An Option Contract Framework for an Economy Based Grid Resource Management System

by

Purnachander R Parupally

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science
Faculty of Graduate Studies
University of Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a MSc thesis entitled:

An Option Contract Framework for an Economy Based Grid Resource Management System

submitted by: *Purnachander R. Parupally*

in partial fulfillment of the requirements for the degree of: *MSc*

____

Dr. Peter Graham, Advisor

____

Dr. Pourang Irani

____

Dr. Robert McLeod
Electrical and Computer Engineering

Date of Oral Examination: *May 30, 2007*

The student has satisfactorily completed and passed the MSc Oral Examination.

____

Dr. Peter Graham, Advisor

____

Dr. Dean Jin
Chair of MSc Oral

____

Dr. Pourang Irani

____

Dr. Robert McLeod
Electrical and Computer Engineering

(The signature of the Chair does not necessarily signify that the Chair has read the complete thesis.)

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

AN OPTION CONTRACT FRAMEWORK FOR AN ECONOMY BASED GRID RESOURCE MANAGEMENT SYSTEM

BY

Purnachander R. Parupally

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of

Manitoba in partial fulfillment of the requirement of the degree

MASTER OF SCIENCE

Abstract

Large scale distributed computational environments, such as computational grids, are growing in popularity for solving large and complex problems in many fields including science, engineering, medicine and commerce. A major function of grid systems is managing the allocation of shared resources (e.g. computational capacity, storage space and files) between grid users. In the recent past, pay-for-use systems have been developed and deployed for solving large computational problems which have lead to the development of new resource management strategies, some based on economic models from the real world.

In pay-for-use systems, demand for computational resources may not be uniform. There are certainly seasonal changes that need to be considered and, in some cases, unpredictable demand changes may also occur. These affect the price of resource use. Prices for resources may also change due to other factors (e.g. when a new generation of processors are released and deployed, resource providers may be motivated to increase prices). Existing economic models used in grid systems to allocate available resources do not provide a way to hedge against price increases/decreases for resources. In this research, I introduce an options-like contract framework for use in economy based grid resource management systems. The use of such a contract allows users to hedge against price increases for resources and service providers to hedge against price decreases.

The efficiency and practicality of a simple grid resource management system are evaluated with and without the use of my options-like framework. Resource alloca-

tion efficiency is estimated for three different scenarios (resource trading for immediate use by consumers, resource trading with advanced reservation and resource trading involving options). These three scenarios are measured and compared to each other, most importantly for two extreme situations: high resource demand with low supply of resources and high resource supply with low demand.

Results show that the system supports improved resource utilization and improved price certainty for both resource providers and consumers.

Dedication

Dedicated to my parents.

Acknowledgements

I would like to thank my supervisor Dr. Peter C. J. Graham, for all his ideas, motivation and encouragement throughout this research. I really appreciate the way he helped me to organize the research work, which not only ended in the completion of my thesis but also in learning the right direction of setting things up.

I am also grateful to all my committee members for having provided a good environment during my defense, and for all the valuable suggestions on my thesis draft.

I am very thankful to Dr. Ruppa Thulasiram who allowed me to register in the Computational Finance reading course where I leaned financial concepts.

I am thankful to Dr. Pourang Irani for his valuable advice on carrying out statistical analysis of my research.

I am grateful to my parents for their encouragement and for extending their support always, without which I would not be what I am today. I would also like to extend my gratitude to my sister and brother-in-law for having put the insight into me of obtaining higher education in Canada. I would also like to thank my friends for extending their support at all times.

