Moving Markets to the Grid

– Full Paper –

1 Introduction

The increasing interconnection between computers has created the vision of a Computational Grid. Within this network, computer resources are accessible to anyone participating in the Grid. This has major ramifications since organizations that have computational demand are not required to purchase and maintain computer resources for their own. Instead, it is possible that computation can be performed spontaneously by other resources in the Grid that are not under the control of the (temporary) user. The corresponding suppliers of computation can be resource owners that have computer resources available (e.g. computer center operators).

Most of the research carried out in the area of Grid has been devoted to the hardware and software infrastructure, such that from the technical point of view the access to resources is dependable, consistent, pervasive and inexpensive (Foster and Kesselman 2004). The technical infrastructure is a necessary requirement to implement Computational Grids in practice. Technical feasibility, however, is not tantamount to actual realization, as also economic issues are important: Resource owners will only offer their computational resources, if they are adequately compensated. Compensation requires a functioning billing and accounting.

In science, it is often referred to a resource sharing model, where organizations can take part if they are sharing idle resources as a part of a fixed entry fee. This model has been adapted from file sharing known from Peer-to-Peer networks like Gnutella. The idea is that everyone contributes to the network and consumes if necessary. This rather cooperative model works only insufficiently, as the incentives to contribute more than a minimum resource endowment is not given. Grids operating under resource sharing as billing model thus suffer under meager contributions to the Grid. The main flaw in those sharing Grids is that computational resources are inherently private goods. From economics it is well known that private resources are characterized by two main properties: the principles of exclusion apply and there is rivalry in consumption. In essence, the first property is advantageous for the application of resource sharing model, as those participants who fail to comply with the minimum sharing requirement can be excluded from the Grid. Nonetheless, rivalry-in-consumption makes the model inapplicable. Rivalry-in-consumption refers to the fact that the consumption of the good by one participant prevents another user from consuming it at a time. Stated differently, the good "computational resource" is scarce, where the scarcity can be expressed by a price.

The sharing model typically solves conflicts of demand for the same resources by first-come first serve scheduling algorithms, which are inefficient as the values the resource consumers have for the computational resources are not incorporated in the scheduling decision. It should be noted at this point that file sharing does not exhibit rivalry condition; the download of files can be performed by as many participants as possible, the only limitation refers to the bandwidth but not by the good itself. Files are hence not scarce; accordingly a market price does not exist.

In summary, the use of sharing models in Grid has two main shortcomings: Firstly, resource owners offer only a small fraction of their available resource on the Grid. Secondly, scheduling is highly inefficient as first-come first serve mechanisms are used that do not account values of the jobs. Unfortunately the first shortcoming makes the second one more severe. As only a minimum of resources are contributed to the Grid and demand for resources is very large, there will be a situation of an extremely large excess demand for resources.

Commercial Grids are accordingly skeptical concerning the resource sharing model. Resource owners are seeking for compensation possibilities in exchange for sharing their resources. There are two common mechanisms that have been established in the past. Firstly, resource owners offer flat rates for sharing resources. These flat rates have the negative ramification that users order the resources for the entire time span specified by the flat rate, even though they are not used for the total time. This leads to idle resources that cannot be allocated in the meantime. The second mechanism refers to Service Level Agreements (SLA). In essence, SLAs specify a service (e.g. Web hosting), the quality of service criteria, the price the consumer has to pay for the service and the penalty if the SLA is violated. SLAs accordingly aggregate computational resources that are underlying the provided service. Service Level Agreements are a promising instrument for billing the Grid, as they incorporate values to the services.

Currently, SLAs are manually negotiated among the participants, which can be very inefficient and costly. To avoid often recurring negotiation costs the SLAs are defined in a way that they cover a longer time period.

In this paper, we argue in favor of short-term services, even on-demand that can be traded over an Open Grid Market. Thus, the paper is structured as follows: In the second section, we motivate *why this change in designing services* is reasonable. In the third section, a requirement list upon market mechanisms in Grid will be given. Section four gives a brief overview over market mechanisms for the Grid. It will be shown that not all mechanisms are adequate for Grid. Hence, in section five a new market architecture is introduced, which will be implemented and preliminary evaluated in section six. Section seven closes with a short summary.

