Bridging the Grid Adoption Gap – Developing a Roadmap for Trading Grids

Dirk Neumann  
Information Management and Systems, University of Karlsruhe, neumann@iism.uni-karlsruhe.de

Jochen Stoesser  
Information Management and Systems, University of Karlsruhe, stoesser@iism.uni-karlsruhe.de

Christof Weinhardt  
Information Management and Systems, University of Karlsruhe, weinhardt@iism.uni-karlsruhe.de

Follow this and additional works at: http://aisel.aisnet.org/bled2007

Recommended Citation

http://aisel.aisnet.org/bled2007/22
Bridging the Grid Adoption Gap – Developing a Roadmap for Trading Grids

Dirk Neumann, Jochen Stoesser, Christof Weinhardt
Information Management and Systems, University of Karlsruhe (TH), Germany
{neumann, stoesser, weinhardt}@iism.uni-karlsruhe.de

Abstract
This paper argues that the technology of Grid computing has not yet been adopted by enterprises due to the lack of viable business models. While in academia, Grid technology has already been taken up, the sharing approach among non for-profit organizations is not suitable for enterprises. In this paper, the idea of a Grid market is taken up to overcome this Grid adoption gap. Although this idea is not new, all previous proposals have been made either by computer scientists being unaware of economic market mechanisms or by economists being unaware of the technical requirements and possibilities. This paper is unique as it derives an economically sound set of market mechanisms based on a solid understanding of the technical possibilities.

Keywords: Electronic markets, grid computing, business models

1 Introduction

Despite much criticism in the past, Grid computing is increasingly gaining traction in a number of application areas. Grid computing denotes a computing model that distributes processing across an administratively and locally dispersed infrastructure. By connecting many heterogeneous computing resources, virtual computer architectures are created, increasing the utilization of otherwise idle resources (Foster et al. 2001).

By means of Grid computing, it is possible to set up virtual supercomputers via the connection of normal servers that cost no more than $3000 each. Adding and removing servers is simple granting extreme flexibility in building up infrastructures. The most salient example of what Grid computing can achieve is illustrated by its prominent predecessor SETI@home. SETI@home connected over one million computers spread across 226 countries to reach processing power of 418.6 TFLOPS in its peak time (Cirne et al. 2006). For comparison, the world’s fastest supercomputer, IBM BlueGene/L, has an estimated total processing power of between 280 and 267 TFLOPS (http://www.top500.org/), while Google’s search engine system can muster between 126 and 316 TFLOPS estimated total processing power. While Grids allow enormous potential by aggregating idle
resources that are geographically distributed, for businesses it has furthermore been projected that Grids may decrease total IT costs by 30 % (Minoli 2004).

The report “Grid Computing: A Vertical Market Perspective 2005-2010” (Insight Research 2004) estimates an increase of worldwide Grid spending from $714.9 million in 2005 to approximately $19.2 billion in 2010. Despite existing Grid technology and commercial needs, up till now, almost all research efforts have been focusing on using Grids within academic communities. One of the most prominent activities in academia is the EGEE project (Enabling Grids for eScience, http://www.eu-egee.org/) which is funded by the European Commission. EGEE is developing a service Grid infrastructure which is suitable for any scientific research especially where the time and resources needed for running the applications are considered impractical when using traditional IT infrastructures (e.g. weather forecasts, protein folding etc). With a funding of over 30 million €, the established EGEE Grid comprises over 20,000 CPUs and 5 Petabytes of storage.

While the adoption of Grid technology in eScience has been prospering, the adoption by companies has been slow, mainly due to the lack of viable business models coupled with chargeable Grid services. What is needed is a set of mechanisms that enable users to discover, negotiate, and pay for the use of Grid services on-demand. According to The451Group, one of the leading Grid research institutes, the application of resource trading and allocation models is one of the crucial success factors for establishing commercial Grids (Fellows et al. 2007).

Quite recently, Sun Microsystems has adopted the idea of trading resources, believing that companies will eventually stop maintaining their own infrastructure and instead buy computing power off a Grid1. To put weight on this idea, Sun plans to build up an electronic marketplace for trading resources. Sun started off with offering a fixed price for computing services of $1 per CPU hour. Amazon is currently launching comparable initiatives with Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3).

