A Multiattribute Combinatorial Exchange for Trading Grid Resources

Björn Schnizler, Dirk Neumann, Daniel Veit, Christof Weinhardt

Information Management and Systems, University of Karlsruhe (TH)
{schnizler,neumann,veit,weinhardt}@iw.uni-karlsruhe.de

Abstract: The Computational Grid is a promising technology for providing access to distributed high-end computational capabilities. It enables the execution of complex and computationally demanding applications such as simulations or multimedia-renderings. However, one of the key problems in the Computational Grid is to decide which jobs are to be allocated to which resources at which time. In this context, the use of market mechanisms for scheduling and allocating Computational Grid resources is a promising approach to solve these problems. This paper describes the design of a mechanism for allocating and scheduling resources like processors or storage space having multiple attributes in Computational Grids according to the agents' bids. As such a clearing and pricing model for a multiattribute combinatorial exchange model is formulated, which supports bids on bundles, quality, and time attributes. We have evaluated the designed mechanism according to its computational tractability by means of a simulation.

Keywords: Grid, Resource Allocation, Combinatorial Exchanges, Market Engineering

Acknowledgements: This work has partially been funded by EU IST progamme under grant 003769 "CATNETS"

1 Introduction

The increasing interconnection between computers has created the vision of a Computational Grid. Within this Grid, 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 one. 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 computation resources available (e.g. computer center operators).

Most of the research 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. 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 in Grid will be given. It will be shown that all mechanisms are not adequate for Grid. Hence, in section five the new market mechanism MACE is introduced, which will be implemented in section six. Section seven closes with a short summary.

2 On-demand Services and Open Grid Markets

With the introduction of the standardized1 description framework WSRF (Web Service Resource Framework), it is possible to describe stateful services within a well-accepted framework. The possibility to describe services is the prerequisite for the services to become more standardized. As a consequence, services itself are becoming more and more standardized, or in short they become a commodity – though they still remain domain specific. This standardization process opens up the way to Open Grid Markets, where services are traded as commodity. No longer is it necessary to negotiate long-term contract. Instead, it is also conceivable to trade short term services. Short term services depend on the actual resource consumption not on the average. For resource owners, it is risky to

---

[1] Note that WSRF is not a de-iure standard, but recommendation by the Global Grid Forum. Currently, WSRF is more or less the de-facto standard. In the following, we will refer to WSRF without entering the discussion concerning concurring efforts.

engage into a long term SLA, since the resource demand that is needed for meeting the QoS requirements may exceed their capacity. If they could instead offer services that correspond to their actual available resources, this risk is shifted to the resource consumers. The second scenario, where resource owners offer on-demand services directly motivates the establishment of an Open Grid Market. Over the Open Grid Market resource-near services are traded. The resulting price equilibrating demand and supply reflects the valuations of the buyer for the service and the reservation price of the sellers of services. The first scenario also gives rise for an Open Grid Market. Since it can always happen that the SLAs exceed the resource capacities of the resource owner, an Open Grid Market makes an immediate purchase of resources possible thus avoiding the payment of the penalty.

The establishment of an Open Grid Market 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 Requirements and Desirable Properties

The theoretical basis for designing auctions has emerged from a part of game theory called mechanism design(Milgrom 2004). A mechanism $\mathcal{M}$ specifies the available messages and the rules how to resolve it via clearing and price rules. Formally, a mechanism $\mathcal{M}$ is a pair (M, y_M) where M is the language and y_M the resulting allocation h and prices p. For any message profile $m \in M$, the mechanism $\mathcal{M}$ computes the resulting allocation and prices as an equilibrium solution. Within the scope of practical mechanism design, it is the primary goal to investigate a mechanism that is applicable in certain situations and which attains an allocation that has desirable properties. As such, for tailoring an adequate mechanism for Computational Grid, it is necessary that the mechanism accounts for the requirements on the outcome (what is to achieve?) and on the mechanism itself (how it is achieved). While the requirements on the outcome reflect standard economic measures, the requirements on the mechanism take the specific environment of Computational Grid into consideration.

3.1 Requirements on the outcome

In literature, there are many standard economic measures. For Computational Grid the following typically apply:

- *Allocative efficiency*
  Efficiency is a focal concept of economics. *Pareto* efficiency requires from the mechanism to attain an allocation, for which no other allocation exists, that makes at least one agent better off without making at least one agent worse off. If utility is transferable among all agents, a mechanism that maximizes the sum of individual utilities (i.e. the sum of surpluses conditional on the given information set) is called allocative efficient.

  Allocative efficiency can be defined in an ex-post and ex-ante sense. Ex-ante efficiency takes preferences over *expected* allocations in consideration, whilst ex-post analyzes preferences over *realized* allocations.
- *Incentive Compatibility*

  Incentive compatibility refers to the validity of the messages the agents place. It is said a mechanism is incentive compatible if the agents report their preferences truthfully. Agents may have an incentive to untruthfully report their preferences in order to increase their individual utility.
- *Individual Rationality*
  The constraint that the mechanism is individual rational requires that the utility after participating in the mechanism must be higher than before. Otherwise the agent would decide not to take part in the mechanism. This individual rationality constraint is thus sometimes termed participation constraint (Wurman 1999; Fudenberg and Tirole 2000).
- *Budget Balance*

  The concept of *budget balance* is concerned with whether the mechanism requires payments from outside the system. A mechanism is said to be budget balanced if the amount of prices sum up to 0 over all agents. In this case the mechanism ‘merely’ redistributes the payments among the agents. Neither funds from the system are removed nor is the system subsidized from outside. Budget balance is a nice property since the resource allocation can be performed at no costs. In the case the mechanism runs a deficit the mechanism must be subsidized by some outside source and is thus not per-se feasible. (Parkes 2001; Jackson 2002).
- *Computational tractability* considers the complexity of computing the outcome of a mechanism from agent strategies (Kalagnanam and Parkes 2003). With the size of the message space the allocation problem can become very demanding. Computational constraints may delimit the design of choice and transfer rules.

