proceedings of the 5th International Workshop on Grid Economics and Business Models, Gecon 2008. pp. 58-72, Las Palmas, Spain (August 2008).

7. The Popcorn Project, http://www.cs.huji.ac.il/~popcorn/, (2008).

8. The gridbus project, http://www.gridbus.org/, (2008).

9. Rahman, M., Buyya, R.: An Autonomic Workflow Management System for Global Grids. In: Proceedings of the Eighth IEEE International Symposium on Cluster Computing and the Grid. CCGrid 2008. pp.578-583. IEEE Computer Society, Washington, DC, USA (May 2008).

10. Buyya, R., Abramson, D., Giddy. J.: Nimrod-G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. In: Proceedings of the 4th International Conference on High Performance Computing in Asia-Pacific Region. HPC Asia 2000. IEEE Computer Society Press, Beijing, China, (May 2000).

11. Venugopal, S., Buyya, R., Winton, L.: A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids. In: Proceedings of the 2nd Workshop on Middleware For Grid Computing. MGC 2004. vol. 76, pp.75-80. ACM. Toronto, Canada, (October 2004).

12. Buyya, R., Murshed, M.,: GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. Journal of Concurrency and Computation: Practice and Experience. vol.14, no.13-15: pp.1175-1220, Wiley Press, USA, (2002).

13. Calheiros, R. N., Ranjan, R., De Rose, C.A. F., Buyya, R.: CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services. ICPP, 2009.

14. Roberston, D.: A Lightweight Method for Coordination of Agent Oriented Web Services. In: Proceedings of the AAAI Spring Symposium on Semantic Web Services. (2004).

15. European Energy Exchange, http://www.eex.com/en/, (2009).

16. Risch, M., Altmann, J.: Cost Analysis of Current Grids and its Implications for Future Grid Markets. In: Proceedings of the Grid Economics and Business Model Workshop. Gecon 2008. pp.13-27. Springer LNCS. Heidelberg, 2008.

17. Risch, M., Altmann, J., Makrypoulias, Y., Soursos, S.: Economics-Aware Capacity Planning for Commercial Grids. In: Collaborations and the Knowledge Economy. vol. 5, pp. 1197-1205. IOS Press. Amsterdam, Netherlands, (2008).

18. Risch, M., Altmann, J.: Capacity Planning in Economic Grid Markets. In: Proceedings of the 4th International Conference on Grid and Pervasive Computing 2009. GPC 2009. pp. 1-12. Springer LNCS. Geneve, Switzerland, (May 2009).

19 Neumann, D., Stößer, J., Weinhardt, C., Nimis, J.: A Framework for Commercial Grids - Economic and Technical Challenges. Journal of Grid Computing. Vol. 6, no. 3, pp. 325-347, (2008).

20. Schnizler, B., Neumann, D., Veit, D., Weinhardt, C.: Trading Grid Services - A Multi-Attribute Combinatorial Approach. European Journal of Operational Research. Vol. 187, no. 3, pp. 943-961, (2008).

TEMEP Discussion Papers

2009-26: Jae Young Choi, Yeonbae Kim, Yungman Jun, and Yunhee Kim, "A multivariate probit analysis of Korean firms' information system adoption: An empirical analysis on the determinants of the adoption and complementarity among the information systems"

2009-27: Flávio Lenz-Cesar and Almas Heshmati, "Determinants of Firms Cooperation in Innovation"

2009-28: Firas H. Al-Hammadany and Almas Heshmati, "Determinants of Internet Use in Iraq"

2009-29: Jeong-Dong Lee, "Industry Dynamics and Productivity Research"

2009-30: Nam TRAN-QUY and Almas HESHMATI, "Measurements and Determinants of Pay Inequality and its Impacts on Firms Performance in Vietnam"

2009-31: Nils Karlson, Marcus Box and Almas Heshmati, "Generality, State Neutrality and Unemployment in the OECD"

2009-32: Chang Seob Kim, Yoonmo Koo, Ie-jung Choi, Junhee Hong, and Jongsu Lee, "Consumer Preferences for Automobile Energy Efficiency Grades"

2009-33: Jongsu Lee, Jae Young Choi and Youngsang Cho, "A forecast simulation analysis of the next-generation DVD market based on consumer preference data"

2009-34: Jungwoo Shin, Chang Seob Kim, and Jongsu Lee, "Model for Studying Commodity Bundling with a Focus on Consumer Preference"

2009-35: Yuri Park, Hyunnam Kim, and Jongsu Lee, "Empirical analysis of consumer preferences for mobile phones and mobile contents in the presence of indirect network effects"

2009-36: Jörn Altmann and Juthasit Rohitratana, "Software Resource Management Considering the Interrelation between Explicit Cost, Energy Consumption, and Implicit Cost: A Decision Support Model for IT Managers"

2009-37: Marcel Risch, Ivona Brandic, and Jörn Altmann, "Using SLA Mapping to Increase Market Liquidity"

2009-38: Amit K. Bhandari and Almas Heshmati, "Willingness to Pay for Biodiversity Conservation"

2010-39: Marcel Risch, Jörn Altmann, Li Guo, Alan Fleming, and Costas Courcoubetis, "The GridEcon Platform: A Business Scenario Testbed for Commercial Cloud Services"