On the simulation of Grid market coordination approaches

Werner Streitberger, Sebastian Hudert and Torsten Eymann
University of Bayreuth, Chair of Information Systems Management, Universitaetsstrasse 30, 95440 Bayreuth, Germany

Bjoern Schnizler
University of Karlsruhe, Department of Economics and Business Engineering, Englerstrasse 14, 76131 Karlsruhe, Germany

Floriano Zini
IRST - Fondazione Bruno Kessler, via Sommarive 18, 38100 Trento, Italy

Michele Catalano
Universita delle merche Ancona, Dipartimento di Economia Piazzale, Martelli 8, 60121 Ancona, Italy

April 15th, 2007, accepted on October 18th, 2007

Abstract. Grid computing has recently become an important paradigm for managing computationally demanding applications, composed of a collection of services. The dynamic discovery of services, and the selection of a particular service instance providing the best value out of the discovered alternatives, poses a complex multi-attribute n:m allocation decision problem, which is often solved using a central resource broker. However, decentralized approaches to this service allocation problem represent a much more flexible alternative, thus promising improvements in the efficiency of the resulting negotiations and service allocations. This paper compares centralized and decentralized service allocation mechanisms in grid market scenarios according to a defined set of metrics.

Keywords: Grid Economics, Economic Resource Allocation, Grid Simulation

1. Introduction

Grid computing allows the secure and coordinated sharing of globally distributed resources spanning several physical organizations (Foster and Kesselman, 1999). Service-Oriented Architectures (SOAs) underlie several of the current Grid initiatives and reflect the current Grid computing infrastructures, where participants offer and request application services. SOA defines standard interfaces and protocols that enables developers to encapsulate resources of different complexity and value as services that clients access without knowledge of their internal workings (Foster, 2005). Grid systems have therefore increasingly been structured as networks of inter-operating services that communicate with one another via standard interfaces. Such infrastructures of services provided to an a priori unknown set of consumers can be efficiently organized as markets analogously to traditional service markets in real world economies. This article describes an investigation in implementing an electronic Grid Market based on the "'Catallaxy"' concept of F. A. von Hayek (Hayek, 1945). Catallaxy describes a "'free market"' economic self-organization approach for electronic services brokerage, which can be implemented for realizing service markets within service-oriented global computing infrastructures. These comprise network concepts, such as Grid and Peer-2-Peer (P2P) systems, which overlay the existing physical Internet topology. In service-oriented global computing infrastructures, participants offer and request actual application services and computing resources for providing such services, of different complexity and value - creating interdependent markets.

In this article, the complex interdependencies are broken down into two interrelated markets:

- a service market - which involves trading of application services, and
- a resource market - which involves trading of computational and data resources, such as processors, memory, etc.

This distinction between resource and service is necessary to allow different instances of the same service to be hosted on different resources. It also enables a given service to be priced, based on the particular resource capabilities that are being made available by some hosting environment. In such interrelated markets, allocating resources and services on one market inevitably influences the outcome on the other market. This concept of two interrelated markets takes the current Grid concept one step further. Current Grids just start to shift from providing raw computing resources to actual services offered within the system. Our approach does not only incorporate services as basic units provided to consumers within the Grid, but also defines a resource market trading the actual computational resources needed for implementing those services.

A common approach of many other Grid market concepts is to allocate resources and services by relying on the presence of central resource/service broker. However, the complex reality could turn such approaches useless, as the underlying problem is computationally demanding and the number of participants in a service-oriented global computing infrastructure can be huge. The research question taken up in this article is *how* to develop a Grid realization of an economic concept, which describes the ability to trade electronic services in a decentralized fashion, a free-market economy to adjudicate and satisfy the needs of participants who are self-organized and follow their own interest.

The Catallaxy concept represents a coordination approach for systems consisting of such autonomous agents, and is based on constant negotiation and price signalling between agents (Eymann et al., 2003b). Every individual (agent) assigns a value to service access information, and by exchanging bids for service access, the price signals carry information between individuals (agents) about the knowledge of others (Hayek, 1945). This exchange of information applies even across markets, as changing availability on the resource market will be reflected by cascading price changes for those basic services which rely on such resources. Hayek called this feature a "'telecommunication"' system in its literal sense. The huge size of Grids to be controlled, and the availability of software agent technology, makes implementing Hayek's Catallaxy an alternative to a central allocation approach, using the ensuing "'spontaneous order"' as a concrete proposal for both the design and coordination of information systems. The resulting multi-agent system will be highly dynamic, thereby leading to Grid networks which behave in a P2P fashion. The term P2P should be interpreted not as a specific system architecture, but as a general approach for distributed system design.

This paper investigates the efficiency and general outcome of decentralized resource negotiations in grid systems. For this purpose a particular grid environment is implemented and used for the actual simulation runs. Using Grid simulation software, different economic settings are investigated. The simulation results are evaluated using a defined set of efficiency metrics. This paper concludes with discussing the resulting metrics.

2. Related Work

The use of market mechanisms for allocating computer resources is not a completely new phenomenon. Regev and Nisan propose within the scope of the POPCORN project the application of a Vickrey auction for the allocation of computational resources in distributed systems (Regev and Nisan, 2000). The Vickrey mechanism achieves truthful bidding as a dominant strategy and hence results in an efficient allocation (Vickrey, 1961).

Buyya was among the first who motivated the transfer of market-based systems from distributed systems to Grids (Buyya et al., 2002). Wellman et al. model single-sided auction protocols for the allocation and scheduling of resources under consideration of different time constraints (Wellman et al., 2001).

Wolski et al. compare classical auctions with a bargaining market (Wolski et al., 2003). As a result, they come to the conclusion that the bargaining market is superior to an auction based market. This result is less surprising, as the authors only consider classical auction formats, where buyers cannot express bids on bundles (Wolski et al., 2003). Eymann et al. introduce a decentral bargaining system for resource allocation in Grids, which incorporates the underlying topology of the Grid market (Eymann et al., 2003a).

Subramoniam et al. account for combinatorial auctions by providing a *tâtonnement* process for allocation and pricing (Subramoniam et al., 2002). Combinatorial auctions are multi-item auctions, where an agent can submit bids on multiple heterogeneous resources as a bundle (Cramton et al., 2006). A bundle consists of logical *AND* concatenated bids on a set of resources. Such bids ensure that an agent is allocated to either all resources of the bundle or to none of it. A practical example for a combinatorial auction is to bid for a bundle that comprises CPU, memory and hard disk capacity. If the agent would only receive the CPU without the memory and the hard disk capacity, the allocation would be useless. By means of bidding on these items in the form of a bundle, the agent can ensure that he gets all or none of the resources. Furthermore, combinatorial auctions allow expressing sub-additive valuations. This is realized by allowing multiple XOR concatenated bids on a set of bundles, where the XOR operator ensures that at most one bundle is allocated to an agent. Extending the aforementioned example, this allows the agent to bids on both, a resource bundle of resource provider as well as a resource bundle of resource provider B (Schnizler, 2007).

Conen goes one step further by designing a combinatorial bidding procedure for job scheduling including different running, starting, and ending times of jobs on a processing machine (Conen, 2002). However, these approaches are single-sided and favor monopolistic sellers or monopsonistic buyers in a way that they allocate greater portions of the surplus. Installing competition on both sides is deemed superior, as no particular market side is systematically put at advantage. Competition assumes that there is a ”contest between rivals”. As an example for competitive behaviour of agents suppose two rival companies that compete for the access to one single supercomputer.

