The Catallaxy Approach for Decentralized Economic-based Allocation in Grid Resource and Service Markets

Oscar Ardaiz[1], Pau Artigas[1], Torsten Eymann[2], Felix Freitag[1], Leandro Navarro[1], Michael Reinicke[2]

[1] Computer Architecture Department, Polytechnic University of Catalonia, Spain

{oardaiz, partigas, felix, leandro}@ac.upc.es

[2] Institute for Computer Science and Social Studies, Albert-Ludwigs-University Freiburg, Germany

{eymann, reinicke}@iig.uni-freiburg.de

Abstract

Efficient resource allocation in dynamic large-scale environments is one of the challenges of Grids. In centralized economic-based allocation approaches, the user requests can be matched to the fastest, cheapest or most available resource. This approach, however, shows limitations in scalability and in dynamic environments. In this paper, we explore a decentralized economic approach for resource allocation in Grid markets based on the Catallaxy paradigm. Catallactic agents discover selling nodes in the resource and service Grid markets, and negotiate with each other maximizing their utility by following a strategy. By means of simulations, we evaluate the behavior of the approach, its resource allocation efficiency and its performance with different demand loads in a number of Grid density and dynamic environments. Our results indicate that while the decentralized economic approach based on Catallaxy applied to Grid markets shows similar efficiency to a centralized system, its decentralized operation provides greater advantages: scalability to demand and offer, and robustness in dynamic environments.

Keywords: Decentralized Allocation, Economic-based Allocation, Grid Markets

1. Introduction

Grids deal with the pooling and coordinated use of large sets of distributed and heterogeneous resources [1]. These resources can be computing power, storage of data or content, network bandwidth or sensor and actuators. A number of successful applications like Cactus [2] for distributed scientific problem-solving aggregate computational and data resources, or DataGrid [3] for analysis of distributed large-scale databases have demonstrated the feasibility of this approach.

Future Grid applications will have a very large and more complex resource pool combined with dynamic environments [4]. Efficient resource allocation in such future Grid applications requires more intelligent decisions than we have in today's systems; in particular, where the service is provided from and which computing resources should be used. It is known that delivery of computing or services based on planned resources, often based on worst-case scenarios, lead to poor resource utilization. Thus the need is for mechanisms which outsource resources in a dynamic manner, and therefore have the potential to achieve efficient on-the-fly provisioning of resources.

In this paper, we explore the use of the economic paradigm of the Catallaxy [5] as an approach of decentralized service provision targeting at scenarios of large and dynamic Grid environments. In Catallactic coordination, the decisions of the participants of the Grid are based on economic principles, where each participant's goal is to maximize his or her individual profit. Intelligent agents negotiate dynamically to provide service to the members of a virtual community. Awareness that resources like bandwidth, processing power and storage are limited is inherent to such a market based coordination approach.

The potential of the economic concept lies in the self-regulation of demand and supply, which is an emergent feature of market-based coordination mechanisms. The decentralized concept also achieves scalability of the resource allocation mechanism. Applying such a decentralized economic coordination is expected to lead to a system which is robust to dynamic distributed environments.

In this work our goal is to evaluate the performance of the Catallactic paradigm for coordinating the service provision in large-scale dynamic Grid markets. For this purpose, we have built a simulator that simulates the characteristics of diverse Grid scenarios and the underlying TCP/IP based computer networks. In simulation experiments, we evaluate the performance of the proposed Catallactic coordination of Grid markets, characterized in terms of node density and node dynamics, and compare its performance with that of a centralized baseline approach.

This paper is organized as follows: Section 2 outlines the motivation for the decentralized and economic approach. In section 3 we describe the characteristics of the Catallactic approach and its potential for resource allocation. In section 4 we describe our simulation framework. Section 5 provides the performance evaluation of Catallactic coordinated networks and the comparison with a centralized approach. Finally, section 6 presents the conclusions of this work and future outlook.

2. Decentralized and Economic-based Allocation approaches

Grids are concerned with the pooling and coordinated use of large sets of distributed and heterogeneous resources. These resources can consist of computing power, storage of data or content, network bandwidth and sensor or actuators. Grid applications enable users to be provided with services such as information search, content download, parallel processing, or data storage, by connecting a large number of distributed resources. In order to keep such a service operational and efficient, resource allocation mechanisms are required. Most of the current research relies on the existence of a centralized resource allocator. LEGION [6], Condor-G [7], and most Globus-based implementations [8] typically use a centralized matchmaker instance to evaluate the resource candidate list. The matchmaker instance selects the apparently optimal match from the list, according to global optimization considerations on computer load, network latency and usage, or storage space usage. The requesting client receives one singular matching partner. Clients and service providers update the centralized resource broker in a continuous frequency about their requests and effective availability. Performing resource allocation with a centralized coordinator, however, entails several difficulties. One of the requirements for the central coordination instance to decide correctly is that the environment should not change between the beginning and end of the control and assignment process. However the Internet is a very dynamic and fast changing system, and service demands and node connectivity changes are very frequent. A continuously updating mechanism would be needed to reflect the changes in the environment. A second difficulty of a centralized coordinator is that it requires global knowledge about the state of the system. However, as the scale of the systems grows the central coordinator becomes a bottleneck and a single point of failure.

Decentralized approaches are attractive for large dynamic systems, since with a high number of participants the time to centrally compute a solution increases. This is due to the fact that the solution space formed by the number of possible combinations grows. In decentralized approaches, problems are solved locally without having global information available. Decisions are independent from the instantaneous behavior of many other participants, since they are outside the scope of the individual participant. Peer-to-peer (P2P) applications connect a high number of end user application programs from frequently disconnected end user computers to achieve certain functionality without any global state or global coordinator, therefore resource allocation has to be decentralized. Some examples: Gnutella [9] and Freenet [10] provide file and content download, aggregating end-user storage capacity; SETI@home [11] or Entropia [12] facilitate highly parallel computational intensive applications, aggregating end-user CPUs power. However, P2P systems provide only one kind of application, which aggregate only one kind of resource at a time. In contrast, our target is Grid systems that aggregate large sets of distributed and heterogeneous resources.

Economic models are known to be mechanisms for regulating demand and supply. They give incentives to providers to share resources and encourage their consumers to use resources efficiently. In economic models, participants or agents own their resources, which they buy and sell from each other. Agents requiring resources are willing to pay for them. Communication between the agents is the means by which process are negotiated. Successful negotiations between the agents lead to the provision of services. Applying Economic concepts to allocating or scheduling resources in computing systems is not a new idea [13][14]. An early attempt at using economic ideas is Agoric Open Systems (AOS) [15]. AOS were defined as software systems that use market mechanisms for resource allocation; they encapsulate information, access paths and resources in objects traded by economic actor processes. Similar projects are Mariposa [16], Popcorn [17], and Spawn [18]. The basic problem can be characterized by having a number of processors, supplying computing power to a demand situation composed of computation jobs. The particular question is how supply and demand can be matched to each other if the situation on both sides is unclear. In closed environments, e.g. parallel computing, this question can usually be simplified, as the number of processors is fixed and the arrival of computational jobs is deterministic. Most of these approaches rely on using a centralized auctioneer and the explicit calculation of an equilibrium price as a valid implementation of the mechanism. In models with centralized components, collecting and assigning bids can create a bottleneck when operating in a large-scale system. Another point is that centralized components can represent a single point of failure.

Recent research in Grid computing has also recognized the value of price generation and negotiation for resource allocation and job scheduling, and in general the use of economic models for trading resources and services in increasingly large-scale and complex Grid environments. Nimrod/G Resource Broker [19], G-COMMERCE [20] DataGrid Optorsim study [21] are examples given for those grid economies. However these systems implement centralized approaches such as auctioneers or comparable electronic marketplace instances [22][23], which collect bids and offers from the Grid nodes, and match supply and demand in the same way as a stock market mechanism does. For example, G-COMMERCE [20] comprises agents (producers and consumers), the commodity objects (tagged with a price), and a centralized institution called "The First Bank of G". This institution combines a "tâtonnement" [24] sequential auctioneering approach with a polynomial method for finding general market equilibria. The price-setting of the individual producers and consumers uses local knowledge about the resources, and the single-variable utility functions are expressed in budget units only.

Research is being carried out to improve P2P systems by applying economic concepts: The computer power market [25] proposes a market-based resource management system for P2P computing. However they their resource traders are based on centralized grid economic based brokers which cannot scale. The PeerMart [26] approach implements double auctions (based on well-known economic mechanisms [27][28]) to match client request with file hosting nodes. Such double auctions occur on top of a P2P overlay network to distribute broker load of an otherwise centralized auctioneer onto clusters of peers, each being responsible for brokering a certain number of goods. Evaluation of such mechanisms is on-going work.

