Resource Allocation in Computational Grids - A Market Engineering Approach

Björn Schnizler, Dirk Neumann, Christof Weinhardt

Information Management and Systems, University of Karlsruhe

{schnizler, neumann, weinhardt}@iw.uni-karlsruhe.de

Abstract

The Computational Grid is a promising technology for providing access to distributed high-end computational capabilities. However, deciding which jobs are allocated to which resources is one the key problems in Computational Grids. In recent times, researchers have increasingly suggested to employ market mechanisms for scheduling and allocating Grid resources. This paper outlines the design of a market for allocating and scheduling resources in Computational Grids. The design of the market is based on the Market Engineering process, which provides methods and design principles for the development of market institutions. First, the characteristics and the requirements of the market participants as well as the Grid resources are elicited. Furthermore, a combinatorial clearing formulation is introduced as a primal mixed integer problem, which supports bids on bundles, quality, and time attributes. Followed by a performance simulation, the runtime of the clearing model is simulated.

Keywords: Combinatorial Exchanges, Grid Computing, Market Engineering

1 Introduction

In the mid 1990s the term Grid was introduced for describing a distributed computing infrastructure for advanced science and engineering [12]. In the meantime, considerable attention has been spent on constructing such Grid infrastructures (e.g., Condor [14], Globus [11], Gridbus [1]). Since that time, the term Grid itself has also undergone a change. Different from the traditional meaning, the expression has been broadened to embrace almost any kind of distributed networks. In this context Foster and Kesselman raise the question whether the term Grid has any real substance and meaning [13]. In order to avoid this philosophical question, this paper focuses on the so-called Computational Grid. According to Foster and Kesselman, the Computational Grid ”[...] is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities” [12]. Apparently, the Computational Grid enables the execution of complex and computationally demanding applications such as simulation, large-scaled optimization, Monte Carlo computing, image processing, and rendering.

Although in theory, Grid users could utilize resources from multiple locations (e.g. CPU cycles, memory capacities, network bandwidth, or storage media), this potential is, in practice, rarely exploited. Typically, Grid users avoid reaching far beyond their home institutions, whose resources are often inadequate to conduct these computationally demanding applications. This seeming contradiction is reasoned by the significant barriers that arise when organizational boundaries are crossed. These barriers mainly reflect different mechanisms and policies of different organizational units. Overcoming these barriers - impending the wide adoption of the Grid - requires that Grid users can discover, acquire, and manage reliably computational resources [14].

Typically, the resource management in Grids is operated as batch queuing system. Accordingly, access of the resources takes place through a scheduling of jobs, which reflect finite, known demands of resources. The scheduling employs a strategy determining which jobs are to be executed, at which time, and on which resources. Most of the current systems for Grid resource management (such as NetSolve [5], Legion [7], Condor AppLeS PST [6]) employ cost functions for the scheduling, where the cost function relies on system-centric parameters. In recent times, researchers have increasingly suggested to employ market mechanisms for scheduling and allocating Grid resources [4], [34].

From an economic point of view, Markets are - under certain circumstances - an effective institution to allocate resources efficiently, due to their extraordinary information processing capability inherent to the price system [18]. As such, the application of markets to the Computational Grid as an allocation and scheduling mechanism is deemed promising.

For Computational Grids, however, traditional market mechanisms, namely auctions, incur several drawbacks that prohibit their applicability.

The resources in Computational Grids are typically characterized by several attributes, where the configuration of the attributes can vary considerably. For instance, different computational resources as CPUs, hard disks, and memory are hosted on heterogenous machines such as workstations, clusters, or mobile devices. These resources can be connected to the Computational Grid in several ways (e.g. LAN, glass fibre, UMTS, or modems). However, most of the known market mechanisms can handle only single attribute resources.

Furthermore, the allocation problem is aggravated by the dynamic nature of demand and supply. The natural dynamism of demand is further increased by the fact that load balancing of resources (e.g. during scientific simulations) can vary considerably over time. Supply in Grids is also highly volatile, as any single resource is highly unreliable. Since any resource (e.g. a notebook) can join and leave the Grid in a rather unpredictable way, supply is subject to high dynamism as well. Comprising, at the moment there exists no perfect market mechanism, which is directly applicable in the Grid environment.

This paper attempts to tailor a market mechanism for this environment, which tries to meet the requirements in the Grid. As designing market mechanisms is rather difficult - success is per-se not warranted - the design follows the lines of the systematic Market Engineering approach [32], [23].

The remainder of the paper is structured as follows: In section 2, the market engineering process is briefly introduced. In section 3, the process is applied to the problem of resource allocation and scheduling in Computational Grids. As a result, a prototypical system is introduced that solves the clearing problem in Computational Grids. In section 4, the proposed approach is discussed and compared with related work. Section 5 closes with a summary and future work.

2 The Market Engineering Process

As aforementioned, designing an adequate market mechanism that meets the requirements is a complex task involving several activities. In principle, any design endeavor can be either intuitive or discursive.