2 Why Markets for the Grid?

Today's resource management systems in Grids have recognized the need of expressing values by including user priority, weighted proportional sharing, and service level agreements that set upper and lower bounds on the resources available to each user or group (Irwin et al. 2004). Maximizing the utility (i.e. the sum of valuations), however, is only possible if the resource manager knows the attached valuations or the exact relative weights, respectively at any point of time. Knowing the valuations at any time is a very demanding requirement, as users typically have no incentive to report decreases in their valuation, because they loose priority and correspondingly value by not getting their computation done.

Hence, value-oriented approaches are not sufficient per-se to achieve an efficient solution. Only if all participants are willing to report their priorities and values honestly, these algorithms (e.g. Proportional Share) will work well. This is where markets for Grid enter the discussion. Markets have the ability to set the right incentives for users to reveal their true valuation as well as for resource owners to provide those resources that are scarcest in the Grid. With the introduction of prices, incentives will be given to the users to substitute the scarce resource (e.g. number of CPUs) with less scare resource (memory). For instance, a fixed pricing scheme which requests \$10 for one CPU and \$1 for memory, sets the incentives to reduce the number of used CPUs in favor of the cheaper memory.

A fixed pricing scheme, however, is not enough to achieve an efficient allocation: Suppose the fixed price of a resource provider as shown in Figure 1 (c.f. (Lai 2005)). Demand changes

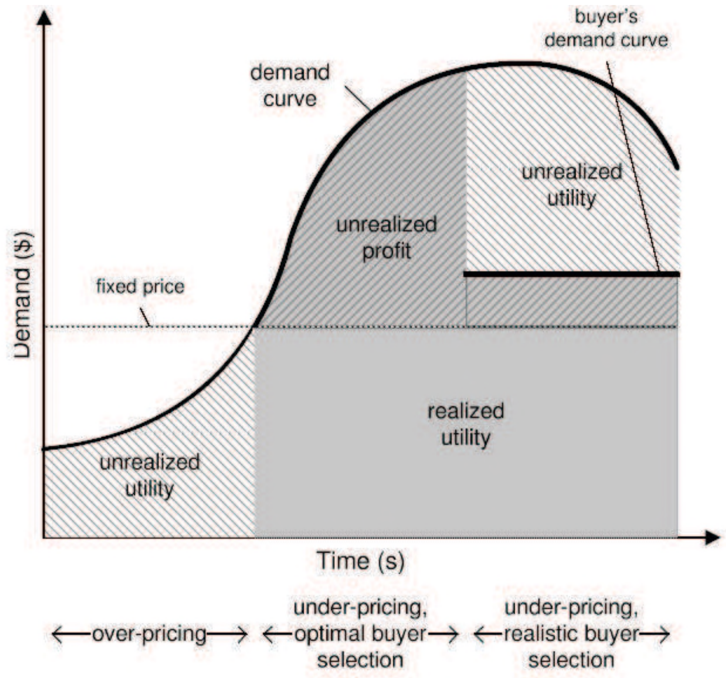

Figure 1: Fixed Pricing

over time - this will be depicted by the parabola - without loss of generality the costs of supplying resources are assumed to be zero. If demand is below the fixed-price (left end of the graph), no resource will be requested. This is because the value for the resource, represented by the demand curve is below the price. As a consequence, there is a loss of utility, which is denoted by the area below the demand curve. In the middle part of the Graph there could be more buyers willing to pay the fixed price, as their demand exceeds the price. If the seller allocates the resource efficiently to the user who values the resource most, there will be unrealized profit for the seller indicated by the striped gray area, which refers to the difference between the demand curve and the fixed price.

A market mechanism has the ability to set the prices in a way that most, if not the entire utility under the demand curve can be realized; resulting in an efficient allocation, where a system designer cannot extract more utility. The gist of a market is to couple price discovery with the valuations of the users, which are expressed in the form of bids.

Comprising, markets in Grid can, on the one hand, mean a business model for the resource owners, as a way to get money for their resource provision. On the other hand, markets are an adequate way to determine the most efficient solution making the Grid attractive for users.

The establishment of so-called Open Grid Markets is not just a theoretical game in rhetoric's but a promising scenario for the future. Potential major players in the Open Grid Markets are preparing their engagement. For example the major telecommunication companies (e.g. British Telecom) have a great need for a liquid Open Grid Market in the future aiming at in-sourcing the entire IT hardware of their clients as new business model. Also resource owners like SUN or IBM are frequently investing into the development of Open Grid Markets. Nonetheless, the design of an Open Grid Market is associated with several obstacles that need to be solved before the vision of an Open Grid Market materializes. In the following, we emphasize the development of the market mechanism being the key functionality of the market.