Generally, the first goal allocative efficiency qualifies for objective functions the mechanism designer wants to achieve, while the remaining categories principally are constraints upon the objective function. Hence, a market mechanism for Computational Grid is intended to maximize total utility(Krishna and Perry 1998). The remaining criteria usually impose constraints on the maximization problem.

3.2 Requirements upon the mechanism

Beside the standard economic measures, the mechanism must also account for the underlying environment. For Computational Grid the requirements stemming from the environment are as follows:

- Double-sided mechanism

  The mechanism apt for Computational Grid allows many resource owners (henceforth sellers) and many resource consumers (buyers) to act simultaneously. Principally, it is also conceivable to establish several one-sided mechanisms emulating a double-sided-market. In the following, this premise is translated into providing a double-sided mechanism, since this installs competition on both sides and is thus deemed promising to yield an adequate (allocative efficient) allocation.

- Language includes bundles bids

  Buyers demand a combination of resources as a bundle to perform a task (Subramoniam, Maheswaran et al. 2002). Apparently, resources in the Grid are complementarities. Complementarities are goods with super-additive valuations ($v(A)+v(B)\leq v(AB)$), as the sum of the valuations for the single resources is less than the valuation for the whole bundle. If any component of the bundle, say the CPU, is not allocated to him, the remaining bundle has no value for him since the rendering cannot be processed without the CPU. In order to avoid the exposure risk (i.e. receiving only one leg of the bundle without the other), the mechanism must allow for bids on bundles. Likewise, the seller can also express bids on bundles.

  The buyer may want to submit more than one bid on a bundle but many that are excluding each other. In this case, the resources of the bundles are substitutes. This means that the buyer has sub-additive valuations ($v(A) + v(B) \geq v(AB)$) for the resources. For instance, the buyer is willing to pay a high price for a job during the day and a low price if the job is executed at night. However, this job must be computed only once. As such, the mechanism must support XOR-bids to express substitutes. For simplicity we restrict a bid of a seller to a set of OR-bids. This simplification can be assumed by the fact, that the resources in the Computational Grid are usually no substitutes for the sellers.

- Language includes bids on quality attributes
  Resources in Grids are typically not completely standardized. Similar resources can differ in their quality. A hard disk be characterized by its quality attributes capacity (in Gigabyte GB), access time (in milliseconds (ms)), and data throughput (in bits per second (bits/s)). While a rendering job requiring a minimum amount of GB, say 250 GB, can be conducted by a 500 GB hard disk, but not by a 100 GB hard disk. As such, minimum quality requirements must be met, while similar resources of superior quality work as well.
- Language includes bids on multiple time slots

  Buyers usually require resources only for a certain time span. Having conducted the computation, there is typically no further use for the resources. The exact timing of the computation is not always that important for the buyer. For instance, the buyer may be indifferent whether the job is performed at 10 a.m. or at 11 a.m., as long as the job is finished at certain time, say 3 p.m. Therefore, the mechanism must allow for placing bids on time ranges.
- Clearing and pricing rules that exploit the full-range of the language

  Note only the language must comprehend the peculiarities of the Computational Grid environment, but also the clearing and pricing rules. But he design of the clearing and pricing rules that (1) impute a desirable allocation (allocative efficient) and (2) make usage of all information of the language is rather tempting.

4 Related Work

The use of market mechanisms for allocating computer resources is not a completely new phenomenon. Regev and Nisan propose within the scope of the POPCORN project the application of a Vickrey auction for the allocation of computational resources in distributed systems (Nisan, London et al. 1998). The Vickrey mechanism achieves truthful bidding as a dominant strategy and hence results in an efficient allocation. Buyya (Buyya, Stockinger et al. 2001),Wolski et al. (Wolski, Brevik et al. 2003), and Subramoniam et al. (Subramoniam, Maheswaran et al. 2002) first motivated the use market-based mechanisms such as auctions or electronic negotiations for Computational Grid. In their first attempt, Wolski et al. (Wolski, Brevik et al. 2003) suggested the use of traditional auction formats such as English auctions. The use of traditional auction formats in the Grid environment, however, is conceivably delimited, as the trading objects are traded as unbundled standardized commodities. As a consequence, those traditional auction formats fail to express demand on bundles – exposing the buyers and sellers, respectively, to the risk of receiving only one leg of the bundle without the other. To avoid such an exposure risk of the buyers, Subramoniam et al. (Subramoniam, Maheswaran et al. 2002) employs the use of ascending bundling auctions. Nonetheless, the resources are still considered to be *standardized commodities*. Standardization of the resources would either imply that the number of resources are extremely limited compared to the number of all possible resources or that there are extremely many mechanisms, which are likely to suffer under meager participation. Both implications result in rather inefficient allocations.

Reviewing the requirements upon the mechanism (described in section 2), it becomes obvious that the previous described mechanisms fail to satisfy these requirements. Especially the negligence of *time attributes* for bundles and *quality constraints* for single resources diminish the use of the proposed market mechanisms. The introduction of time attributes redefines the Grid allocation problem to type of scheduling problem.

