Market Based Coordination in the Computational Grid - A Market Engineering Approach

Author: Björn Schnizler

University of Karlsruhe (TH), Information Management and Systems

Englerstraße 14, 76131 Karlsruhe, +49 (0) 721 608 8385

bjoern.schnizler@iw.uni-karlsruhe.de

Supervisor: Prof. Dr. Christof Weinhardt

University of Karlsruhe (TH), Information Management and Systems

Englerstraße 14, 76131 Karlsruhe, +49 (0) 721 608 8370

christof.weinhardt@iw.uni-karlsruhe.de

Summary: The Computational Grid is an infrastructure for providing access to distributed high-end computational resources. Therefore, it enables the execution of computationally demanding applications. One of the key problems in building such an infrastructure is to decide which jobs are to be allocated to which resources at what time. A promising approach for solving this problem is the use of market mechanisms for allocating and scheduling these resources. This contribution outlines the design and implementation of a family of market mechanisms for allocating and scheduling computer resources in the Computational Grid.

Keywords: Computational Grid, Combinatorial Exchange, Multiattribute Auctions, Market Engineering

1 Introduction

1.1 Motivation and Problem

The increasing interconnection between computers through the Internet has created the vision of Computational Grids. In analogy to the power grid, computer resources such as processors or hard disks can be accessed in a plug-and-play environment. A user has access to a reliable virtual computer, which consists of many heterogeneous computer resources. These resources are not visible to the user – such as a consumer of electric power is unaware of how the demanded electricity is being generated and thereafter transmitted to the power socket.

Connecting multiple computer resources to a virtual computer enables the execution of complex and computationally demanding applications such as large-scaled simulations or real-time risk-analysis. Thus, the Computational Grid is expected to have a beneficial impact on scientific organizations as well as business companies.

At the moment, most of the research in the area of the Computational Grid focuses in particular on the hardware and software infrastructure, such that from a technical point of view, “the access to resources is dependable, consistent, pervasive, and inexpensive” [FoKe04]. Nonetheless, there are still barriers preventing the deployment of large-scaled Computational Grid infrastructures.

One of the key issues in building a Computational Grid is to determine which computer resources are allocated to which applications and scheduled at what time. Most Computational Grid infrastructures such as Legion[1] or Condor[2] employ optimization algorithms, which allocate and schedule resources based on static system specific cost functions [BuAV04].

However, static system specific cost functions typically lead to economically inefficient allocations. These functions do not guarantee that those demanding users will receive their supplied resources who value them highest. Furthermore, these functions ignore the fact that users owning resources only have incentives offering resources, if they are adequately compensated. Compensation requires determining how the supplied resources are allocated among potential buyers and how the prices for the resources are set. Both aspects are crucial for implementing economic efficient Computational Grid infrastructures.

1.2 Objective and Outline

Recently, researchers have increasingly suggested employing market mechanisms for allocating and scheduling resources in the Computational Grid [BuAV04; WPBB03]. According to Hurwicz, markets can be an effective institution to allocate resources (Pareto-) optimal [Hurw72]. This is achieved by the interplay of demand and supply and due to the information feedback inherent to the price system. As such, the application of market mechanisms to the Computational Grid as an allocation and scheduling mechanism is deemed promising.

[1] http://legion.virginia.edu/
[2] http://www.cs.wisc.edu/condor/

The objective of the thesis is to design, to implement, and to evaluate a market which is capable of determining an efficient allocation and schedule for computer resources in the Computational Grid.

The difficulty in designing such a market is that the underlying mechanism through which the participants act can have a profound impact on the results of that interaction [Jack02]. For instance, in a sealed bid auction mechanism the participants do not learn as much about the valuations of the other participants as in an open cry auction. The higher information feedback may affect the bidding behavior of the market participants and could therefore lead to different outcomes. Furthermore, the market infrastructure (e.g. ways of communication) as well as the business structure (e.g. trading fees) can also influence the behavior of the participants and therefore the outcome of the mechanism [WeHN03].

The Market Engineering approach manages these influences by means of a structured, systematic, and theoretically founded procedure of analyzing, designing, evaluating, and introducing electronic market platforms [WeHN03; Neum04].

Using the methodology of Market Engineering - which will be introduced in section 2 - the following objectives constitute the outline of this paper:

1. **Environmental Analysis:** In a first step, the market participants, their preferences as well as the traded objects have to be analyzed. Based upon this, requirements for a market mechanism for the Computational Grid will be elicited (section 3.1).

2. **Design of a market mechanism:** In the second step, a market mechanism has to be designed. The difficulty in this stage arises out of the fact, that the rules of the mechanism have to be set in a way that the desired outcome specified in the requirement analysis can be realized (section 3.2.1).

3. **Implementation:** After designing the market mechanism, the formally specified trading rules have to be implemented into a software system.

4. **Evaluation:** In the fourth step, the market mechanism has to be evaluated whether the defined requirements are achieved or not (section 3.3).