Above all, I would like to thank the God for giving this wonderful life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recently, the popularity of the Internet together with rapid improvements in processor and networking technologies have led to the development of high-performance, wide area computing systems based on what is known as "grid computing". A grid is a collection of distributed, high-end computing resources assembled dynamically to solve large-scale problems in science, engineering, medicine, etc. Grid resources (i.e. computers, storage, databases, scientific instruments, etc.) are usually located at geographical dispersed locations and will have different access policies, capabilities, etc. Managing these resources in grids poses many challenging issues such as maintaining site autonomy, supporting the heterogeneous nature of the network substrate, resource allocation and/or co-allocation issues, the need for on-line control and support for the sale and purchase of resource use. It is this last point that is the focus of the research presented in this thesis.

Many existing grid systems address some of the identified challenges in grid resource management, most with the notable exception of support for "computational economies". A computational economy is a framework in which resource providers can sell access to their resources and resource consumers can buy access to the resources to solve their computational problems. As grids mature, the need for such "charge-for-use" models is increasing. The use of a computational economy motivates resource owners to offer their resources for others to use and this will ultimately help to facilitate the development and management of large-scale computational grids.

In a grid economy, both resource providers and resource consumers can exploit various economic models (such as commodity market models, etc.) for "trading" resources and establishing resource pricing strategies based on their objectives. While there are a few existing economic models used in grid systems, they do not yet provide a way to hedge against price increases/decreases for resources. In addition, the existing economic models do not provide an effective solution for dealing with variation in supply-and-demand for resources over time. It is these problems that I will address.

Many high performance computing applications (for example, in drug discovery and high energy physics, etc.) require very long term computations and have generally predictable future computational needs in terms of time and amount. (e.g. "I know that I will need 500 CPUs for 3 weeks at the beginning of September.") Such applications will benefit from economic models which not only provide flexible resource pricing but also offer effective risk tolerance. One way to provide these benefits is to introduce an option-like mechanism for pre-purchasing the right to use grid resources in the future. We know from the use of options in the real-world that they provide an effective means of risk tolerance. A grid economy model that supports options should provide flexible resource pricing and offer risk-tolerance to users during periods of variation in supply-and-demand for resources over time.

The goal of my research is thus to introduce and evaluate a framework for supporting options in grid resource management systems. I have implemented a simple, centralized "Grid Market Directory (GMD)" that supports a variant on a form of options from the world of commodity trading. An evaluation of the framework has also been done using a combination of the prototype implementation and simulation. The efficiency and practicality of my simple grid resource management system has been shown by comparing simulation results with and without the use of options for two key usage scenarios. Additionally, support for allowing resource users to re-sell unneeded resources has also been added and assessed. My experimental results show that the use of options in grid environments will provide benefits to resource providers and resource consumers allowing them to hedge against price increases/decreases for resources in a wide range of scenarios.

1.1 Organization

The rest of my thesis is organized as follows: Chapter 2 reviews key research related to my thesis. Chapter 3 motivates my work and describes the problem underlying it. Chapter 4 explains my proposed option contract approach for use in grid environments. Chapter 5 presents some of the prototype implementation details. A description of the simulations done is provided in Chapter 6. Chapter 7 presents my experimental results and discusses their statistical significance. Finally, Chapter 8 concludes the thesis and discusses some possible directions for future research.

Chapter 2

Related Work

This section gives an overview of grids, resource management/allocation in grids, grid economy frameworks, economy/market based resource management strategies as well as financial derivatives (such as options and futures).

2.1 Grids

A grid is a collection of distributed, typically high-end computing resources assembled dynamically to solve large-scale computational problems. Grids [9, 18] enable wide-area sharing through the selection and aggregation of geographically distributed autonomous resources. Such grid resources can be computers, databases or scientific instruments all of which may be used to solve large-scale problems in such fields as science, engineering, medicine and commerce. Grid resources are often selected dynamically at run-time based on resource availability, cost, capability and users' quality-of-service requirements. The dynamic selection of resources in grids poses challenges to resource management and scheduling complicated by the fact that the resources are typically geographically distributed and have different access policies, costs of use, capabilities and availability.