- **Intuitive Approaches:** Intuitive approaches involve creativity in the form of complex associations of ideas. As such, intuitive approaches aim at increasing the flow of ideas such as brainstorming, the Delphi-method, or others. Despite the fact that intuitive approaches have been leading to many excellent solutions, a purely intuitive approach incurs major disadvantages. The crux of intuitive approaches is that good ideas are not discovered or undiscovered; they just come. Therefore, the right idea might not come at the right moment. Furthermore, the results of intuitive approaches are strongly dependent on the designer's expertise, skills, and experiences. It also cannot be excluded that the intuitive ideas are not already circumscribed by the education and experience of the designer [23], [26]. Apparently, for complex design problems, intuitive approaches may fail to achieve satisfactory solutions.

- **Discursive Approaches:** It is likely that not a single design approach solves the entire but parts of the design problem. Accordingly, a design strategy is necessary which decomposes the complex overall design problem into several smaller, less complex problems. Problems are not tackled in their totality, but transformed into smaller problems. For those smaller problems, stronger design methods - including intuitive methods - may exist that solve them [1]. As such, the design strategy applies a deliberate, 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 [16].

In the context of designing market mechanisms, the approach of Market Engineering aims at the discursive, goal-oriented development of market institutions [2]. At heart of Market Engineering stands the process model that gives structure to the process.

Figure 1 sketches the higher-level phases of the Market Engineering process [32]. The objectives and the strategy that governs the Market Engineering approach stand at the outset of the Market Engineering process.

In the first stage - the environmental analysis - the requirements of the new market mechanism (i.e. Who are the potential participants, what are their preferences, endowments and constraints?) are deduced. The design and implementation stage is more or less a container for several design phases.

In analogy to the engineering design process from mechanical engineering [26], the design stage is decomposed into four major phases being the conceptual design, embodiment design, detail design, and implementation. In the conceptual design, the market mechanism is deduced

[1] The better structured the problems are, the more powerful are the applicable design methods [25].

[2] The market institution is thereby defined in a broad manner to capture not only the trading rules (i.e. the market microstructure), but also the technical infrastructure as well as potential business models [32], [23]

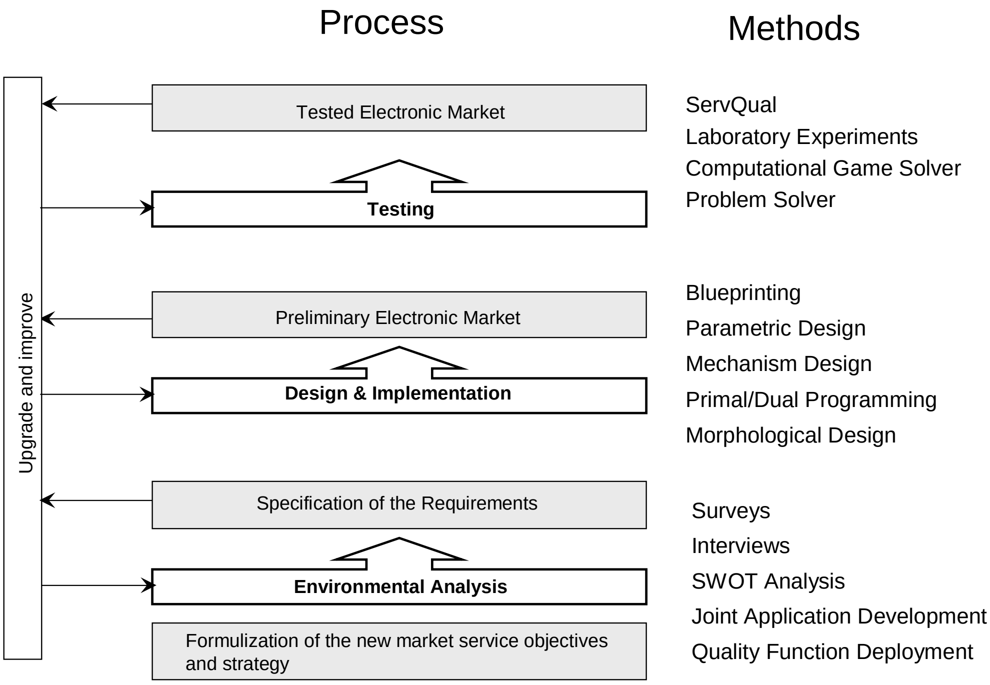

Figure 1: Design Process and Design Methods

as an allocation and payment function. These purely conceptual functions are refined in the embodiment design phase into an auction scheme which is subsequently transformed into a formal process model (e.g. using UML). In the detail design phase, all remaining design issues are tackled and subsequently implemented.

Having implemented the appropriate market mechanism, it is tested upon its economic properties and its operational functionality.

At any stage of the Market Engineering process, there is a decision whether to proceed with the next step or better to repeat the prior one. The use of prototypes is again possible at any stage of the process, such that they are left out in the picture. The Market Engineering process not only structures the design process but also provides the designer with a whole array of methods that may support separate sub-tasks.

In Figure 1, several design methods are associated with any phase of the process. Those methods are sometimes substituting or complementing each other. For instance, while the methods of primal/dual programming [20], mechanism design [19], and parametric design [23] are all addressing the same problem - namely the conceptual design of the market mechanism

- the method of blueprinting [24] aims at the transformation of the conceptual design model into a software model that is free of implementation details. For a detailed description of how these methods can be sequenced effectively see [23] and [17].