Finally, section 4 summarizes the contribution and gives an overview on future work.

2 Market Engineering

In the general context of engineering, a design method refers to a way, procedure, or technique for solving an individual design problem. These design methods can be either intuitive or discursive. Intuitive approaches involve creativity in the form of complex associations of ideas and aim at increasing the flow of ideas. However, the results of intuitive approaches strongly depend on the designer's expertise, skills, and experiences. As such, intuitive approaches may fail to achieve suitable solutions for complex problems. Discursive approaches are strategies which decompose a complex design problem into several smaller, less complex problems. The design strategy intends to describe a step-by-step procedure to aid the designer in the matching of the unique problem situation along the overall design process with the available design methods.

The design of a market mechanism is a complex and interdependent task and therefore, the approach of Market Engineering aims at the discursive, goal-oriented development of market institutions. The corresponding Market Engineering process model and a selection of associated methods are outlined in Figure 1 [WeHN03] and briefly described based on the work of Neumann [Neum04].

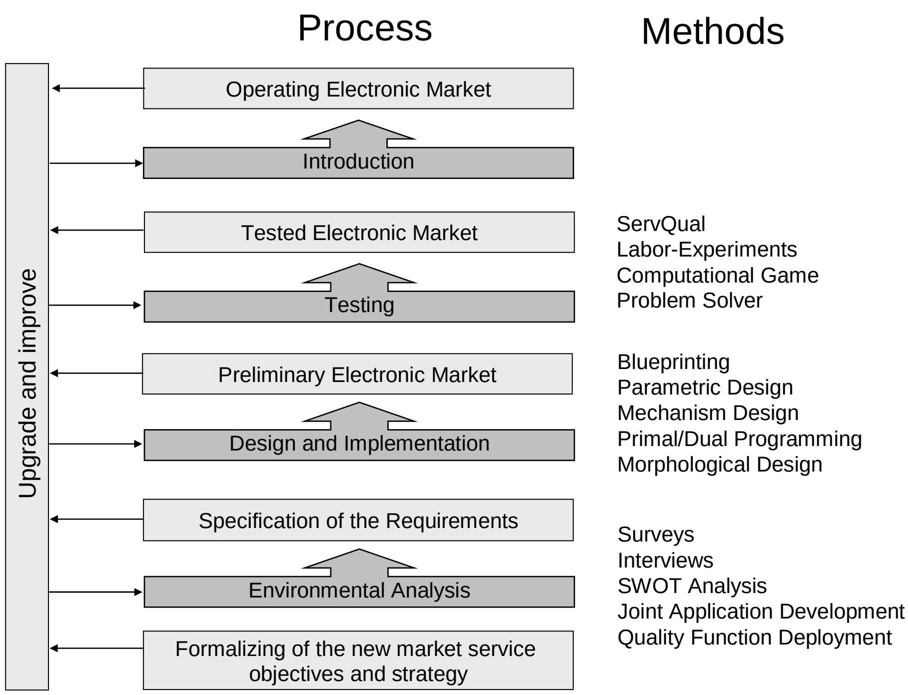

Figure 1: Design Process and Methods

In the first stage - the environmental analysis - the objectives and the strategy of the new electronic market are formalized. The stage comprises two different phases: the environment definition and the requirement analysis: The goal of the environment definition is to collect information about the transaction objects (e.g. what resources are traded), the participants (e.g. who are the potential participants), as well as their preferences, endowments, and constraints. Subsequently, the needs and requirements of the participants are extracted in the requirement analysis phase.

The objective of the second stage - the Design and Implementation stage - is the conceptual and formal design of an electronic market and its transformation into an information system[3].

In the third stage - the testing stage - the implemented electronic market is tested upon its functionality and its economic properties. After the testing phase, the electronic market can be introduced.

At any stage of the process there is a decision (e.g. supported by prototypes), whether to proceed with the next step or to repeat the prior one.

3 A Market for the Computational Grid

3.1 Step 1 – Environmental Analysis

The environmental analysis comprises information about the potential participants, their characteristics and needs, the traded resources, as well as their endowments. As described in section 2, the environmental analysis comprises the environment definition and the requirement analysis.

3.1.1 Environmental Definition

Surveying the literature and following Czajkowski et al., the market place for trading Computational Grid resources can be represented as shown in Figure 2 [CzFK04]. The market is spanned around resource owners as sellers (e.g. server farms), resource consumers as buyers (e.g. scientists at universities), and intermediaries as middleware components (e.g. Globus Toolkit [FoKe04]).

[3] This stage is decomposed into four major phases being the conceptual design, embodiment design, detail design, and implementation. For further details, the reader is referred to [Neum04].