Hitherto, electronic marketplaces for Grid resources have not yet taken off – there are very few customers using Sun’s fixed price offer. But what is the reason for this limited success of Grid markets? Almost every large computer hardware manufacturer like HP, Sun, or Intel has already worked on or at least pondered the options for Grid markets, but still no Grid market has successfully been launched, creating a Grid adoption gap.

This paper attempts to explain why Grid markets initiatives have failed so far in the conception phase of development already. The explanation mainly focuses on the object that is being traded on Grid markets. As a result of this analysis, it is discovered that not only one single Grid market is necessary but a set or catalogue of different marketplaces satisfying the diverse needs of different market segments. This paper provides a roadmap towards such a catalogue of market mechanisms. As such, this paper provides guidance for potential Grid market operators (e.g. Telecom companies such as France Telecom for Mobile Grids, or hardware vendors such as Sun for Computational and Data Grids) in the choice of

---

1 The idea of introducing markets for scheduling computer resources is by far not a new idea. First suggestions have been made in the mid 60s (Smidt 1968). But despite that markets have not yet made their way into applied scheduling mechanisms.
the market mechanisms that are needed to gear up Grid computing for enterprise usage.

The remainder of this paper is structured as follows. Section 2 discusses possible ways to define the trading object in Grid markets and thus forms the background of the discussion on market mechanisms. Section 3 explores whether one marketplace alone is sufficient for meeting the needs of the Grid providers and users. As a result of section 3, a two-tiered market structure with two classes of markets is proposed as a viable solution for enterprise Grids. Section 4 discusses the use of different market mechanisms for this two tiered market structure and shows which mechanisms are most adequate for which kind of application. Section 5 concludes the paper with a summary and points to future work.

2 Background

Applications can be deployed either by directly accessing resources that are distributed over the network or by invoking a Grid service which encapsulates the respective resources behind standardized interfaces. These two alternative ways of deployment give rise to totally different requirements for potential markets.

From a technical point of view, resources are simple to describe as there exists only a finite set of resources. A resource may be defined by the operating system (e.g. Linux OS), number of CPUs (e.g. 4 * x86 CPU), memory (e.g. 128MB RAM) etc. The GLUE schema provides a standardized vocabulary for describing computing elements. The standardization of resources offers a simple way to semantically describe them. This in turn alleviates resource discovery, as matchmaking is straightforward.

Services on the other hand can be extremely difficult to describe as the service space is infinite due to the myriads of variants in service design. For the description of complex services on the collective layer domain-dependent ontologies can be used, if existent. In the case of raw services, i.e. resource-near services, standardized languages such as JSDL exist. Nonetheless, the indefinite search space tremendously exacerbates service description and likewise service discovery of complex services.

Both resources and services are provided on the basis of Quality-of-Service (QoS) assertions. For resources, the QoS description is simple, as only standardized properties and the duration of resource access matter. For services, however, QoS is more difficult, as not only time aspects play a role but also precision and accuracy of the services. The definition of precision and accuracy depends on the service and cannot be standardized. This also has ramifications on monitoring. While monitoring resource access is relatively simple, the monitoring of very complex services becomes particularly demanding when services are intertwined.

Figure 1 shows the different aggregation levels of services and resources with respect to the layers of the Grid Protocol Architecture. At the bottom are the physical resources on which the services are executed. These resources may be CPUs, memory, sensors, other hardware and software or even aggregated resources such as clusters (e.g. Condor cluster) and designated computing nodes. Raw services are resource-near services (such as storage services). They access resources via standardized interfaces. Raw services also comprise standard application services which could potentially be standardized. Non standardized
application services are complex services which are so diverse that they cannot reasonably be standardized.

![Grid Protocol Architecture](image)

**Figure 1: Grid Protocol Architecture and the different aggregation levels of services and resources**

The deployment of applications is orchestrated via Grid services using existing Grid middleware (e.g. Globus). Typically, services are allocated by the global resource broker that can act as a meta-scheduler to local nodes, where the job requests are queued and executed by batch schedulers of the local resource manager.