3. The Catallactic approach applied to Grid markets

3.1 Motivation for using the Catallaxy paradigm

Catallaxy is an alternative word for “free economy” Hayek’s notion of the Catallaxy [29]. The proven ability of a free-market economy to adjudicate and satisfy the conflicting needs of millions of human agents recommends it as a decentralized resource allocation principle. The Catallaxy concept is based on the explicit assumption of self-interested actions of the participants, who try to maximize their own utility and choose their actions under incomplete information and bounded rationality. The term Catallaxy comes from the Greek word "katallatein”, which means, "to barter” and at the same time, "to join a community.” The goal of Catallaxy is to arrive at a state of coordinated actions, the "spontaneous order”, which comes into existence through the bartering and communicating of the Community members with each other, thus achieving a community goal that no single user has planned for [29]. Catallaxy is opposed to “plan economy” where a central entity has global knowledge of the system and commands all entity decisions. In Catallaxy no entity can have global knowledge. In fact a central presumption is "constitutional ignorance”, assuming that it is impossible (and incurable) to know everything. In Catallaxy competition is the norm: entities are selfish and work in their own interest to gain income. In Catallaxy coordination among entities occurs as a result of price signals sent among entities when negotiating for goods. Another interesting property of Catallaxy is that it also coordinates interrelated markets. In real world market economies end-users or clients request a service or manufactured product from an intermediary agent such as a wholesaler or service provider. The wholesaler or service provider purchases or manufactures the desired product with resources or goods from the manufacturer. The price for the client charged by the service provider includes the benefit added by the service provider for elaborating the product. Also, clients do not normally have direct access to the manufacturer, but purchase from service providers.

Some of the requirements for Catallaxy to appear in a system are [30]: that agents work in their own interest to gain income; that they subjectively weigh and choose preferred alternatives, and that they have access to at least one markets. A central presumption is "constitutional ignorance”, assuming that it is impossible (and incurable) to know each and every circumstance that determines the agent's action. Participants communicate using commonly accessible markets, where they barter about access to resources held by other participants. Whether they are increasing or decreasing, the development of prices for specific goods leads buyers to look for alternative sources of procurement and thus enhance the dynamics of the market. Note that a market here is nothing more than a communication bus – it is not a central entity of its own, which collects all information and matches market participants using some optimization mechanisms, which would contradict "constitutional ignorance”. If there are different system agents which can constitute different markets then Catallaxy can hide one market from others while achieving coordination.

Catallaxy can be applied to digital economies [5]. So far Catallaxy has been applied to digital economies where goods being sold and bought in a digital market are real life objects: wood, boards, panels and tables. In this work we show that Catallaxy can be applied to Grid markets where computational resource and services are bought and sold.

Catallaxy can be implemented with multi-agents systems software technology [5]. Software agents can act selfishly as their human owners do, maintaining their private strategy and taking autonomous decisions, they can communicate directly, bypassing centralized control institutions and they can access markets by implementing some location or discovery mechanisms.

3.2 Catallactic Agents and Mechanisms

Catallactic agents follow the economic goal of profit maximization. They try to buy input goods for less and to sell output goods for more. Depending on the current market situation, its equity, and stock, the agent decides autonomously which action to take next – whether to buy, sell, produce, move or self-terminate.

The agent lifecycle "follows the money": if the agent has finished goods in stock, it tries to sell. If the agent has no goods, but input factors in stock, it simulates the production of output goods (by waiting an appropriate length of time). If the agent has no input factors in stock, it tries to buy some. If the market situation is completely satisfying , e.g. if there are no offers or demands within a certain time span, the agent tries to move to another marketplace. If the agent has spent its entire budget or all marketplaces are shut down, it has to terminate – every few milliseconds the agent has to pay utilization fees to the market anyway, so doing nothing is never a rewarding strategy.

In the case of buying or selling, the software agent goes through the three stages of a market transaction: information, agreement, and settlement [31]. In the information phase of any transaction, a buyer or seller has to identify his or her potential trading partners. The buyer agent initiates the agreement phase by communicating with a supplier. Both software agents negotiate using a monotonic concession protocol [32], where propose and counter-propose messages with subsequent price concessions are exchanged. If the negotiation process is successful, both agents will reach a compromise price agreement; otherwise, someone will sooner or later drop out of the negotiation. In this event, the agents will restart with other partners. In the final settlement phase, the transaction is carried out and monitored. Sellers and buyers exchange goods and money, respectively.

To maximize the spread between input and output prices, and thus its utility, the agent follows a certain negotiation strategy. Comparable automated negotiation efforts in multi-agent systems can be found in the research context of agent-mediated electronic commerce [33; 34] and market-oriented programming [35]. Human negotiation uses parameters such as demand level, concession, and concession rate [36]. A bargainer's demand level can be thought of as the level of benefit to the self associated with the current offer or demand. A concession is a change of offer in the supposed direction of the other party's interests that reduces the level of benefit sought. Concession rate is the speed at which demand level declines over time. In the real world, the values are influenced by determinants such as expectations about the other's ultimate demand, position and image loss, limit and level of aspiration and time pressure. Electronic agents use an adaptive strategy based on a stochastic finite state automaton, in which action paths are taken depending on stochastic probes against certain internal parameters. A combination of six variables, collectively called the Genotype, with values from a continuous value range between 0 and 1, describes the strategy. As an example, the acquisitiveness (p_acq) defines the probability of maintaining the agent's own position. Whether an agent concedes in an actual negotiation step, is subject to a stochastic probe against this parameter – the lower the acquisitiveness value, the higher the average concession rate. If the agent concedes, the In-negotiation delta price change (del_change) parameter calculates the amount of the price concession between two negotiation steps. Both partners calculate a percentage from the price difference of their original offers. If the buyer has del_change = 50% and the supplier del_change = 0%, the agents will reach agreement after two negotiation steps at the initial price of the supplier, if no agent drops out. To maximize their income, the agents will try to raise their initial demand level between different negotiations by the value of Pre-negotiation delta price change (del_jump). The Satisfaction parameter (p_sat) determines if an agent will drop out from an on-going negotiation. The more steps the negotiation takes, or the more excessive the partner's offers are, the sooner the negotiation will be discontinued. Effectively, this parameter creates time pressure. Information from past performance in earlier successful and unsuccessful negotiations computes into a subjective market price for each agent, which modifies the parameter Memory using a weighted exponential average with weight w_memory (w_mem).

3.3 Description of Grid markets

If nodes participating in a Grid are utility maximizing entities then a Grid economy appears. The Grid economy we consider in this study consists of three types of entities: resource providers, service providers, and clients. The resource providers sell resources, which can be bandwidth, storage, CPU cycles. Today, Grids are assumed to consist of a number of resource providers, whose clients are resource consumers, in the sense that clients make use of raw computational, network or storage resource to execute their private application code with their private data. After the OGSA Open Grid Service Architecture was proposed [37], service providers are presented as new actors in Grids. Service provider aggregate raw computational, network and storage resources from resource providers, and compose a service with a generic functionality demanded by several clients. The specified resource requirement depends on the given services. There could be simple services like a PDF conversion service [38], a data mining service or a whether-forecasting service, and services with a complex specification including several dependent tasks. Some services will be very specialized using proprietary algorithms, and thus will be provided only by a single service provider. But many other services will be provided by several service providers. The client requests a service from the service provider and is willing to pay for it, since the service represents a value for the client.

In such a service provision scenario, two electronic markets are operated: one market is for resources and the other market is for services, Figure 1. Each market has buyers and sellers. At the service market, service providers sell and clients buy service instances. At the resource market, resource providers sell, and service provider buy resources. In economic terms, the separation of the market for resources and services means that the end-user knows neither the market for resources nor the prices for the resources being purchased and combined to deliver a service. The resources, on the other hand, do not know the prices of the service sold to the end-user. The service provider, however, negotiates in both markets: It negotiates with the resource providers and with the end-users to sell a service.