3 Desirable Properties and Requirements

The objective of an adequate market mechanism for the Grid is the efficient and reliable provision of resources to satisfy demand. Economic approaches explicitly assume that users act strategically in a way that does not reveal their true valuations for resources aggregated in services. Incentive compatible mechanisms are accordingly advantageous, as they encourage users to report their true valuations. This allows the mechanism to maximize allocative efficiency (i.e. the sum of the individuals' utilities).

A critical step in designing markets is to understand the nature of the trading object. This paper considers services which respect resource functionalities (e.g. storage) and quality characteristics (e.g. size), dependencies and time attributes. Relying on services instead of computational resources removes many technical problems. For instance, the resource CPU may technically not be offered without an appropriate amount of hard disk space on the same computer, while a computation service offering CPU cycles already includes the complementary resources. In the remainder of the paper, it is abstracted from those technical details by treating resources and services as synonyms.

The following presents the design objectives and the Grid specific requirements for the market mechanism.

3.1 Design Objectives

The theoretical basis for designing mechanisms has emerged from a branch of game theory called mechanism design (Milgrom 2004). Within the scope of practical mechanism design, the primary design objective is to investigate a mechanism that has desirable properties. The following comprises common economic properties of a mechanism's outcome:

- *Allocative efficiency:*

  An allocation is efficient if the sum of individual utilities is maximized. It is assumed that utility is transferable among all participants. A mechanism can only attain allocative efficiency if the market participants report their valuation truthfully. This requires incentive compatibility in equilibrium.

- *Incentive compatibility:*

  A mechanism is incentive compatible, if every participant's expected utility maximizing strategy in equilibrium with every other participant is to report its true preferences (Parkes et al. 2001).

- *Individual rationality:*

  The constraint of individual rationality requires that the utility following participation in the mechanism must be greater or equal to the previous utility.

- *Budget balance:*

  A mechanism is said to be budget balanced if the prices add up to zero for all participants (Jackson 2002). In case the mechanism runs a deficit, it must be subsidized by an outside source and is therefore not feasible per-se.

- *Computational tractability:*

  Computational tractability considers the complexity of computing a mechanism's outcome. With an increasing number of participants, the allocation problem can become very demanding and may delimit the design of choice and transfer rules.

It is clear that the objective allocative efficiency meets the general design goal that the mechanism designer wants to achieve, whereas the remaining categories are constraints upon the objective (Neumann 2004).

3.2 Domain-Specific Requirements

In addition to those mechanism properties pertaining to the outcome, the mechanism must also account for the underlying environment. The constraints of the market participants impose very rigid requirements upon the design:

- *Double-sided mechanism:*

  A Grid middleware usually provides a global directory enabling multiple service owners to publish their services and multiple service requesters to discover them. Since a market mechanism replaces these directories, it has to allow many resource owners (henceforth sellers) and resource consumers (buyers) to trade simultaneously.

- *Language includes bids on attributes:*
  Participants in the Grid usually have different requirements for the quality characteristics of Grid services and require these in different time spans. For example, a rendering job could require a storage service with at least 250 GB of free space for four hours in any slot between 9 a.m. and 4 p.m. The Grid community takes these requirements into account by defining service level agreement protocols, e.g. WS-Agreement (Ludwig et al. 2004). To facilitate the adherence of these agreement protocols, a market mechanism fitting the Grid is required to support bids on multiple quality attributes of services as well as their time objectives.

- *Language includes combinatorial bids:*
  Buyers usually demand a combination of different Grid services as a bundle in order to perform a task (Subramoniam et al. 2002). As such, Grid services are complementarities, meaning that participants have super-additive valuations for the services, since the sum of valuations for single services is less than the valuation for the whole bundle $(v(A)+v(B) \leq v(AB))$. Suppose a buyer requires services for storage, computation and rendering. If any service, e.g. the storage service, is not allocated to him, the remaining bundle has no value for him. In order to avoid this exposure risk (i.e. receiving only a subset of the bundle), the mechanism must allow for bids on bundles.

  The buyer may want to submit more than one bid on a bundle as well as many that exclude each other. In this case, the resources for the bundles are substitutes, i.e. participants have sub-additive valuations for the services $(v(A) + v(B) \geq v(AB))$. For instance, a buyer is willing to pay a high price for a service during the day and a low price if the service is executed at night. However, this service may only be computed once. To express this, the mechanism must support XOR[1] bids to express substitutes. For the sake of simplicity, a seller's bid is restricted to a set of OR[2] bids. This simplification can be justified by the fact that Grid services are non-storable commodities, e.g. a computation service currently available cannot be stored for a later time.