To account for time attributes, Wellman et al. model single-sided auction protocols for the allocation and scheduling of resources under consideration of different time constraints (Wellman, Walsh et al. 2001). Conen goes one step further by designing a combinatorial bidding procedure for job scheduling including different running, starting, and ending times of jobs on a processing machine (Conen 2002). Both approaches are, however, single-sided and thus do not create competition on both sides. Demanding competition on both sides suggests the development of a combinatorial exchange. In literature, Parkes et al. introduce the first combinatorial exchange as a single-shot sealed bid auction (Parkes, Kalagnanam et al. 2001). As payment scheme, Vickrey discounts are approximated. Biswas and Narahari (Biswas and Narahari 2003) propose an iterative combinatorial exchange based on a primal/dual programming formulation of the allocation problem. By doing so, the preference elicitation problem can be alleviated, as the bidders can restrict their attention to some preferred bundles in contrast to all $2^{|\mathcal{G}|-1}$ possible combinations.

Obviously, both approaches neither accounts for time nor for quality constraints are thus not directly applicable for the Grid allocation problem. This paper intends to tailor a mechanism for allocating Grid by converting the aforementioned approaches into a combinatorial exchange that also incorporates time and quality constraints.

5 Introducing an Auction for Computational Grid

As mentioned above, the design of mechanisms mainly affects two components: (i) The communication language which defines how bids can be formalized and (ii) the outcome determination by means of allocation and pricing rules.

In the following, a bidding language will be introduced which fits the requirements specified in section 2. Furthermore, a winner determination model (allocation rule) and a family of pricing schemes will be introduced.

The following auction format follows common assumptions of mechanism design: Participants are assumed to be risk neutral, have linear utility functions, and have independent private valuations and reservation prices. Hence, the sellers' reservation prices can be linearly transformed to any partial execution of any bundle.

(i) Bidding Language

Allowing participants to submit multi-attribute combinatorial bids requires a formalized bidding language which will be introduced in the following:

Let N be a set of buyers and M be a set of sellers, where $n \in N$ defines an arbitrary buyer and $m \in M$ an arbitrary seller. Furthermore, there is a set G of discrete resources and a set of w bundles $S = \{S_1,\dots,S_w\}$ with $S_i \in S$ and $S_i \subseteq G$ as a subset of resources. A resource g has a set of l cardinal quality attributes $A_g = \{a_{g,1},\dots,a_{g,l}\}$ where $a_{g,j} \in A_g$ represents the j^{th} attribute of the resource g. Resources in form of bundles $S_i \in S$ can be assigned to a set T of discrete time slots, where $t \in T$ specifies one single time slot.

A buyer n can specify the minimal required quality characteristics for a bundle $S_i \in S$ with $q_n(S_i,g,a_{g,j}) \geq 0$, where $g \in S_i$ is a resource of the bundle S_i and $a_{g,j} \in A_g$ is a corresponding attribute of the resource g. Accordingly, a seller m can specify the maximal offered quality characteristics with $q_m(S_i,g,a_{g,j}) \geq 0$.

Furthermore, a buyer n can specify the minimum required number of time slots $s_n(S_i) \geq 0$ for a bundle $S_i \in S$. The earliest time slot for any allocatable bundle can be specified by $e_n(S_i) \geq 0$, the latest possible allocatable time slot by $l_n(S_i) \geq 0$.

A buyer can express the valuation for receiving a single slot of a bundle S_i by $v_n(S_i) \geq 0$, which determine the maximal price for which buyer n is willing to trade.

As an order, a buyer n can submit a set of XOR bundle bids $B_{n,1}(S_1) \oplus \cdots \oplus B_{n,u}(S_j)$, where u is the number of bundle bids in the order. A single bundle bid $B_{n,f}(S_i)$ is defined as the tuple

$$B_{n,f}(S_i) = (v_n(S_i),$$

$$(q_n(S_i,g_1,a_{g_1,1}),\dots,q_n(S_i,g_j,a_{g_j,c})),s_n(S_i),e_n(S_i),l_n(S_i)).$$

As an example, suppose a bundle $S_1 = \{CPU, HDD\}$ where each good has one attribute $A_{CPU} = \{SPEED\}$ and $A_{HDD} = \{SPACE\}$. The bid $B_{n,1}(\{CPU,HDD\}) = \{2,(700,300),6,2,10\}$ would express that a buyer wants to buy the bundle $S_1 = \{CPU, HDD\}$ with a CPU having at least 700 MHz and a hard disk having at least 300 GB of space. The buyer requires 6 slots of this bundle, which have to be fulfilled within a time range of slot 2 and slot 10. The valuation for a single slot for this bundle is $v_n(\{CPU,HDD\}) = 2$.

