A Greedy Grid - The Grid Economic Engine Directive

Sudharshan Vazhkudai (1) and Gregor von Laszewski (2)

(1) Department of Computer and Information Science
The University of Mississippi
University, MS 38677
chucha@john.cs.olemiss.edu

(2) Mathematics and Computer Sciences Division
Argonne National Laboratory
Argonne, IL 60439
gregor@mcs.anl.gov

January 21, 2001

Abstract

The advent of national-scale "Computational Grid" infrastructures has helped deploy advanced services, beyond those taken for granted in today's Internet, such as: remote access to computers, wide area resource management, authentication, and directory services, thus enabling access and utilization of a variety of heterogeneous resources distributed over multiple domains. The availability of these services represents an opportunity to implement advanced services utilizing these basic Grid services. In this paper, we explore issues related to defining services that are based on economy and financial models in order to encourage further resource sharing among the administrative domains while also considering commodity PC's that are part of todays Internet. We propose an extendable architecture, The Grid Economic Engine Directive (Greed), that allows integration of various economy models within the same framework, exposing the services through secure protocols and policies. The services provided by this framework include, for example, a bartering service, a bidding service, and a trading service. We intend to develop components that can be integrated within a customizable Portal simplifying access to many of the services and propose to integrate the Greed economic middleware into the existing Globus metacomputing toolkit, thus enabling the application of economic paradigm to the Globus Computational and Data Grids. We further illustrate the applicability of the proposed Greed services by building a prototype business model as a higher-level application, using the economic middleware.

Keywords: Grid, Computational Economy, Commodity Computing, Data Grids

1 Introduction

Using economy based models to gain access to remote resources is not a new concept in a Grid based environment. Renting, Leasing, and establishing contracts between the application users and supercomputing centers based on the usage of compute time is a well known fact. In the last decade, the scientific supercomputing centers have experimented with combining their computational resources in order to build metacomputing systems[1]. These have been extended to include many other resources such as networks and storage systems. This integrated agglomeration of resources is today referred to as the Grid.

We foresee the advantage of designing Grid services that allow the integration of economy models within the concept of resource sharing. Introducing such economy models have multiple beneficial aspects, including better utilization of the resource, and the mo-

[1] "Grid Environments are often referred to as "Metacomputing Systems" or "Wide Area Distributed Systems". In this paper, we will use these terms interchangeably.

tivation for integration of resources by providers that traditionally are not participating in the Grid.

Due to the heterogenous nature of the demands between users and the offers provided by resource providers, we suspect that multiple economy models should be supported in such a grid environment to ease the differences between supply and demand. Before we describe the architecture in more detail in a later section, we first present a variety of economy models that we consider for integration. Some of these economy models are today already used successfully on the Internet. Nevertheless, an integrated economy-based service that provides the ability for exchanging Grid specific services and resources has yet to be defined.

2 Economic Models

The real world is replete with a variety of economic and business models, suited for various purposes. Not all of them are applicable in a metacomputing environment. In this section, we will discuss those models that are applicable to the Grid environment. We have attempted to classify the various models into broad categories such as market model, community model, etc., by citing popular example implementations of these models. Figure 1 represents a sample classification of the community based model and its various implementations. A similar structure applies to market models and various others.

2.1 Taxonomy of Economy Models applicable to Metacomputing

1. **Ownership / Shareholders** - The simplest model is the total ownership of a resource by a resource provider. Application users have to apply to the resource owner in order to negotiate the terms of usage of this resource. In case a resource is owned partially amongst a group of resource providers, application users may negotiate with a resource provider for obtaining partial access or with the group to obtain full access to the resource for a fraction of its use. Examples for such ownership models are industry and research based supercomputing centers.

2. **Market Model** - In a market model owners of resources advertise their abilities on a market to which consumers are attracted based on the variety of offers and the consumer's ability to choose from a variety of alternative sources. Usually a fixed or approximate exchange is determined before the service is offered in the market. Having a market and a value attached with resources may simplify the contract and negotiation phase of exchanging heterogenous resources. Examples are the JavaMarket [9], POPCORN [18] market, etc.

3. **Auction Model** - An auction [16] is a variety of services that are offered to the highest bidder. Auction and bidding strategies for load balancing have been used quite predominantly in earlier research in local area distributed systems [20]. Recent applications of auctioning and bidding include popular commercial systems such as: amazon.com [1] and eBay [3]. These strategies have been quite successful in earlier attempts and can be extrapolated to a metacomputing scenario where resource owners and consumers auction and bid services.