Building a successful grid requires low-level services (e.g. security, authentication, information discovery, and quality of service) as well as high-level services (tools for application development, resource management and scheduling) [18, 8].

Globus [17] is an example of grid software that provides basic infrastructure (i.e. low-level services) which can be used to construct higher level services such as programming tools and schedulers. The main component of the Globus system is the Globus Metacomputing [1] Toolkit (GMT). The GMT offers mechanisms such as communication, resource information management, resource location, resource allocation, authentication and data access. In particular, the Globus toolkit provides a service called the Metacomputing Directory Service (MDS) to manage resource information (including resource configuration details, resource performance information and application specific-information). Globus services are not aimed at direct application use. Instead Globus services are designed to facilitate the construction of higher level services which applications may then use.

2.2 Resource management

Resource management [18] is focused primarily on the processes of resource discovery and resource selection. Resources are discovered based on resource characteris-

[1]Metacomputing seeks to combine geographically dispersed and, typically, large-scale computing resources to solve very large and/or complex problems.

tics required by a potential resource consumer. Resources are then selected (from the discovered resources) based on criteria such as availability, reliability, performance, cost, etc. Resource management in grids poses many challenging issues [16] such as maintaining site autonomy, supporting the heterogeneous nature of the network substrate, policy extensibility, resource allocation and/or co-allocation issues, the need for on-line control and support for computational economies [10]. A number of systems have been developed to address some of the challenging issues associated with using grids. These include Condor, Condor-G, NetSolve/GridSolve, Nimrod, Nimrod/G etc.

2.2.1 Condor

Condor [27] is a resource management system mainly designed to support high throughput computing (HTC) rather than high performance computing (HPC) [2]. Condor locates idle workstations and then schedules jobs onto those idle workstations. The job scheduling in Condor lies between a static centralized approach and a distributed approach. Condor uses the Remote Unix (RU) [27] facility to execute remote jobs. The main feature of RU is its checkpointing mechanism which ensures job progress by saving the state of a running program when the remote machine executing the job fails or is forced to stop execution. The checkpoint can then be used to allow the job's execution to be resumed on another machine using the state saved in the checkpoint. Condor provides fair access to resources using the Up-Down algorithm [27, 28]. The Up-Down algorithm allows users to access resources

[2]HTC focuses on getting many sequential jobs done where as HPC focuses on getting one or more parallel jobs done quickly.

based on workload so that heavy users will get steady access to resources whereas light users will get fair access to resources. The Up-Down algorithm achieves this by maintaining a scheduling index at each workstation. Initially, the index for each workstation is set to zero and it will be updated periodically based on the load assigned to that particular workstation (i.e. the index is increased when a remote job is assigned to a particular workstation and the index is decreased when a workstation wants more remote jobs - this occurs when a workstation completes one or more of its current jobs). When a workstation with a high index needs to run a new local job and it has insufficient capacity, then the lowest priority remote jobs on the workstation are preempted and check pointed so that the corresponding compute capacity can be assigned to execute the local user's jobs.

2.2.2 Condor-G

Condor-G [19] combines features of both Globus and Condor to harness multi-domain resources [3]. Condor-G uses the Globus toolkit protocols to support remote resource access and Condor features to support computation management and remote job execution. Resource allocation in Condor-G is performed using the matchmaker framework [31]. This matchmaking framework is based on a classified advertisement (classad) mechanism.

The Matchmaker framework is divided into five components; classad specification, advertisement protocol, matchmaking algorithm, matchmaking protocol and claiming protocol. The matchmaker framework uses a semi-structured data model

[3]Resources from different administrative domains with different access policies.

for specifying classified advertisements (classads) which allow providers/consumers to specify resource configurations/requirements and to publish queries for desired resources. The matchmaking mechanism performs resource allocation in two phases, the matching phase and the claiming phase. In the matching phase, the matchmaker algorithm matches providers and consumers based on the constraints specified in their classads and then informs the relevant entities of the match. In the claiming phase, the matched entities contact each other directly for further negotiations (if there are any) and then they co-operate with each other to effect the required service.