When relying on resources, the executables need to be transferred as well. Resources can be deployed as services – in this case the resource providers have to guarantee the completion of the service at a given point of time. Likewise services can be deployed as resources, where the executables need to be transferred to the resource fabric (e.g. Condor cluster). Typically, state-of-the-art Grid middleware does only support limited resource management functionalities. In most cases, the middleware does not enforce any policies concerning how many resources a job can consume because there is no way for the local administrators to specify the degree to which resources can be shared. Trading resources is thus difficult to achieve by means of Grid middleware. Trading resources is possible when it happens on the operating system level, which supports effective resource management. So-called Grid Operating systems, henceforth Grid OS, (e.g. MOSIX) support resource management on the OS Kernel level, and are potentially available for setting up markets (Stoesser et al. 2007). The nice thing about Grid OS is that applications need not be altered to get them ready for the Grid, as is the case when Grid middleware are being used.

From an economic perspective resource markets are promising for automation via an organized electronic market. There are standardized items for sale that potentially attract many buyers and sellers. Complex services have again a disadvantage as demand is highly specialized and distributed across niche markets, such that only few potential buyers are interested in the same or related services. Nonetheless, demand for complex services is huge compared to demand for resources as most of the potential users are interested in getting their services executed, no matter how many physical resources will be needed.
3 One market fits all?

In this section, it will be explored whether it is sufficient to build up and operation one single market for Grids. Based on the previous background description, the answer to this question is straightforward: Designing one Grid market for all kinds of resources, from physical resources such as processing power, memory and storage running on native platforms to sophisticated virtual resources or services, bundling and enriching such physical resources, seems inappropriate due to both technical and economical factors.

a) Technical factors

From the technical point of view, differences in the monitoring and deployment of services and resources exist such that it is very difficult to devise a generic system that is capable of supporting all kinds of resource and service trading. Even worse, different deployment mechanisms impose different requirements on the market mechanism.

b) Economical factors

From the economic point of view, market mechanisms need to achieve the following standard objectives in mechanism design (Stößer, Neumann et al. 2007, cf. Table 1).

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocative efficiency</td>
<td>Allocative efficiency is the overall goal of market mechanisms for Grid resource allocation. A mechanism is allocatively efficient if it maximizes the utility across all participating users (welfare or overall “happiness”).</td>
</tr>
<tr>
<td>Budget-balance</td>
<td>A mechanism is budget-balanced if it does not need to be subsidized by outside payments.</td>
</tr>
<tr>
<td>Computational tractability</td>
<td>The market mechanism needs to be computed in polynomial run time in the size of the number of resource requests and offers.</td>
</tr>
<tr>
<td>Truthfulness</td>
<td>Truthfulness means that it is a (weakly) dominant strategy for users to reveal their true valuations to the mechanism.</td>
</tr>
<tr>
<td>Individual rationality</td>
<td>A mechanism is individually rational if users cannot suffer a loss in utility from participating in the mechanism, i.e. it is individually rational to participate.</td>
</tr>
</tbody>
</table>

Table 1: Economic Objectives

As mentioned in the background discussion, trading resources and services imposes totally different requirements on the market: while resources are more or less a commodity for which auction mechanisms seem to work well, (complex) services are inherently non-standardized making auction-like mechanisms inapplicable. But even for trading resources it substantially matters how they are deployed.

From this brief discussion, it can be stated that a one-size-fits-all market for Grids is infeasible from both the technical as well as from the economic side. Instead, due to their heterogeneous properties, Grids can be divided into two different types of markets: resource markets and service markets spanning out a “two tiered
*Grid market structure*[^2]. In the following, these two classes of markets are analyzed in terms of their requirements on the market structure[^3] (cf. Figure 2).

![Figure 2: Two-tiered market structure](image)

- **Tier 1-Markets**

On the physical resource market, low-level resources are traded such as processing power, memory, and storage. Demand in the resource market is enfolded by complex services that need resources to be executed. This setting poses special requirements on the used market mechanism (Schnizler et al. 2006). However, as aforementioned, the requirements also depend on the way resources are being deployed due to technical restrictions.