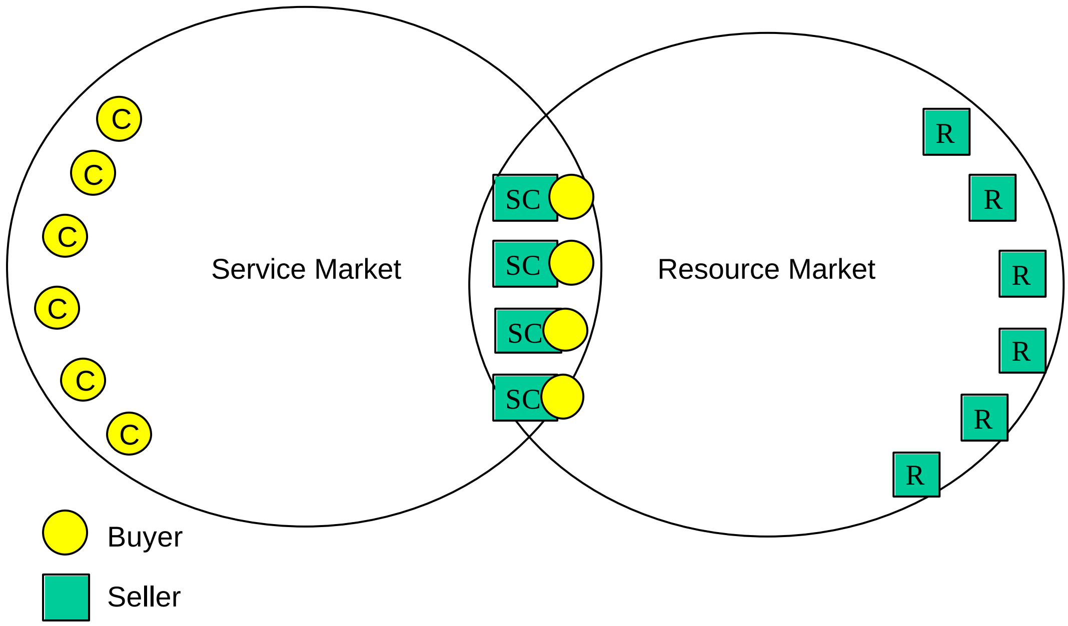

Fig. 1. Grid service and resource markets. Service clients (C) buy services at the service market from service providers (SC), different service providers can sell the same service type. Service providers (SC) need raw computational, network and storage resources to provide its service. Service providers (SC) buy resources at the resource market from resource providers (R).

3.4 Catallactic Agents and Mechanisms applied to Grid markets

A Grid market can contain one of the following three types of catallactic agents:

1) Client (C): An agent behaving as a client, or consumer, of a service. As such it needs to access the service, use it for a defined time period, and then continue with its own program sequence. The functionality required by the clients is a generic service. Clients request services, which can be the PDF conversion service, frequency analysis service, data-mining services, etc. Client agents incorporate a Catallactic agent with an internal strategy. Such an agent must be capable of discovering service provider agents and negotiate with them.

2) Service-copy (SC): An agent behaving as a service copy offers a service as an intermediary. Service-copies offer the service to requesting clients. Service-copies agents represent an autonomous instance of a service provider, but several different service-copies can provide the same service, therefore there is a market where clients can negotiate. Several service-copies provide the same service. However they act autonomously when negotiating with clients. Service copies do not own the resources which are needed to provide the service. Service-copies must be allocated a resource to be able to provide the service. Service-copies buy resources at the resource market, where they act autonomously when negotiating and purchasing resources in the resource market. Each service-copy agent has an internal strategy. Since service-copy agents negotiate in both markets, its strategy must aim at maximizing profit in both markets simultaneously.

3) Resource (R): An agent, the owner of a resource. This resource, for instance, can be storage or processing power. Resource agents act as the owners of resources, which can be bandwidth, storage or CPU cycles.

The self-interest of the clients is to access a service at the lowest cost and/or the fastest time. In an environment where services have to be paid for access, the utility gain of clients is the difference between their private value (of what the access is worth) and the actually paid transaction price. Service-copies strategy must attempt to provide access to clients, such that a minimum number of service demands have to be rejected. A valid assumption is that service providers can charge for the access to a service. In this case, the service provider receives a charge price from the client. Service-copies are interested in increasing revenue by either increasing turnover (more service accesses in the same time) or profit (more profit by each service access). Lastly, resources incur costs, and the resource agents aim to fill these costs and to make profits by increasing the usage of the resources. As more resources are used, resource agents could obtain more physical resources or they can increase the price paid by the service provider.

Client agents require a discovery mechanism to locate service copies, also service-copies require a discovery mechanism to locate resources. There exist different service and resource discovery mechanisms in real systems with different properties: in Grids there exist distributed index services such as Globus MDS [39], in peer-to-peer networks there exist more decentralized mechanisms such as Gnutella's flooding [9] or distributed hash tables [40]. Any discovery mechanism could work in some situations. However for Catallaxy, as a totally decentralized approach, discovery mechanisms that are decentralized are preferred.

The negotiation and money flow in the catallactic coordinated Grid market is shown in Figure 2. A client who has discovered a service-copy that can provide the service of its interest, initiates a negotiation with such service-copy. If the service-copy does not have any resource allocated, the service-copy has to locate an appropriate resource agent and initiate negotiations with the resources to obtain the provision of resources. After successful negotiations, the service-copy offers the service to the client. If the client accepts, then the service-copy provides the service by means of the contracted resources.

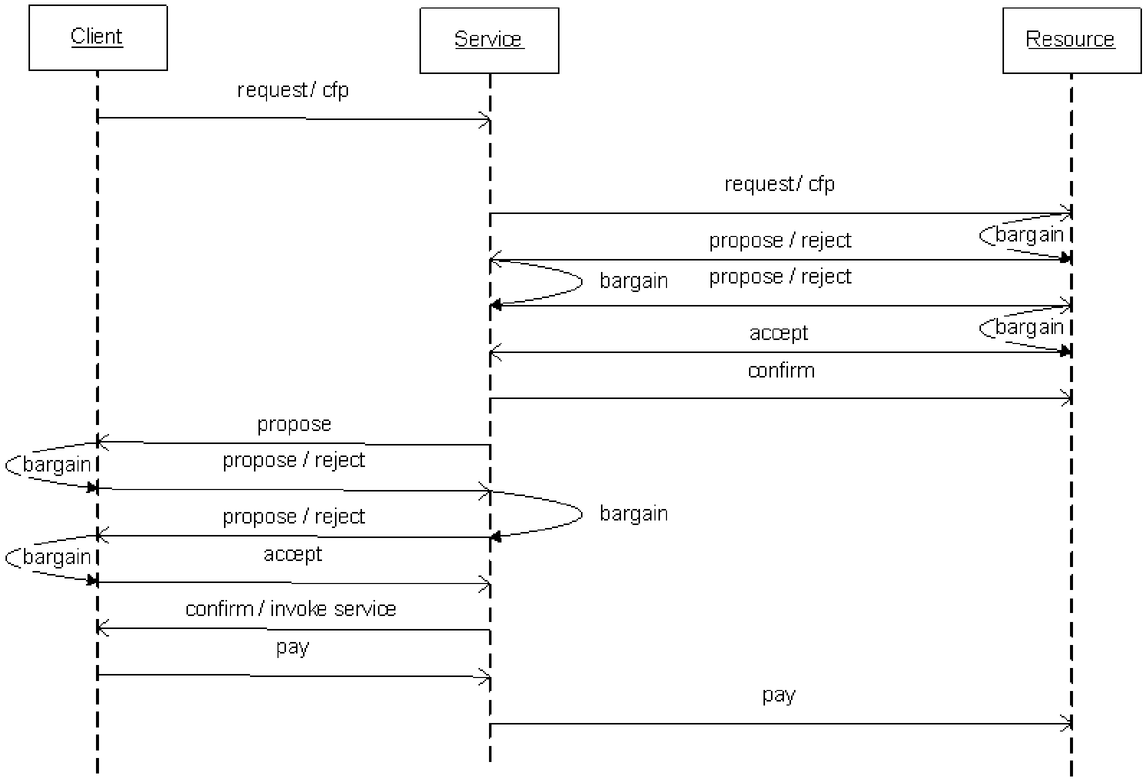

Figure 2. Catallactic agents direct negotiation mechanisms in Grid markets. A catallactic client agent negotiates for service access with a catallactic service agent, which in turn has to negotiate for resource provision with catallactic resource agent. After successful negotiations, the service-copy offers the service to the client.

4 Experimental framework

We wish to evaluate how well the Catallactic coordinated Grid markets achieve service provision to the requesting clients. We use simulations because we are interested in evaluating the behaviour in different Grid scenarios. The main goal is to measure the allocation efficiency of the Grid coordinated by the Catallaxy paradigm. A second goal is to compare the obtained results with that of a centrally coordinated baseline system.

4.1 A Simulator for Grid, agents and interaction mechanisms

The simulator must meet three main requirements: first, it must simulate grid resources and service allocation mechanisms as well as client requests, but it does not need to simulate client access to services. Second, it is necessary to simulate autonomous entities with individual strategies and capable of taking autonomous decisions and interacting among them; therefore the simulator must be able to support multiple concurrent agents and inter-agents message passing mechanism. Finally, since we are interested in observing the behaviour of the negotiations in a Grid environment, the simulator must simulate an Internet network topology and its communications characteristics.