2.2.3 NetSolve

Casanova and Dongarra [15] designed a system called NetSolve, which allows users to access resources distributed over a network and hides all the implementation details from the users. One of the major goals of NetSolve was to provide interactive as well as programming interfaces. Interactive interfaces support tools (such as MATLAB) and therefore require little or no programming effort from the user thus facilitating easy use by non-computer scientists. Programming interfaces support programming languages (such as FORTRAN and C) but require more programming effort from the user.

The NetSolve system has three main components. They are the NetSolve Client, the NetSolve Agent and NetSolve resources. The NetSolve Client sends a service request to the NetSolve Agent. The NetSolve Agent chooses the best available

NetSolve resource(s) [4], "solves a problem and returns the answer to the client [15]". The service request is structured in the form of triple: <name, inputs, outputs> where name is a character string, inputs is a list of input objects and output is a list of output objects. In NetSolve, an object is decribed as <objecttype, data> where objecttype can be MATRIX, VECTOR etc. and data can be any standard scalar data type.

Over time, NetSolve has been extended to address various limitations such as request sequencing (corresponding to task execution dependencies) and security. Although, the current version of NetSolve [35] deals with many issues, some desirable enhancements were not adequately addressed, including support for use in large-scale distributed systems. Thus, NetSolve was re-designed to create a new system known as GridSolve [39].

GridSolve employs NetSolve as a primary building block to create the necessary middleware to access widely distributed resources. The major goal of GridSolve is to provide a mechanism to interact with existing/emerging grid services. GridSolve can either have a client communicate with various grid services or can have server which acts as a proxy to other grid services. In GridSolve, the client-side approach is supported by the GridRPC API [34] and in the server-side approach, GridSolve delegates the scheduling to specialized scheduling and execution services such as Condor, etc.

[4]NetSolve Resource(s) are chosen based on the size and nature of the problem.

2.2.4 Nimrod

Abramson et al. [3] introduced a tool called Nimrod which is used to perform parameterised simulations [5] on distributed workstations. Nimrod provides features for remote job execution (like Condor) as well as a declarative parametric modeling language that is used to specify the various parameters for the set of simulations. Nimrod takes overall responsibility for distributing the simulation jobs and gathering their results after performing the remote computations.

Nimrod has a number of limitations when it comes to grid computing. Nimrod can only be used with a static set of resources since it does not support dynamic discovery of resources. Also, unlike, Nimrod/G (discussed next), Nimrod does not provide mechanisms to allow users to specify deadlines for executing applications. In addition, Nimrod lacks advanced security and access control mechanisms.

2.2.5 Nimrod/G

Abramson et al. [2, 11] also developed a system called Nimrod/G to overcome the limitations of Nimrod. Nimrod/G is a version of Nimrod that uses Globus [17] services to discover resources dynamically and to provide security and access mechanisms. Nimrod/G uses static cost methods in selecting resources whereby the system tries to discover resources based on user specified costs and deadlines. If no match can be found, the system tries to find the next cheapest resources that meet cost and deadline constraints. In Nimrod/G, static cost methods are a major drawback to the system as the cost of resources may change over time due to

[5] Application executions over a range of parameters.

variation in supply-and-demand for resources.

Buyya et al. [10, 1] propose a new version of Nimrod/G to overcome the limitations of the use of static cost methods in selecting resources. Their version of the Nimrod/G toolkit/resource broker uses Globus [17] services to provide essential low-level mechanisms and uses GRACE [10, 12] (Grid Architecture for Computational Economy) services (explained later) for trading resources based on cost and deadlines. The major difference in this version of the Nimrod/G [10] architecture compared to earlier versions of Nimrod/G [11] is the "trading manager" which is responsible for resource negotiation.