**Deployment as resource**

The technical requirements stemming from the way of deployment are listed in Table 2.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-attributes</td>
<td>Physical resources, either deployed as raw service or as resource, have quality attributes such as speed of the CPU, the operating platform or bandwidth. Thus the mechanism needs to cope with multi-attributes.</td>
</tr>
<tr>
<td>Bids on bundles</td>
<td>Generally, users require a combination of resources to execute a job (e.g. CPU and memory). If the mechanism does not account for bids on bundles, the user is facing to the risk of obtaining only one leg of this bundle without the other (the so called “exposure risk”). The market mechanism thus needs to support requests for bundles of</td>
</tr>
</tbody>
</table>

[^2]: We use the term „market structure“ to denote the configuration of marketplaces.

[^3]: It should be noted that there could be n-intermediate markets. We consider only the extreme ones, as they exhibit different characteristics.
The allocation of the mechanism needs to be made instantaneously, as the market assumes the role of operating system schedulers. The mechanism thus needs to be a lightweight mechanism that requires little computation time. Being an online mechanism is crucial because in the case of a decrease of the performance of an application requires an adaptation to a new execution state, new resources need to be found and scheduled for immediate execution.

Typically, Grid OS are migrating jobs from one resource to another. The mechanism needs to be split-proof in a sense that users cannot improve their priority by splitting jobs into more parts.

Likewise, the mechanism needs to assure that no users have advantages through the merger of jobs.

### Table 2: Requirements for trading resources

**Deployment as raw service**

The technical requirements stemming from the way of deployment are listed in Table 3.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-attributes</td>
<td>See above</td>
</tr>
<tr>
<td>Bids on bundles</td>
<td>See above</td>
</tr>
<tr>
<td>Time attributes</td>
<td>When raw services are traded the market mechanism needs to take time attributes into account. The requesters need to specify their demand, so that the market mechanism can efficiently schedule the job requests according to availability of resources and to the price. This differs from trading resources, where the market mechanism executes jobs upon availability.</td>
</tr>
<tr>
<td>Co-Allocation</td>
<td>Capacity-demanding Grid applications usually require the simultaneous allocation of several homogenous service instances from different providers. For example, a large-scale simulation may require several computation services to be completed at one time. This situation where simultaneous allocation of multiple homogenous services is called co-allocation. A mechanism for the service market has to enable co-allocations and provide functionality to control it.</td>
</tr>
<tr>
<td>Coupling</td>
<td>For some applications, it may be logical to couple multiple raw services of a bundle in order to guarantee that these are allocated from the same seller and will be executed on the same machine. The mechanism ideally offers this functionality.</td>
</tr>
<tr>
<td>Resource Isolation</td>
<td>Security and performance considerations lead to the requirement of resource isolation. In fact, it can only be assured that an application satisfies a given QoS or security level, respectively in case the resources on which the application is executed are committed only to one party.</td>
</tr>
</tbody>
</table>
**Tier 2-Markets**

In the service market, applications demand the execution of their constituting complex services. Along the lines of the two-tiered market structure, complex services can be decomposed into smaller services that can in turn be translated into resources that are necessary for executing them. E.g., some complex service might require a basic XML Transformer service which in turn needs processing power, memory, storage etc. Buyers in such a service market request a complex service – the provider of this service, the service integrator, is responsible for obtaining the required raw services and physical resources in turn, thus hiding parts of the Grid’s complexity from the buyer. Such a hierarchical shading of complexity seems to be an appropriate approach since service requesters typically have no clue how much resources the complex service will consume (Eymann et al. 2006).

The requirements on the market mechanism are totally different than for resource markets. Complex services are rarely used by two different companies – installing competition hence does not make sense. The service market rather imposes the difficulty to find a counterpart that is exactly offering the capabilities to execute the job. The market mechanism is more search-oriented such as bilateral or multi-lateral negotiation protocols (cf. Table 4).