[1]A XOR B (A $\oplus$ B) means either $\emptyset$, A, or B, but not AB

[2]A OR B (A $\vee$ B) means $\emptyset$, A, B, or AB

- *Language includes co-allocation constraints:*
  Capacity-demanding Grid applications usually require the simultaneous allocation of several homogenous service instances from different providers. For example, a large-scale simulation may require several computation services to be completed at one time. Research literature often refers to the simultaneous allocation of multiple homogenous services as co-allocation. A mechanism for the Grid has to enable co-allocations and provide functionality to control it. In this context, two cases must be considered: Firstly, it is desirable to limit the maximum number of service co-allocations, i.e. the maximum number of service divisions. Secondly, it may be logical to couple multiple services of a bundle in order to guarantee that these resources are allocated from the same seller and – more importantly – will be executed on the same machine.

An adequate market mechanism for the Grid must satisfy these requirements stemming from the economic environment and ideally meet the design objectives.

4 Related Work

Buyya et al. (2002) were among the first researchers motivating the transfer of market-based systems from distributed systems to Grids. Nonetheless, they propose classical one-sided auction types which cannot account for combinatorial bids. Wolski et al. (2001) compare classical auctions with a bargaining market, coming to the conclusion that the bargaining market is superior to an auction-based market. This result is less surprising since the authors only consider classical auction formats where buyers cannot express bids on bundles. Eymann et al. (2003) introduce a decentralized bargaining system for resource allocation in Grids. In their simulation the bargaining systems work fairly well; however, bids on bundles are largely ignored.

Subramoniam et al. (2002) account for combinatorial bids by providing a tâtonnement process for allocating and pricing Grid resources. Furthermore, Ng et al. (2005) propose repeated combinatorial auctions as a microeconomic resource allocator in distributed systems. Nonetheless, the resources are still considered to be standardized commodities. Standardization of the resources would either imply that the number of resources are limited compared to the number of all possible resources or that there are many mechanisms which are likely to suffer due to meager participation. Both implications result in rather inefficient allocations.

Additionally, state-of-the-art mechanisms widely neglect time attributes for bundles and quality constraints for single resources. Hence, the use of these mechanisms in the Grid environment is considerably diminished. The introduction of time attributes redefines the Grid allocation problem as a scheduling problem. To account for time attributes, Wellman et al. (2001) model single-sided auction protocols for allocating and scheduling resources under different time constraint considerations. However, the proposed approach is single-sided and favors monopolistic sellers or monopsonistic buyers in a way that allocates greater portions of the surplus. Installing competition on both sides is deemed superior, since no particular market side is systematically given an advantage.

Demanding competition on both sides suggests the development of a combinatorial exchange. In research literature, Parkes et al. (2001) introduce the first combinatorial exchange as a single-shot sealed bid auction. As payment scheme, Vickrey discounts are approximated. The approach results in approximately efficient outcomes; however, it neither accounts for time nor for quality constraints and is thus not directly applicable to the Grid allocation problem.

Counteractively, Bapna et al. (2005) propose a family of combinatorial auctions for allocating Grid services. Although the mechanism accounts for quality and time attributes and enables the simultaneous trading of multiple buyers and sellers, there is no competition on the sellers' side as all orders are aggregated to one virtual order. Moreover, the mechanism does not take co-allocation constraints into account.

In reviewing the related mechanisms according to the requirements presented in section 3, it is revealed that no market mechanism installs competition on both sides, includes combinatorial bids, allows for time constraints, manages quality constraints, or considers co-allocation restrictions. This paper intends to address these deficiencies by outlining the design of a market based Grid architecture.

5 Market-Oriented Service Allocation

Tackling an adequate architecture for enabling market-oriented service allocation in Grids requires the design of open interfaces as well as flexible components. Thus, the relevant components of the proposed architecture as well as their interaction are based on Web service standards in order to provide connectivity, efficiency, flexibility, immediacy, and interoperability of the whole system (Lawler et al. 2004).