Demanding competition on both sides suggests the development of an exchange. In literature, Parkes et al. introduce the first combinatorial exchange as a single-shot sealed bid auction (Parkes et al., 2001). As payment scheme, Vickrey discounts are approximated. Biswas and Narahari propose an iterative combinatorial exchange based on a primal/dual programming formulation of the allocation problem (Biswas and Narahari, 2004). 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 $2^G - 1$ possible combinations.

Several Grid simulators allow the modeling and simulation of Grid resources and allocation policies; examples include OptorSim (Bell et al., 2003) (Cameron et al., 2004), GridSim (Buyya and Murshed, 2002), MicroGrid (Song et al., 2000) and SimGrid (Legrand et al., 2003). OptorSim provides the features needed to model and evaluate the data transfer and replication strategies in data Grids. It is a discrete event simulator implemented in Java and follows the abstraction of data resources. The main goal of this simulator is to provide a means for evaluation of data transfer strategies in a data Grid, and so does not provide a service-oriented application model. MicroGrid enables the emulation of a Grid environment. A user can run her Grid application on this emulated environment, while the simulator intercepts the exchanged messages. Although it is possible to simulate service-oriented applications, MicroGrid does not provide a decoupling of the service and resource layers that allows the design and evaluation of different strategies or economic mechanisms for each layer. SimGrid provides a set of abstractions and functionalities that can be used to build simulators for several application domains. The core functionalities can be used to model and evaluate parallel application scheduling on distributed computing platforms. SimGrid also provides emulation facilities for running distributed and parallel applications in an emulated Grid environment. SimGrid is a trace based event simulator and, like the simulators previously described, uses the abstraction of ’resources’.

GridSim (Buyya and Murshed, 2002) is a Java-based Grid simulation toolkit that provides features for application composition, information services, and the ability to model heterogeneous computational resources of variable performance. In addition, GridSim provides an auction framework that allows the design and evaluation of auction protocols for Grid systems. By using these features, it is possible to model and evaluate the scheduling of jobs on Grid resources and evaluate the impact of the allocation policies. GridSim has the features necessary to design and model the resource layer. The features provided by GridSim enable the modeling and simulation of intricate Grid environments. However, it does not provide a service framework for simulating service-oriented Grid applications. In this work we opted for leveraging the existing features of GridSim and providing a service framework that enables the modeling and evaluation of service provisioning policies, resource allocation policies and multiple economic mechanisms for service negotiation and resource management.

All these simulation environments allow the allocation computing or data resources using simple economic and technical allocation approaches, but don't support service-oriented application models. Therefore, new simulation models are needed to overcome these drawbacks of current Grid simulation environments. The next Section describes an abstract simulation model for service-oriented applications.

3. Simulation Model

This section describes the formal model used in simulation to abstract from concrete networks, which defines the Grid network (GN). A Grid network is defined by a connected non-oriented graph

$$GN=\langle S,L\rangle$$

where $S=\{1,\ldots,n\}$ is a set of network sites and $L=\{\langle i_1,j_1\rangle,\ldots,\langle i_m,j_m\rangle\}$ is a set of links which connect sites. Each site i is characterised by:

- a *failure probability* fs_i which models the unreliability of the site s_i in sending messages to other sites;
- a triple

  $$\langle CSA_i, BSA_i, RA_i\rangle$$

  where CSA_i is a set of *Complex Service Agents (CSAs)*, BSA_i is a set of *Basic Service Agents (BSAs)*, and RA_i is a set of *Resource Agents (RAs)*. In every site there can be zero or more complex/basic service agents and zero or more resource agents, that is

  $$|CSA_i|\geq 0, |BSA_i|\geq 0, |RA_i|\geq 0$$

  A node with no associated agents is a *router*.

Each link $\langle i,j\rangle$ is characterised by a bandwidth $bw_{i,j}$ which defines the maximum amount of information (bits/second) that can be transmitted along the link.

Complex Service Agents. CSAs are entry points to the Grid system and are able to execute *Complex Services (CSs)* for Grid clients. A CS is defined as a set of *Basic Services (BSs)*. CSAs are not specialised: they accept any type of complex service request and take care of the execution of the component basic services.

Basic Service Agents. BSAs provide CSAs with the BSs they need to furnish their complex services to Grid clients. BSs have two *attributes*: *name* and *quality*. Name is a unique identifier whose intended semantics (i.e., the provided service) is shared among all agents. Values for quality are from a discrete set. The intended semantics of quality values is shared among all agents.

For example, there can be a basic service named *pdf_converter* whose quality assumes values in the set $\{bronze, silver, gold, platinum\}$. Another example is a basic service named *PrinterService* with quality in the set $\{silver, gold\}$. Given these BSs, an example CS is

$$CS_1 = \{\langle PDFConverter, gold\rangle, \langle PrinterService, silver\rangle\}$$

As a CS is defined as a *set* of BS, there are no assumptions on the order BSs are executed. In other words, the notion of *workflow* is not considered in the definition of CSs.
Every BSA has an associated BS. This means that BSAs are specialised and able to execute specific basic services. Multiple BSAs for the same basic service can co-exist in the Grid network. For example, in an Grid system there might be two or more BSAs providing a the BS $\langle pdf\_converter, silver\rangle$.

Resources. Resources have a *name*, for example *storage* or *cpu*, or *ram*. Name is a unique identifier whose intended semantics is shared among all agents. Every resource is also characterised by a *quantity* whose value is a positive integer. The intended semantics of quality values is shared among all agents. For example, the resource *storage* might have $quantity$=50 while quantities for resources *cpu* and *ram* could be 100 and 150 respectively. The unit for each resource is assumed to be virtual: for each resource the value of *quantity* represents the maximum available amount of resource expressed in the virtual unit.

Resource Bundle. A resource bundle is a set of pairs

$$\langle resource\_name, resource\_quantity\rangle$$

Examples of resource bundles are:

$$RB_1 = \{\langle cpu, 70\rangle, \langle storage, 40\rangle\}$$
$$RB_2 = \{\langle cpu, 50\rangle, \langle storage, 50\rangle, \langle ram, 75\rangle\}$$

Every basic service has an associated resource bundle. The bundle defines which resources and respective quantity are necessary for service provision. For example, the association

$$\langle PDFConverter, gold\rangle \longrightarrow RB_1$$

specifies that for the execution of a pdf conversion having quality gold there is the need of a 70 CPU units and 40 storage units.

Resource Agents. RAs are “proxies” for aggregations of resources. Their task is to provide BSAs with resources needed for the execution of basic services. Every RA has an associated *available resource bundle*. An available resource bundle is similarly defined to a resource bundle but the resource quantities define the maximum resource amounts which are available from the RA. Example of available resource bundles are

$$ARB_1 = \{\langle cpu, 100\rangle, \langle storage, 40\rangle\}$$
$$ARB_2 = \{\langle cpu, 100\rangle, \langle storage, 40\rangle, \langle ram, 150\rangle\}$$

Multiple RAs for the same available resource bundle can co-exist in the Grid system.

3.1. Service and resource allocation

The allocation of services and resource takes place in two separate markets, the *service market* and the *resource market*. As the meaning of "allocation" in the two markets is not the same, here its two semantics are described.

3.1.1. *Service Market*

Basic services are provided by single BSAs. A CSA receiving a request for CS provision starts a "service allocation process" for each of the BSs included in the CS. Every "service allocation process" produces the selection of a single BSA able to provide the BS. Service allocation abortion is possible.