2.2.6 Limitations of Existing Resource Management Systems

Existing systems such as Globus, Condor and Condor-G, etc. address some of the challenging issues associated with using grids. However, none of them provide a "computational economy" framework in which resource providers can sell their resources and resource consumers can buy resources to solve their computational problems. Computational economies motivate resource owners to contribute their resources for others to use thereby facilitating development and management of large scale computational grids. They also offer the potential to support various types of economic models for trading resources and establishing resource pricing strategies.

Existing systems such as Condor and NetSolve follow a system-centric approach (scheduling decisions are based on system-centric parameters such as timeframe for executing jobs) rather than a user-centric approach (scheduling decisions are based on user's quality of service requirements such as cost for executing jobs). The system-centric approaches aim to increase system throughput and utilization ignoring access costs and thereby, sometimes, end-user concerns. On the other hand, user-centric approaches aim to increase system utilization while considering user's quality of service requirements including cost and deadlines, etc. User-centric approaches are generally driven by economy-based resource management.

Finally, most existing systems do not consider the regulation of demand and supply for resources, which can be very important in grid resource management because the cost of resources may change depending on demand-and-supply for resources over time.

2.3 Grid Computational Economy Models

A grid economy [10, 12] offers incentives (i.e. payment) for resource providers to contribute their resources for others to use and also provides a basis upon which resource consumers can choose resource providers to best meet their requirements.

Some of the benefits of grid economies (i.e. economy-based resource management) are [10]:

- A grid economy offers an incentive for resource providers to contribute their resources for others to use thereby facilitating development of large scale grids.
- A grid economy facilitates resource consumers being able to express their requirements and objectives such as cost, deadline, etc. for solving problems.

- A grid economy provides a means to regulate supply and demand of resources.
- A grid economy facilitates the development of user-centric scheduling policies instead of system-centric scheduling policies.
- A grid economy provides mechanisms for resource trading that can be used to ensure efficient allocation, management and use of resources.

In a grid economy, resource providers want to maximize resource utilization and profit while resource consumers want to solve their problems at low cost and within a required timeframe. Various economic models [13] for trading resources and establishing resource pricing strategies based on their objectives can be considered for use in grids.

2.3.1 Economic Models in Grid Computing

Based on existing economic principles, numerous economic models have been considered [13] for trading resources and establishing resource service pricing strategies in computational environments (including grids). They include:

- Commodity Market Model: In this model, the resource providers competitively set resource prices based on supply and demand for resources and charge resource consumers according to resource consumption.
- Posted Price Model: In this model, the resource providers set special offer prices for resources to attract resource consumers and then charge the resource consumers as per their resource consumption.

- Bargaining model: In this model, resource providers and resource consumers negotiate with each other to determine resource service prices and resource usage to meet their objectives.

- Tender/Contract Model: In this model, resource consumers invite "sealed bids" from resource providers (or resource providers invite "sealed offers" from resource consumers) and then resource consumers select the resource providers who offer the best services according to the resource consumers' objectives (or vice versa).

- Auction Model: There are also several possible auction models based on either closed bidding (where participants do not know the bid values of other participants) or open bidding (where participants do know the bid values of other participants) that may be applicable. They are:

  - English Auction (First-Price, Open Cry): In this type of auction, participants are free to place bids exceeding other participants' bids and the auction ends when no more participants are ready to place bids. The highest bid participant is the winner.

  - First-Price, Sealed-bid Auction: In this type of auction, participants can place only one bid without any knowledge of other participants' bids and the highest bidder is the winner.

  - Vickrey Auction (Second-Price, Sealed-bid Auction): In this type of auction, participants can place only one bid without any knowledge of other participants' bids. The highest bidder is the winner and pays the bid price placed by the second highest bidder. Thus, Vickery auctions encourage bidders to place high bids as bidders already know that they will not pay the higher value but will pay a price closer to market valuation.