4. **Tender Model** - A business entity (consumer) that requires a particular service places a specification of its requirements which includes a price quote. In response to this, service providers submit their tenders attempting to match the specifications. The consumer selects the best candidate from among the tenders submitted, which could be the service provider that charges the least amount of money and also satisfies all of the constraints mentioned in its specification.

5. **Community Model** - The most successful model today in the Internet to mobilize resources to build large community clusters is based on the fact that a community is established through a common beneficial task to be solved. Providers contribute their resources voluntarily as they

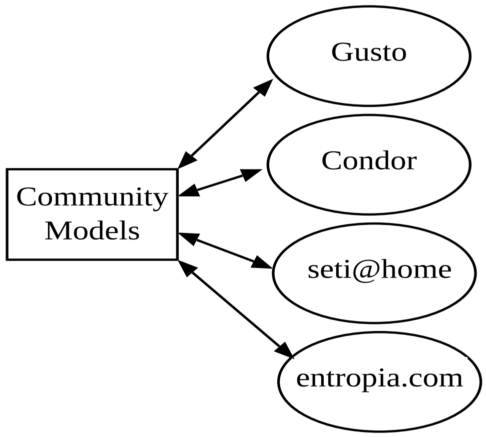

Figure 1: A sample Economic Model

consider it beneficial to participate in this community activity. Successful examples for this activities are seti@home [8], entropia.com [5], and the Condor [2] system.

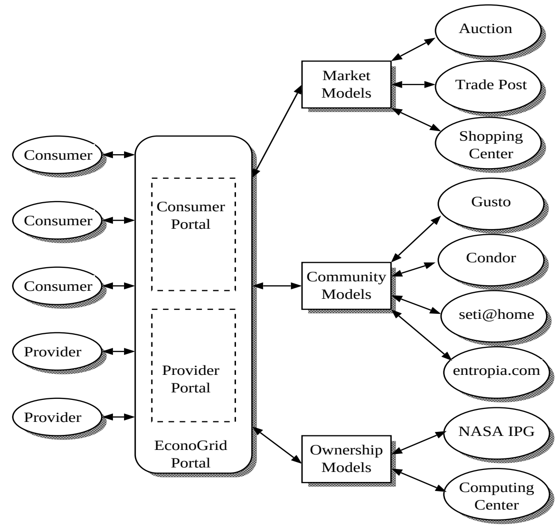

Figure 2: EconoPortal - The EconoGrid Portal provides access to various economy models that allow consumers and producers to decide which model is best suited for their application or their resource contribution.

2.2 A Case for the Economic Portal

In this section, we will present a case for the need for an economic portal. In Figure 2, we describe the essential architecture for the economic portal. We propose that the economic portal abstracts the various implementations that are based on multiple economic models, thereby providing a uniform interface and access to different implementations.

1. From the discussion above and from Figure 1, it is evident that various models have their associated merits and application domains which make them indispensable. Supporting just one model would restrict us to a particular domain of applications which is not desirable. Thus it would be beneficial if we had a generic implementation that supported multiple economic models thereby facilitating a wider range of applications.

2. Also evident from Table 1 is the overlapping nature of these economic models. We can easily predict that these models have inherent similarities that can be tapped in order to construct a meta-model.

3. Another advantage of this approach is the portal it provides for service providers and consumers. The portal would channelize access to various economic models. Service providers and consumers can choose models that are beneficial to their purposes. For example: a service provider might decide to contribute his resource to an implementation based on the auction model, attempting to profit from user bids. Likewise, a consumer can choose a model best suited for his purpose. Thus, with this approach, both parties are provided with the freedom to select the economic models based on their current needs and preferences.

4. As use of this approach progresses, the portal can transform into an intelligent system, suggesting various implementations automatically

Economy Model	Condor	Seti@Home	entropia.com	JavaMarket	eBay
total ownership	X	X	X	X	
auction					X
idealistic community	X	X	X		
regulated economy	X	X	X	X	
market model				X	X

Table 1: Taxonomy of Grid based Economic models.

based on consumer and service provider preferences.

5. Yet another feature is the scalability of the economic portal. As further implementations evolve, they could be easily plugged into the portal.