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-attributes</td>
<td>See above</td>
</tr>
<tr>
<td>Workflow support</td>
<td>To support complex services, distributed resources such as computational devices, data, and applications need to be orchestrated while managing the application workflow operations within Grid environments. The market-mechanism needs to account for this during design time and run time of the workflow.</td>
</tr>
<tr>
<td>Scalability</td>
<td>Scalability considers how the properties of a protocol change, as the size of a system (i.e. the participants in the Grid) increases. The market mechanism needs to be scalable per se in order to be applicable.</td>
</tr>
<tr>
<td>Co-Allocation</td>
<td>See above</td>
</tr>
</tbody>
</table>

The problem with current markets for Grid is that they are designed as resource markets. For example Sun’s $1/CPU-hour advertisement campaign aims at selling physical resources, i.e. CPU hours. This type of market is, however, by design not relevant for enterprise customers who have deadlines to meet until when a job needs to be executed and have no idea of how many physical resources are required to meet the deadline. While enterprises typically have time critical jobs to execute, applications in academia are less time dependent. As such, resource markets where would be viable business models – here the users have to wait until the queued jobs are being executed. But clearly, the issue of payment for resources is controversially discussed in academia. In the future, even for the EGEE Grid, billing and payment will soon become an issue as demand exceeds...
supply. It seems that resource markets will become an adequate model for academia Grids such as EGEE or D-Grid, the German Grid initiative.

Grid markets that will be widely accessed by enterprises need to be of the form of service markets on which complex services are offered. For example a manufacturer is interested in executing a Computer Aided Design application and deploying it on a computation intense platform. This complex service showcases a very specific service which is likely to be demanded from one single requester only. To accommodate this complex service, the service must be decomposed into its constituting raw services and further on to the physical resource demand. Integrators are needed to facilitate this decomposition process, where integrators denote companies which are specialized in aggregating and disaggregating services into resources. Specialization stems from experience, allowing the identification of the service needs by comparing it with similar services in the past, where similarity is established in terms of algorithms, data structures and sizes, etc. Telecommunication companies and hardware producers seeking to virtualize IT infrastructures naturally have a strong interest and the competency to become such integrators.

Table 5 summarizes the discussion above.

<table>
<thead>
<tr>
<th>Trading Object</th>
<th>Physical Resource</th>
<th>Complex Application Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Structure</td>
<td></td>
<td>Tier 2-market</td>
</tr>
<tr>
<td>Deployment Description</td>
<td>Resource</td>
<td>Raw Service</td>
</tr>
<tr>
<td>languages</td>
<td>GLUE</td>
<td>JSDL/RSL</td>
</tr>
<tr>
<td>Time limits</td>
<td>No</td>
<td>yes</td>
</tr>
<tr>
<td>Application areas</td>
<td>Academia</td>
<td>Integrators and resource</td>
</tr>
<tr>
<td>Requirements</td>
<td>Multi-attributes,</td>
<td>providers</td>
</tr>
<tr>
<td></td>
<td>Bids on bundles,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Online mechanism,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Split-proof,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Merge-proof</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Grid Market Types

4 A Roadmap for Grids

In the previous sections, we have motivated that Grids require not something like a Global Grid market where all Grid requests and supplies are collected, but a more complex two-tiered market structure. Figure 2 summarizes the market structure. On the resource market complex services require physical resources either plainly deployed or accessed via service interfaces. Applications demand the execution of several complex services. Integrators assume the responsibility of mediating between the applications being unaware of their resource need and the resources themselves.
In the following we will set up a taxonomy of known market mechanisms which support different types of Grid applications. This taxonomy is conceived to be a roadmap for further Grid Market developments that help bridging the adoption gap.

1.1. Market Mechanisms for Trading Resources

For Grids, where resources (and not services) are traded, no time restrictions apply. Usually, not the resources itself are traded but shares of computing units (e.g. nodes). The idea is that bidding determines the share the user receives from the pool of available nodes. The more resources a user obtains, the faster the application is completed. Since no time restrictions apply, batch and interactive applications can be served by the same market mechanisms.

The following online mechanisms apply for resource-based systems:

- Fair Share (Kay and Lauder 1988)
- Proportional Share (Lai et al. 2004)
- Pay-as-bid (Sanghavi and Hajek 2005; Bodenbenner et al. 2007)

Obviously, there are mechanisms available that serve resource markets on the Grid OS level. All three groups of mechanisms, however, have the same drawback, as they can be used in scenarios only where one resource provider serves several consumers. There is no competition among the providers. In case all resources are under fully centralized control, this is unproblematic. But in Grids, the underlying idea is to cross administrative boundaries. Apparently, there is a need for market mechanisms that satisfy all requirements of the resource market.