3.1.2. *Resource Market*

A BSA requests a bundle of resources on the resource market in order to be able to execute its specific basic service. Issuing such a request initiates a "resource bundle allocation process". This process can have a number of outcomes:

1. abortion, if there are no RAs able to provide the needed resources;
2. one single RA provides the total amount of every resource in the requested bundle;
3. multiple RAs partially provide all resources in the requested bundle;

In the simulations for this paper, we assume that one single RA provides the total amount of every resource in the bundles requested by BSAs and therefore there is no co-allocation of resources.

4. Simulation Environment

This section describes the simulation architecture of two allocation mechanisms for the simulation model presented in the previous section. As a reference for the catallactic approach, two auctions mechanisms are designed and implemented. On the service market, a continuous double auctions fits best to the simulation model and on the resource market a combinatorial auction is developed. The simulation architecture for central and catallactic approach shows the design of the simulated Grid scenarios and the interaction sequence of the agents.

4.1. AUCTIONS

Formerly, auctions have been successfully applied to trade a variety of different commodities such as financial shares, electricity, or logistic scenarios. Auctions are institutions with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants (McAfee and McMillan, 1987). As auctions can achieve economically efficient outcomes, their application to the Grid scenario is considered a promising approach. In this subsections, an auction schema for the service market and another schema for the resource market are introduced.

4.1.1. *Service Market*

In the service market, we apply a double auction market institution. Buyers and sellers are services, which require other auxiliary services. That is, we distinguish basic services as sellers and complex services as buyers. Basic services offer one or more specific auxiliary services. Hence, they are responsible for providing the auxiliary services to the buyers as well as for acquiring the required resources for the services on the resource market. Obviously, the products traded on the service market are completely standardized.

For example, a complex service receives a request for data analysis. Two basic service are needed for this task, a basic service for data mining and a basic service for data conversion. Both basic services sell their standardized service to a complex service which is able to fulfill his data analysis task now.

In a double auction market, a large number of participants trade a common object and can submit bids (buy orders) and asks (sell orders). Trading in double auctions is organized by means of order books, each for a set of homogeneous goods. An order book is responsible for storing non executed orders of the agents. For instance, in the service market there will be *n* different order books, each for one of the *n* different services. Buyers and sellers submit their bids in a sealed envelope to the auctioneer. The auctioneer aggregates the bids to form supply and demand curves. Once these curves are aggregated, they are used to set a specific price for trading – the price at which supply equals demand[1].

A key consideration in double auctions is the timing of the clearing process, i.e. the timing of determining the auction winners and thereby the allocation of the resources. Double auctions can be either cleared continuously (Continuous Double Auction) or periodically (Periodic Double Auction, Call Market): A Continuous Double Auction (CDA) is a double auction where buyers and sellers simultaneously and asynchronously announce bids and offers. Whenever a new order enters the market, the auctioneer tries to clear the market immediately. Thus, the CDA is advantageous especially in terms of immediacy. A Call Market is a double auction with periodic uniform clearing, e.g. the auctioneer clears the market every fives minutes. All orders in a period are collected in an order book and will be cleared periodically. Assuming none time-critical resources, the call market is advantageous in terms of enhancing the overall welfare in a market. A short time period may increase the overall welfare of a market; considering the immediate service allocation, a continuous clearing would be superior. The effects of both concepts have to be evaluated for the service market scenario by means of simulations.

4.1.2. *Resource Markets*

In the resource market, participants are the basic services as resource consumers (buyers) and resource services (sellers) offering computational services having specific capacities, e.g. processing power. The same resources (e.g. CPU) can differ in their quality attributes, e.g. a hard disk can have 30GB or 200GB of space.

Continuing the data mining example from the previous section, the data mining basic service needs an application environment to execute his service. The basic service specifies the amount of resources and bids for its needed resources. Resource services offer their resources to the basic service.

An adequate market mechanism for the resource market has to support simultaneous trading of multiple buyers and sellers, as well as an immediate resource allocation. Furthermore, the mechanism has to support bundle orders – i.e. all-or-nothing orders on multiple resources – as basic services usually demand a combination of computer resources. For comprising the different capacities of the resources (i.e. resources can differ in their quality), the mechanism has to support bids on multi-attribute resources.

Reviewing the requirements and surveying the literature, no classical auction mechanism is directly applicable to the resource market. Instead, a multi-attribute combinatorial exchange (MACE) is applied that satisfies the described requirements (Schnizler, 2006).

[1] Price tunnels (i.e. ranges where any price will be acceptable because the supply and demand curves overlap) are resolved using the *k*-pricing schema as presented in (Friedman, 1991).

MACE allows multiple buyers and sellers simultaneously the submission of bids on heterogonous services expressing substitutabilities (realized by XOR bids) and complementarities (realized by bundle bids). Furthermore, the mechanism is capable of handling cardinal attributes as well as an immediate execution of given orders as the clearing can be done continuously. For instance, a resource consumer can bid on a bundle consisting of a computation service and a storage service. The computing service should have two processors where each processors should have at least 700MHz. Furthermore, the storage service should have 200MB of free space. The bids can be formulated as WS-Agreement offers (Ludwig et al., 2004) and thereby comply with standard resource negotiation mechanisms applied in current Grid systems. After the participants submitted their bids to the auctioneer, the allocation (winner determination) and the corresponding prices are determined.

The objective of the winner determination problem in MACE is the maximization of social welfare, i.e. the difference between the buyers' valuations and the sellers' reservation prices. The problem is formulated as a linear Mixed Integer Programm (MIP) and thus can be solved by optimization solvers such as CPLEX[2]. The winner determination is, however, a generalization of the combinatorial allocation problem (CAP) and thus $\mathcal{NP}$ complete. For large-scaled scenarios, the use of approximations has to be evaluated (Mito and Fujita, 2004). Nevertheless, the application of such a complex problem seems to be promising, as the number of different bundles in the resource market is restricted.

The outcome of the winner determination model is allocative efficient, as long as buyers and sellers reveal their valuations truthfully. The incentive to set bids according to the valuation is induced by an efficient pricing mechanism. With respect to the economic objective of achieving an efficient allocation, a pricing scheme based on a Vickrey-Clarke-Groves (VCG) mechanism would attain this objective. Moreover, VCG mechanisms are the only allocative-efficient and incentive-compatible mechanisms (Green and Laffont, 1977).

The basic idea of a VCG mechanism is to grant a participant a discount on its bids. This discount reflects the impact of that bid on the social welfare. A VCG mechanism is efficient, incentive-compatible, and individual rational for participants with quasi linear utility functions (Parkes et al., 2001). However, (Myerson and Satterthwaite, 1983) proved that it is impossible to design an exchange, which is incentive-compatible, (interim) individually rational, and budget-balanced that achieves efficiency in equilibrium. In MACE, a VCG mechanism is efficient and individual rational, however, not budget-

[2] CPLEX is a commercial solver for optimization problems (`http://www.ilog.com/`).

balanced. In this case, the auctioneer has to endow the exchange, which is practically not realizable.

Relaxing the efficiency property, a possible implementation of a budget-balanced pricing rule for MACE is the *k*-pricing scheme. The underlying idea of the *k*-pricing scheme is to determine prices for a buyer and a seller on the basis of the difference between their bids (Sattherthwaite and Williams, 1993). For instance, suppose a buyer wants to buy a computation service for 5 and a seller wants to sell a computation service for at least 4. The difference between these bids is π=1, i.e. π is the surplus of this transaction and can be distributed among the participants. This schema can be applied to MACE and results in an approximately efficient outcome (Schnizler et al., 2006).