2.3 Advantages of Economic Models

The advantages of applying economic models to resource management in a Grid environment are as follows:

- **Competitive approach** - Traditionally, distributed systems have followed a "Cooperative approach" to resource management, wherein nodes interact with each other in a cooperative manner in order to improve system throughput, performance and resource sharing. Each node is configured to improve the entire system's throughput; Whereas a competitive approach to resource management introduces the notion of competition among the nodes, wherein a node's primary intent is to increase its own throughput or profit. Each node, thus behaves in a selfish manner trying only to maximize its own profit. The argument that goes in favor of a competitive approach is: since every node is concerned with its own profit, this inherently improves the system throughput as a whole [20].

- **Motivation for collaboration** - One of the other problems to be addressed is that of motivation for collaboration. Most systems assume that nodes can associate and help each other, but the question is, "Why should they help each other? What is in it for them?". Different systems have different approaches. For example, systems that follow a "Community model", say Condor, follows the policy, "If a machine contributes its resources to the pool, it can in turn receive service". Another approach is the introduction of "Computational Economy". A strong motivating factor is the association of monetary gain in return for service provided.

2.4 Disadvantages of Economic Models

Implementing an economy based market model on the shoulders of the unique user community of scientific applications may pose certain risk factors which have to be considered in order to be able to continue pursuing scientific explorations and free research.

- **Hinders Basic Research** - One of the problems to note is that an introduction of money based services may pose a danger to basic research programs. These research programs (creation of Internet, for example) are often initiated with long term goals that are not fulfilled or considered by many market oriented goals. Putting in the foreground a short term "marketable" goal may eliminate an advance altogether due to lack of funds that has to be provided over a long period of time.

- **Monopoly** - Another important aspect is the inherent possibility of monopolies that can control the price of a particular resource or service. If an economy based model is implemented a monopoly on services and resources should be prevented so that "arbitrary-price-forwarding" to consumers can be avoided.

3 A Greedy Grid

3.1 Motivation for Greed

Application of economic principles to metacomputing environments is still in its embryonic stages. There have been a couple of proposals and implementations such as: GRACE [11], JavaMarket, etc. Most of these approaches apply economic principles only to the job execution environment, and follow specific economic models. In our approach, we provide a generic framework encompassing popular economic models applicable to the metacomputing scenario and apply economic concepts uniformly to all grid related services.

3.2 The Globus Philosophy

Globus [6] builds the fundamental technology necessary for building computational grids. The Globus metacomputing toolkit[2] comprises of a set of modules, each of which provides an interface that higher-level layers can invoke. Globus thus provides a mechanism for building various higher-level services based on low-level mechanisms. The toolkit contains of modules supporting (Figure 3): Resource Management (GRAM); Communication (Nexus); Security (GSI); Unified Resource Information Service (GRIS); Remote Data Access (GASS); Executable Management (GEM).

GRAM provides an interface to local resource management strategies (like Condor, loadleveler, Load Sharing Facility - LSF, etc); Nexus provides a uniform interface to a diverse set of communication protocols (like UDP, TCP, etc); GSI, Globus Security Infrastructure, provides APIs for authentication; GRIS provides tools for information publishing, retrieval and access; GASS, Globus access to secondary storage, provides uniform access to secondary storage devices in a metacomputing environment; GEM, Globus Executable Management, provides APIs for executable staging [10].

[2] We use the terms "toolkit", "framework" and "middleware" interchangeably.

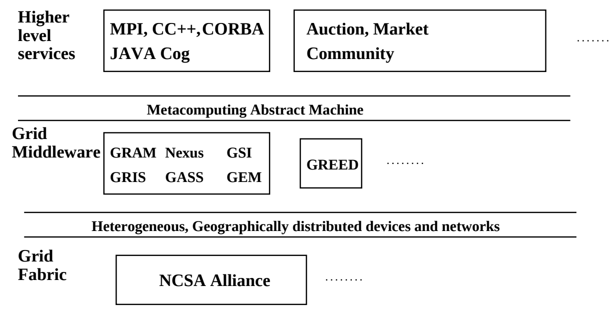

Figure 3: Globus Toolkit with Economic Additions.

3.3 Greed Overview

Following the above mentioned Globus architecture, it is evident that if we wished to incorporate economic considerations into the Grid computing environment, it should be done in such a way as to abide by the spirit of the existing infrastructure. Globus stresses more on providing a uniform interface for a service so that various implementations could coexist harmoniously. Our implementation of a "Greedy Grid" takes into consideration this architecture of Globus, which implies that it would take its place as a module in the Globus toolkit, providing a uniform interface for multiple economic model implementations.

In (Figure 3), we show the economic inclusions into the existing Globus infrastructure. The relevant module for economics in the Globus toolkit is Greed, The Grid Economic Engine Directive. This ideally encompasses and abstracts the various computational economic issues that are involved in the various models, such as: the market model, the community model and the ownership model, thereby providing a generic interface. In the higher level services, we show various implementations currently available that are based on various economic models.