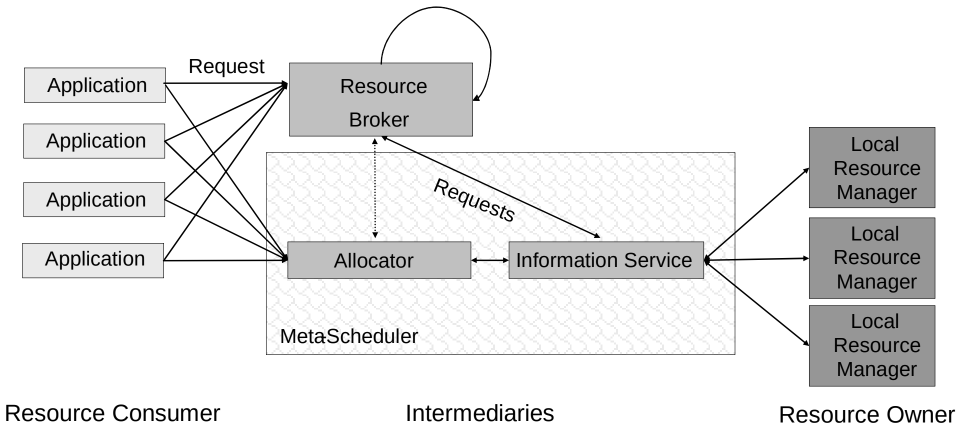

Figure 2: The Market for Computational Grids

The intermediaries technically provide the resource management infrastructure for exploiting remote resources. According to Czajkowski et al, the intermediary level consists of three basic components [CzFK04]:

- **The Resource Broker** is responsible for resource discovery, selection, aggregation, and subsequently for the data and program routing.
- **The Information Services** provides persistent access to information about the current availability and capability of resources.
- **The Allocator** coordinates the allocation of resources at multiple sites. The allocator and the information service have the responsibility of scheduling jobs.

Based upon this view on the intermediary layer, the market mechanism for the Computational Grid can be sketched as follows: The allocation and scheduling performed by the Resource Broker can be transferred to the market mechanism. Instead of sending requests to the information service, the resource broker translates the user requirements into bids. Those bids expressing demand and supply build the base for allocating and scheduling resources by the market mechanisms [ScNW04].

3.1.2 Requirement Analysis

Based upon this view of a market mechanism for the Computational Grid, the necessary requirements for a mechanism can be specified. Firstly, these requirements are desirable requirements derived from the theory of Mechanism Design [Jack02] and, secondly, domain specific requirements from the Computational Grid environment [ScNW04].

Within the scope of classical Mechanism Design it is the goal to investigate a mechanism that is applicable in certain situations. To achieve this objective, the designed mechanism has to fulfill the following requirements [Jack02; Park01; Neum04]:

- **Allocative efficiency** should be determined by maximizing the total value over all participants.
- **Budget balancy** means that all payments made to the mechanism are redistributed among the participants. A relaxed requirement is weak budged balance, i.e. net payments are made from the participants to the mechanism, but no vice versa.
- **Individual rationality** ensures that a participant achieves at least as much utility from participating in the mechanism as without participating.
- **Incentive compatibility** requires that all participants report their preferences truthfully.
- **Computational tractability** considers the complexity of computing the outcome [Park01].
- **Communication tractability** considers the minimization of communication effort that is required to converge on a desirable global outcome [Park01].

The requirements elicited through mechanism design theory do, however, not fulfill all requirements for a Computational Grid mechanism. The underlying environment and its properties stem from domain specific requirements as described in the following:

- **Simultaneous trading** of multiple sellers and multiple buyers is required by the mechanism and can be realized by a double-sided mechanism.
- **Trading dependent resources** is required by the mechanism. Buyers usually demand a combination of computer resources as a bundle to perform a task [SuMT02]. This is based on the fact that resources in the Computational Grid are complementarities ($v(A)+v(B) \leq v(AB)$) . Thus, the mechanism is required to support bids on bundles. Furthermore, a 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, i.e. the buyer has subadditive valuations ($v(A)+v(B) \geq v(AB)$)[4]. As such, the mechanism must support XOR[5]-bids to express substitutes. Bids of the sellers have to be restricted to a set of OR[6]-bids, as computer resources are usually no substitutes for the sellers. This can be derived from the fact, that computer resources are non-storable commodities, i.e. the CPU power available at a present time cannot be stored for a later time. Furthermore, a seller may be also willing to

[4] For instance, a 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.

[5] A XOR B (A $\oplus$ B) means either $\phi, A, B,$ but not AB.

[6] A OR B (A $\vee$ B) is defined as $\phi, A, B,$ or AB.

sell just a part of the offered resources. Thus, the possibility of a partial executing on the sellers' side must be given.

- **Support for multiattribute resources** should be realized by the mechanism, as resources in the Computational Grid are typically not completely standardized. Similar resources can differ in their quality, e.g. a hard disk by its capacity (in GB) or access time (in ms). Hence, minimum quality requirements must be met, while similar resources of superior quality work as well.
  Furthermore, buyers usually require resources for a certain time span. However, the exact timing of the computation is not always important. For instance, a 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 a certain time, e.g. 3 p.m. Thus, the mechanism must allow for placing bids on time attributes[7].