3 An Application for Grid Markets

For the design of a market mechanism for Computational Grids, the Market Engineering process is supported by the following methods: The environmental analysis is guided by a comprehensive literature survey to elicit a set of ad-hoc requirements. Afterwards, the market mechanism is modelled using primal/dual programming. The intuition for using primal/dual programming instead of mechanism design or other methods is that the market is supposed to be large enough such that the single market participants cannot influence the market price. As such, game-theoretic modelling of the market mechanism becomes obsolete [20]. The clearing algorithm is implemented in a prototype using the problem solver CPLEX [3] for clearing the market.

3.1 Stage 1 - Environmental Analysis

When designing a market mechanism, the designer has to be aware of the underlying environment for which the market mechanism is to be designed. Thereby, the environment description embraces information about the potential market participants, their needs, the characteristics of the traded resources, and their endowments. Once the environment description is available, the designer can elicit the requirements for the market mechanism from the potential participants of that particular environment [23].

Corresponding with the engineering design process, the design stage comprises two different phases: the environment definition and the requirement analysis.

[3] CPLEX is a software tool for solving linear and integer programming problems (`http://www.cplex.com/`).

Environment Definition

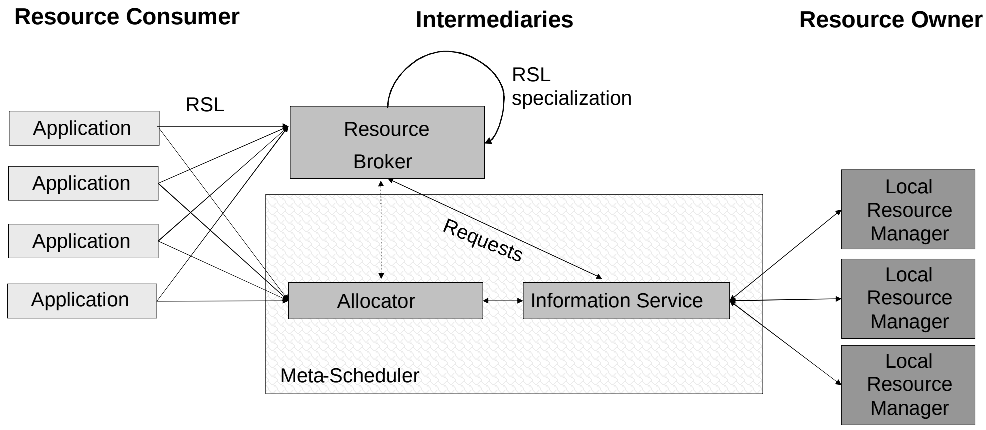

Figure 2: The market for Computational Grids

Surveying Grid literature and following [10], the simplified picture in Figure 2 can be adapted as a market place for Computational Grid resources. In essence, the market for Computational Grid is spanned around the resource owners as sellers (e.g. IBM or Sun with their computer centers), the resource consumers as buyers (e.g. scientists at universities, rendering or the biochemical firms) and some intermediaries (e.g. Condor, Gallop, Legion etc.). The intermediaries technically provide the resource management infrastructure for exploiting remote resources. According to the resource management architecture proposed by [10] the intermediary layer consists of three basic components:

Resource Broker: The resource broker components are responsible for resource discovery, selection, aggregation, and subsequently for the data and program transportation [8]. By transforming the resources to the consumers' requirements (which are specified for instance in the Resource Specification Language (RSL)) into a set of jobs that are self-reliantly scheduled on the appropriate resources (i.e. RSL specialization) and subsequently managed, the complexity of the Grid is concealed [31]. For the market participants, the resource broker is apparently more of a black box.

Resource Information Manager: The resource information manager provides pervasive access to information about the current availability and capability of resources [10].

Allocator: The allocator coordinates the allocation of resources at multiple sites. Obviously, the allocator and the information service assume the responsibility of (meta-) scheduling the jobs.

Based upon this view on the intermediary layer, the market mechanism for Computational Grid can be sketched as follows: The transition from an intermediary layer to a mediated market mechanism is not too far. In essence, the scheduling performed by the resource broker can be shifted to the market mechanism. Instead of sending requests to the information service, the resource broker can translate the user requirements into bids. Those bids expressing demand and supply situation are subsequently cleared by the market mechanisms.

Requirement Analysis

Setting up a market mechanism requires the designer to define three components:

- a bidding language, which specifies how bids can be formulated,
- a clearing scheme, which determines who gets which resource, and
- a payment scheme, which defines the payments the individual users have to make depending on the allocation.

Ideally, the resulting market mechanism reflects the users' requirements. Based upon the literature survey the following seven ad-hoc requirements can be elicited for the Grid market.

Requirement 1: Trading Exchange: The market for Computational Grid is characterized by many resource owners (henceforth sellers) and many resource consumers (buyers). Providing an exchange installs competition on both sides and is thus deemed promising to yield an adequate allocation [3].

Requirement 2: Bundle bids: Typically, buyers demand a combination of resources as a bundle to perform a task [30]. Apparently, resources in the Grid are complementarities. Complementarities are goods with superadditive 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.

Suppose, for example, a buyer intending to render images requires hard disk, CPU, and memory. If any component, 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 market mechanism must allow for bids on bundles. Likewise, the seller can also express bids on bundles.

Requirement 3: XOR-bids on the buy side: 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 subadditive 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 market mechanism must support XOR [4] bids to express substitutes.