The orders of the sellers are formalized in a similar way as the buyers' orders are. A seller can express the reservation price for a single slot for a bundle S_i by $r_m(S_i) \geq 0$, which determines the minimum price for which seller m is willing to trade. An order is defined as a concatenated set of OR bundle bids $B_{m,1}(S_1) \vee \cdots \vee B_{m,u}(S_j)$, where u is the number of bundle bids. A single bid $B_{m,f}(S_i)$ is defined as the tuple

$$B_{m,f}(S_i) = (r_m(S_i), (q_m(S_i, g_1, a_{g_1,1}), \ldots, q_m(S_i, g_j, a_{g_j,c})), e_m(S_i), l_m(S_i)).$$

(ii) Winner Determination

For formalizing the winner determination model, the decision variables $x_n(S_i)$, $z_{n,t}(S_i)$, and $y_{m,n,t}(S_i)$ have to be introduced first. The binary variable $x_n(S_i)$ denotes, whether the bundle S_i is allocated to the buyer *n* $(x_n(S_i) = 1)$ or not $(x_n(S_i) = 0)$. Furthermore, the binary variable $z_{n,t}(S_i)$ is assigned to a buyer *n* and is associated in the same way as $x_n(S_i)$ with the allocation of S_i in time slot *t*. For a seller *m*, the real-valued variable $y_{m,n,t}(S_i)$ with $0 \leq y_{m,n,t}(S_i) \leq 1$ indicates the percentage contingent of the bundle S_i allocated to buyer *n* in time slot *t*. For example, $y_{m,n,t}(S_i) = 0.5$ denotes that 50 percent of the quality characteristics of bundle S_i are allocated from seller *m* to buyer *n* in time slot *t*. Thus, a partial allocation of a 700 MHz CPU with just 350 MHz would be expressed by $y_{m,n,t}(CPU) = 0.5$.

By means of these decision variables, the winner determination model can be formulated as described in Schnizler et al. (Schnizler, Neumann et al. 2004):[2]

$$max \sum_{n\in N}\sum_{S_i\in S}\sum_{t\in T} v_n(S_i) z_{n,t}(S_i) - \sum_{m\in M}\sum_{n\in N}\sum_{S_i\in S}\sum_{t\in T} r_m(S_i) y_{m,n,t}(S_i) \tag{1}$$

$$s.t. \sum_{S_i\in S} x_n(S_i) \le 1, \forall n \in N \tag{2}$$

$$\sum_{t\in T} z_{n,t}(S_i) \le x_n(S_i) s_n(S_i), \forall n \in N, \forall S_i \in S \tag{3}$$

$$\sum_{n\in N} y_{m,n,t}(S_i) \le 1, \forall m \in M, \forall S_i \in S, \forall t \in T \tag{4}$$

$$\begin{aligned} &\sum_{S_i\ni g, S_i\in S} x_n(S_i) s_n(S_i) q_n(S_i, g, a_{g,j}) \le \\ &\qquad \sum_{m\in M}\sum_{S_i\in S}\sum_{t\in T} y_{m,n,t}(S_i) q_m(S_i, g, a_{g,j}), \forall n \in N, \forall g \in G, \forall a_{g,j} \in A_g \end{aligned} \tag{5}$$

$$\begin{aligned} &\sum_{S_i\ni g, S_i\in S} z_{n,t}(S_i) q_n(S_i, g, a_{g,j}) \le \sum_{S_i\ni g, S_i\in S}\sum_{m\in M} y_{m,n,t}(S_i) q_m(S_i, g, a_{g,j}), \\ &\qquad \forall n \in N, \forall S_i \in S, \forall t \in T \end{aligned} \tag{6}$$

$$(t - e_n(S_i)) z_{n,t}(S_i) \ge 0, \forall n \in N, \forall S_i \in S, \forall t \in T \tag{7}$$

$$(l_n(S_i) - t) z_{n,t}(S_i) \ge 0, \forall n \in N, \forall S_i \in S, \forall t \in T \tag{8}$$

$$(t - e_m(S_i)) \sum_{n\in N} y_{m,n,t}(S_i) \ge 0, \forall m \in M, \forall S_i \in S, \forall t \in T \tag{9}$$

$$(l_m(S_i) - t) \sum_{n\in N} y_{m,n,t}(S_i) \ge 0, \forall m \in M, \forall S_i \in S, \forall t \in T \tag{10}$$

$$x_n(S_i) \in \{0,1\}, \forall n \in N, \forall S_i \in S \tag{11}$$

$$z_{n,t}(S_i) \in \{0,1\}, \forall n \in N, \forall S_i \in S, \forall t \in T \tag{12}$$

$$y_{m,n,t}(S_i) \ge 0, \forall n \in N, \forall m \in M, \forall S_i \in S, \forall t \in T \tag{13}$$

The objective function (1) maximizes the surplus V^* which is defined as the difference between the sum of the buyer's valuations $v_n(S_i)$ and the sum of the sellers' reservation prices $r_m(S_i)$. Assuming truthful bidders, the objective function reflects the goal of maximizing the social welfare in the economy. The first constraint (2) guarantees that each buyer n can be allocated only to one bundle S_i. This constraint is necessary to fulfill the XOR constraint of a buyer order. Constraint (3) ensures that for any allocated bundle S_i, the buyer receives

---

[2] The mechanism is a generalization of the combinatorial allocation problem (CAP) and therefore NP-complete.

not more than the required slots $s_n(S_i)$ within the time set T. For each time slot t, constraint (4) guarantees that each seller cannot allocate more than the seller possesses. For each resource, constraint (5) ensures that the sum of the supplied quality characteristics for all attributes over all sellers is greater than the demanded quality for each attribute of each buyer. Furthermore, it is guaranteed that for any allocated bundle in time slot t, all required resources have to be fulfilled in the same slot in at least the demanded qualities (constraint (6)). Finally the constraints (7)-(10) indicate that slots cannot be allocated before the earliest and after the latest time slot of neither any buyer (constraint (7), (8)), nor any seller (constraint (9), (10)). The constraints (11), (12), and (13) define the decision variables of the optimization problem.