We have built a simulator that models grids on top of the J-Sim simulator [41]. The J-Sim simulator is build based on the Autonomous Component Architecture (ACA), which enables simulation of multiple concurrent autonomous entities that can communicate with each other directly, therefore providing for the second requirement. Grid resource allocation mechanisms are also easily implemented in such architecture. Furthermore, it provides a module for simulating Internet communications links and topologies. Our simulator allow different types of agents to be created, which form Grid markets and implement application layer protocols. On a lower network level, J-Sim simulates a general TCP/IP network. Thus, our simulator experimental studies of different resource allocation mechanisms at the application level can be conducted while realistic values of variables that depend on network characteristics can also be measured.

4.2 Configuration of Grid markets and catallactic agents and mechanisms

A physical network topology was configured in the simulator with characteristics similar to the Internet: a number of core nodes highly connected in a mesh topology by high capacity long latency links (these connections represent international links, each country is usually connected to several countries forming a mesh), each core nodes is connected to a hierarchy of edge nodes by links with less capacity and less latency (in each country one or several operators run a network with a hierarchical topology of national and local links).

Agents are instantiated on each node of the network. Agents are one of the three types described previously (Client, ServiceCopy, or Resource). Depending on the particular experiment, a node may contain several agents or none at all. In the latter case, the node acts simply as a network router. The number and location of resource and service-copy agents varies according to the density setting of the experiment, which will be explained later.

Resource agents represent a individual machine or a cluster of machines providing computational and network resources. A resource unit is equivalent to a computational capacity of 1 Gflops and a network capacity of 1 Mbits. Resource agents simulate resources nodes with a capacity ranging from 60 resource units (i.e. A high capacity computational center with 60 Gflops computational power and 60 Mbits Inet conectivity) to 4 resource units (i.e. a low capacity server with 4 Gflops computational power and 4 Mbits Inet connectivy). Resource agents simulating high capacity computational centers are situated at highly connected core nodes in the physical topology similar to real computational centers. Small capacity resource agents are placed at edge nodes. All resource agents sell equal types of resources. Resource agents were configured to sell each resource unit at a minimum price of 20 monetary units and at a maximum price of 40 monetary units. Values of the internal strategy parameters of each resource agents are as follows: concession probability, p_acq, is set to 0.75; in-negotiation delta price change, del_change, is set to 0.5; pre-negotiation delta price change, del_jump, is set to 0.15; continuation probability, p_sat, is set to 0.75; and weight of previous offers, w_mem, is set to 0.3. This values are taken from a previous study of the Catallaxy [44] were those values were found to be good in a different electronic market scenario. There may be other values that give better results in Grid markets, and we plan to conduct further research to show the effect of varying internal agents strategy values.

Service-copy agents represent an instance of a service. Three different services, with different service ID, are provided by different service-copies. Several service copies provide the same service, with the same service ID. Service copy agents do not simulate service functionality, they only negotiate with resource agents to obtain those resources required to provide a service, and negotiate with clients to provide their service. In the simulator, each service copy requires a resource unit to provide service to each client request, a computational capacity of 1 Gflops and a network capacity of 1 Mbits. There are many real services that require such resource levels, e.g. a media transcoding service requires high network capacity to upload a video to be transcoded, and transcoding applications are computationally intensive, or a data-mining service requires high network capacity to upload databases to be data-mined, and data-mining applications are computationally intensive. Initially a service copy is installed at each node in the topology where resources are available. We do not consider that service-copies could migrate or replicate towards resources where they are not previously installed. Service-copy agents are configured to try to obtain a profit for each service unit sold of: a minimum of 20 monetary units and a maximum of 30 monetary units.

Values of the internal strategy parameters of each service-copy agents are similar to resource agents: concession probability, p_acq, is set to 0.75; in-negotiation delta price change, del_change, is set to 0.5; pre-negotiation delta price change, del_jump, is set to 0.15; continuation probability, p_sat, is set to 0.75; and weight of previous offers, w_mem, is set to 0.3.

Client agents are distributed on the outer leaves of the topology in every experiment. Client agents simulate service demand requests. Client agents are configured to buy each service unit at a maximum price of 50 monetary units; its expected minimum value is set to 23 monetary units. Values of the internal strategy parameters of each client agents are similar to other agents: concession probability, p_acq, is set to 0.75; in-negotiation delta price change, del_change, is set to 0.5; pre-negotiation delta price change, del_jump, is set to 0.15; continuation probability, p_sat, is set to 0.75; and weight of previous offers, w_mem, is set to 0.3.

The discovery mechanism simulated is a flooding distributed search mechanism such as those implemented in Peer-to-Peer networks as Gnutella [9]. Clients send requests for services with a hop limiting counter; requests for services are forwarded among agents that decrease such a counter, until it reaches a null value. The hop counter is configured with a maximum value of 5. This discovery mechanism is simple to implement but may generate a large number of messages in topologies with high degree nodes as in many realistic networks.

Each negotiation is implemented as a contract-net protocol [42]. Buying agents, either clients or service-copies, negotiate with the first selling agent that is discovered, since it should provide a good service or resource quality because of its network proximity. If such negotiation fails they continue negotiating with the second selling agent discovered. Negotiations only start after a demand request for a service arrives, we do not consider negotiations to pre-allocate resources before clients demands. We do not simulate bundle transactions where buyers buy resources transactionally from several providers, but such a possibility can be accommodated easily within the framework.

4.3 Baseline system for comparison

In the simulations, we wish to compare Catallaxy with a centralized baseline system which should behave near to the optimal in some scenarios because it aims at obtaining total knowledge of the system. Therefore the entities configuring the baseline system and its interactions are those shown in Figure 3. A centralized instance called the Master-service-copy (MSC), receives all requests and is informed by all resources and service copies of its availability. The MSC decides which resource and service-copy provide the service to each client request, and sends back an accept message to the client. The resource allocates the required resource units and the service-copy provides the service to the Client. The MSC is located in the central node of the topology. So as to compare both systems with economic parameters, the MSC implements an economic-based centralized allocation model based on economic double auctions [27][28]. The centralized auctioneer computes an equilibrium price every 50 msec. with requests and bids arriving from every party. Those client requests which have offered a price above the equilibrium prices are accepted, others are rejected.

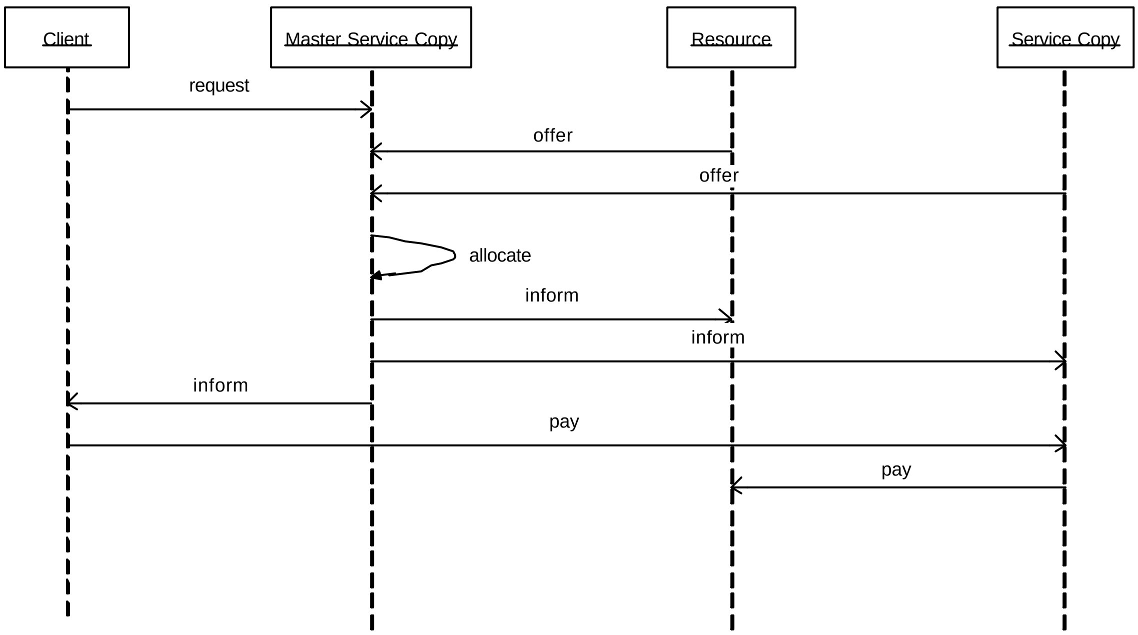

Figure 3. Baseline system: A master service copy (MSC) receives request and offers, allocates and informs clients, resources and service copies of its decision.