Requirement 4: No XOR-bids on the sell side: For simplicity we restrict an order of a seller to a set of OR [5] bids. This simplification can assumed by the fact, that the resources in the Computational Grid are usually no substitutes for the sellers.

Requirement 5: Bids on quality attributes: Resources in Grids are typically not completely standardized. Similar resources can differ in their quality. For instance, a hard disk can

[4] A XOR B ($A \oplus B$) means either A or B but not both

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

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 that requires a minimum amount of GBs, 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.

Requirement 6: 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 market mechanism must allow for placing bids on time ranges.

Requirement 7: Co-Allocation: Scientific applications (e.g. genetic disease research) require more resources than any single seller could provide. As such, an allocation from multiple sites (or in short co-allocation) is indispensable.

3.2 Stage 2 - Design and Implementation

In the following, a programming model for allocating and scheduling resources in the Computational Grid, that meets the above mentioned requirements is presented. That is, a combinatorial clearing formulation for bundles of resources which includes quality and time attributes. First, the notation for the model is introduced. Afterwards, a specific clearing mechanism is presented. The section concludes with an example of the model and a performance simulation.

Notation

Let $N = \{n_1, \ldots, n_{|N|}\}$ be a set of buyers and $M = \{m_1, \ldots, m_{|M|}\}$ a set of sellers, where $n \in N$ is an arbitrary buyer and $m \in M$ a seller. Furthermore, there is a set of discrete resources $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$ and a set of bundles $\mathcal{B} = \{S_1, \ldots, S_{|\mathcal{B}|}\}$ with $S \in \mathcal{B}$ and $S \subseteq \mathcal{G}$ as a set of resources. Each resource g has a set of x cardinal attributes $A_g = \{a_{g,1}, \ldots, a_{g,x}\}$ with $a_{g,j} \in A_g$ which represent the qualities of the resource. Bundles can be assigned to discrete times-slots [6] which are represented by a set $T = \{1, \ldots, t_{|T|}\}$, where $t \in T$ defines a single time slot.

For any bundle S, any resource $g \in S$, and any attribute $a_{g,j} \in A_g$, each buyer n can specify the minimal required quality [7] $q_n^b(S, g, a_{g,j}) \geq 0$. For example, in the context of Grid resources, a quality attribute can determine the minimum number of FLOPS [8] of a requested CPU. Furthermore, for each bundle S, a buyer can determine the number of required time slots with the parameter $s_n^b(S) \geq 0$, as well as a time range of possible allocations. For each bundle S, the parameter $e_n^b(S) \geq 0$ denotes the earliest time slot for allocating the bundle S whereas $l_n^b(S) \geq 0$ indicates the latest possible time for any allocatable slot for S. The valuation of a buyer n per slot for any bundle S is represented by $v_n(S)$ which is assumed to be $v_n(S) \geq 0$ (no negative valuation).

A buyer n can submit a set of XOR bids $(B_{n,1}^b(S_1) \oplus \ldots \oplus B_{n,r}^b(S_j))$ on bundles as an order, where r is the number of bundle bids. A single bid $B_{n,i}^b(S)$ is defined as the tuple

$$B_{n,i}^b(S) = (v_n(S), (q_n^b(S, g_1, a_{1,1}), \ldots, q_n^b(S, g_x, a_{x,z})), s_n^b(S), e_n^b(S), l_n^b(S))$$

where x is the number of resources in bundle S and $a_{x,z}$ is the last attribute of the resource x.

[6] In the model, possible dependencies between single resources and time slots are ignored. For example, a CPU providing $6 * 10^6$ FLOPS in two slots could be replaced by a CPU capable of serving $12 * 10^6$ FLOPS in a single slot. However, these dependencies cannot be generalized to all resources (e.g. hard disk capacity) and will therefore be neglected.

[7] In the following, the upper index b indicates that the corresponding variable α^b is associated with a buyer and α^s indicates the association of α^s with a seller

[8] FLOPS is an acronym for Floating Point Operations per Second and is used as a measure of a computer's processing speed.

For example, the bid $B^b_{n,1}(\{CPU, HDD\})$ including two goods each with one single attribute $q^b_n(\{CPU, HDD\}, CPU, SPEED)$ and $q^b_n(\{CPU, HDD\}, HDD, SPACE)$ with

$$B^b_{n,1}(\{CPU, HDD\}) = \{1, (3 * 10^6, 30), 4, 2, 10\}$$

expresses that a buyer n wants to buy a bundle $S = \{CPU, HDD\}$ with at least $3 * 10^6$ FLOPS of CPU processing speed and 30 GB hard disk space. The buyer requires 4 slots for 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 $v_n(\{CPU, HDD\}) = 1$.