3.2 Meeting the Requirements

A number of mechanisms have been proposed that attempt to solve the allocation problem in Computational Grid systems or related architectures.

Most of these mechanisms are central in nature in a way that the allocation problem is solved by a central entity using global optimization algorithms without the employment of prices. This central entity requires detailed information about the demand and supply situation in order to be effective. As information is dispersed among the buyers and sellers, central allocation algorithms may not enfold their power, because this information requirement is not even closely met. Market-based approaches incorporate incentives for truthful information revelation by implementing prices.

Wolski et al. suggest the use of traditional auction formats such as English auctions [WPBB03]. The use of traditional auction formats in the Grid environment, however, is delimited, as the trading objects are traded as unbundled standardized commodities without any quality characteristics. 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 a part of the bundle without the other. To avoid this, Subramoniam et al. employ the use of ascending bundling auctions [SuMT02]. Nonetheless, the resources are still considered to be standardized commodities, as no quality characteristics are supported.

7 As a technical requirement, it is required that jobs can be interrupted at any time. Furthermore, the specification of available time slots is restricted to buyers, although this could be useful for a seller. Solving this problem will be addressed in future work.

Reviewing the requirements on the mechanism, it becomes evident 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.

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 [WWWK01]. The approach, however, is single-sided and thus do not create competition on both sides.

Parkes et al. introduce the first combinatorial exchange as a single-shot sealed bid auction [PaKE01]. Biswas and Narahari [BiNa03] propose an iterative combinatorial exchange based on a primal/dual programming formulation of the allocation problem. However, both approaches neither account for time nor for quality constraints and are thus not directly applicable for the Computational Grid allocation problem.

The work presented in this contribution intends to tailor a mechanism for allocating Computational Grid resources by converting the aforementioned requirements into a combinatorial exchange that additionally incorporates time and quality constraints.

3.3 Step 2 – Design and Implementation

3.3.1 Design

Designing a combinatorial exchange mainly concerns two phases: winner determination and pricing. The winner determination allocates resources of sellers among buyers. Afterwards, the pricing mechanism determines the net payments to the participants.

The following exchange model[8] follows common assumptions of mechanism design: Participants are assumed to be risk neutral, have linear utility functions, and have independent private valuations. Hence, the sellers' reservation prices can be linearly transformed to any partial execution of any bundle.

(i) Bidding Language

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

[8] For simplicity the communication costs of the Computational Grid are neglected. Instead it is assumed that each agent can submit jobs to any resource. Including the communication costs is addressed to future work.

Let N be a set of $|N|$ buyers and M be a set of $|M|$ sellers, where $n \in N$ defines an arbitrary buyer and $m \in M$ an arbitrary seller. Furthermore, there is a set of $|G|$ discrete resources $G = \{g_1, \ldots, g_{|G|}\}$ and a set of $|S|$ bundles $S = \{S_1, \ldots, S_{|S|}\}$ with $S_i \in S$ and $S_i \subseteq G$ as a subset of resources. A resource g_k has a set of $|A_k|$ cardinal quality attributes $A_{g_k} = \{a_{g_k,1}, \ldots, a_{g_k,|A_k|}\}$ where $a_{g_k,j} \in A_{g_k}$ represents the j^{th} attribute of the resource g_k. Resources in form of bundles $S_i \in S$ can be assigned to a set of maximal $|T|$ discrete time slots $T = \{0, \ldots, |T|-1\}$, 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 $q_n(S_i, g_k, a_{g_k,j}) \geq 0$, where $g_k \in S_i$ is a resource of the bundle S_i and $a_{g_k,j} \in A_{g_k}$ is a corresponding attribute of the resource g_k. Accordingly, a seller m can specify the maximal offered quality characteristics with $q_m(S_i, g_k, a_{g_k,j}) \geq 0$.

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

$$B_{n,f}(S_i) = (v_n(S_i), (q_n(S_i, g_1, a_{g_1,1}), \ldots, q_n(S_i, g_j, a_{g_j,|A_j|})), s_n(S_i), e_n(S_i), l_n(S_i)).$$

As an example, suppose a bundle $S_1 = \{CPU, HDD\}$ where each good has one attribute $A_{CPU} = \{SPEED\}$ and $A_{HDD} = \{SPACE\}$. $\mathrm{B}_{n,1}(\{\mathrm{CPU}, \mathrm{HDD}\}) = \{2, (6 \cdot 10^6, 300), 6, 2, 10\}$ would express that a buyer wants to buy the bundle $S_1 = \{CPU, HDD\}$ with a CPU having at least $6 \cdot 10^6$ FLOPS 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 determine 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

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

(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*. For instance, 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. [ScNW04]:

$$\max \sum_{n\in N}\sum_{S_i\in S}\sum_{t\in T} v_n(S_i) z_{n,t}(S_i) - \sum_{m\in M}\sum_{n\in N}\sum_{S_i\in S}\sum_{t\in T} r_m(S_i) y_{m,n,t}(S_i)^{9} \tag{1}$$