(iii) Example

As an example of the model suppose there are two buyers n_1, n_2, three sellers m_1, m_2, m_3, and two resources $G = \{CPU, HDD\}$ each with one single attribute. The participants can submit bids on the bundles $S_1 = \{CPU\}, S_2 = \{HDD\}$, and $S_3 = \{CPU, HDD\}$. These bundles can be allocated within a time range of 5 slots, i.e. $T = \{0,...,4\}$. As shown in Table 1, the buyers submit a set of XOR orders, the sellers a set of OR orders.

N	S_i	$v_n(S_i)$	q_n	e_n	l_n	s_n	M	S_i	$r_m(S_i)$	q_n	e_n	l_n
n_1	HDD	2	100GB	2	4	1	m_1	CPU	2	1400MHz	0	3
	CPU, HDD	4	1000MHz, 150GB	0	4	4		HDD	2	100GB	0	3
n_2	CPU	2	400MHz	1	4	2	m_2	CPU	2	1000MHz	1	3
	CPU, HDD	4	800MHz, 200GB	0	4	3		HDD	3	200GB	1	3
							m_3	CPU, HDD	3	1000MHz, 200GB	3	4

Table 1: XOR orders of the buyers n_1, n_2 and OR orders of the sellers m_1, m_2, m_3.

The optimal solution of the mechanism is to allocate $S_3 = \{CPU, HDD\}$ to buyer n_1 ($x_{n_1}(S_3) = 1$) and to allocate $S_1 = \{CPU\}$ to buyer n_2 ($x_{n_2}(S_1) = 1$). The corresponding welfare is $V^* = 5.9$. The schedule and the seller's allocations are given in Table 2.

M	S_i	**0**	**1**	**2**	**3**	**4**
m_1	**CPU**		n_1,**1000MHz** n_2,**4000MHz**	n_1,**1000MHz** n_2,**4000MHz**	n_1,**250MHz**	
m_2	**HDD**		n_1,**150GB**	n_1,**150GB**		
m_3	**CPU,HDD**				n_1,**750MHz**, **100GB**	n_1,**1000MHz**, **200GB**

Table 2: Allocation schedule

Buyer n_1 receives the bundle $S_3=\{CPU,HDD\}$ from seller m_1, m_2, and m_3 in the time slots 1,2,3, and 4. For instance, buyer n_1 gets the CPU from seller m_1 and the HDD from seller m_3 in the time slots 1 and 2. Although an allocation from seller m_3 to buyer m_1 would realize a higher welfare in these time slots, seller m_3 cannot allocate the bundle before time slot 3. In time slot 3, however, this is realized by a co-allocation of seller m_1 and m_3 to buyer n_1. Finally, buyer n_1 gets the complete bundle from seller m_3 in time slot 4, because seller m_1 cannot allocate any bundle in this slot. Seller m_3 allocates the complete bundle, because partial executions of goods in a bundle are not possible[3] (e.g. a partial execution with 150GB of the hard disk).

Similarly, buyer n_2 receives the bundle $S_1=\{CPU\}$ from seller m_1 in two time slots.

Allocative Efficiency of the mechanism is assured, 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 schedule.

(iv) Pricing

The pricing problem in combinatorial exchanges is to determine the payments made by the participants and made by the exchange to agents after the exchange clears (Parkes, Kalagnanam et al. 2001).

Reviewing the requirements on the outcome as described in section 2, a pricing scheme based on a Vickrey-Clarke-Groves (VCG) mechanism would suffice the requirements. Moreover, Groves mechanisms are the only allocative-efficient and incentive compatible mechanism (Green and Laffont 1977).

---

[3] This restriction is required because complementary goods in a bundle cannot be priced individually.

The basic idea of a VCG mechanism is to grant a participant a discount on its bids based on the impact of that bid on the social welfare. A VCG mechanism is efficient, incentive-compatible, and individual rational for participants with quasi linear utility functions (Parkes 2001). However, Myerson and Satterthwaite proved that it is impossible to design an exchange which is incentive compatible, (interim) individually rational, and budget balanced that achieves efficiency in equilibrium (Myerson and Satterthwaite 1983). Hence, a VCG pricing schema is firstly introduced as a benchmark and secondly adapted by a VCG approximation mechanism to achieve budget balance.

(a) Vickrey Pricing

Let N' be a set of buyers and M' be a set of sellers who are part of the allocation (i.e. $x_n(S_i)=1$ with $n \in N'$ and $y_{m,n,t}(S_i)>0$ with $m \in M'$). The union of both sets is defined as $W = N' \cup M'$, where $w \in W$ is a participant who is part of the allocation.

Let V^* be the maximized value of the winner determination problem and $(V_{-w})^*$ be the maximized value of the allocation without participant w. Therefore, the Vickrey discount for a participant w can be calculated by $\Delta_{VICK,w}=V^*-(V_{-w})^*$, i.e. the impact of w's bid on the welfare V^*. In consideration of the Vickrey discounts, the Vickrey price $p_{VICK,n}(S_i)$ for a bundle S_i and a buyer n can be calculated by

$$p_{VICK,n}(S_i)=v_n(S_i)s_n(S_i)-\Delta_{VICK,n},$$

and the Vickrey price $p_{VICK,m}(S_i)$ for a bundle S_i and a seller m can be calculated by

$$p_{VICK,m}(S_i)=r_m(S_i)\sum_{n\in N'}\sum_{t\in T} y_{m,n,t}\ (S_i)+\frac{\Delta_{VICK,n}}{\alpha}.$$

where α denotes the number of bundles with which the seller m is part of the allocation4.