The sellers' bids are formalized in a similar way as the buyers' bids are. For each bundle S, a time range can be specified denoting the earliest time slot $e^s_m(S) \geq 0$ and the latest possible time slot $l^s_m(S) \geq 0$ for which the seller m can provide its resources. The maximum available quality for each resource $g \in S$ and each concatenated attribute $a_{g,j} \in A_g$ can be determined by $q^s_m(S, g, a_{g,j}) \geq 0$. For a single time slot t and a bundle S, the reservation price can be represented by $r_m(S) \geq 0$. Therefore, a bid $B^s_{m,i}$ of a seller m can be formalized by the tuple

$$B^s_{m,i}(S) = (S, r_m(S), (q^s_m(S, g_1, a_{1,1}), \ldots, q^s_m(S, g_x, a_{x,z})), e^s_m(S), l^s_m(S))$$

with x indicating the number of resources in the bundle and $a_{x,z}$ representing the last attribute of the resource x. As above mentioned, an order of a seller is restricted to a set of OR bids:

$$(B^s_1(S_1) \vee \ldots \vee B^s_x(S_x))$$

Clearing Model

Exchanges can be cleared either periodically (e.g. at five minutes intervals) or continuously (any feasible trade will be cleared immediately). Parkes et al. [27] describe that a periodical clearing may lead 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. For simplicity, this tension between efficiency and immediacy is not further elaborated.

Our clearing model follows a common assumption of the auction theory that buyers and sellers are risk neutral and have linear utility functions [21]. Therefore, it is assumed that the sellers' reservation prices can be linearly transformed to any partial executions of any bundle. For example, suppose a seller m offering a single resource g having one attribute $a_{g,1}$ as a bundle $S = \{g_1\}$ with the quality $q^s_m(S, g, a_{g,1}) = 10$. For this bundle S, the seller has a reservation price of $r_m(S) = 2$. A partial execution of S with a quality of 5 would lead to a proportional transformation of the reservation price to $r_m(S)\frac{5}{q^s_m(S,g,a_{g,j})} = 1$.

It is furthermore assumed that all jobs can be interrupted at any time. Following [15], the network topology and the communication costs are neglected. Instead, it is assumed that each user can submit jobs to any resource. For example, this assumption allows the co-allocation that CPUs and hard disk capacities can be allocated to a single buyer from different located sellers.

Let $x_n(S)$ denote a binary variable with $x_n(S) = 1$ if the bundle $S \in \mathcal{B}$ is allocated to a buyer n. Otherwise the value of the variable is set to $x_n(S) = 0$. Furthermore, each buyer n has a binary variable $z_{n,t}(S)$ which is associated in the same way as $x_n(S)$ with the allocation of S in time slot t.

For each seller m, the variable $y_{m,n,t}(S)$ indicates the percentage contingent of the bundle S allocated to buyer n in time slot t. For example the variable $y_{m,n,t}(S) = 0.5$ denotes that 50 percent of the qualities of the bundle S is allocated from seller m to buyer n in time slot t.

By means of these variables, the clearing model can be formulated as the following mixed integer program:

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

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

$$\sum_{t\in T} z_{n,t}(S) \leq x_n(S) s^b_n(S) \ \forall n \in N, \forall S \in \mathcal{B} \tag{3}$$

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

$$\sum_{S \ni g} x_n(S) s_n^b(S) q_n^b(S,g,a) \leq \sum_{m \in M} \sum_{t \in T} \sum_{S \in \mathcal{B}} y_{m,n,t}(S) q_m^s(S,g,a) \; \forall n \in N, \forall g \in \mathcal{G}, \forall a_g \in A_g \tag{5}$$

$$\sum_{S \ni g} z_{n,t}(S) q_n^b(S,g,a) \leq \sum_{S \ni g} \sum_{m \in M} y_{m,n,t}(S) q_m^s(S,g,a) \; \forall n \in N, \forall a_g \in A_g, \forall g \in \mathcal{G}, \forall t \in T \tag{6}$$

$$(t - e_m^s(S)) y_{m,n,t}(S) \geq 0 \; \forall n \in N, \forall m \in M, \forall S \in \mathcal{B}, \forall t \in T \tag{7}$$

$$(l_m^s(S) - t) y_{m,n,t}(S) \geq 0 \; \forall n \in N, \forall m \in M, \forall S \in \mathcal{B}, \forall t \in T \tag{8}$$

$$(t - e_n^b(S)) z_{n,t}(S) \geq 0 \; \forall S \in \mathcal{B}, n \in N, t \in T \tag{9}$$

$$(l_n^b(S) - t) z_{n,t}(S) \geq 0 \; \forall n \in N, \forall S \in \mathcal{B}, \forall t \in T \tag{10}$$

$$x_n(S) \in \{0,1\} \; \forall S \in \mathcal{B}, \forall n \in N, \forall t \in T \tag{11}$$

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

$$y_{m,n,t}(S) \geq 0 \; \forall m \in M, \forall S \in \mathcal{B} \tag{13}$$

The objective function (equation (1)) maximizes the surplus which is defined as the difference between the sum of the buyer's valuations $v_n(S)$ and the sum of the sellers' reservation prices $r_m(S)$. The objective function reflects the goal of maximizing the social welfare w in the economy.

The first constraint (equation (2)) guarantees that each buyer n can be allocated only one bundle S. This equation is necessary to fulfill the XOR constraint of a buyer order. Equation (3) ensures that for any allocated bundle S, the buyer receives all required slots $s_n^b(S)$ within the time set T. For each time slot t, equation (4) guarantees that each seller cannot allocate more than the seller possesses. For each resource equation (5) ensures that the sum of the supplied quality for all attributes and 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 (equation (6)). Finally the equations (7) - (10) indicate that slots cannot be allocated before the earliest and after the latest time slot of neither any seller (equation (7), (8)), nor any buyer (equation (9), (10)).