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

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

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

$$\sum_{S_i \ni g_k, S_i\in S} x_n(S_i) s_n(S_i) q_n(S_i, g_k, a_{g_k,j}) \leq \tag{5}$$

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

$$\sum_{m \in M} \sum_{S_i \in S} \sum_{t \in T} y_{m,n,t}(S_i) q_m(S_i, g_k, a_{g_k,j}), \forall n \in N, \forall g_k \in G, \forall a_{g_k,j} \in A_{g_k}$$

$$\sum_{S_i \ni g_k, S_i \in S} z_{n,t}(S_i) q_n(S_i, g_k, a_{g_k,j}) \leq \sum_{S_i \ni g_k, S_i \in S} \sum_{m \in M} y_{m,n,t}(S_i) q_m(S_i, g_k, a_{g_k,j}), \quad \forall n \in N, \forall S_i \in S, \forall t \in T \tag{6}$$

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

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

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

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

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

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

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

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 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 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.

As an example for the model suppose there are two buyers n_1, n_2, three sellers m_1, m_2, m_3, and two resources $G = \{g_1, g_2\}$ each with one single attribute. The participants can submit bids on the bundles $S_1 = \{g_1\}$, $S_2 = \{g_2\}$, and $S_3 = \{g_1, g_2\}$. These bundles can be allocated within a time range of 5 slots, i.e. $T = \{0,...,4\}$. The buyers submit a set of XOR orders (shown in table 1) and the sellers a set of OR bids (shown in table 2).

Buyer	S_i	$v_n(S_i)$	q_n	e_n	l_n	s_n
n_1	$\{g_1\}$	2	$\{g_1, a_1, 500\}$	0	3	3
	$\{g_2\}$	2	$\{g_2, a_1, 10\}$	2	4	1
	$\{g_1, g_2\}$	4	$\{g_1, a_1, 500\}$, $\{g_2, a_1, 15\}$	0	4	4
n_2	$\{g_1\}$	2	$\{g_1, a_1, 200\}$	1	4	2
	$\{g_2\}$	1	$\{g_2, a_1, 15\}$	0	4	3
	$\{g_1, g_2\}$	4	$\{g_1, a_1, 400\}$, $\{g_2, a_1, 20\}$	0	2	1

Table 1: XOR orders of buyer n_1 and n_2

Seller	S_i	$r_m(S_i)$	q_n	e_n	l_n
m_1	$\{g_1\}$	2	$\{g_1, a_1, 700\}$	0	3
	$\{g_2\}$	2	$\{g_2, a_1, 10\}$	0	3
m_2	$\{g_1\}$	2	$\{g_1, a_1, 500\}$	1	3
	$\{g_2\}$	3	$\{g_2, a_1, 20\}$	1	3
m_3	$\{g_1, g_2\}$	3	$\{g_1, a_1, 500\}$,$\{g_2, a_1, 20\}$	3	4

Table 2: OR orders of seller m_1, m_2, and m_3

The optimal solution of the mechanism is to allocate $S_3 = \{g_1, g_2\}$ to buyer n_1 ($x_{n_1}(S_3) = 1$) and to allocate $S_1 = \{g_1\}$ to buyer n_2 ($x_{n_2}(S_1) = 1$). The welfare is $V^* = 5.9$. The corresponding schedule and the seller's allocations are given in table 3.

Buyer n_1 receives the bundle $S_3 = \{g_1, g_2\}$ 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 good g_1 from seller m_1 and the good g_2 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[10] (e.g. a partial execution with 15GB of the hard disk).

Similarly, buyer n_2 receives the bundle $S_1 = \{g_1\}$ from seller m_1 in two time slots.

	S_i	0	1	2	3	4
m_1	$\{g_1\}$		$\{n_1,500\}$,	$\{n_1,500\}$,	$\{n_1,125\}$	
m_2	$\{g_2\}$		$\{n_1,15\}$	$\{n_1,15\}$		
m_3	$\{g_1,g_2\}$				$\{n_1,375;15\}$	$\{n_1,500;20\}$

Table 3: Allocation schedule

(iii) Pricing

The pricing problem in combinatorial exchanges is to determine the payments made by the participants to the exchange based on the participants' bids.

Considering the definitions in section 3.1.2, a pricing schema based on a Vickrey-Clarke-Groves (VCG) mechanism would suffice the requirements. The basic idea of a VCG mechanism is to grant a participant a discount on the participant's bids based on the impact of that bid on the social welfare. A VCG mechanism is efficient, incentive-compatible, individual rational and budget balanced for participants with quasi linear utility functions [Park01]. However, Myerson and Satterthwaite proofed that it is impossible to design an exchange which is incentive compatible, interim individually rational, and budget balanced that achieves efficiency in equilibrium [MaSa83].