4.2. Bargaining mechanism

Having defined a formal model for using a central economic allocation mechanism in a Grid network, this section describes an alternative, decentral approach. The bargaining mechanism introduced here, implements the selection decision in the requesting client itself. Related realizations of decentral approaches are found in P2P Networks, where Gnutella (Adar and Huberman, 2000) is a typical example. An optimization of network performance is out of the scope of the clients behavior; in contrast, the selfish conduct of each peer leads to performance and congestion problems in the P2P network, which are principally hard to solve (Adar and Huberman, 2000). Gnutella uses a flooding algorithm for service discovery. The catallactic approach also uses flooding for decentral service and resource discovery.

In decentral matchmaking models, agents communicate directly with each other, decide on their own, and do not take the system state into account. In the Edgeworth process (Varian, 1994), economic subjects trade bilaterally with each other only if their utility is supposed to increase after the barter. In that case, the sum of all utilities increases after each successful barter; the final state is Pareto-optimal and has maximum system utility.

A theoretical fundament for how dynamic market processes, heterogeneous agents and choice under incomplete information work together can be found in Neo-Austrian Economics, in particular in Friedrich August von Hayek's Catallaxy concept (Hayek et al., 1989). Catallaxy describes a state of spontaneous order, which comes into existence by the community members communicating (bartering) with each other and thus achieving a community goal that no single user has planned for. The implementation of Catallaxy uses efforts from both agent technology and economics, notably agent-based computational economics (Tesfatsion, 1997).

An iterative bilateral negotiation protocol, similar to a contract-net, is used as no complete information is available (Smith, 1980). Both agents approximate to the trade-off point in iterative steps exchanging offers and counter-offers. This process is described as monotonic concession protocol (Rosenschein, 1994). If an agent receives an offer or counter-offer, it decides to either make a concession or send the same price as in the last negotiation until the negotiation ends with an accept or a reject.

After the negotiation, the autonomous agents adapt their negotiation strategies using a feedback learning algorithm. The learning concept used in this simulation is derived from so-called gossip learning. This means that the agents learn from received information about other transactions in the market. This information may not be accurate or complete, but serves as an indication about the gross direction of the market. In our implementation, this gossip information is created and broadcast by a successful agent, in analogy to issuing an adhoc information in stock market periodicals.

In economic simulations lots of research efforts on evolutionary algorithms can be found. We selected the STDEA (Smith Taylor Decentral Evolutionary Algorithm) (Smith and Taylor, 1998). The STDEA is a decentral evolutionary algorithm, which has no global evaluation metric (fitness value), used in genetic algorithms to separate the under performing participants (Goldberg, 1993). A fundamental quality of the mechanism is the decentral communication and fitness evaluation, using local available data. Every agent sends a plumage object after a successful transaction, advertising its average income (fitness) and its genes (genotype) to all agents of the population after an evaluation phase, i.e. after it has carried out a certain number of negotiations with this genotype. If an agent receives a plumage object from other agents, it decides using a blindness probability, whether the plumage object is evaluated, avoiding premature unification of the genotype. Sender and recipient remain anonymous. If a certain maturity threshold of received plumage is exceeded, the agent replaces his old genotype with the evolved version after the completion of evaluation, selection, recombination and mutation phases as in normal genetic algorithms. The mutation rate is also influencing the algorithm, which determines the frequency and the extent of explorative behavior of the population.

Ongoing communication by using price signalling leads to constant adaptation of the system as a whole and propagates changes in the scarcity of resources throughout the system. The resulting patterns are comparable to those witnessed in human market negotiation experiments (Kagel and Roth, 1995)(Smith, 1962)(Pruitt, 1981).

4.3. Overall simulation architecture

The CATNETS simulation environment must be able to simulate the behaviour of both the Catallactic distributed service/resource allocation mechanism and the auction-based central service/resource allocation mechanism described above. We adopt a system whose high level architecture is depicted in Figure 1. There are three main components:

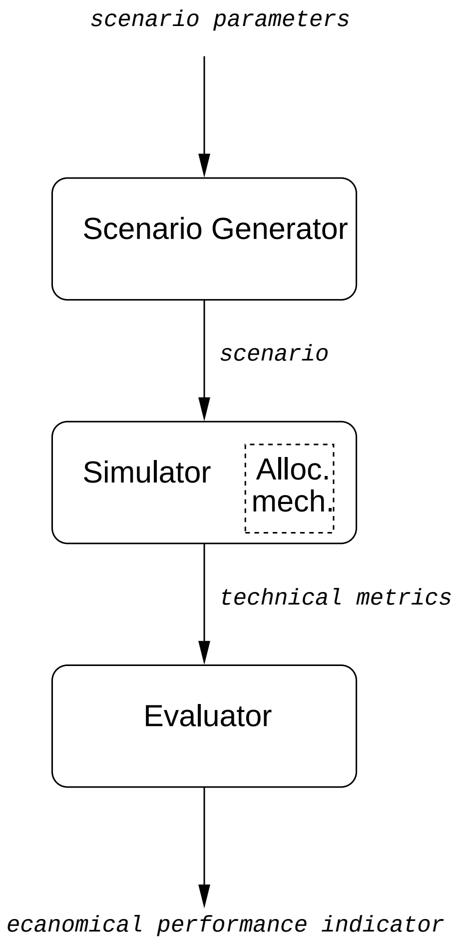

Figure 1. Architecture of the simulation environment.

Scenario Generator: This component takes a set of scenario parameters as an input and produces a scenario to be simulated as an output.

Simulator: It takes a scenario as an input and execute it by using either the auction-based or catallactic service/resource allocation mechanism. The output of a simulation is a set of technical metrics.

Evaluator: This component takes a set of technical metrics as an input and calculates an economical performance indicator for the allocation mechanism under observation.

This paper focuses on the high level description of the CATNETS Grid simulator. The evaluator component is presented in (Streitberger et al., 2007).

4.4. SIMULATOR

The CATNETS simulator is, based on OptorSim (Bell et al., 2003) (Cameron et al., 2004), a Data Grid simulator which was initially developed in the framework of the European DataGrid project. At the time of selecting the simulation environment, OptorSim gave the best opportunities for simulating the decentral allocation mechanism. OptorSim offered already parts of the requirements like a P2P communication layer, a scalable simulation core and an easy-to-use configuration. This significantly reduced the effort of implementing the CATNETS scenario into OptorSim. In (de Assuncao et al., 2007), a framework is presented, which extends GridSim with the CATNETS scenario. New allocation policies were introduced supporting the catallactic market mechanism and the communication was extended for supporting decentralised allocation mechanisms.

The general process of using the CATNETS Grid simulator passes a configuration step, followed by the simulation of the the scenario and finishes with an evaluation. Given as input: (1) a Grid network configuration (produced by a *Scenario Generator*), (2) a set of simulation parameters included into a configuration file, and (3) an allocation mechanism (*auction-based* or *catallactic*, the CATNETS Grid simulator runs a number of *Complex Services* on the simulated Grid network. During simulation it records metrics used for an off-line evaluation of the allocation performance.

This section gives an overview of the architecture of the simulator. Details are given on how the integration of the central and catallactic mechanisms into OptorSim has been performed.

4.4.1. *Simulator architecture*

The architecture of the CATNETS simulator is depicted in Figure 2.