The real world is so diverse in that, it has evolved from a simplistic "Bartering" approach to sophisticated applications such as stock markets and auction, which are inherently complex. It would not be realistic to generalize all of these complex models. As a starting point, we consider those economic models that are applicable to the metacomputing environment (Figure 2), and have proven to be successful; attempt to generalize the concepts involved; abstract out the common features; provide for possible enhancements as additional models are considered.

3.4 The Greed Framework

We now consider several metrics to analyze the various economic models, in order to build a framework .

- **Pricing Strategy**

  1. **Entities to be priced** - The most important aspect of the pricing strategy is deciding what to price or estimate. In a Grid based resource management environment there are a variety of entities that producers and consumers would like to share among themselves. For example: CPU cycles, Storage space, Transfer Bandwidth, Amount of time the resource is required, Secure transactions, Access to particular software (PUNCH [7] like systems), etc.
  2. **Cost of a particular service** - How does one decide the cost of a particular service? More precisely, "What is the cost per unit of storage?" or "What is the cost per unit of CPU time?". "Is there a central entity that decides the price of a particular service?" or "Are prices decided more on a supply-demand-infrastructure basis, i.e., "Is the price of a service dictated by the availability or non-availability of it?". A supply-demand scenario in a Grid environment would mean a dynamic self-stabilizing system that adjusts itself based on the market needs. There are several strategies to decide the pricing models. For example, the "Exchange based economy" and the "Price based economy" [21].

- **Plan Generation**

  1. **Cost Scheme** - This would involve devising a cost scheme for the service. Typically, the requirements and capabilities would be preprocessed through the economic engine in order to generate a cost plan. The requirements and capabilities can be specified in a resource description language, for example: a classified advertisement, ClassAd [15] (from Condor). A typical capability statement of a service provider, say a storage service provider, could be, "capability = storagespace < 20G && avgrdbandwidth < 100K". A cost plan for the above mentioned statement would be the (cost per unit storagespace * 20G) + (cost per unit avgrdbandwidth * 100K).
  2. **Discount** - As part of the plan generation the economic engine could also specify discount strategies to enhance profit. These discount strategies could be based on past transactions and historical data.

- **Trading**

  This should encompass and generalize the trading methodologies involved in the various economic models. For example: The manner in which service providers announce their price (tender model, auction model, market model); The manner in which consumers submit their bids / tenders; Matching providers specifications with that of the consumers. The above mentioned items vary significantly among economic models. For example, the price announcing scheme in a market model is to have service providers report their prices to a central market; In an auction model, the auctioneer announces the price by requesting bids from consumers, etc.

- **Negotiation**

  The Greed framework in the service provider and the requester would be tuned towards maximizing the profit for either side. The service provider attempts to maximize the profit by increasing the cost while the service requester attempts to maximize the profit by minimizing the cost. Obviously there has to be some intermediary cost scheme that is mutually acceptable. There are a variety of schemes proposed in the literature on price negotiations. A strategy that is suited to the Grid environment is the bidding scheme.

- **Accounting**

  Some of the details in accounting include: Maintaining information about each transaction; Track depletion of resource / service; Monitor the ratio of resource usage to profit gained; Monitor funds availability from a service consumer perspective (maintain balances); Check if money pre-paid equates to the service delivered; pay back if not; Strategies for Billing and Payment.

- **Electronic Currency**

  The moment we introduce economic models, we have to introduce the notion of money. Initially, in the real world economic model, there was bartering, but that was soon replaced by the notion of money due to problems associated with bartering such as: mapping services as equivalent. We have to address the following, among others [12]: How to represent money in the digital environment: E-cash [4], Netcash [17]; For this system to be valid, both parties should be able to trust the currency supplied; Obtaining Digital currency - Third parties or Banks - provide insurance; These digital currencies could be digital certificates that are duly signed by an entity trusted by both parties.

- **Quality of Service**

  Economic models become more significant when quality of service requests are placed. In such cases, service providers can charge an extra amount for QOS deadlines. When consumers make QOS requests they are typically willing to pay the extra charge in return for the reliable service. This also includes issues such as: leasing, bulk purchases and reservations.

4 A Higher level Service based on Greed

In this and the subsequent sections, we will present a prototype implementation of a higher level service we have built that would use the Greed framework. We have modeled the Greed framework in accordance with existing Globus standards, i.e., providing an economic middleware upon which a variety of services can be built. This provides the necessary flexibility and extensibility.