For illustrating this theorem, a VCG pricing schema is firstly introduced, followed by a VCG approximation mechanism.

(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 par-

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

ticipant *w* can be calculated by $\Delta_{VICK,w} = V^* - (V_{-w})^*$. 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

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

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

Applying the VCG schema to the above presented example ($V^* = 5.9$) determines the prices and discounts shown in table 4.

Participant	$(V_{-w})^*$	$\Delta_{VICK,w}$	$v_n(S_i); r_m(S_i)$	$p_{VICK,w}(S_i)$
n_1	2.86	3.04	16	12.96
n_2	3.04	2.86	4	1.14
m_1	0	5.9	4.36	10.25
m_2	0	5.9	4.50	10.39
m_3	0	5.9	5.25	11.14

Table 4: Vickrey discounts and prices

A VCG mechanism is efficient and individual rational, however, in an exchange not budget balanced. Aggregating the net payments of the example results into a negative value with 12.96+1.14-(10.25+10.39+11.14) = -17.96. In this case, the auctioneer has to endow the exchange, which is practical not realizable.

Holding most of the VCG properties, a possible implementation of a budget-balanced pricing schema for exchanges is the so-called approximated VCG pricing mechanism introduced by Parkes et al. [PaKE02].

(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 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 [PaKE01]:

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

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

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

$$\Delta_w \geq 0, \forall w \in W \tag{17}$$

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. [PaKE02] indicate among others the following distance functions $L(\Theta, \Theta_{VICK})$ for this problem: The quadratic error function $L_2(\Theta, \Theta_{VICK}) = \sum_{w \in W} (\Delta_{VICK,w} - \Delta_w)^2$, the squared relative error function $L_{RE2}(\Theta, \Theta_{VICK}) = \sum_{w \in W} (\Delta_{VICK,w} - \Delta_w)^2$, and the product error function $L_{\Pi}(\Theta, \Theta_{VICK}) = \prod_{w \in W} (\frac{\Delta_{VICK,w}}{\Delta_w})$.

	L_2		L_{RE2}		L_{Π}	
	Δ_w	$p_w(S_j)$	Δ_w	$p_w(S_j)$	Δ_w	$p_w(S_j)$
n_1	0	16	0.76	15.24	1.18	14.82
n_2	0	4	0.71	3.29	1.18	2.82
m_1	1.96	6.32	1.47	5.83	1.18	5.54
m_2	1.96	6.46	1.47	5.97	1.18	5.68
m_3	1.96	7.21	1.47	6,.2	1.18	6.43

Table 5: Approximated Vickrey discounts and prices

Applying these approximations on the above presented example determines the approximated discounts and the prices in table 5. In this case, the exchange does not have to endow the participants as it fulfills the weak budget balance property.

In Schnizler et al. a simulation was conducted, which compared these different mechanisms. As a result, the approximated Vickrey discounts were at least 13.5% smaller than the original discounts [SNVW05].

3.4 Step 3 – Testing

3.4.1 Implementation

Simulation tools are supporting the Market Engineering approach in testing and evaluating the designed mechanisms. Thus, the presented mechanism is implemented in a Java based simulation environment as shown in figure 3. 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 4 sketches briefly the main components of the market simulator.

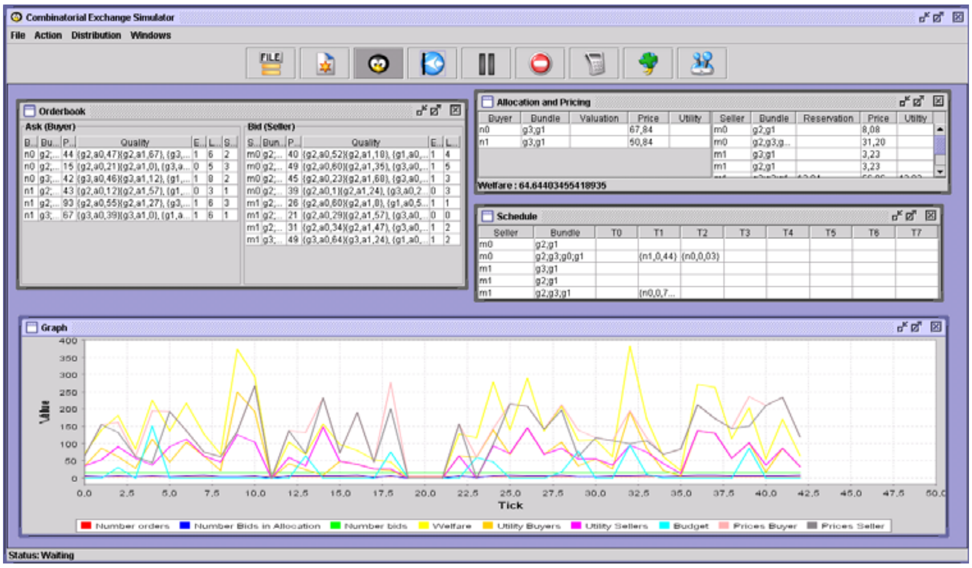

Figure 3: Combinatorial Exchange Simulator

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*[11]). 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).