- Dutch Auction: In this type of auction, the auctioneer starts the auction process with the highest desired (by the provider) price and lowers the price until one of the participants takes the item for the lowered price.
- Double Auction: In this type of auction, resource providers ask for a given price for their resources and resource consumers offer bids for the resources. The auction process continues (double or twice in this context) until an auctioneer matches a resource providers' asking price with a resource consumers' bid.

In addition to the economic models just described which are based on human activities, there are also models that have been proposed for computational systems alone. They are:

- Bid-based Proportional Resource Sharing Model [13]: In this model, the resource share given to a resource consumer is proportional to the bidding value of the resource consumer compared to other resource consumers' bids. This model is very popular in single administrative environments.
- Community/Coalition/Battering Model: In this model, all participants act as both resource consumers and resource providers sharing each other's resources thereby creating a cooperative resource sharing environment. The resource owners contributing their resources to a common environment will

also get access to the resources (from the common environment) when needed. Resource share to the contributors can be decided by employing any other economic model.

- Monopoly/Oligopoly Model: In the monopoly model, a single resource provider for a particular service dominates the market and sets the price of resources. The resource consumers do not have the opportunity to negotiate the resource price. In the oligopoly model, a few resource providers exist and collectively set the resource price. Such approaches commonly occur in services like telecommunications and Internet access, etc.

2.4 Market Based Resource Management Systems

A number of systems have been developed that, to some extent, support market based resource management. These include POPCORN, Java Market, JaWs, GRACE, and the Compute Power Market.

2.4.1 POPCORN

Nisan et al. [30] developed a system called POPCORN which provides a market-based model for buying/selling CPU time for computations. POPCORN uses Java applet technology to implement the POPCORN programming paradigm (where computations are decomposed into sub-computations called "computelets" to execute on remote processors). This approach "supports portability, inter-operability and safety in global computing [30, 25]".

POPCORN has three entities [32, 30]: markets, CPU-time buyers, and CPU-time sellers. A market matches the buyers and sellers of CPU-time based on sealed-bid auction mechanisms. The buyer writes an application in Java using the POPCORN paradigm and the seller visits a URL to offer resources to the buyer.

Unfortunately POPCORN cannot be used with non-Java applications. Also, POPCORN neither provides machine profiles of sellers nor supports incremental development to include additional functionality in the system.

2.4.2 Java Market

Amir et al. [5] have introduced a market-oriented system called Java Market which allows organizations or Internet users to harness the computational power of idle machines. Java and Java capable browsers are the primary building blocks of Java Market.

Java Market consists of three main entities as shown in Figure 2.1; Resource Managers, Task Managers and Market Managers. A resource manager is responsible for keeping track of producers' machines, a task manager is responsible for keeping track of consumers' tasks, and a market manager acts as a mediator between resource managers and task managers.

Once a resource provider registers with the market by visiting its URL, the resource provider deploys a "Launch Applet". The launch applet informs the resource managers about the provider's machine's power and other characteristics. Similarly, once a resource consumer registers with the market by visiting the URL of the market, the resource consumer deploys a "Request Applet". A request applet collects information about the tasks to be performed and forwards it to a task