4.4 Grid scenarios for evaluation

Our goal is to evaluate the Catallactic decentralized allocation approach. We expect Catallaxy to maintain good performance at low density P2P Grids and at highly dynamic Grids, in contrast to centralized approaches that cannot scale and adapt to changes. Thus by simulation we aim to explore the design space of Grid along two dimensions: Grid density and Grid dynamics.

Grid density measures the concentration of Grid resources. The highest density is reached if only one service/resource providing node in the whole grid exists (with all capacity concentrated in it, i.e. a high performance computing center). The lowest density has every grid node providing the described service to others with small capacity (i.e. P2P Grid systems). Current Grids correspond to intermediate values of density. We will simulate Grids with different densities. A high density Grid formed by several high performance clusters situated at highly connected core nodes in the physical topology similar to real computational centers (HPC Grid). A low density Grid will contained a higher number of small capacity nodes placed at edge nodes of the network topology. (P2P Grid). To compare results among diffent scenarios the total Grid capacity will be the same in all experiments: 300 resource units, i.e., the total Grid capacity is 300 Gflops and 300 Mbits. To decrease node density between different scenarios, the number of nodes must be increased so that the total Grid capacity is the same in all simulation scenarios. Resource units are distributed uniformly among nodes, but at lower density scenarios resource nodes are place in locations nearer to the edge of the network: in the high density experiments each node can provide 60 service units each, and there are 5 service-copy and 5 resource agents. In medium-high density each node can provide 30 service/resource units each, and there are 10 service-copies and 10 resources. In medium-low density each node can provide 12 service/resource units each, and there are 25 service-copies and 25 resources, and in the low density scenario, each node can provide 4 service units each, and there are 75 service-copies and 75 resources nodes.

Node dynamics measure the continuous availability of service/resource-providing nodes in the grid. Low dynamics mean an unchanging and constant availability (ASP or HPC centers); high dynamics are attributed to a grid where nodes start up and shut down with high frequency (case of end-user dial-up PCs). The values of node dynamics are varied in the different simulations. During the experiments, the availability of each resource changes every 200ms with a certain probability, where this probability starts at 0% in the static case, and increases in steps of 10% up to 30% for the highly dynamic case. Node dynamics simulate the availability of service/resource-providing nodes in the grid. Availability of resources represents resource nodes which become unavailable due to network disconnections because they use a dial-up connection or a wireless connection. Low dynamics mean an unchanging and constant availability resource nodeas in the case of computational and data center; high dynamics are attributed to a grid where nodes start up and shut down with high frequency as in the case of end user dial-up or wireless computers.

4.5 Demand Workload

Demand traces are generated by a random demand trace generator. The demand trace generator is configured to produce traces whose requests are generated with an average inter-request time of 25, 50, 100 or 200 milliseconds, i.e. a demand load of 40, 20, 10 and 5 requests per second respectively. Demand requests are randomly assigned among 75 client nodes. Each generated client request will request one service unit. One service unit requires 1 Gflops computational power and 1 Mbits bandwidth capacity. The service ID of each request corresponds to service IDs of existing service copies. Finally each request has an average duration of 5 seconds i.e. service copies and corresponding resources are allocated for 5 seconds.

5. Performance evaluation

5.1 Measurement criteria

In this study, we apply both technical and economic parameters for the evaluation. Technical parameter are usually interesting only to some entities of the system, i.e. communication cost is important for network operators, whereas economic parameters can measure how well all participants of the system fulfil their interest. The technical parameters we evaluate are resource allocation efficiency (RAE), response time (REST), and communication cost (CC). The economical parameter we evaluate is social welfare (SWF).

Resource Allocation Efficiency (RAE)

Resource Allocation Efficiency (RAE) is our main technical criterion. It indicates the ratio of service demands, for which the Grid provides a service, to all sent service demands. In other words, it measures how many requests are allocated a service copy which has been allocated resources. A higher RAE is deemed to be better both for the client agent and for the Grid as a whole. However, this parameter does not capture the cost nor the effects derived from the service provision.

Response Time (REST)

The Response Time (REST) measures the time from when a request is produced in a client until the request service instance has been granted. It does not depend on the type of service, since service time is included. It is only influenced by the necessary mechanisms to establish a service session between client and service, and by the physical network response time. For comparing different coordination mechanisms, a lower average REST is considered to be better.

Bandwidth utilization/Communication Cost (CC)

This parameter measures the cost of the control messages sent in the network to provide a service. In our simulator, we use different message types to fulfil the negotiation protocol. To assess the bandwidth utilization, however, we treat the message size as a constant which is the same for all control message types, and use a global hop counter, which computes the total number of hops needed for control messages to complete the service provision for a certain demand trace.

Social Welfare (SWF)

Social welfare utility (SWF) measures how well all participants of an economic system can maximize their individual utility. Basically, utility measures the fulfillment of self-interest of the participants. The equation for SWF in its easiest form sums up the individual utility profits u_i:

$$U_{SWF} = \sum u_i \qquad (1)$$

Individuals are everyone who participate in the economic environment. For a Grid, the individual players are the clients (Clients which demand and pay for service access), the service provider instances (ServiceCopies which offer access and receive payments) and finally the resources providers (Resources which offer bandwidth and storage to the service instances). Thus, the equation can be written as:

$$U_{SWF} = \sum u_i^C + \sum u_j^{SC} + \sum u_k^R \qquad (2)$$

The self-interest of the clients is to access a service at the lowest cost and/or the fastest time. Utility can thus be measured either using costs or time. In this work we consider that services must be paid for access, and services are allocated enough resources to be accessed in a set time (if service are not accessed in that time the Service Level Agreement must be considered as unfulfilled). The utility gain of clients is the difference between their private value of what the access is worth (p), and the actually paid transaction price ($v_{i,p}$):

$$u_i^C = v_{i,p} - p \qquad (3)$$

The self-interest of the service provider is always to provide access to some service instance. Each client request for a service requires one service copy to buy resources at the lowest cost from resource providers. Like the clients, the service providers also have a private value for service access (p^{sa}) . In addition, there is private value for buying resource access from the hosting node (p^{ra}), therefore the utility of the service provider is the utility gain in the service market plus the utility gain in the resource market:

$$u_j^{SC} = (p^{sa} - v_{p,j}^{sa}) + (v_{p,j}^{ra} - p^{ra}) \qquad (4)$$

The self-interest of the resources providers is to make profits by increasing the usage of the resources. The utility gain of resources is the difference between the actually sold resource price ($v_{p,k)}$ and their private value of what the resource costs (p):

$$u_k^R = p - v_{p,k} \qquad (5)$$

The "maximum social welfare-criterion" (SWF) balances both costs and revenue incurred by the nodes and allows comparison of different variants of the Catallaxy and Baseline implementations. It should be pointed out that SWF solutions are a subset of "Pareto-efficient" ones; once the sum of the payoffs is maximized, an agent's payoff can increase only if another agent's payoff decreases.

5.2 Results

Each experiment runs for 2000 seconds simulated time. After 10 repetitions of the experiments, the uncertainty of measuring any parameter was found to be sufficiently small (stddev < 2 %).

Performance in highly loaded Grids

The behaviour of a system when demand load increases is saturation; no more requests can be allocated. Further load will produce congestion; fewer requests are allocated than at saturation point.

Figure 4 shows the behaviour of a catallactic-operated grid when demand load increases from 5 requests per second to 10, 20 and 40. The grid density and dynamics of those simulations are the high density, low dynamics scenario corresponding to high-capacity continuous available nodes (HPC Grid).

Resource allocation efficiency RAE drops to 60% of successful allocations due to resource saturation. For the catallactic-operating mode the response time REST increases very fast at high loads since the flooding discovery mechanisms take longer to find a node with available resources. For the baseline system it the response time is similar at different loads, since the computational time to match demand and offer is not simulated. The response time is the average time to transmit a request to the central allocator, plus a constant value to compute such allocation, plus the time to send back the decision. For both mechanisms the communication cost CC is lower at high demand loads because we only measure the communication cost of successful allocations. Finally, the social welfare utility SWF is lower at high demand loads because competition among clients decreases clients' utility, since they are forced to buy resources at a higher cost. When demand load is very high, service copies and resources increase their prices to such an extent that many clients cannot buy, therefore overall utility drops.