Example

The notation and clearing model can be illustrated by the following example: There are two resources $\mathcal{G} = \{g_1, g_2\}$ each with single quality attributes ($a_{g_1,1}$, $a_{g_2,1}$). In addition, there are three bundles $\mathcal{B} = \{S_1, S_2, S_3\}$ with $S_1 = \{g_1\}$, $S_2 = \{g_2\}$, and $S_3 = \{g_1, g_2\}$. The bundles can be allocated within 8 available slots in a time range of $T = \{t_0, \ldots, t_7\}$. Each buyer submits a single XOR order which is shown in table 1. Each seller order consists of two OR bids which are given in table 2.

Buyer	Bundle	Valuation	Quality	Earliest	Latest	Slots
n_1	$S_1 = \{g_1\}$	3	$(g_1, 500)$	2	7	4
	$S_2 = \{g_2\}$	2	$(g_2, 10)$	3	5	1
	$S_3 = \{g_1, g_2\}$	5	$(g_1, 500), (g_2, 15)$	1	4	4
n_2	$S_1 = \{g_1\}$	1	$(g_1, 300)$	1	7	2
	$S_3 = \{g_1, g_2\}$	4	$(g_1, 300), (g_2, 15)$	2	6	3

Table 1: XOR orders of buyer n_1 and n_2

Seller	Bundle	Reservation	Quality	Earliest	Latest
m_1	$S_1 = \{g_1\}$	3	$(g_1, 700)$	0	7
	$S_2 = \{g_2\}$	2	$(g_2, 10)$	1	4
m_2	$S_1 = \{g_1\}$	2	$(g_1, 260)$	2	6
	$S_2 = \{g_2\}$	2	$(g_2, 25)$	2	6

Table 2: OR orders of seller m_1 and m_2

The optimal allocation is $x_1(S_1) = 1$ and $x_2(S_3) = 1$ with a maximized social welfare of $w = 7.63$.

The corresponding schedule for this allocation is given in table 3.

M	S	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
m_1	g_1			$n_1, 500$	$n_1, 500$	$n_2, 300$	$n_2, 300$	$n_1, 400$ $n_2, 300$	$n_1, 500$
m_2	g_1							$n_1, 100$	
	g_2					$n_2, 15$	$n_2, 15$	$n_2, 15$	

Table 3: Allocation schedule

For example, the bundle S_1 is allocated from seller m_1 to buyer n_1 in the time slots t_2, t_3, t_6, and t_7. These single allocations satisfy the time range constraints of the bid $B^b_{n_1}(S_1)$ from buyer n_1 [9] .

The co-allocation of the mechanism is demonstrated in time slot t_6. Both bundles, S_1 assigned to buyer n_1 and S_3 assigned to buyer n_2, have to be allocated simultaneously to complete in time $(l^b_1(S_1) = 7, l^b_2(S_3) = 6)$. However, seller m_1 cannot fulfill both buyers with their requested qualities. Due to the fact that the valuation of buyer n_2 for the bundle S_3 is higher than the valuation of n_1 for S_1, buyer n_2 is fully served by the cheaper seller m_1. The buyer n_1 is co-allocated with seller m_2, who has a higher reservation price as m_1.

Consider the bid of buyer n_1 for the bundle S_3 as an example for the time-constraints. Although the buyer has a higher valuation for this bundle, both buyer n_2 and his own valuation for the bundle S_1, there is no allocation because the resource g_2 cannot be assigned in time.

3.3 Stage 3 - Testing

The clearing mechanism is implemented as a Java based prototype. As optimization engine for solving the mixed integer program, CPLEX 9 in combination with GAMS 21.3 [10] is used. As a consequence, exact and optimal solutions are obtained.

[9] Recall that interruptible jobs are assumed.

[10] GAMS is a modelling system for mathematical programming problems.

The clearing model is - in analogy to the general combinatorial allocation problem - $\mathcal{NP}$-complete, as it is an instance of the set packing problem. With an increasing number of traded resources, this problem becomes intractable from a computational point of view. However, in the Computational Grid environment, the number of different resources is limited. The complexity of the problem could therefore be tractable. Hence, the computational performance of the model is simulated in a first step [11].

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

The bundles of each buyer 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 $\beta = 0.75$. This proceeding is iterated until a resource is not added or the bundle includes all resources [29]. Sandholm et al. show, that the Decay distribution can lead to hard instances of general combinatorial allocation problems [28].

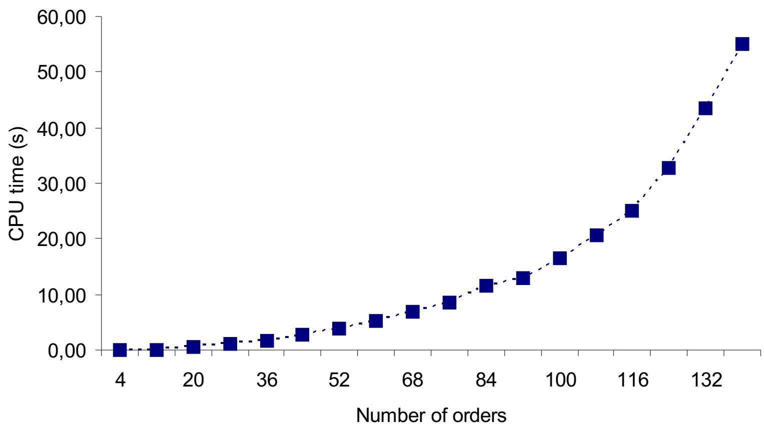

Figure 3: Performance simulation results

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 (including 303 bids) takes over 50 seconds.