The component at the top of the figure simulates a Grid system user who submits requests for the execution of Complex Services. The sequence of submitted requests is determined by a specific *pattern*, which is a parameter of the simulation. The component called *ComplexServiceDispatcher* performs the dispatching of complex service requests to Complex Service Agents (CSAs). Various dispatching policies are available.

The bottom of Figure 2 represents the simulated Grid system. In the simulator, Complex Service Agents, Basic Service Agents (BSAs), and Resource Agents (RAs) are implemented as threads. In every simulated Grid

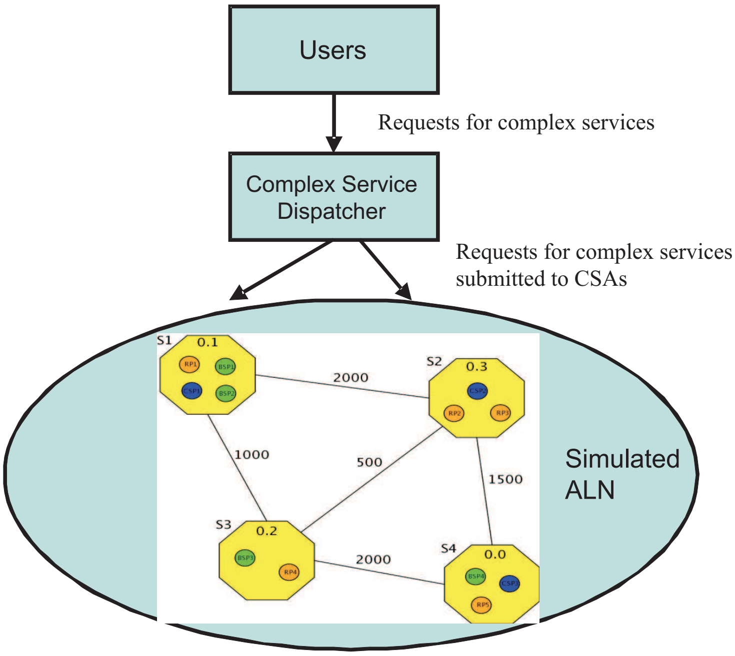

Figure 2. High level architecture of the CATNETS Grid simulator.

site there is also a components called *P2P mediator*, whose task is to manage the exchanging of messages between Grid sites. P2P Mediators are also implemented as threads.

The structure of the simulated Grid network differs depending on the allocation mechanism. When the central mechanisms is adopted, the structure is shown in Figure 3.

The Grid network includes a special site where only the central auctioneers are located. The acronym SMAA stands for *Service Market Auctioneer Agent*, while RMAA for *Resource Market Auctioneer Agent*. This site is fully connected to all other Grid sites and all the allocation requests for services or resources are sent to it via the P2P mediators.

When the catallactic decentral mechanism is simulated, this special site is not present in the Grid network. As shown in Figure 4, P2P Mediators perform a propagation of messages over the Grid network and agent interaction is likely to a "classical" P2P model.

4.4.2. *Agent behaviour*

The behaviour of users and agents in the Grid network depends on the adopted service/resource allocation mechanism.

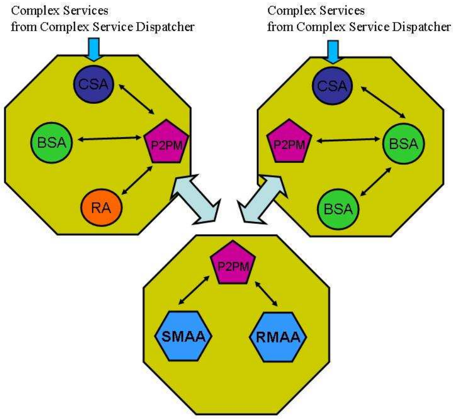

Figure 3. Grid network structure for central mechanism.

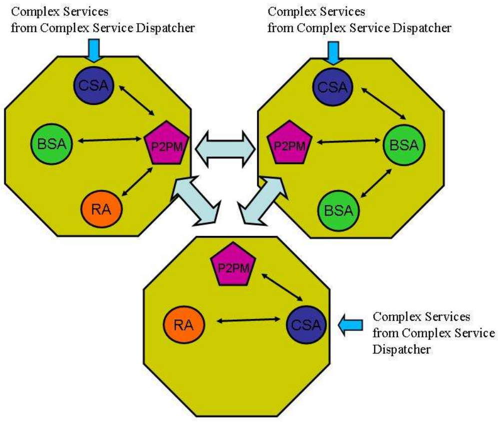

Figure 4. Grid network structure for catallactic mechanism.

Catallactic mechanism. When this mechanism is adopted the behaviour of Grid users is proactive while CSAs, BSAs, and RAs are reactive. The intended meaning of proactive is that users or agents are able to “take the initiative” in the interaction with other agents while by reactive we mean that they respond to external messages from other agents or users.

The behaviour of users and agents can be summarised as follows:

– Grid users submit requests for complex services to complex service agents.

- CSAs react to requests for complex service by issuing a sequence of requests for the involved basic services over the Grid network; they are also able to react to messages from BSAs while bargaining for basic services.
- The behaviour of BSAs is also reactive: they (1) respond to requests for basic services by possibly generating an offer for the requested basic service, (2) react to messages from CSAs while bargaining for basic services, and (3) issue requests for the associated resource bundle after a negotiation for a basic service ends successfully.
- The behaviour of RAs is reactive: They (1) respond to requests for resource bundles form BSAs by possibly generating an offer for the whole or part of the bundle and (2) react to messages from BSAs while bargaining for resources.

Centralised mechanism. When this mechanism is adopted, users and all agents in the Grid network except CSAs work actively. In both, the service and resource markets, the matching of requests and offers is performed by a dedicated central auctioneer.

- Grid users submit requests for complex services to complex service agents;
- CSAs react by submitting requests for the involved basic services to the centralised service auctioneer (SMAA);
- BSAs which want to sell a basic service send offers for this service to the centralised service auctioneer (SMAA). Moreover, when their service is successfully allocated, they submit requests for resource bundles to the centralised resource auctioneer (RMAA) in order to get the resources which are necessary for service provision.
- RAs send offers for resource bundles to the resource central auctioneer.

5. Evaluation of the Central and Catallactic Approach

This section presents a preliminary evaluation of the central and Catallactic allocation approach. The scenario generator produces a random network and distributes services and resources. This results in the simulation setup presented in subsection 5.1. The following subsection describes the simulation execution, whereas the last subsection discusses two selected metrics, allocation time and utility of the agents.

5.1. Simulation Setup

The scenario generator creates a Grid scenario described in section 3. The size of the network is set to 20 nodes, the failure probability is 0. The nodes and the network have 100% availability. The links between the nodes have an available bandwidth of 1GBit/s and the message size was set to 10kb. In the central allocation approach, the scenario generator randomly selects one node and places the auctioneers on this node. The remaining network and agent setup described below stays the same in both the central and Catallactic allocation mechanism.

A set of 10 agents is assigned to the network nodes using a uniform distribution. There is one CSA, which is able to provide the following set of complex types to the requesting client:

$$CS_1 \longrightarrow \{\langle BS_3\rangle\}$$

$$CS_2 \longrightarrow \{\langle BS_3\rangle, \langle BS_2\rangle, \langle BS_1\rangle\}$$

$$CS_3 \longrightarrow \{\langle BS_2\rangle, \langle BS_1\rangle\}$$

The Grid user randomly selects every 50ms one of these complex services and submits his request to the complex service agent using the complex service dispatcher. The complex service agent applies a sequential access policy allocating the set of basic services. This means, a request for complex service CS_2 allocates first BS_2 and second BS_1. The execution time each of the basic services is set to 100ms. This enables an isolated analysis of the central and Catallactic allocation approach without taking into account effects of varying execution times.