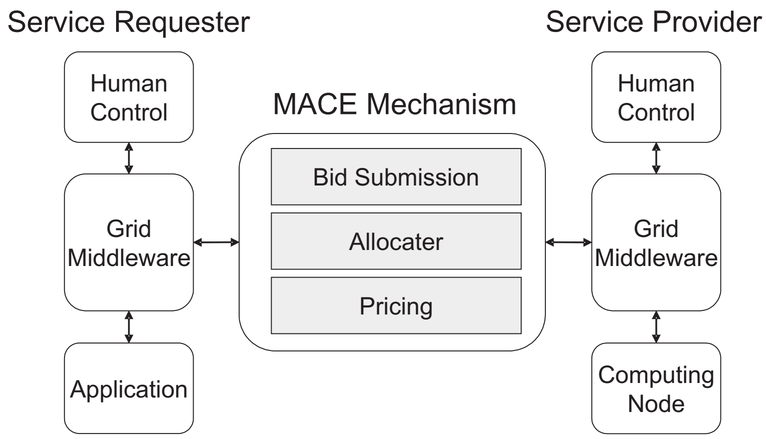

Figure 2: Open Grid Market Architecture

A high level overview on the proposed Open Grid market architecture is depicted in figure 2. In essence, the innovation in this architecture pertains to the Grid market and to the middleware extensions, which are necessary to access the market. Resource sharing is technically realized by standard Grid middleware such as Globus Toolkit (Foster 2005) and is thus not in the center of attention.

Following the Globus framework, the sharing of resources is not realized by sharing physical resources (e.g. CPU), but by providing services as aggregated resources (e.g. CPU cycles). Within this architecture, the market consists of service requesters (e.g. scientists running a simulation application), service providers (e.g. computer centers with idle resources), and a market mechanism capable of aggregating bids from resource providers and users and subsequently determining an outcome.

The human control interface can be used to specify user policies manually. In the context of Grids, service policies govern by whom and under which conditions services may be allocated (Lamparter and Agarwal 2005). For instance, a policy may state that the communication between a service provider and a particular consumer must be encrypted with a key of at least 1024 bit. Having specified the underlying service policies, requests and offers are initiated by an Application, e.g. an application which is integrated into a Grid workflow system. Applications specify detailed service type requirements (e.g. a storage service) and quality of service constraints (e.g. a storage service with at least 300GB free space for four hours). It is assumed that approximate quality and time constraints of the requested services can be determined using prediction models (e.g. (Kee et al. 2004)). The service requirements are encoded using standard description languages such as the Resource Specification Language (RSL) or the Classified Advertisements.

The market mechanism component stands centrally for the market architecture which implements a trading mechanism for Grid Services. The component is defined as a Web service in order to provide interoperable invocation facilities to the bidding agents. The market service interface is specified by means of the Web Service Description Language (WSDL) and comprises a list of provided operations including messages that are accepted and returned. For instance, the interface provides operations to retrieve status information (e.g. the current highest bid) and to submit bids.

The underlying institutional mechanism of the market component is a multi-attribute combinatorial exchange (MACE), as proposed by Self Citation (2005a). MACE allows multiple bidding agents (representing service requesters and providers) the simultaneous submission of bids on heterogonous services expressing substitutes (realized by XOR bids) and complements (realized by bundle bids). Furthermore, the mechanism is capable of handling cardinal attributes. For instance, a resource consumer can bid on a bundle consisting of a computation service and a storage service. The computing service should have two processors. Each processor should have at least 700MHz and the storage service should have at least 300 GB of free space. Bids – encoded as WS-Agreement offers – are submitted to an auctioneer. Regularly (in case of a call market) or continuously (in case of continuous matching), the auctioneer determines an allocation (winner determination) and the corresponding prices.

In the following, the market mechanism components are introduced in more detail.

5.1 Bid Submission

Meeting the requirements specified in section 3 requires a formal bidding language that enables buyers and sellers to submit combinatorial bids, including multiple attributes and co-allocation constraints. Furthermore, the bidding language has to be understood by common Grid middleware frameworks, such as the Globus Toolkit.

A prominent agreement specification for Grids is the WS-Agreement specification defined within the Global Grid Forum (GGF). It defines a language and a protocol for advertising the capabilities of service providers and creating agreements based on creational offers, and for monitoring agreement compliance at runtime (Andrieux et al. 2005).