[11]The simulation was performed on a Pentium IV 2.3 GHZ with 2GB of main memory.

The performance simulation shows that the clearing problem is computationally very demanding. As previously noted the implementation of the mixed integer problem is very complex. As such in complex scenarios, it has to be examined to use different order structures or heuristics. Alternatively, the clearing schedule can take place as a periodic event.

4 Related Work

A number of resource allocation mechanisms has been developed that attempting to solve the allocation problem in Grid systems or related architectures. Most of these mechanisms are, however, central in nature in a way that the allocation problem is solved by a central entity. 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.

As a consequence, Buyya [4], [3], Wolski et al. [34], and Subramoniam et al. [30] motivate the use market-based mechanisms such as auctions or electronic negotiations for the allocation problem.

In a first approach, Wolski et al. [35] propose the use of traditional auction formats such as English auctions. The use of traditional auction formats in the Grid environment, however, is delimited, as the resources are traded as unbundled standardized commodities. Consequently, in traditional auction formats, it is impossible to express demand for bundles without running the risk of receiving only one leg of the bundle without the other. In order to avoid such an exposure risk of the buyers, Subramoniam et al. [30] extend the proposed auctions to bundles. Nonetheless, Subramoniam et al. consider the resources to be standardized commodities. Standardization of the resources would either imply that the number of resources is extremely limited compared to the number of all possible resources or that there are extremely many market mechanisms that are likely to have only few participants. Both implications result in rather inefficient allocations. The work presented in this paper introduces time attributes for bundles as well as quality constraints for single resources to allow trading in a single venue. This introduction redefines the Grid allocation problem to a scheduling problem.

In this context, Wellman et al. model single sided auction protocols for the allocation and scheduling of resources under consideration of different time constraints [33]. Conen designs a combinatorial bidding procedure for job scheduling including different running, starting, and ending times of jobs on a processing machine [9]. 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 the literature, Parkes et al. introduce the first combinatorial exchange as a single-shot sealed bid auction [27]. As payment scheme, Vickrey discounts are approximated. Biswas and Narahari [2] 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. However, both approaches neither support quality nor time constraints and are thus not completely capable of solving the allocation problem. As such neither of these approaches are specifically targeted to the Grid environment.

5 Concluding Remarks

This paper has outlined the design of a market for the Computational Grid environment that incorporates bidding on bundles. Different to other algorithms, the proposed approach also accounts for time and quality constraints, which are very important for the Grid environment. Following the structured design process of Market Engineering, the market was firstly defined in terms of the market participants (buyers and sellers) and their respective characteristics. Subsequently, the requirements of the market participants upon the market mechanism were elicited. Based on the technique of primal/dual programming, the clearing problem was formulated as a primal problem. Our market mechanism was shown to allocate the resources efficiently given the true valuations on the expense of very high computational costs.

For the future, the pricing scheme has to be added as the dual formalization of the clearing problem. Assuming myopic best response strategies will achieve that the buyers and the sellers reveal their true valuations and reserve prices, respectively.

Alternatively, it is also possible to follow the Parkes et al. [27] approach by approximating the Vickrey discounts. Once, the pricing scheme is determined, the concrete process rules of the auction can be designed (e.g., whether the clearing time is periodical or continuous). Although the mechanism may have desirable properties, it is still uncertain, whether the implemented market system can in deed attain these properties in real world settings. As such, the use of laboratory experiments is in addition to simulations necessary to demonstrate that the proposed properties still hold. Other directions of future work aim at the reduction of the bid space introducing additional structure on the bid space for example by means of ontologies.

Acknowledgements

Research reported in this paper has been financed by the EU in the IST project CATNETS, the Alexander-von-Humboldt foundation, the Federal Ministry of Education and Research, and the Social Sciences and Humanities Research Council of Canada.

References

[1] BAKER, M., BUYYA, R., AND LAFORENZA, D. Grids and Grid technologies for wide-area distributed computing. *Software Practise and Experience 32*, 15 (2002), 1437–1466.

[2] BISWAS, S., AND 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).

[3] BUYYA, R., ABRAMSON, D., GIDDY, J., AND STOCKINGER, H. Economic models for resource management and scheduling in grid computing. *The Journal of Concurrency and Computation: Practice and Experience 14*, 13-15 (2002), 1507–1542.

[4] BUYYA, R., STOCKINGER, H., GHIDDY, J., AND ABRAMSON, D. Economic models for management of resources in peer-to-peer and grid computing. Technical report, Department of Computer Science and Software Engineering, The University of Melbourne, Australia, 2001.

[5] CASANOVA, H., AND DONGARRA, J. NetSolve: A network-enabled server for solving computational science problems. *The International Journal of Supercomputer Applications and High Performance Computing 11*, 3 (1997), 212–223.

[6] CASANOVA, H., OBERTELLI, G., BERMAN, F., AND WOLSKI, R. The apples parameter sweep template: user-level middleware for the grid. In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing* (2000), IEEE Computer Society.