As described in the general scenario, a basic service buys resources on a resource market for its service execution. In this scenario, there are three basic services with its resource bundle needs:

$$BS_1 \longrightarrow \{\langle R_1, 28\rangle, \langle R_2, 12\rangle\}$$

$$BS_2 \longrightarrow \{\langle R_2, 21\rangle, \langle R_3, 1\rangle\}$$

$$BS_3 \longrightarrow \{\langle R_1, 77\rangle, \langle R_2, 23\rangle, \langle R_3, 35\rangle\}$$

For example, basic service BS_1 needs a resource bundle with resource R_1 with amount 28 and resource R_2 with amount 12. It is assumed, that this resource bundle has to be provided by one single resource service. An additional constraint is the exclusive resource usage. The resource service is not allowed to share his remaining resources to any other basic service. In detail, the list of available resource services are:

$$ARB_1 \longrightarrow \{\langle R_2, 90\rangle, \langle R_3, 78\rangle\}$$

$$ARB_2 \longrightarrow \{\langle R_2, 21\rangle, \langle R_3, 29\rangle\}$$

$$ARB_3 \longrightarrow \{\langle R_1, 86\rangle, \langle R_2, 84\rangle, \langle R_3, 92\rangle\}$$

$$ARB_4 \longrightarrow \{\langle R_1, 93\rangle, \langle R_2, 43\rangle, \langle R_3, 35\rangle\}$$

There are 6 resource agents in the network, two resource agents of resource type ARB_1 and ARB_2 and one resource agent for type ARB_3 and ARB_4. This enables a basic service to select between different resource providers.

5.2. Simulation Execution

The simulation execution is defined as a set of 10 simulation runs. In each simulation run, 50 complex services from the complex service list are requested. First, the allocation on the service market takes place and second the resources are allocated. If one allocation fails the whole complex service request terminates with a cancelation message.

The central auction mechanism uses two different clearing policies: a continuous double auction on the service market and a call market clearing policy with a clearing interval of 100ms on the resource market. The decentralized approach uses the same bilateral negotiation protocol and the same parameter setup on both markets. The timeout for selecting a proposal is set to 500ms.

We selected two metrics for analysis, which reflect the time needed for allocation and the agent's utility. In detail, the metrics are:

- Allocation Time: The allocation time metric is defined as the time needed until the basic service and resource are available for consumption by the requestor. In detail, the start time of the allocation is in the central case the time between submitting a bid to the auctioneer and receiving the message, which signals the end of the auction. In the decentralized case, the agent starts to measure the allocation time between submitting a call-for-proposal message and the termination of the bilateral negotiation.

- Agent's Utility: This metric measures the utility of each complex service agent, basic service agent and resource agent. The metric is defined for a seller (basic service agent on the service market and resource agent)

$$utility_{seller} = 1 - \frac{valuation}{price}$$

and for a buyer (complex service agent and basic service agent on the resource market)

$$utility_{buyer} = 1 - \frac{price}{valuation}.$$

Both buyer and seller don't trade below their individual valuation of the product. If they pay their valuation for the product, they don't make profit and their utility is 0.

5.3. Metrics Evaluation

Figure 5 and Figure 6 show the allocation times for trades on the service and resource market. The allocation times are averaged over every successful basic service allocation. The basic service is a complete standardized product with a constant execution time. Therefore, no differentiation between the different basic service types is made. Figure 5 compares the continuous double auction of the central allocation approach with the decentralized bilateral bargaining approach. The auction outperforms the Catallactic approach up to 35%. The computation of an outcome for the central allocation exhibits less overhead than the bilateral message exchange in the Catallactic approach. This results in lower allocation time for the central matchmaking. The mean allocation times of the different simulation runs show similar results. Both allocation mechanisms indicate stable mean allocation times on the service market.

In Figure 6, the central allocation approach on the resource market changes. Here, a multi-attributive combinatorial auction, which is able to allocate resource bundles, replaces the continuous double auction. The decentralized allocation approach remains the same (except some small adaption for handling resource bundles). Like on the service market, the decentralized approach shows almost the same allocation time. Also, the mean allocation times are stable between different simulation runs. Compared to the auction, the bilateral bargaining shows shorter allocation times in 60% of the experiment runs.

In these experiments, the computationally demanding central matchmaking on the resource market needs more allocation time on average. However, the central allocation approach is able to outperform the decentralized case in 40% of the simulation runs. The worst case in experiment 8 with 2000ms

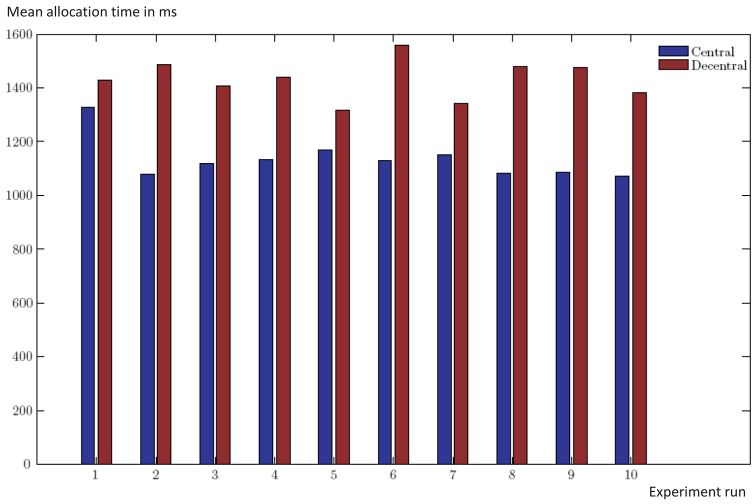

Figure 5. Mean Basic Service allocation time in milliseconds for 10 experiment runs. Compared mechanisms: Continuous Double Auction and Catallactic

and the best case in experiment 9 with 1000ms show a significant deviation between the simulation runs. One reason for this result is the number of messages transferred over the Grid network. The bilateral bargaining approach has to transfer a pair of messages in each negotiation round. Coming to an agreement usually takes 3 rounds on average. If the auctioneer in the central case is able to compute an fast outcome, the central approach outperforms the decentralized case in this scenario due to its less complex messaging.

Figure 7 and Figure 8 show the mean utility of the complex service agents and the basic service agents. The valuation of the traded good changes according to the market price. In the centralized approach, the market price is computed by the auctioneer, whereas the decentralized allocation approach can only estimate the market price using former experience and his learning algorithm. Both, the centralized and decentralized mechanism, adapt their valuation after every allocation using the market price (central) or the price signals they receive (decentralized).

The service buyer in the central approach shows a high utility between 0.35 and 0.48. He is able to buy below his valuation of the good. In the decentralized allocation approach, the complex service agent optimizes his valuation using his past experience and the evolutionary learning algorithm. The algorithm applies mutation to explore possible better outcomes, which lead to high utility deviation between the experiments.