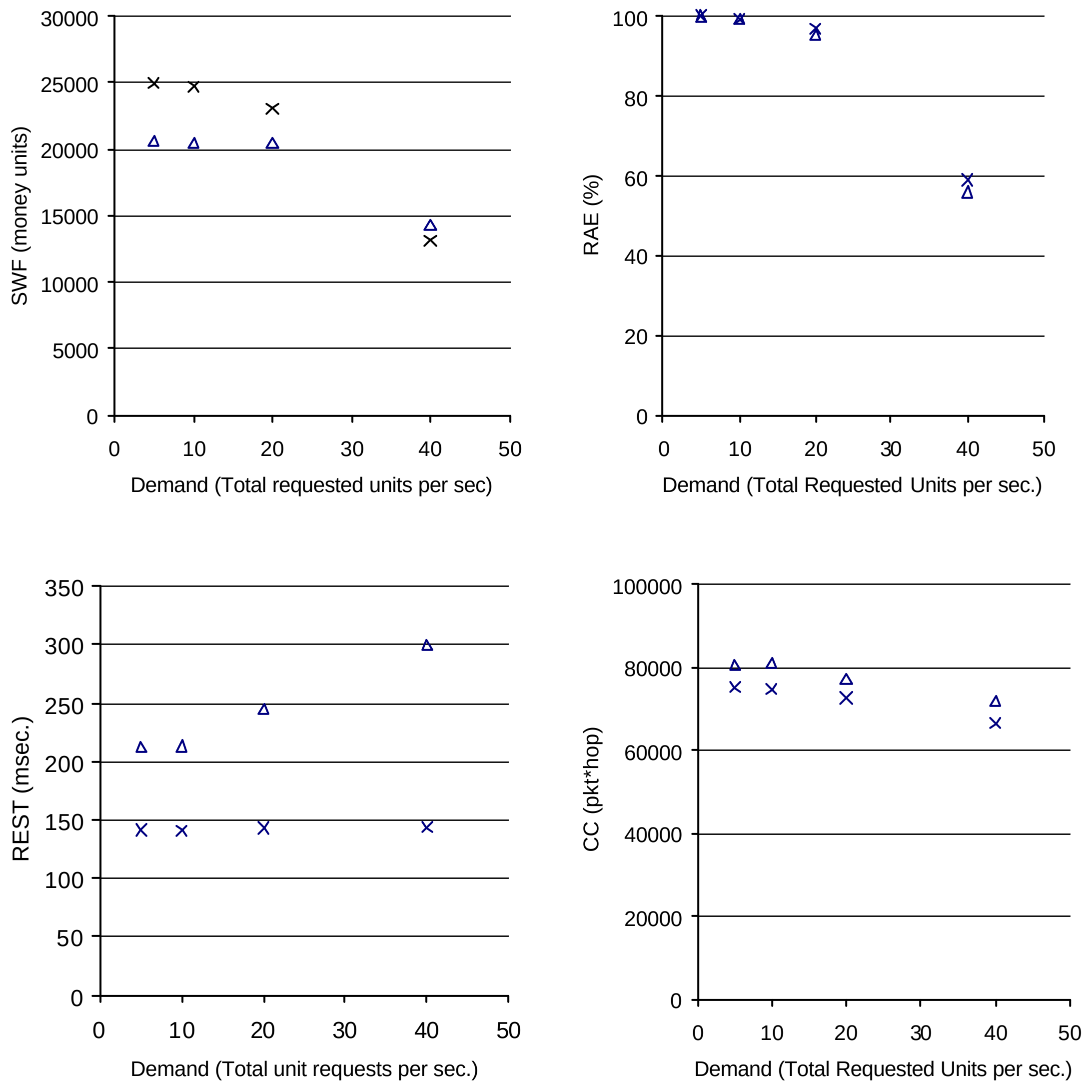

Figure 4. Catallactic (triangles) and baseline (crosses) behavior with demand loads of 5 requests units per second to 10, 20 and 40.

Performance in high and low density Grids

The expected behaviour as density decreases is that it becomes more difficult to allocate requests because resources and service copies are more dispersed; for the same total number of resource units there are more resource and service copy nodes.

Figure 5 shows the behaviour of catallactic and baseline operated grids when density decreases from high capacity nodes to low capacity nodes. Total grid capacity is maintained by increasing the number of nodes. The total demand load corresponds to a medium-highly loaded demand with 20 requests per second. Node dynamics is set to zero; nodes are always available.

Resource allocation efficiency drops for both catallactic and baseline operated grids. However, the baseline system shows worse behavior because it must monitor an increasing number of nodes that leads to a situation where more requests are unmatched. On the other hand, the decentralized catallactic system can cope better with higher number of nodes. The overall utility is higher for the catallactic system than for the baseline system. However, it also decreases as node number increases due to higher competition among service-copy and resource nodes. As expected, the total communication cost increases for both mechanisms. The response time increases for the baseline system as density decreases, since it becomes busier monitoring an increasing number of nodes. The response time for the catallactic system is higher than the baseline system in all density and it is dependent on the placement of nodes. In high density Grids the response time is influenced most by the discovery time, it takes less time to discover a free resource as nodes are more dispersed in the network, i.e. as density is lower. However in very low density Grids the response time presents a high overhead due to longer negotiation times with nodes that are place in the edge of the network, therefore it increases compared to intermediate values of density.

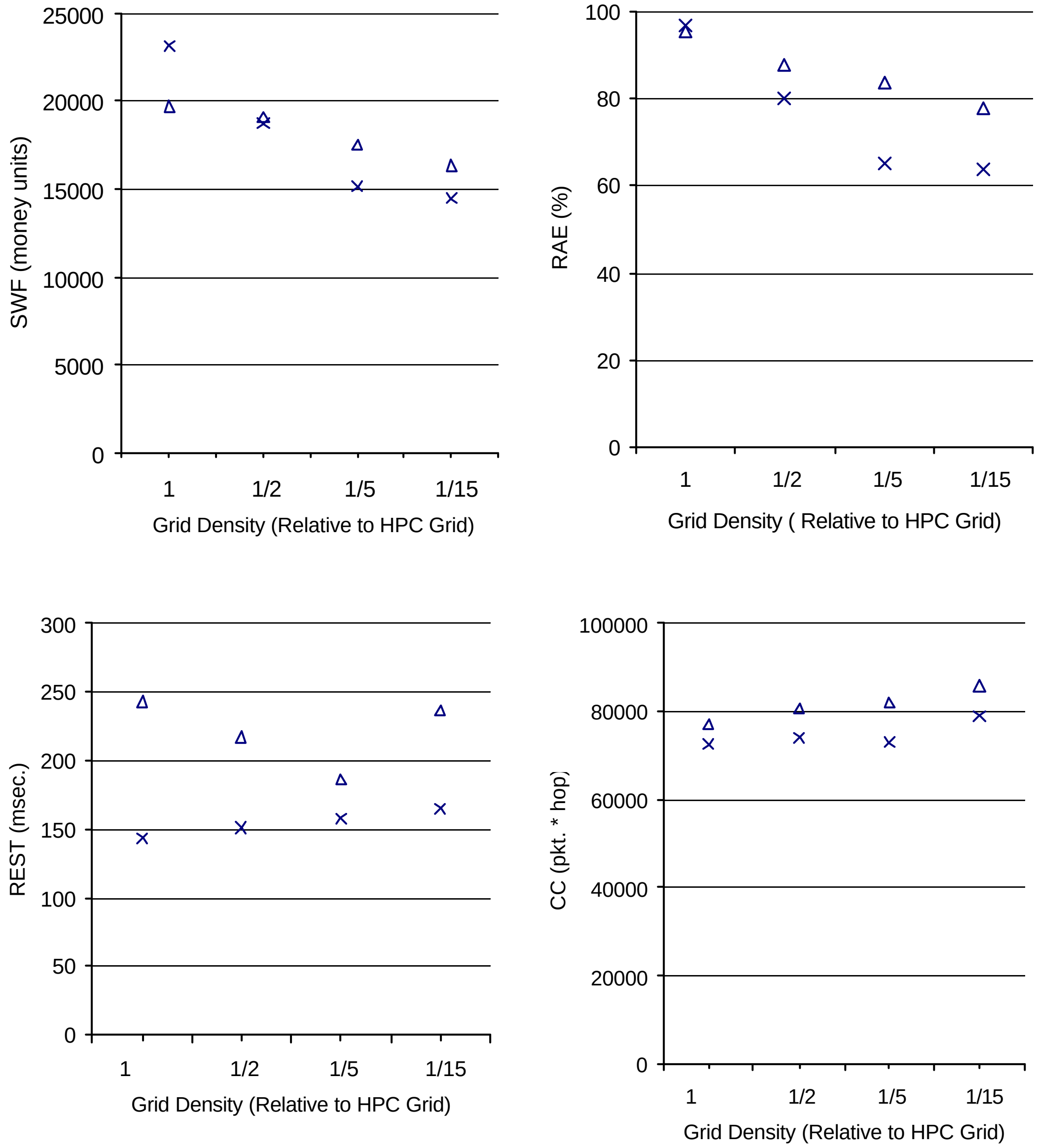

Figure 5. Catallactic (triangles) and baseline (crosses) behavior in Grids with different density. The HPC Grid, relative density of 1, has little number of high capacity nodes (5 resources nodes with 60 resources units each) and P2P Grids have increasing number of nodes and decreasing node capacity relative to the HPC Grid (10, 25 and 75 nodes with 30, 12 and 4 resource units each respectively). Total Grid capacity has been maintained for all scenarios. Besides placement of resources nodes changes from central location in the network at the HPC Grid to edge locations of the network in the scenario with low density, P2P Grid.