Service requesters and service providers can use WS-Agreements in order to specify their bids and submit them to the market mechanism. For example, a storage service provider might specify the offer shown in figure 3. Here the quality characteristics comprise a capacity of 285GB as well as a storage duration of 20s. The provider has a reservation of 27 for the service.

```
<wsag:AgreementOffer>
 <wsag:Terms>
  <wsag:All>
    <wsag:ServiceDescriptionTerm wsag:Name="readFile"
        wsag:ServiceName="StorageService">
      <job:arguments>/some/file/to/read</job:arguments>
    </wsag:ServiceDescriptionTerm>
    <wsag:All>
     <wsag:ServiceDescriptionTerm wsag:Name="storageSpace"
        wsag:ServiceName="StorageService">
        <job:realMemorySize>285</job:realMemorySize>
     </wsag:ServiceDescriptionTerm>
   </wsag:All>
   <wsag:GuaranteeTerm wsag:Name="Valuation">
     <wsag:ServiceScope>
       <wsag:ServiceName>StorageService</wsag:ServiceName>
     </wsag:ServiceScope>
     <wsag:BusinessValueList>
      <wsag:CustomBusinessValue>
        <mace:reservation>27</mace:reservation>
       </wsag:CustomBusinessValue>
     </wsag:BusinessValueList>
   </wsag:GuaranteeTerm>
 </wsag:Terms>
</wsag:AgreementOffer>
```

Figure 3: WS-Agreement Example

5.2 Allocator

After the bids are submitted to the auctioneer, an allocation is computed, i.e. it is determined which sellers provide services and which buyers receive them.

In the proposed multi-attribute combinatorial exchange, this allocation problem is formulated as a mixed integer program (MIP). This is advantageous, as standard optimization solvers such as CPLEX[3] can be applied.

The objective of the allocation process in MACE is the maximization of social welfare, i.e. the difference between the buyers' valuations and the sellers' reservation prices. The winner determination is, however, a generalization of the general combinatorial allocation problem (CAP)

[3]CPLEX is a commercial product and is currently the state of the art optimization engine. See `http://www.ilog.com/` for details.

which is proven to be $\mathcal{NP}$ complete (Rothkopf et al. 1998). For large-scaled scenarios, the use of approximations have to be evaluated (Mito and Fujita 2004). Nevertheless, the application of such a complex problem seems to be promising, as the number of different bundles in the resource market is restricted.

The outcome of the allocation process is allocative efficient as long as buyers and sellers reveal their valuations truthfully. The incentive to set bids according to the valuation is induced by an adequate pricing mechanism.

5.3 Pricing

The question how to determine payments made by participants to the exchange and vice versa after the mechanism has determined the winners is referred to as pricing problem. With respect to the objective of achieving an efficient allocation, a pricing scheme based on the well-known Vickrey-Clarke-Groves (VCG) mechanism would attain this objective (Vickrey 1961; Clarke 1971; Groves 1973). Moreover, VCG mechanisms are the only allocative-efficient and incentive compatible mechanisms (Green and Laffont 1977). However, Myerson and Satterthwaite (1983) proved that it is impossible to design an exchange which is incentive compatible, (interim) individually rational, and budget balanced that achieves efficiency in equilibrium. The theorem is comprehensive and also applies to the presented mechanism. Obviously, the VCG pricing schema cannot be applied since it runs a permanent deficit requiring outside subsidization.

Relaxing the requirement of having an efficient allocation opens up the possibility for a second-best mechanism that is budget balanced. These ideas gave rise to the development of a k-pricing scheme and to determine prices for buyers and sellers on the basis of the difference between their bids. For instance, presume a buyer wants to buy a computation service for \$5 and a seller wants to sell a computation service for at least \$4. The difference between these bids is $\pi = 1$, i.e. π is the surplus of this transaction and can be distributed among the participants. This schema can be applied to MACE and results in an approximate efficient outcome (Self Citation 2005a). The outcome of the mechanism, i.e. the allocation and the corresponding prices, is subsequently sent to the service providers and requesters as WS-Agreement.