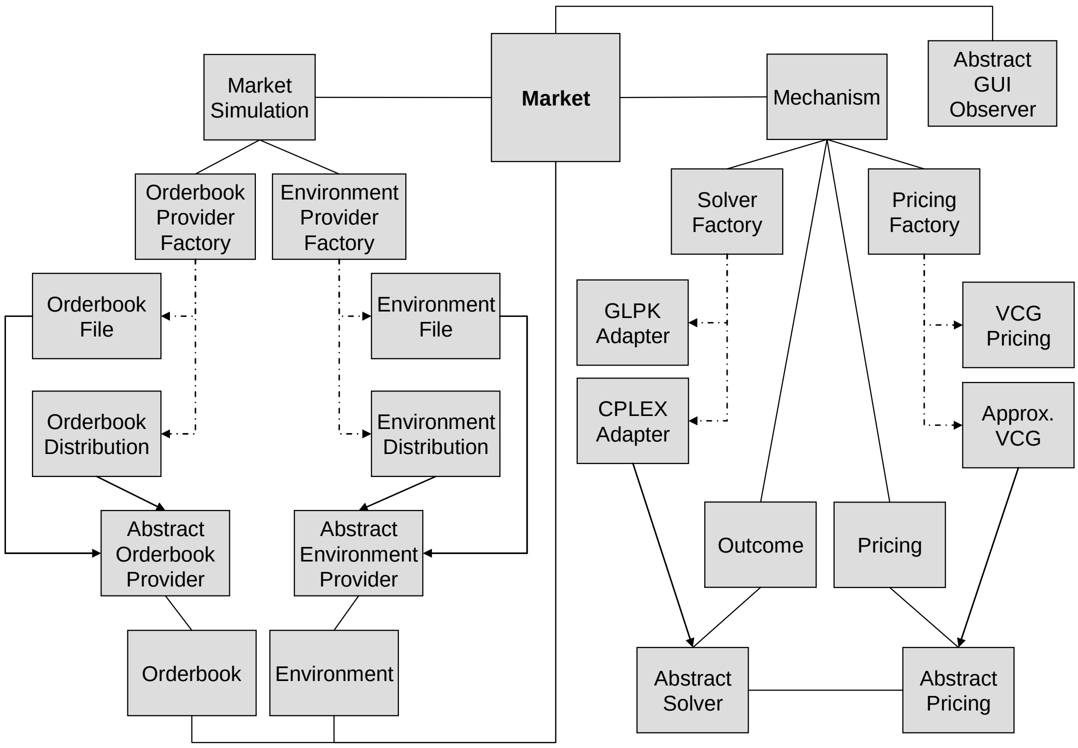

Figure 4: Architecture of the Simulation Tool

3.4.2 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 runtime simulation.

[11] CPLEX is a mathematical optimization engine for solving linear programs (http://www.ilog.com/).

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 [SSGL02].

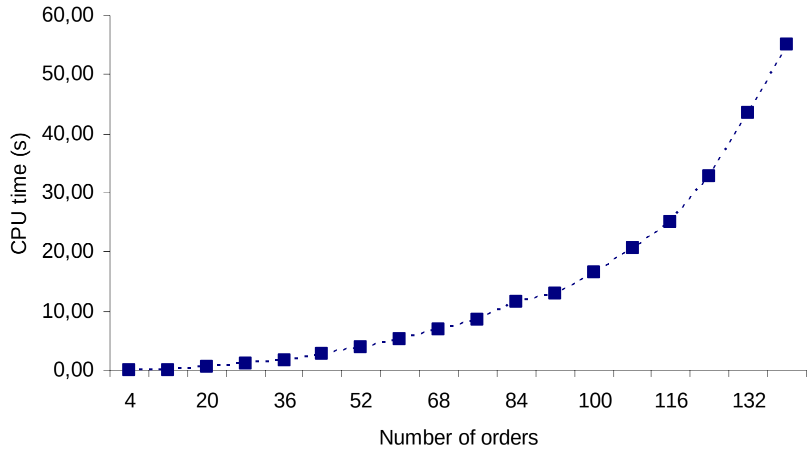

Figure 5: Performance simulation results

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 have to be examined. Further tests of the economic requirements are adressed to future work.

4 Outlook

This paper has outlined the design and the prototypical implementation of a market for trading computer resources in the Computational Grid.

Based upon the Market Engineering design process, the environment and the requirements a market mechanism has to fulfill were elicited. In consideration of these requirements, the winner determination problem was formulated as a mixed integer program. As a pricing mechanism, an approximated Vickrey approach was adapted. Subsequently, the market mechanism was implemented in a simulation prototype which enabled first evaluations of the mechanism by means of a run-time analysis.