Similar results show the service sellers (Basic Service Agents) in Figure 8. The utility levels between the experiments show stable values for the sellers in the central approach and high variance in the Catallactic approach. At the

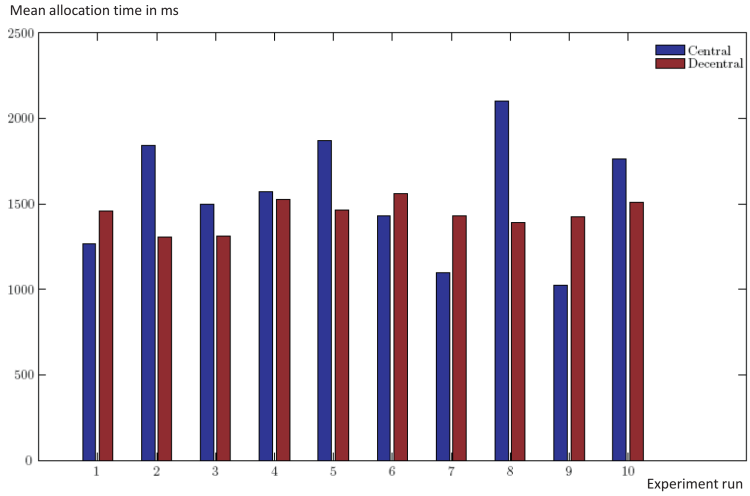

Figure 6. Mean Resource Service allocation time in milliseconds for 10 experiment runs. Compared mechanisms: Combinatorial Auction (central) and Catallactic (decentral)

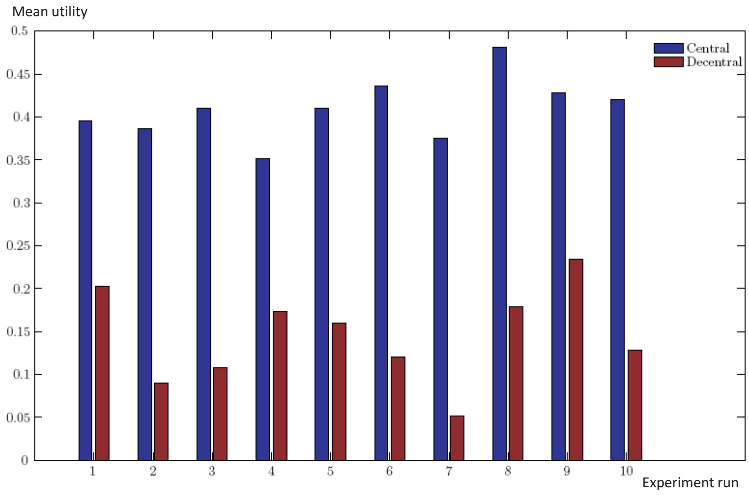

Figure 7. Mean Utility for 10 Complex Service Agents (service buyer) and 10 experiment runs. Compared mechanisms: Continuous Double Auction (central) and Catallactic (decentral)

current stage, the absolute values of the utility metric still need further examination. Future simulations runs will address this problem by increasing the number of Grid network sites as well as the number of resource and services available.

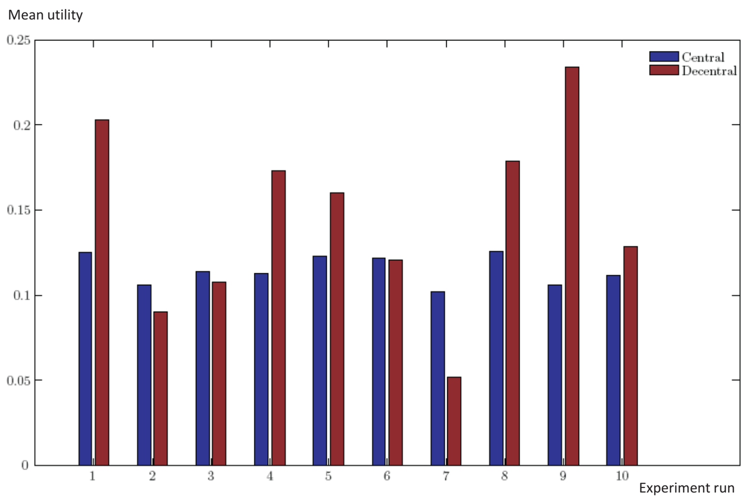

Figure 8. Mean Utility 10 Basic Service Agents (service seller) and 10 experiment runs. Compared mechanisms: Continuous Double Auction (central) and Catallactic (decentral)

6. Conclusion

This paper has introduced a framework for the simulation of service-oriented global computing infrastructures. Our Grid simulation model realizes two interdependent markets, a service and a resource market. On the service market, Complex Service Agents procure Basic Services to fulfil the complex demands of their human users. On the resource market, Basic Service Agents try to allocate Grid resources in order to operate their Basic Service offers. Basic Service Agents are thus buyers (of resources) on one market and sellers (of basic services) on the other. Any changes in prices or market structure will propagate, but it is also possible to investigate the market behavior in isolation.

On these markets, we have simulated the impact of different market mechanisms for resource allocation, a centralized and a decentralized approach.In the centralized case, the service market is designed as a double auction market. The resource market is carried out as a combinatorial exchange with a k-pricing approach. In the decentralized case, both markets are populated by agents executing bilateral bargaining using heterogeneous, heuristic strategies. These approaches are implemented into the same simulation framework, executed in several simulation runs and evaluated with two metrics, allocation time and a utility metric.

The lessons learnt from implementing and executing this Grid simulation framework are:

– A final statement on the performance of the Catallaxy approach to a centralized auction approach is difficult to obtain. A main problem lies in the technical aspects of the implementation. Catallaxy works best in large scale scenarios, but a sufficient simulation for Catallaxy needs larger technical resources. On the other hand, the simulation time needed for the centralized approach increases dramatically with growing simulation size. The largest practical simulation so far runs 200 agents.

– From an implementation perspective, even the moderate complexity of the heuristic bargaining strategy leads to a noticeable variance of the simulation results, when compared with the predictable results of the auctioneer's algorithm. The calibration of the simulation, using a working prototype, became an important task in the CATNETS project.

– The simulation framework includes an automated scenario generator, the simulator itself and an ex-post evaluation component. The automated scenario generator is necessary to generate the Grid scenarios. A manual scenario generation is practically impossible because of the high number of technical parameters related to Grid simulation as such, in addition to the parameters for the two interrelated markets which are the focus of our work.

– With regard to the Grid market parameters, we have achieved various possibilities for adaptation to real world settings. Virtualized resources and resource bundles are supported. It is possible to simulate dedicated and shared resource (co-)allocation. Also different quality levels of resources and services can be modelled. Bargaining models and continuous double auctions are both supported in the simulations models.

Regarding the two metrics measured, the higher number of messages exchanged in the bilateral bargaining approach leads initially to higher allocation times. The centralized auction implementation is a highly efficient clearing mechanism for small or medium scenarios, which can also handle bundled goods. However, when increasing the size of the simulation, the computation time of the centralized allocation approach increases. When the Grid network size reaches a certain threshold, the decentralized bargaining approach is more favorable, however, still at the expense of a larger number of messages needed. For the utility metric, our results do not show a clear picture yet, and further investigation is needed.

Acknowledgements

This work has partially been funded by the EU in the IST programme "Future and Emerging Technologies" under grant FP6-003769 "CATNETS".

References

Adar, E. and B. A. Huberman: 2000, 'Free riding on gnutella'. *First Monday* **5**(10).

Bell, W., D. Cameron, L. Capozza, A. Millar, K. Stockinger, and F. Zini: 2003, 'OptorSim - A Grid Simulator for Studying Dynamic Data Replication Strategies'. *Int. Journal of High Performance Computing Applications* **17**(4).