Applying the VCG schema to the above presented example ($V^*=5.9$) determines the prices and discounts shown in Table 3.

---

[4] A seller can be part of the allocation with multiple bundles, as the seller's orders are OR concatenated bundle bids.

	$(V_{-w})^*$	$\Delta_{VICK,w}$	$v_n(S_i)/r_m(S_i)$	$p_{VICK,w}(S_i)$
n_1	2.86	3.04	16	12.96
n_2	3.04	2.86	4	1.14
m_1	2.91	2.99	4.36	7.35
m_2	3.36	2.54	4.50	7.04
m_3	3.36	2.54	5.25	7.79

Table 3: Vickrey discounts and prices

According to Myerson and Satterthwaite's impossibility theorem, a VCG mechanism is efficient and individual rational, but not budget balanced (Myerson and Satterthwaite 1983). Aggregating the net payments of the example results into a negative value with 12.96+1.14-(7.35+7.04+7.79) = -8.08. In this case, the auctioneer has to subsidize the double auction, which is practically not sustainable.

Retaining most of the VCG properties, a possible implementation of a budget-balanced pricing scheme for double auctions is the so-called approximated VCG pricing mechanism introduced by Parkes et al. (Parkes, Kalagnanam et al. 2001).

(b) Approximated VCG Pricing

The idea is to cleave on the budget balance and individual rational constraints and approximate the Vickrey discounts resulting in a relaxation of the incentive compatibility and thus allocative efficiency requirement. This is realized by minimizing a function L which denotes the distance between the set of the original Vickrey discounts Θ_{VICK} and the set of the approximated discounts Θ, where $\Delta_{VICK,w} \in \Theta_{VICK}$ is the original Vickrey discount and $\Delta_{w\in} \in \Theta$ is the approximated discount for participant w.

Parkes et al. formulate this problem as the following linear program (Parkes, Kalagnanam et al. 2001):

$$\min_{\Theta} L(\Theta, \Theta_{VICK}) \tag{14}$$

$$s.t. \sum_{w \in W} \Delta_w \leq V^* \tag{15}$$

$$\Delta_w \leq \Delta_{VICK,w}, \forall w \in W \tag{16}$$

$$\Delta_w \geq 0, \forall w \in W \qquad (17)$$

The objective (14) minimizes a distance function between the original Vickrey and the approximated discounts. The first constraint (15) guarantees the budget-balance property, so that the exchange never has to transfer net payments to the participants. The second constraint (16) ensures that no participant gets more than the original Vickrey discount. The last constraint (17) guarantees the individual rational property.

Parkes et al. (Parkes, Kalagnanam et al. 2001) indicate among others the following distance functions $L(\Theta,\Theta_{VICK})$ for this problem: The quadratic error function $L_2(\Theta,\Theta_{VICK}) = \sum_{w \in W} (\Delta_{VICK,w} - \Delta_w)^2$ (Threshold approximation), the squared relative error function $L_{RE2}(\Theta,\Theta_{VICK}) = \sum_{w \in W} \frac{(\Delta_{VICK,w} - \Delta_w)^2}{\Delta_w}$ (Fractional approximation), and the product error function $L_{\Pi}(\Theta,\Theta_{VICK}) = \prod_{w \in W} (\frac{\Delta_{VICK,w}}{\Delta_w})$ (Reverse approximation).

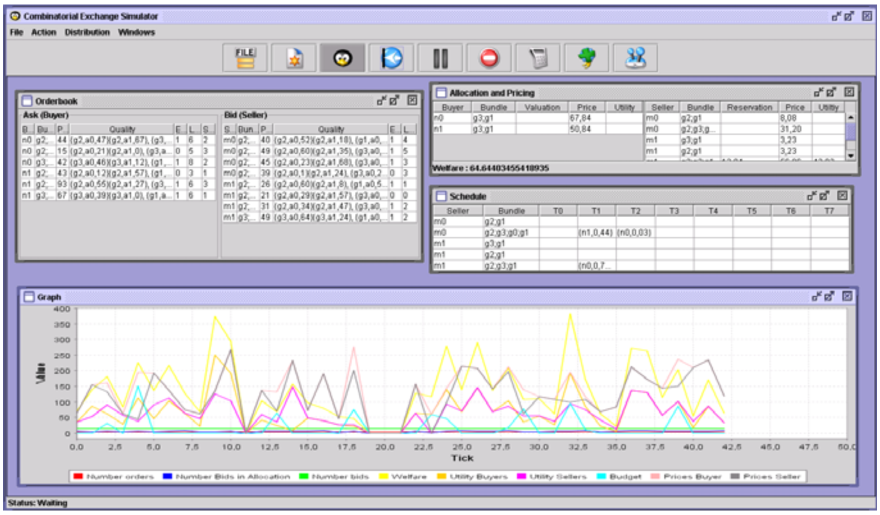
Applying these approximations on the above presented example determines the approximated discounts and the prices in Table 4. In this case, the exchange does not have to endow the participants as it fulfills the weak budget balance property.

w	L_2		L_{RE2}		L_{Π}	
	Δ_w	$p_w(S_i)$	Δ_w	$p_w(S_i)$	Δ_w	$p_w(S_i)$
n_1	1.42	14.58	1.28	14.72	1.18	14.82
n_2	1.24	2.76	1.21	2.79	1.18	2.82
m_1	1.38	5.74	1.26	5.62	1.18	5.54
m_2	0.92	5.42	1.07	5.57	1.18	5.68
m_3	0.92	6.17	1.07	6.32	1.18	6.43

Table 4: Approximated Vickrey discounts and prices

6 Implementation

The presented mechanism is implemented in a Java based simulation environment as shown in figure 1. The simulation tool is capable of generating different environments (different number of participant, resources, and bundles) and different order flows and therefore enables the evaluation of economical and technical properties. Figure 2 sketches briefly the main components of the market simulator.