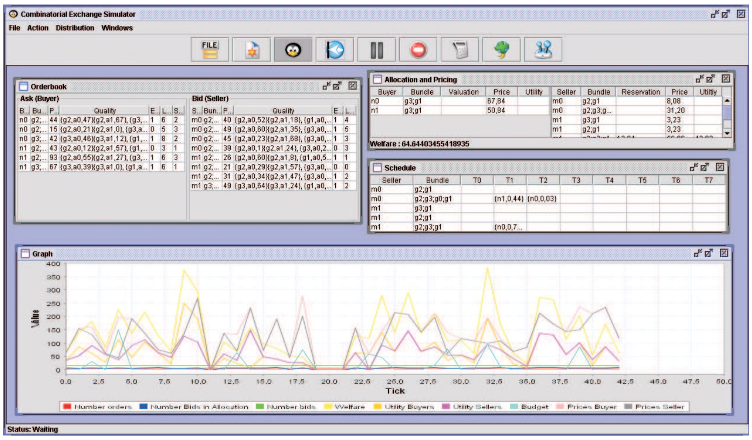

Figure 4: jCase – Java Combinatorial Auction Simulation Environment

6 Evaluation

The presented market mechanism is implemented in the Java Combinatorial Auction Simulation Environment (jCASE)[4], a toolkit for simulating combinatorial mechanisms shown in figure 4. In a first step, the computational tractability of the mechanism is analyzed by means of a run-time simulation. The allocation problem is solved using a Pentium XEON with 3.2GHz and CPLEX 9.1. No special CPLEX parameters are used.

6.1 Data Basis

For generating the bids, the Decay distribution is applied as proposed by Sandholm et al. (2002). In the Decay distribution, each bundle consists firstly of one random resource. Afterwards, a new resource is added randomly with a probability of $\alpha = 0.75$. This proceeding is iterated until a resource is not added or the bundle includes all resources. Sandholm et al. (2002) show, that the Decay distribution can lead to hard instances of general combinatorial allocation problems.

For the price, time, and quality attributes, a uniform distribution is used. Each order of a buyer consists of a uniformly distributed number (1 to 4) of bundle bids, which can be allocated within a time range of 8 different time slots.

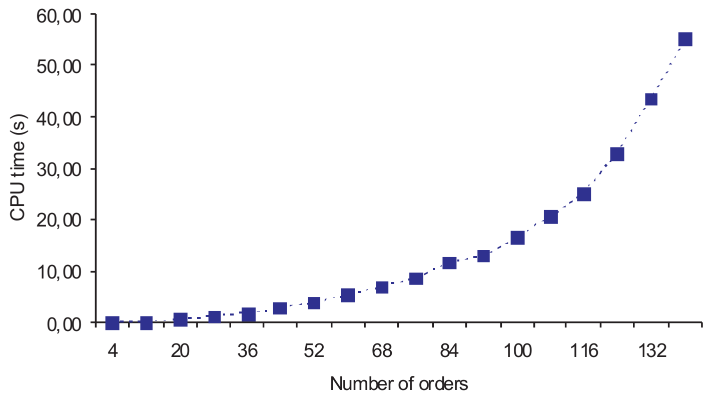

Figure 5: Performance simulation results

6.2 Performance Evaluation

Figure 5 shows the CPU time of CPLEX as a function of the number of orders. With 68 orders in the market for example, 132 bids on bundles are generated, and 6.832 seconds of processing time are required. In the worst case, the solving of 140 orders (with 303 bids) takes over 50 seconds using a Pentium IV 2.3 GHZ. The performance simulation shows that the winner determination problem is computationally very demanding. For more complex scenarios, the use of approximations such as genetic algorithms have to be examined. For further tests of the economic requirements, the reader is refered to Self Citation (2005b).

7 Conclusion

The increasing standardization of Grid services arranges for the commoditization of computational resources. Aggregations of basic computational services are increasingly becoming a utility like energy. In analogy to energy, those Grid services can also be traded over Open Grid Markets.

As Grid middleware like Globus Toolkit or Unicore strive for virtualizing the underlying processes and the corresponding involved computational resources, it is indeed possible – though to a certain extent – to trade computational resources. Using market mechanisms offers the advantage that an efficient allocation of resources can be attained.

Currently, the canon of market mechanisms that is available only insufficiently accounts for the peculiarities of Grid and are thus widely inapplicable. In this paper, we motivate the

[4] `http://self_citation`

design of a fully-fledged market mechanism that is tailored to the needs of the Grid. As such, the market mechanism considers combinatorial bids, multiple and negotiable quality attributes beyond the price, and time attributes. There is strong evidence that the envisioned $k-$pricing auction achieves fairly good results, but the proof still remains for future work.

References

Andrieux, A., K. Czajkowski, A. Dan, K. Keahe, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu (2005). Web Services Agreement Specification (WS-Agreement). Gwd-r, Global Grid Forum.

Bapna, R., S. Das, R. Garfinkel, and J. Stallaert (2005). A Market Design for Grid Computing. Technical report, Department of Operations and Information Management, University of Connecticut.

Buyya, R., D. Abramson, J. Giddy, and H. Stockinger (2002). Economic Models for Resource Management and Scheduling in Grid Computing. *The Journal of Concurrency and Computation: Practice and Experience 14*(13-15), 1507–1542.