Various business models can be built based on the Greed middleware. A few examples are: A Market Model, An Auction Model, and A Business Tender Model. In the subsequent section, we will present the prototype of a business tender model that we have built.

4.1 Buyer's Market - A Business Tender Model

In this section, we will present a description of a Buyer's market, a tender based business model (Figure 4). This model makes use of the Greed framework to build a tender based market, wherein a consumer collects tenders from different service providers to decide upon a best provider.

The consumer consists of an Ad Agency that generates an advertisement, a requirement specification, based on the service request. This request is processed through the Greed framework to obtain the corresponding cost limitations for the service requests. The advertisement now consists of the requirements and their associated price ceilings, which are announced using a candidate probing technique to the service providers. This is processed through the "Trading" protocol of the Greed framework. In response, service providers submit their tenders to the consumer. The service provider's corresponding Greed component decides the appropriate cost for use of its resource. The consumer's "Tender Processing Unit" gathers all submitted tenders and performs a match against its own specification in order to obtain a suitable service provider. It might happen that no suitable candidate is obtained during the matching

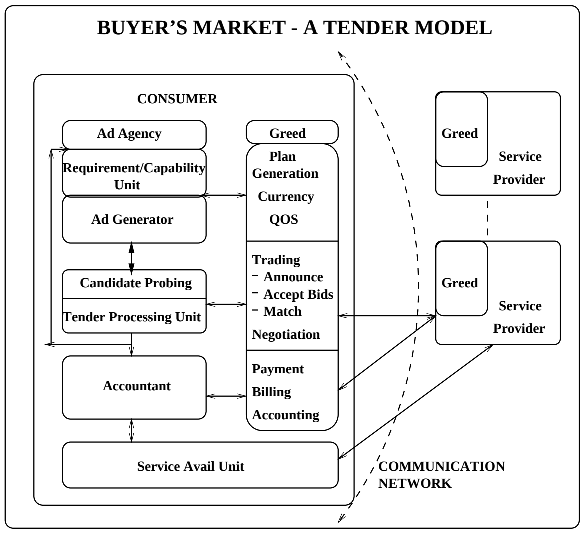

Figure 4: A Buyer's Market

process. In such a case, the consumer typically revises its specification by repeating the above process. Once a service provider has been identified, the accounting and the payment procedures are initiated through the Greed framework, which is responsible for keeping track of balance, overdue, etc. Finally, the service is availed by the consumer.

Based on this model, a variety of systems can be built that provide various services.

4.2 A Case Study

A variety of applications, in diverse domains in grids, can be built based on the business tender model. A few domains of applicability are:

- **The Globus Computational Grid** - The computational Grid is an environment where computational jobs are scheduled across multiple sites that are spread across the grid. In such an environment, a compute resource broker can be built, based on the business tender model, that applies economic principles based on the Greed middleware.

- **The Globus Data Grid** - The Data Grid [14] is an environment where massive storage devices are spread across the grid. For such an environment, a storage resource broker can be built based on the tender model.

In the following sections, we will discuss the prototype implementation of a storage resource broker for the Globus Data Grid based on business tender model and the Greed middleware.

4.2.1 Storage Resource Broker

A data grid provides an environment where a community of researchers, with common interests, can share and access data sets efficiently across the network. In such a scenario, it is quite common for researchers to maintain replicas of data sets in locations that provide good performance characteristics. A storage broker is an entity that attempts to find a suitable storage replica server based on an application's requirements. The storage broker can be viewed as a higher-level service that is based on the Globus Middleware toolkit. The storage broker presented here works on the "Tender Model". In this section, we will present briefly, the architecture necessary to realize the above mentioned scenario.

The application, at the client end, presents a ClassAd [15], a resource specification language, to the storage broker, representing its requirements and specifications. The broker contacts the Greed framework to decide upon a cost that the client is willing to pay, thus applying pricing and plan generation schemes according to the Greed middleware. The broker then performs candidate probing on a set of replica servers in the Globus Data Grid environment. The candidate probing mechanism interacts closely with the trading mechanism in the Greed middleware. Trading usually involves negotiations between client and server and requires identification of and agreement on communication protocols that will be used for such interaction. In this case, we have chosen the LDAP [13], a light weight directory access protocol, often used to access and retrieve information from directory structures in a grid environment. The broker announces the requirement of the application to a collection of replica servers by dispatching LDAP probes. Replica servers submit their bids in response to the probes. The bid comprises the servers' cost specifications, obtained through their Greed components and also information about their storage capabilities such as: available space, transfer bandwidths, seek times, their requirements (policies using which their resource can be used), etc. The broker can now perform a match of the application ClassAd against the list of ClassAds obtained. The matching takes into consideration the cost specifications of both parties and returns the best match based on the rank value of the match, i.e., the application can specify a rank value in its ClassAd that associates a higher priority to a replica server with better transfer rates, or one that charges a lower cost. Both consumer and service provider perform the accounting, billing and payment processes. Of course this is not as simple as mentioned here and involves complex charging and authorization schemes. Once the storage replica has been identified, transfer or access can be achieved using, high speed file transfers like, GSIFTP [6] procedures.