Ask (Buyer)

B...	Bu...	P...	Quality	E...	L...	S...
n0	g2;...	44	(g2,a0,47)(g2,a1,67), (g3,...	1	6	2
n0	g2;...	15	(g2,a0,21)(g2,a1,0), (g3,a...	0	5	3
n0	g3;...	42	(g3,a0,46)(g3,a1,12), (g1,...	1	8	2
n1	g2;...	43	(g2,a0,12)(g2,a1,57), (g1,...	0	3	1
n1	g2;...	93	(g2,a0,55)(g2,a1,27), (g3,...	1	6	3
n1	g3;...	67	(g3,a0,39)(g3,a1,0), (g1,a...	1	6	1

Bid (Seller)

S...	Bun...	P...	Quality	E...	L...
m0	g2;...	40	(g2,a0,52)(g2,a1,18), (g1,a0,...	1	4
m0	g2;...	49	(g2,a0,60)(g2,a1,35), (g3,a0,...	1	5
m0	g2;...	45	(g2,a0,23)(g2,a1,68), (g3,a0,...	1	3
m0	g2;...	39	(g2,a0,1)(g2,a1,24), (g3,a0,2...	0	3
m1	g2;...	26	(g2,a0,60)(g2,a1,8), (g1,a0,5...	1	1
m1	g2;...	21	(g2,a0,29)(g2,a1,57), (g3,a0,...	0	0
m1	g2;...	31	(g2,a0,34)(g2,a1,47), (g3,a0,...	1	2
m1	g3;...	49	(g3,a0,64)(g3,a1,24), (g1,a0,...	1	2

Buyer	Bundle	Valuation	Price	Utility
n0	g3;g1		67,84	
n1	g3;g1		50,84	

Seller	Bundle	Reservation	Price	Utility
m0	g2;g1		8,08	
m0	g2;g3;g...		31,20	
m1	g3;g1		3,23	
m1	g2;g1		3,23	

Welfare : 64.64403455418935

Seller	Bundle	T0	T1	T2	T3	T4	T5	T6	T7
m0	g2;g1								
m0	g2;g3;g0;g1		(n1,0,44)	(n0,0,03)					
m1	g3;g1								
m1	g2;g1								
m1	g2;g3;g1		(n0,0,7...						

Figure 1: Combinatorial Exchange Simulator

The central component is the *Market* class which instantiates an *Environment* (storing participants, goods, and bundles), an *Orderbook* (storing bids), a *Mechanism* (implementing the winner determination and pricing mechanism), and *GUIObserver* components (responsible for visualization).

The *Environment* and the *Orderbook* can be filled by either XML based files (*EnvironmentFile*, *OrderbookFile*) using several distributions (*EnvironmentDistribution*, *OrderbookDistribution*). The decision whether to use file based or distribution generated data is made by the *EnvironmentProviderFactory* and the *OrderbookProviderFactory*.

The *Mechanism* encapsulates the market mechanism by instantiating an *Outcome* class and a *Pricing* class. The *Outcome* class is responsible for the winner determination problem and uses a specific solver (e.g. *CPLEXAdapater*). The *Pricing* class instantiates a pricing mechanism (e.g. *VCGPricing*) that determines the net payments.

Finally, the *AbstractGUIObserver* is an abstract component for visualizing the data (e.g. the order book).

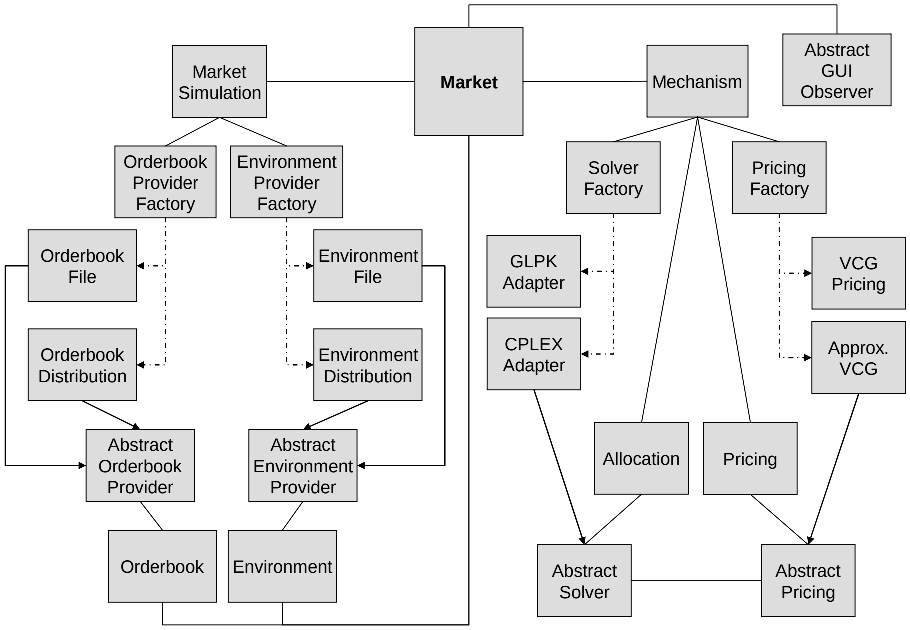


Figure 2: Architecture of the Simulation Tool

Computational Tractability

Having implemented the mechanism, it is tested upon its conformance to the requirements. In a first step, the computational tractability of the mechanism is analyzed by means of a run-time simulation.