Performance in highly available and disconnecting Grids

The two main consequences of node disconnections are: first in the catallactic mode negotiations are aborted and in the baseline system there are allocations at disconnected nodes; and second, service sessions are interrupted. Since we do not simulate service sessions this second effect is not taken into account.

Figure 6 shows the behaviour of catallactic and baseline operated grids when node dynamics increase from always connected nodes to nodes highly disconnected with probabilities increasing up to 30 % of time. The total demand load corresponds to a medium-highly loaded demand with 20 requests per second. Node density is set to the high capacity nodes.

The main effect of node dynamics increase is a decrease of resource allocation efficiency, since an increasing number of negotiations in the catallactic case are aborted, whereas in the baseline case there are an increasing number of allocations to disconnected nodes. The response time for the catallactic system is influenced by node dynamics, because after an agent has aborted a negotiation it can restart another negotiation with other agent, thus extending the response time. Nevertheless, in the baseline system, if an agent is allocated a service copy or resource node that becomes disconnected, it does not issue a new allocation request. Therefore the allocation efficiency is lower than for the catallactic system. The social welfare is lower for the catallactic system because nodes tend to decrease their negotiation prices to achieve a successful negotiation. Nodes do not know why a negotiation has failed. A node discontinuing to negotiate and a node being disconnected produce the same effect.

Finally, we do not simulate the baseline master node being disconnected, which would yield far worse results for the baseline system, as for any centralized system subject to failures.

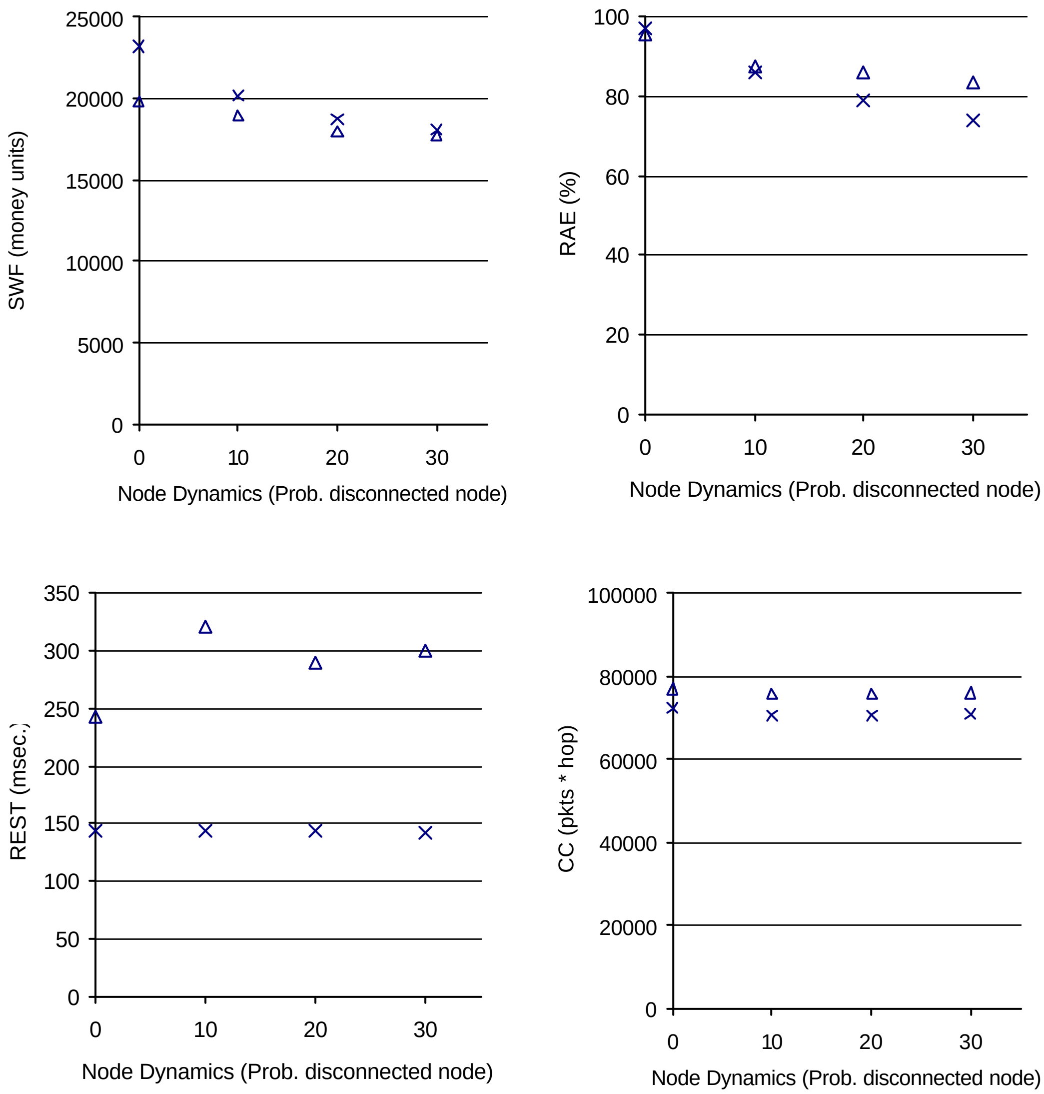

Figure 6. Catallactic (triangles) and baseline (crosses) behavior in a highly available Grid, 0 % probability of being disconnected, and in disconnecting Grids with 10, 20 and 30 probability of node being disconnected.

6. Conclusions and outlook

In this work we show how to apply the Catallaxy decentralized economic-based allocation model to Grids. The Catallaxy paradigm applied to Grids is implemented by three types of agents that represent clients, service providers and resource providers. These agents try to maximize their utility according to their internal strategy negotiating with each other in two different but interrelated Grid resource and service markets. We simulate such Grid markets and catallactic agents in order to evaluate their behavior with different demand loads in a number of Grid density and dynamic environments. We compare these results with a baseline system implementing a centralized double auction allocation mechanism.

The behavior of the catallactic allocation mechanism when the demand load increase is similar to the baseline system; there is a decrease of allocation efficiency due to resource saturation. The response time increases considerably, since it takes longer to discover free resources. However, in the baseline system the central allocator will suffer from demand requests overload, which can cause problems, whereas in a Catallactic system such demand load is distributed throughout the Grid. In low Grid density scenarios the performance of the catallactic and the baseline allocation mechanism degrades due to the dispersion of nodes, nevertheless the catallatic system shows higher resource allocation efficiency and overall utility than the baseline system. The central allocator of the baseline system must monitor more nodes and the complexity of the allocation grows. Whereas the catallactic system copes better with higher number of nodes, however utility decreases due to higher competition. The response time for the catallactic system shows a peculiar influence on the placement of resources, intermediate Grid densities scenarios show the lower response times. Finally, when node dynamics are evaluated by simulating node disconnections from the network, the

Catallaxy approach also shows good behavior since autonomous nodes respond better to failures than centralized allocators. In every scenario the response time of the catallactic system has been worst than in the baseline system, though in the baseline system the computational time to match demand and offer was not simulated.

In summary, the Catallactic system shows similar efficiency to a centralized system but provides advantages due to its decentralized operation: scalability to demand and offer, and fault tolerance in dynamic environments. However response time is high due to discovery mechanisms and negotiation time. Efficient discovery mechanism such as distributed hash tables DHT [40] will decrease response time. Catallaxy will benefit most to long running services where allocation response time can be neglected, in favor of higher allocation efficiency and overall utility.