1.2. Market Mechanisms for Trading Services

Dividing the service market into two parts – raw and complex services – however, is too simplistic, as the timing when demand for services occurs has not yet been considered. This timing is determined by the application itself and depends on the task the application is performing. We use the term application model as a characterization of the processing mode of the application. This encompasses in particular the workload of the application as well as the interaction model between applications and the Grid middleware virtualizing the execution platform. Depending on the application model, different requirements upon the market mechanisms emerge.

- **Batch applications** are characterized by a planned execution and expected termination time (e.g. data mining). Execution is serial and resource demands depend on the parameters, such as the size of input data.

- **Interactive applications** are applications that require services on demand, depending on the interactions with users (e.g. online data analysis). Different than batch applications, with interactive applications it is not possible to plan execution and expected termination time far in advance, so there can be unpredictable peaks of requests occurring within a short time.

- **Task-oriented applications** are dynamically composed from tasks to build more complex tasks. Service demand depends on the (work-) flow of requests from multiple users (e.g. transaction system of a bank constitutes).
Most of the market-based approaches relate to batch applications. Batch applications are comparably easy for two main reasons. Firstly, there is no need to consider a whole workflow with different resource demands on each echelon of the workflow. Secondly, the time to determine the allocation can be relatively long; immediacy is not essential. Thus, complex resource allocation computations can be performed without hampering the whole application due to latency times devoted to the calculation of the optimal allocation. Even worse, most of the practical market-based Grid prototypes consider only one single resource type (e.g. CPU only) and thus make use of standard auctions (e.g. English auction). In applications other than pure number crunching those auction types are inadequate as more than one object (e.g. memory) is required at the same time.

4.2.1. Market Mechanisms for Trading Raw Services

As mentioned above, the market mechanisms for raw services depend on the application model. In the following we discuss the mechanisms that are adequate for batch applications.

- Multi-attribute Combinatorial Auction (Bapna et al. 2006)
- Multi-attribute Exchange (Stoesser, Neumann et al. 2007)
- MACE-mechanism (Schnizler et al. 2006)
- Combinatorial Scheduling Exchange (AuYoung et al. 2004)
- Augmented Proportional Share (Stoica et al. 1996)

For interactive applications it is impossible to predict demand for raw services. Thus, the mechanisms need to allocate the raw services continuously. This can be realized by either frequent call mechanisms, where bids are collected for a very short time span and right away cleared. This requires that the mechanism is solvable in few milliseconds. Alternatively, the mechanism could be an online mechanism, which allows the real-time job submission to available resources (e.g. nodes or clusters). A third way is to introduce a derivative market to insure against the risk of supernormal resource demand.

- Multi-attribute Combinatorial Auction Heuristic (Bapna, Das et al. 2006)
- Multi-attribute Exchange (Stoesser, Neumann et al. 2007)
- Online scheduling mechanism (Heydenreich et al. 2006; Stoesser et al. 2007)
- Augmented Proportional Share (Stoica, Abdel-Wahab et al. 1996)
- Derivative Markets (Kenyon and Cheliotis 2002; Rasmusson 2002)

The requirements on market mechanisms that support task-oriented applications are very demanding, as all constituents of the workflow needs to be allocated – otherwise the application has no value to the user. Currently, there are only bargaining protocols available that guides the user in its search for all components of the workflow.

- Bargaining Protocol (Czajkowski et al. 2002)
4.2.2. Market Mechanisms for Trading Complex Services

As stated earlier, trading complex services is very demanding as there are not many providers and requesters. Currently, there is not much research available that aims at developing market mechanisms for trading complex services. Hence, the market mechanisms for batch, interactive and task-oriented applications do not differ substantially.

For batch applications there are three different market mechanisms suitable, where the first one, MACE, provides a sufficiently rich bidding language for supporting complex services. The second mechanism is the bargaining protocol, while the last one refers to take-it-or-leave-it pricing, where the vendor sets the price and the users decide whether or not to purchase.