Biswas, S. and Y. Narahari: 2004, 'An Iterative Auction Mechanism for Combinatorial Logistics Exchanges'. In: *Proceedings of the 9th International Symposium on Logistics, 11.-14.07.2004, Bangalore, India*.

Buyya, R., D. Abramson, J. Giddy, and S. H.: 2002, 'Economic Models for Resource Management and Scheduling'. *Concurrency and Computation: Practice and Experience* **14**, pp. 1507–1542.

Buyya, R. and M. Murshed: 2002, 'GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing'. *Concurrency and Computation: Practice and Experience* **14**, pp. 1175–1220.

Cameron, D., R. Carvajal-Schiaffino, A. Millar, C. Nicholson, K. Stockinger, and F. Zini: 2004, 'Analysis of Scheduling and Replica Optimisation Strategies for Data Grids using OptorSim'. *Journal of Grid Computing* **2**(1), pp. 57–69.

Conen, W.: 2002, '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*.

Cramton, P., Y. Shoham, and R. Steinberg: 2006, 'Introduction to Combinatorial Auctions'. *Combinatorial Auctions, MIT Press 2006* pp. 1–14.

de Assuncao, M. D., W. Streitberger, T. Eymann, and R. Buyya: 2007, 'Enabling the Simulation of Service-Oriented Computing and Provisioning Policies for Autonomic Utility Grids'. In: *Proceedings of the 4th International Workshop on Grid Economics and Business Models (GECON 2007), 28.08.2007, Rennes, France*.

Eymann, T., M. Reinicke, O. Ardaiz, P. Artigas, L. D. de Cerio, F. Freitag, R. Messeguer, and D. R. L. Navarro: 2003a, 'Decentralized vs. Centralized Economic Coordination of Resource Allocation in Grids'. In: *Proceedings of the 1st European Across Grids Conference*.

Eymann, T., S. Sackmann, G. Mueller, and I. Pippow: 2003b, 'Hayek's Catallaxy: A Forward-looking Concept for Information Systems'. *Proc. of American Conference on Information Systems (AMCIS'03)*.

Foster, I.: 2005, 'Service-Oriented Science'. *Science* **308**, pp. 814–817.

Foster, I. and C. Kesselman: 1999, *The Grid: Blueprint for a New Computing Infrastructure*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Friedman, D.: 1991, 'The Double Auction Market Institution: A Survey'. In: D. Friedman and J. Rust (eds.): *The Double Auction Market - Institutions, Theories, and Evidence*. Cambridge MA, Perseus Publishing, pp. 3–26.

Goldberg, D.: 1993, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA, Addison Wesley.

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

Hayek, F., W. Bartley, P. Klein, and B. Caldwell: 1989, 'The collected works of F.A. Hayek'. *University of Chicago Press*.

Hayek, F. A. v.: 1945, 'The Use of Knowledge in Society'. *American Economic Review* **XXXV**(4), pp. 519–530.

Kagel, J. and A. Roth: 1995, *The handbook of experimental economics*. Princeton University Press.

Legrand, A., L. Marchal, and H. Casanova: 2003, 'Scheduling Distributed Applications: the SimGrid Simulation Framework'.

Ludwig, H., A. Dan, and B. Kearney: 2004, 'Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements'. In: *Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC 2004), 15.-18.11.2004, New York City, NY, USA*. pp. 65–74.

McAfee, R. and J. McMillan: 1987, 'Auctions and Bidding'. *Journal of Economic Literatur* **25**(2), pp. 699–738.

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

Myerson, R. B. and M. A. Satterthwaite: 1983, 'Efficient Mechanisms for Bilateral Trading'. *Journal of Economic Theory* **28**, pp. 265–281.

Parkes, D. C., J. Kalagnanam, and M. Eso: 2001, 'Achieving Budget-Balance with Vickrey-Based Payment Schemes in Exchanges'. In: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 04.-10.08.2001, Seattle, Washington, USA*. pp. 1161–1168.

Pruitt, D.: 1981, 'Negotiation behavior'. *Organizational and occupational psychology. New York: Academic Press*.

Regev, O. and N. Nisan: 2000, 'The POPCORN Market: Online Markets for Computational Resources'. *Decision Support Systems* **28**(1-2), pp. 177–189.

Rosenschein, J. S.; Zlotkin, G.: 1994, 'Rules of encounter - designing conventions for automated negotiation among computers'. *MIT Press, Cambridge*.

Sattherthwaite, M. A. and S. R. Williams: 1993, 'The Bayesian Theory of the k-Double Auction'. In: D. Friedman and J.Rust (eds.): *The Double Auction Market - Institutions, Theories, and Evidence*. Addison-Wesley, Chapt. 4, pp. 99–123.

Schnizler, B.: 2006, 'MACE: A Multi-Attribute Combinatorial Exchange'. In: N. Jenning, G. Kersten, A. Ockenfels, and C. Weinhardt (eds.): *Proceedings 06461 Negotiation and Market Engineering, Internationales Begegnungs- und Forschungszentrum (IBFI), 12.-17.11.2006, Schloss Dagstuhl, Germany*.

Schnizler, B.: 2007, *Resource Allocation in the Grid : A Market Engineering Approach*. Universitaetsverlag Karlsruhe.

Schnizler, B., D. Neumann, D. Veit, and C. Weinhardt: 2006, 'Trading Grid Services – A Multi-attribute Combinatorial Approach'. *European Journal of Operational Research, doi:10.1016/j.ejor.2006.05.049*.

Smith, R. E. and N. Taylor: 1998, 'A Framework for Evolutionary Computation in Agent-Based Systems'. In: *Proceedings of the Seventh International Conference on Intelligent Systems, 01.-03.07.1998, Paris, France*. ISCA Press.

Smith, R. G.: 1980, 'The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver'. *IEEE Transactions On Computers* **C-29**.

Smith, V.: 1962, 'An experimental study of competitive market behavior'. *Journal of Political Economy* **70**, pp. 111–137.

Song, H. J., X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K. Taura, and A. Chien: 2000, 'The MicroGrid: A scientific tool for modeling Computational Grids'. *Scientific Programming* **8**(3), pp. 127–141.

Streitberger, W., T. Eymann, D. Veit, M. Catalano, G. Giulioni, L. Joita, and O. F. Rana: 2007, 'Evaluation of Economic Resource Allocation in Application Layer Networks A Metrics Framework'. In: *Proceedings of 8th Internationale Tagung Wirtschaftsinfromatik (WI2007), eOrganisation: Service-, Prozess-, Market-Engineering, 28.02.-02.03.2007, Karlsruhe, Germany*. pp. 477–494.

Subramoniam, K., M. Maheswaran, and M. Toulouse: 2002, 'Towards a Micro-Economic Model for Resource Allocation in Grid Computing Systems'. In: *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering, 12.-15-05.2002, Winnipeg, Manitoba, Canada*.

Tesfatsion, L.: 1997, 'How economists can get alife'. In: *The Economy as a Evolving Complex System II*. Arthur, W.B. and Durlauf, S. and Lane, D.A. (eds.), pp. 533–564.

Varian, H. R.: 1994, *Mikrokonomie*. Oldenbourg.

Vickrey, W.: 1961, 'Counterspeculation, Auctions, and Competitive Sealed Tenders'. *Journal of Finance* **16**, pp. 8–37.

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

Wolski, R., J. Brevik, J. S. Plank, and T. Bryan: 2003, 'Grid resource allocation and control using computational economies'. In: F. Berman, G. Fox, and A. Hey (eds.): *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley & Sons.