Future work required falls into two areas: incorporating machine learning mechanisms into agents to make their strategy adaptable, and incorporating reputation mechanisms for scenarios where nodes can decide not to fulfill their contracts. There are different machine learning mechanisms to enable agents to learn how to improve their strategy. One kind of machine learning algorithm that is completely decentralized, and thus suitable for Catallaxy, are evolutionary algorithms where genotypes suffer mutations and recombination. These algorithms have been applied to Catallaxy in other scenarios [43]. A preliminary solution to the problem of negotiating agents not fulfilling their contracts has been proposed in [44], however a more decentralized solution is currently being investigated.

References

[1] I. Foster, C. Kesselman "Globus: A Metacomputing Infrastructure Toolkit". Intl J. Supercomputer Applications, 11(2):115-128, 1997.

[2] G. Allen, W. Benger, T. Goodale, H. Hege, G. Lanfermann, A. Merzky, T. Radke, and E. Seidel. ”The Cactus Code: A Problem Solving Environment for the Grid”. In Proc. High Performance Distributed Computing (HPDC-2000), 253-260. IEEE Computer Society, 2000.

[3] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, K. Stockinger “Data Management in an International Data Grid Project”, IEEE/ACM International Workshop on Grid Computing Grid'2000 - 17-20 December 2000 Bangalore, India.

[4] I. Foster, A. Iaminitchi. “On Death, Taxes and the Convergence of Peer-to-Peer and Grid Computing”. 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), February 2003, Berkeley, CA.

[5] T. Eymann and B. Padovan. “The Catallaxy as a new Paradigm for the Design of Information Systems”. Proc. of The World Computer Congress 2000 of the International Federation for Information Processing. 2000.

[6] A.S. Grimshaw, W. A. Wulf. et al. “The Legion Vision of a Worldwide Virtual Computer”. Comm. of the ACM, 40(1) 39:45, 1997.

[7] J. Frey, T. Tannenbaum, et al. "Condor-G: A Computation Management Agent for Multi-Institutional Grids." Cluster Computing 5(3): 237-246.

[8] I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, A. Roy, "A Distributed Resource Management Architecture that Supports Advance Reservation and Co-Allocations", In International Workshop on Quality of Service, 1999.

[9] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network Analysis", 1st First Intl. Conference in on Peer-to-Peer Networks, Linköpings Universitet, Sweden, Aug. 2001.

[10] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. "Freenet: A Distributed Anonymous Information Storage and Retrieval System". In Proc. of the ICSI Workshop on Design Issues in Anonymity and Unobservability, International Computer Science Institute, Berkeley, CA, 2000..

[11] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, D. Werthimer, "SETI@home: An Experiment in Public-Resource Computing", Communications of the ACM, Vol. 45 No. 11, ACM Press, USA, November 2002.

[12] A. Chien, B. Calder, S. Elbert, and K. Bhatia, “Entropia: architecture and performance of an enterprise desktop grid system”, Journal of Parallel and Distributed Computing, Volume 63, Issue 5, Academic Press, USA, May 2003.

[13] B.A. Huberman. “The Ecology of Computation. Studies in computer science and artificial intelligence”. Amsterdam: North-Holland 1988.

[14] S.H. Clearwater. “Market-based control. A paradigm for distributed resource allocation”. Singapore: World Scientific 1996.

[15] M. S. Miller, and K. E. Drexler. “Markets and Computation: Agoric Open Systems. The Ecology of Computation”. B. A. Huberman. Amsterdam, North Hol-land: 133-176.

[16] M. Stonebraker, R. Devine, et al.. “An economic paradigm for query processing and data migration in Mariposa”. 3rd International Conference on Parallel and Distributed Information Systems, Austin, TX, 1994.

[17] O. Regev, and N. Nisan. “The POPCORN market - an online market for computational resources”. International Conference on Information and Computation Economies, Charleston, SC, ACM Press 1998.

[18] C. A. Waldspurger, T. Hogg, et al. "Spawn: A Distributed Computational Economy." IEEE Transactions on Software Engineering 18(2): 103-117, 1992.

[19] R. Buyya, D. Abramson, and J. Giddy. “A Case for Economy Grid Architecture for Service-Oriented Grid Computing”. Proc. 10th IEEE International Heterogeneous Computing Workshop (HCW 2001). San Francisco, 2001.

[20] R. Wolski, J. S. Plank, J. Brevik and T. Bryan “Analyzing Market-Based Resource Allocation Strategies for the Computational Grid” The International Journal of High Performance Computing Applications, Sage Science Press, Volume 15, number 3, Fall, 2001, pages 258-281.

[21] H. B. Bell, D. G. Cameron, L. Capozza, A. P. Millar, K. Stockinger and F. Zini “Simulation of Dynamic Grid Replication Strategies in OptorSim“. In Proceedings of the 3rd International Workshop on Grid Computing, Baltimore,USA, November 2002.

[22] J. Gomoluch and M. Schroeder. “Market-based Resource Allocation for Grid Computing: A Model and Simulation”. 1st International Workshop on Middleware for Grid Computing, Rio de Janeiro, Brazil. 2003.

[23] B. Schnizler, D. Neumann, C. Weinhardt “Resource Allocation in Computational Grids - A Market Engineering Approach”, Proceedings of The Third Workshop on e-Business (WeB 2004), Washington, 2004.

[24] J. Q. Cheng and M. P. Wellman, "The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes." Computational Economics 12: 1-24. 1998.

[25] T. T. Ping, G. C. Sodhy, C. H. Yong, F. Haron and R. Buyya “A Market-Based Scheduler for JXTA-Based Peer-to-Peer Computing System” , ICCSA 2004, LNCS 3046, pp. 147–157, 2004.

[26] D. Hausheer, B. Stiller “Decentralized Auction-based Pricing with PeerMart”. 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), Nice, France, May 2005.

[27] R. Myerson, M. Satterthwaite: “Efficient Mechanisms for Bilateral Trading”, Journal of Economic Theory, Vol. 29, pp. 265-281, 1983.

[28] R. Wilson. “Incentive efficiency of double auctions”. Econometrica, 53(5):1101-1116, 1985.

[29] F.A. Hayek, W.W. Bartley, P.G. Klein, and B. Caldwell. The collected works of F.A. Hayek. University of Chicago Press, Chicago, 1989.

[30] E. Hoppmann, “Unwissenheit, Wirtschaftsordnung und Staatsgewalt. Freiheit, Wettbewerb und Wirtschaftsordnung.” V. Vanberg. Freiburg, Haufe Verlag: 135-169, 1999.

[31] B. Schmid and M. Lindemann, “Elements of a Reference Model for Electronic Markets”. Proceedings of the 31st Hawaiian International Conference on Systems Sciences. Los Alamitos, CA: IEEE Press, 1998.

[32] J.S. Rosenschein and G. Zlotkin, “Rules of encounter - designing conventions for automated negotiation among computers”. Cambridge, Mass: MIT Press, 1994.

[33] R. Guttman and P. Maes “Agent-mediated Integrative Negotiation for Retail Electronic Commerce”. Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98). Minneapolis, Minnesota: 1998.

[34] C. Sierra “A roadmap for Agent-Mediated Electronic Commerce”. Sierra, C. and Dignum, F. (eds.). Agent Mediated Electronic Commerce - The European AgentLink Perspective. Heidelberg: Springer, 2001.

[35] M.P. Wellman, “Market-Oriented Programming: Some Early Lessons”. Clearwater, S.H. (ed.). Market-Based Control: A Paradigm for Distributed Resource Allocation, 74-95. Singapore: World Scientific, 1996.

[36] D.G. Pruitt, “Negotiation behavior”. New York: Academic Press, 1981.

[37] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, “Grid Services for Distributed System Integration”. Computer, 35(6), 2002.

[38] T-Online AG: T-Online PDF Creator: Easy document conversion. 2003, http://www.t-online.net/c/07/30/40/730406.html

[39] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman. “Grid Information Services for Distributed Resource Sharing”. Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[40] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, “A Scalable Content-Addressable Network”, Proceedings of ACM SIGCOMM 2001

[41] H.Y. Tyan and C.J. Hou, "JavaSim: A component-based compositional network simulation environment," Western Simulation Multiconference -- Communication Networks And Distributed Systems Modeling And Simulation, January 2001.

[42] R. Smith, “The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver”. IEEE Transactions on Computers. 29(12). December 1980.

[43] B. Padovan, S. Sackmann, T. Eymann, and I. Pippow, “A Prototype for an Agentbased Secure Electronic Marketplace including Reputation Tracking Mechanisms”. Proceedings of the 34th Hawaiian International Conference on Systems Sciences. Outrigger Wailea Resort, Maui: IEEE Computer Society, 2001.

[44] T. Eymann “Co-Evolution of Bargaining Strategies in a Decentralized Multi-Agent System“. In: Proceedings of the AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems, North Falmouth, MA, November 03-04, 2001.