Clarke, E. (1971). Multipart pricing of Public Goods. *Public Choice 2*, 19–33.

Eymann, T., M. Reinicke, O. Ardaiz, P. Artigas, L. D. de Cerio, F. Freitag, R. Messeguer, and D. R. L. Navarro (2003). Decentralized vs. Centralized Economic Coordination of Resource Allocation in Grids. In *Proceedings of the 1st European Across Grids Conference*.

Foster, I. (2005). Globus Toolkit Version 4: Software for Service-Oriented Systems. In *Proceedings of the IFIP International Conference on Network and Parallel Computing*, LNCS 3779, pp. 2–13. Springer-Verlag.

Foster, I. and C. Kesselman (2004). *The Grid 2 - Blueprint for a New Computing Infrastructure*, Volume 2. Elsevier.

Green, J. and J.-J. Laffont (1977). Characterization of Satisfactory Mechanisms for the Revelation of Preferences for Public Goods. *Econometrica 45*(2), 427–438.

Groves, T. (1973). Incentives in teams. *Econometrica 41*, 617–631.

Irwin, D., L. E. Grit, and J. S. Chase (2004). Balancing Risk and Reward in a Market-based Task Service. In *Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing*, pp. 160–169.

Jackson, M. O. (2002). Mechanism Theory. Oxford ,UK. Eolss Publishers.

Kee, Y.-S., H. Casanova, and A. Chien (2004). Realistic Modeling and Synthesis of Resources for Computational Grids. In *ACM Conference on High Performance Computing and Networking (SC2004)*. ACM Press.

Lai, K. (2005). Markets are Dead, Long Live Markets. Technical Report 0502027, HP Labs.

Lamparter, S. and S. Agarwal (2005). Specification of Policies for Automatic Negotiations of Web Services. In L. Kagal, T. Finin, and J. Hendler (Eds.), *Proceedings of the Semantic Web and Policy Workshop at the 4th International Semantic Web Conference*, Galway, Ireland.

Lawler, J., D. Anderso, H. Howell-Barber, J. Hill, N. Javed, and Z. Li (2004). A Study Of Web Services Strategy In The Financial Services Industry. In *The Proceedings of ISECON 2004*.

Ludwig, H., A. Dan, and B. Kearney (2004). Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements. In *Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC 2004)*, pp. 65–74.

Milgrom, P. (2004). *Putting Auction Theory to Work*. Cambridge University Press.

Mito, M. and S. Fujita (2004). On Heuristics for Solving Winner Determination Problem in Combinatorial Auctions. *Journal of Heuristics 10*(5), 507–523.

Myerson, R. B. and M. A. Satterthwaite (1983). Efficient Mechanisms for bilateral Trading. *Journal of Economic Theory 28*, 265–281.

Neumann, D. (2004). *Market Engineering – A Structured Design Process for Electronic Markets*. Ph. D. thesis, Economics and Business Engineering. Karlsruhe, Germany, University of Karlsruhe (TH).

Ng, C., P. Buonadonna, B. N. Chun, A. C. Snoeren, and A. Vahdat (2005). Addressing Strategic Behavior in a Deployed Microeconomic Resource Allocator. In *In Proceedings of the 3rd Workshop on Economics of Peer-to-Peer Systems*.

Parkes, D. C., J. Kalagnanam, and M. Eso (2001). Achieving Budget-Balance with Vickrey-Based Payment Schemes in Exchanges. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 1161–1168.

Rothkopf, M., A. Pekec, and R. Harstad (1998). Computationally Manageable Combinational Auctions. *Management Science 44*, 1131–1147.

Sandholm, T., S. Suri, A. Gilpin, and D. Levine (2002). Winner Determination in Combinatorial Auction Generalizations. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 69–76. ACM Press.

Self Citation (2005a).

Self Citation (2005b).

Subramoniam, K., M. Maheswaran, and M. Toulouse (2002). Towards a Micro-Economic Model for Resource Allocation in Grid Computing Systems. In *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering*.

Vickrey, W. (1961). Counter Speculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance 16*, 8–37.

Wellman, M. P., W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason (2001). Auction Protocols for Decentralized Scheduling. *Games and Economic Behavior 35*, 271–303.

Wolski, R., J. Plank, J. Brevik, and T. Bryan (2001). Analyzing Market-Based Resource Allocation Strategies for the Computational Grid. *International Journal of High Performance Computing Applications 15*(3), 258–281.