- MACE-mechanism (Schnizler, Neumann et al. 2006)
- Bargaining Protocol (Czajkowski, Foster et al. 2002)
- Software-as-a-service4 (SaaS)

Interactive applications make the design of adequate market mechanisms even more difficult. Potentially, the following mechanisms could be used.

- Bargaining Protocol (Czajkowski, Foster et al. 2002)
- Software-as-a-service (SaaS)
- Online scheduling mechanism (Heydenreich, Müller et al. 2006; Stoesser, Roessle et al. 2007)

As mentioned before, task-oriented applications are very demanding due to the exposure risks involved. Mechanisms that could be used for trading are bargaining protocols and SaaS. It should be pointed out that there is currently no research on this field available.

- Bargaining Protocol (Czajkowski, Foster et al. 2002)
- Software-as-a-service (SaaS)

5 Conclusion

This paper argues that the technology of Grid computing has not yet been adopted by enterprises due to the absence of viable business models. In academia, Grid technology has already been taken up, but the sharing approach among non-for-profit organizations cannot be transferred to enterprises. We pick up the idea of a Grid market to overcome the adoption gap. This idea is not new by any means, but hitherto all proposals that have been made by computer scientists unaware of economic algorithms and models or by economists being unaware of the technical possibilities. This paper attempts to derive an economically sound set of market mechanisms based on a solid understanding of the technical possibilities.

Section 2 analyzed the nature of the object that can be traded in Grids. As shown in the paper, the nature of the trading object is closely associated with the deployment of software applications. Deployment as resource or as service has

---

4 Software-as-a-service refers more to model of software delivery where a service provider (e.g. SAP) offers to requesters applications that are specifically implemented for one-to-many hosting.
major ramifications on the trading object and consequently on the requirements on market mechanisms. Resources are essentially commodities, where services can be both standardized commodities (i.e. raw services) and non-standardized unique entities (i.e. complex services).

Based on the subsequent analysis, this paper derived a two-tiered market structure where all markets on each tier demand for different market mechanisms. The first tier comprises the markets for physical resources (e.g. CPU, memory) that can be accessed either as resource or as raw service. The second tier comprises the markets for complex services.

Section 3 showed the requirements on the market mechanisms depending on the tier of the market structure. Section 4 introduced which mechanisms are suitable for what area. Table 6 summarizes the set of market mechanisms that can be used depending on the application and respective tier within the market structure.

<table>
<thead>
<tr>
<th>Application model</th>
<th>Physical Resource</th>
<th>Complex Application Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Resource</td>
<td>Raw Service</td>
</tr>
<tr>
<td></td>
<td>Fair Share</td>
<td>MACE</td>
</tr>
<tr>
<td></td>
<td>Proportional</td>
<td>Multi-attribute</td>
</tr>
<tr>
<td></td>
<td>Share</td>
<td>Combinatorial Auction.</td>
</tr>
<tr>
<td></td>
<td>Sanghavi-Hajek</td>
<td>Augmented Proportional Share</td>
</tr>
<tr>
<td>Interactive</td>
<td>Online Scheduling</td>
<td>MACE</td>
</tr>
<tr>
<td></td>
<td>Multi-attribute</td>
<td>Bargaining Protocols</td>
</tr>
<tr>
<td></td>
<td>Auction</td>
<td>Software-as-a-service</td>
</tr>
<tr>
<td></td>
<td>Augmented</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proportional Share</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Derivative Markets</td>
<td></td>
</tr>
<tr>
<td>Task-oriented</td>
<td>Bargaining</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protocols</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Market Mechanisms Canon for Grids

In essence, the results of this paper suggest several natural and intriguing research avenues:

- Analyze the properties of the proposed mechanisms for the respective application model classes.
- Compare the efficiency of the market mechanisms attributed to the different classes of the market mechanism canon.
- Implement the mechanisms and conduct field studies in order to get real data.
- Develop market mechanisms, where the market mechanism canon is mostly silent (e.g. task-oriented applications).
- Develop sustainable business models for companies that provide market platforms for trading Grid services or resources.
• Identify the size and the potential revenue of the single markets of the two-tiered market structure.
• Identify limits of the use of market mechanisms in Grid.

References