[7] CHAPIN, S. J., KATRAMATOS, D., KARPOVICH, J., AND GRIMSHAW, A. S. The Legion resource management system. In *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph, Eds. Springer Verlag, 1999, pp. 162–178.

[8] CHETTY, M., AND BUYYA, R. Weaving computational grids: How analogous are they with electrical grids? *Computing - Science and Engineering 4*, 4 (2002), 61–71.

[9] CONEN, W. Economically coordinated job shop scheduling and decision point bidding - an example for economic coordination in manufacturing and logistics. In *Proceedings of the 16th Workshop on Planen, Scheduling und Konfigurieren, Entwerfen* (2002).

[10] CZAJKOWSKI, K., FOSTER, I., AND KESSELMAN, C. *Resource and Service Management*, vol. 2. Elsevier, 2004, ch. 18, pp. 259–283. In: The Grid 2 - Blueprint for a New Computing Infrastructure.

[11] FOSTER, I., AND KESSELMAN, C. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing 11*, 2 (1997), 115–128.

[12] FOSTER, I., AND KESSELMAN, C. *The Grid 2 - Blueprint for a New Computing Infrastructure*, vol. 2. Elsevier, 2004.

[13] FOSTER, I., KESSELMAN, C., AND TUECKE, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal Supercomputer Applications 15*, 3 (2001).

[14] FREY, J., TANNENBAUM, T., FOSTER, I., LIVNY, M., AND TUECKE, S. Condor-G: A Computation Management Agent for Multiinstitutional Grids. In *International Symposium on High Performance Distributed Computing* (San Francisco, CA, 2001), pp. 55–67.

[15] GALSTYAN, A., CZAJKOWSKI, K., AND LERMAN, K. Resource allocation in the grid using reinforcement learning. In *Proc. of the International Conference on Autonomous Agents and Multi-Agent Systems AAMAS-03* (2003).

[16] GRANT, D. Design methodology and design methods. *Design Methods and Theories 13*, 1 (1979), 46–47.

[17] HOLTMANN, C. *Organisation von Maerkten - Market Engineering fuer den Wertpapierhandel*. PhD thesis, Economics and Business Engineering. Karlsruhe, Germany, University of Karlsruhe (TH), 2004.

[18] HURWICZ, L. On informationally decentralized systems. In *Decision and Organization*, C. McGuire and R. Radner, Eds. North-Holland Publishing Company, Amsterdam, NL, 1972.

[19] JACKSON, M. O. Mechanism theory. In *Encyclopedia of Life Support Systems (EOLSS)* (Oxford ,UK, 2002), Eolss Publishers.

[20] KALAGNANAM, J., AND PARKES, D. C. *Auctions, Bidding and Exchange Design*. Kluwer Academic Publishing, 2003.

[21] KRISHNA, V. *Auction theory*. Academic Press, San Diego, 2002.

[22] NEUMANN, D. *Market Engineering - A Structured Design Process for Electronic Markets*. PhD thesis, Economics and Business Engineering. Karlsruhe, Germany, University of Karlsruhe (TH), 2004.

[23] NEUMANN, D., AND HOLTMANN, C. Embodiment design in market engineering. In *Proceedings of the Research Symposium on Emerging Electronic Markets (forthcoming)* (Dublin, Irland, 2004).

[24] NEWELL, A., AND SIMON, H. A. Computer science as empirical enquiry: Symbols and search. *Communications of the ACM 19*, 3 (1976), 113–126.

[25] PAHL, G., AND BEITZ, W. *Engineering Design*. The Pitman Press, Bath, UK, 1984.

[26] PARKES, D. C., KALAGNANAM, J., AND ESO, M. Achieving budget-balance with vickrey-based payment schemes in exchanges. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (2001), pp. 1161–1168.

[27] SANDHOLM, T. An algorithm for optimal winner determination in combinatorial auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (1999), Morgan Kaufmann Publishers Inc., pp. 542–547.

[28] SANDHOLM, T., SURI, S., GILPIN, A., AND LEVINE, D. Winner determination in combinatorial auction generalizations. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems* (2002), ACM Press, pp. 69–76.

[29] SUBRAMONIAM, K., MAHESWARAN, M., AND TOULOUSE, M. Towards a microeconomic model for resource allocation in grid computing systems. In *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering* (2002).

[30] VENUGOPAL, S., BUYYA, R., AND WINTON, L. A grid service broker for scheduling distributed data-oriented applications on global grids. Technical report, grids-tr-2004-1, University of Melbourne, Australia, 2004.

[31] WEINHARDT, C., HOLTMANN, C., AND NEUMANN, D. Market engineering. *Wirtschaftsinformatik 45*, 6 (2003), 635–640.

[32] WELLMAN, M. P., WALSH, W. E., WURMAN, P. R., AND MACKIE-MASON, J. K. Auction protocols for decentralized scheduling. *Games and Economic Behavior 35* (2001), 271–303.

[33] WOLSKI, R., PLANK, J., BREVIK, J., AND BRYAN, T. Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications 15*, 3 (2001), 258–281.

[34] WOLSKI, R., PLANK, J., BREVIK, J., AND BRYAN, T. G-commerce: Market formulations controlling resource allocation on the computational grid. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)* (2001).