The design and implementation of a multiattribute combinatorial exchange raises a number of research questions which are addressed in future work:

- **Pricing Mechanism:** The approximated Vickrey payments are weakly incentive compatible; however, the computational effort for calculating them is intractable. In order to minimize the complexity, a more efficient pricing mechanism has to be designed.
- **Trading Frequency:** Exchanges can be cleared either periodically or continuously. The idea is that a periodical clearing process leads to more efficient outcomes in combinatorial exchanges because a larger number of orders can be aggregated. However, the use of periodical clearing is inferior to continuous clearing in terms of immediacy. Determining metrics to measure immediacy, the overall utility of the participants trading continuously and periodically can be evaluated using simulations.
- **Approximation:** The computational complexity of the winner determination problem can be intractable. The adaptation of approximations for solving the problem could minimize the computational effort. The problem is, however, that suboptimal solutions lead to a loss of economic properties (e.g. allocative efficiency, manipulation strategies) which has to be identified, quantified, and evaluated.
- **Expressiveness of the Bidding Language:** Buyers should be able to specify their demanded resources on a conceptual layer (e.g. Pentium processor) and sellers should to be able to specify their supplied resources on an instantiated layer (e.g. Intel Pentium IV). For improving such an expressiveness of the bidding language, the use of ontologies may support this process.
- **Implementation:** The described mechanism is a component of the CATNETS project, which evaluates centralized and decentralized negotiation mechanisms. Within this project, the mechanism will be implemented in a Grid environment.

The problem of Computational Grid allocation illustrates the shift from obedient towards strategically acting peers, resulting in a gradually melting of the disciplines of computer science and economics.

Acknowledgements

Research reported in this paper has been financed by the European Union in the IST project CATNETS (http://www.catnets.org/).

References

[BuAV04]

Buyya, R.; Abramson, D.; Venugopal, S.: The Grid Economy. In Proceedings of the IEEE Grid Computing, 2004

[BiNa03]

Biswas, S; Narahari, Y.: Iterative Dutch Combinatorial Auctions. To appear in: Special issue of Annals of Mathematics and Artificial Intelligence on the Foundations of Electronic Commerce, 2003

[CzFK04]

Czajkowski, K.; Foster, I.; Kesselman, C.: Resource and Service Management. In: *Foster, I.; Kesselman, C.* (eds): The Grid 2 - Blueprint for a New Computing Infrastructure. Elsevier, 2004, Volume 2, p. 259–283.

[FoKe04]

Foster, I.; Kesselman, C.: The Grid 2 - Blueprint for a New Computing Infrastructure. Elsevier, 2004, Volume 2

[Hurw72]

Hurwicz, L.: On informationally decentralized systems. In: *McGuire, C; Radner, R* (eds.)*:* Decision and Organization, North-Holland Publishing Company, Amsterdam 1972

[Jack02]

Jackson, M.: Mechanism theory. In: Encylopedia of Life Support Systems EOLSS, Eolss Publishers, 2002

[MySa83]

Myerson, R.; Satterthwaite, M.: Efficient mechanisms for bilateral trading. In: Journal of Economic Theory, 28 (1983), p. 265-281

[Neum04]

Neumann, D.: Market Engineering - A Structured Design Process for Electronic Markets. Dissertation, Economics and Business Engineering, University of Karlsruhe (TH), 2004

[Park01]

Parkes, D.: Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency. Dissertation, Department of Computer and Information Science, University of Pennsylvania, 2001

[PaKE01]

Parkes, D.; Kalagnanam, J;Eso, M.: Achieving Budget-Balance with Vickrey-Based Payment Schemes in Exchanges. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 2001

[ScNW04]

Schnizler, B.; Neumann, D.; Weinhardt, C.: Resource Allocation in Computational Grids - A Market Engineering Approach. In: Proceedings of the WeB 2004, Washington, 2004

[SNVW05]

Schnizler, B.; Neumann, D.; Veit, D.; Weinhardt, C.: A Multiattribute Combinatorial Exchange for Trading Grid Resources. Submitted to: The Fourth International Joint Conference on Autonomous Agents & Multi Agent Systems, 2005

[SSGL02] *Sandholm, T.; Suri, S.; Gilpin, A; Levine, D.:* Winner determination in combinatorial auction generalizations. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems, 2002

[SuMT02]

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

[WeHN+03]

Weinhardt, C.; Holtmann, C.; Neumann, D.: Market Engineering, In: Wirtschaftsinformatik 45 (2003) 6, p. 635-640

[WWWM01]

Wellmann, M.; Walsh, E.; Wurmann, P.; MacKie-Mason, J.: Auction Protocols for Decentralized Scheduling. In: Games and Economic Behavior 35 (2001), p. 271-303

[WPBB03]

Wolski, R; Plank, J; Brevik J.; Bryan T.: Analyzing Market-Based Resource Allocation Strategies for the Computational Grid, In: International Journal of High Performance Computing Applications 15 (2003) 3, p. 258-281