An in-depth description of the architecture of the storage resource broker can be found in [19].

5 Conclusion

In this paper, we presented a proposal for a generic interface for the incorporation of economic models into the Globus grid middleware services. We introduced the notion of Greed, a Grid Economic Engine Directive, that encompasses economic models applicable to the metacomputing scenario and presented a set of possible metrics that have to be considered while building the framework. We also discussed, a tender based business model and an example storage broker application that we have built for the Globus Data Grid using the Greed framework.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

References

[1] amazon.com. http://www.amazon.com, 2000.

[2] Condor: High throughput computing. http://www.cs.wisc.edu/condor, 2000.

[3] ebay: The world's online marketplace. http://www.ebay.com, 2000.

[4] Electronic cash. http://www.research.microsoft.com, 2000.

[5] Entropia: Community model for grid. http://www.entropy.com, 2000.

[6] Globus: The globus project. http://www.globus.org, 2000.

[7] Punch: Purdue university network computing hubs. http://punch.ecn.purdue.edu, 2000.

[8] Seti@home: Use your pc at home for seti? http://www.seti.org/science/setiathome.html, 2000.

[9] Y. Amir B. Awebuch and R.S. Borgstorm. The javamarket: Transforming the internet into a metacomputer. Technical Report CNDS-98-1, Johns Hopkins University, 1998.

[10] I. Foster C. Kesselman. The globus project: A status report. In *Proceedings of Heterogenous Computing Workshop*, 1998.

[11] R. Buyya D. Abramson and J. Giddy. An economy driven resource management architecture for global computational power grids. In *The 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2000)*. PDPTA, June 2000.

[12] C. Lai G. Medvinsky and B.C Neuman. Endorsements, licensing, and insurance for distributed system services. In *Proceedings of 2nd the ACM Conference on Computer and Communication Security*, pages pp. 170–175. ACM, November 1994.

[13] T.A. Howes and M.C. Smith. *LDAP Programming Directory-Enabled Application with Lightweight Directory Access Protocol.* Technology Series. MacMillan, 1997.

[14] A. Chervenak I. Foster, C. Kesselman and S. Tuecke. The datagrid: Towards an architecture for the distributed management and analysis of large scientific datasets. *To be Published in the Journal of Network and Computer Applications.*

[15] R. Raman M. Livny and M. Solomon. Matchmaking: Distributed resource management for high throughput computing. In *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing.* World Scientific PressHPDC, 1998.

[16] W.E. Walsh M.P. Wellman, P.R. Wurman and J.F. Mackie-Mason. Some economics of market-based distributed scheduling. In *Proceedings of the Eighteenth International Conference on Distributed Computing Systems*, pages pp. 148–157. World Scientific PressICDCS, 1998.

[17] B.C. Neuman and G. Medvinsky. Netcash: A design for practical electronic currency on the internets. In *Proceedings of 1st the ACM Conference on Computer and Communication Security.* ACM, November 1993.

[18] O. Regev and N. Nisan. The popcorn market–an online market for computational resources. In *Proceedings of the first international conference on Information and computation economies*, pages pp. 148–157. ACM, 1998.

[19] S. Vazhkudai S. Tuecke and I. Foster. Replica selection in the globus data grid. In *Proceedings of the First IEEE International Conference on Cluster Computing and Grid.* IEEE, 2001.

[20] D.F. Ferguson Y. Yemini and C. NiKolaou. Microeconomic algorithms for load balancing in distributed computer systems. In *Proc. of International Conference on Distributed Systems (ICDCS'88).* ICDCS, 1988.

[21] D.F. Ferguson Y. Yemini, J. Sairamesh and C. NiKolaou. Economic models for allocating resources in computer systems. In *Market based Control of Distributed Systems.* World Scientific Press, 1996.