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.

The bundles are generated using the Decay distribution. 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. show, that the Decay distribution can lead to hard instances of general combinatorial allocation problems (Sandholm, Suri et al. 2002).

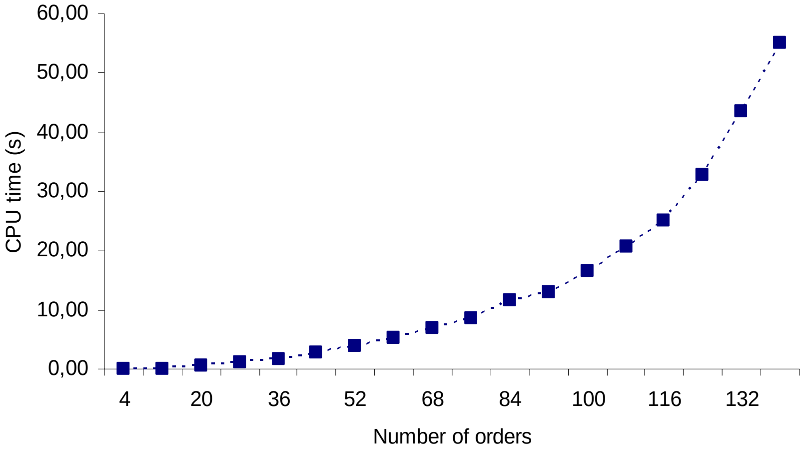


Figure 3: Performance simulation results

Figure 3 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 MACE is computationally very demanding. For more complex scenarios, the use of approximations have to be examined. Tests of the economic requirements are adressed to future work.

7 Concluding Remarks

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 inapplicabel. In this paper, we motivate the 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. At its current state, the economic properties of the market mechanism are untested. There is strong evidence that the envisioned Approximated Vickrey Auction achieves fairly good results, but the proof still remains for future work.

References

Biswas, S. and Y. Narahari (2003). "Iterative dutch combinatorial auctions." Annals of Mathematics and Artificial Intelligence: forthcoming.

Buyya, R., H. Stockinger, J. Ghiddy and D. Abramson (2001). Economic models for management of resources in peer-to-peer and grid computing. International Conference on Commercial Applications for High Performance Computing, Denver, CO.

Conen, W. (2002). Economically coordinated job shop scheduling and decision point bidding - an example for economic coordination in manufacturing and logistics. Workshop on Planen, Scheduling und Konfigurieren, Entwerfen, Freiburg.

Foster, I. and C. Kesselman (2004). The Grid 2. San Francisco, Morgan Kaufmann.

Fudenberg, D. and J. Tirole (2000). Game Theory. Cambridge, MA and London, England, The MIT Press.

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

Jackson, M. O. (2002). Mechanism Theory. Encyclopedia of Life Support Systems, UNESCO -online.

Kalagnanam, J. and D. C. Parkes (2003). Auctions, Bidding and Exchange Design. Supply Chain Analysis in the eBusiness Era. S. D. W. a. Z. M. S. David Simchi-Levi, Kluwer Academic Publishing**:** forthcoming.

Krishna, V. and M. Perry (1998). "Efficient Mechanism Design." Working Paper.

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

Myerson, R. and M. Satterthwaite (1983). "Efficient Mechanisms for Bilateral Trading." Journal of Economic Theory **29**(2): 265-281.

Nisan, N., S. London, O. Regev and N. Camiel (1998). Globally distributed computation over the internet - the popcorn project. 18th International Conference on Distributed Computing Systems, Amsterdam, The Netherlands, IEEE Computer Society.

Parkes, D. C. (2001). Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency. Philadelphia.

Parkes, D. C., J. Kalagnanam and M. Eso (2001). Achieving budget-balance with vickrey-based payment schemes in exchanges. International Joint Conference on Artificial Intelligence1161–1168.

Sandholm, T., S. Suri, A. Gilpin and D. Levine (2002). Winner determination in combinatorial auction generalizations. International Joint Conference on Autonomous Agents and Multiagent Systems.

Schnizler, B., D. Neumann and C. Weinhardt (2004). Resource Allocation in Computational Grids - A Market Engineering Approach. WeB 2004, Washington 19-31.

Subramoniam, K., M. Maheswaran and M. Toulouse (2002). Towards a micro-economic model for resource allocation in grid computing systems. IEEE Canadian Conference on Electrical & Computer Engineering.

Wellman, M. P., W. E. Walsh , P. Wurman and J. MacKie-Mason (2001). "Auction protocols for decentralized scheduling." Games and Economic Behavior **35(**271-303).

Wolski, R., J. Brevik, J. Plank and T. Bryan (2003). Grid resource allocation and control using computational economies. Grid Computing - Making The Global Infrastructure a Reality, John Wiley & Sons**:** chapter 32.

Wurman, P. (1999). Market Structure and Multidimensional Auction Design for Computational Economies. School of Information. Ann Arbor, University of Michigan.