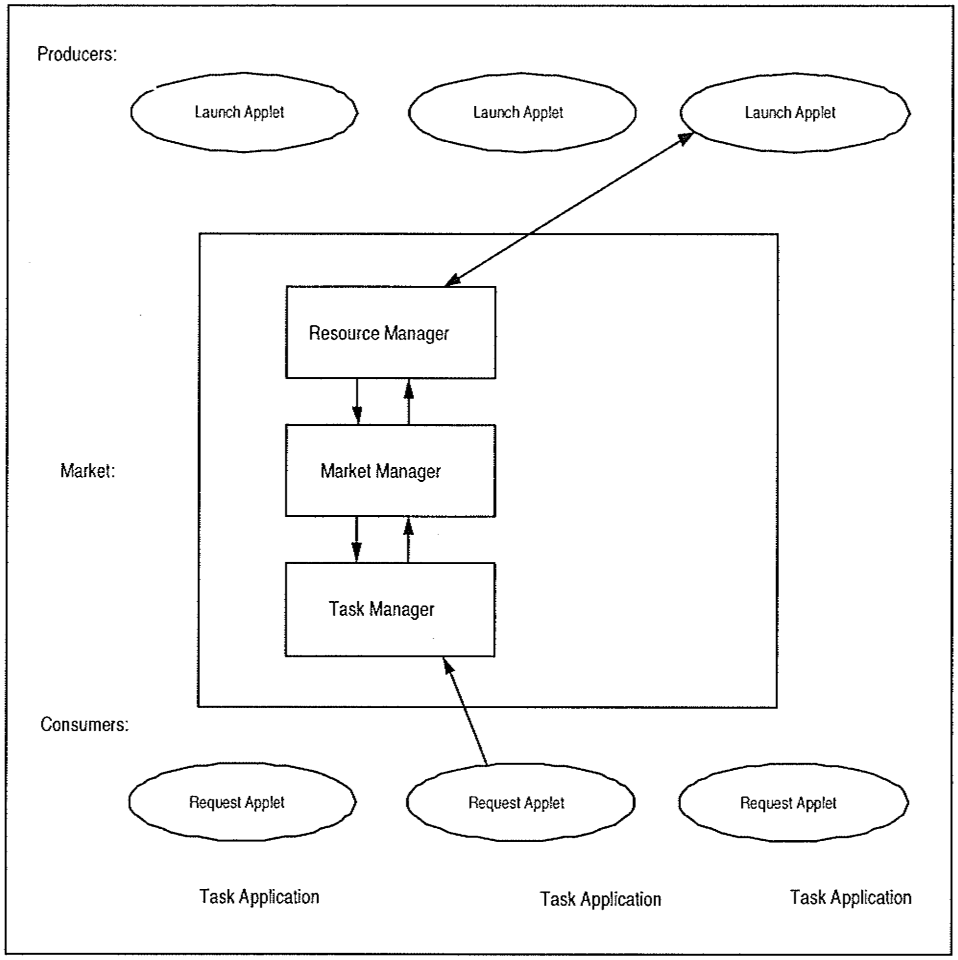

Figure 2.1: The Java Market Architecture, adapted from [5].

manager. The task manager collects all the files related to the user tasks and then passes the tasks to the market manager. The market manager performs admission control and resource allocation (i.e. assigns tasks to providers) using a combination of a cost-benefit framework [4] and winner picking strategy [5, 6] [6].

The cost-benefit framework is mainly designed to maximize the system's per-

[6]Resource providers are selected based on evaluating resource providers previous performance in computing tasks and anticipated future performance

formance. The main concept behind the cost-benefit framework is to evaluate the opportunity cost of a resource and the benefit earned from executing tasks. The cost-benefit framework tries to maximize the benefit and minimize the cost (i.e. if the benefit is higher than its cost then the task is admitted). Similarly, tasks with less benefit will be either rejected or delayed, depending on the system load. Also, using the cost-benefit framework, resource providers are paid according to the services they provide and resource consumers are charged according to the consumption of resources.

2.4.3 JaWs

Lalis and Karipidis [25] presented an economy-based web-computing model called JaWs (Java Web-computing system) which enable producers/consumers to export and access geographically distributed resources easily using their Java framework.

A JaWs framework consists of a market server, resource providers (hosts), host agents, schedulers, and resource consumers (clients). In JaWs, resource providers ask prices for their resources and resource consumers place bids for the resources in the market. The market server finds matches and produces a lease between the resource providers and resource consumers to provide/use resources for a particular amount of time and at a particular price. Resource providers deploy a "host agent" from the market to the provider's resource and the host agent is responsible for downloading and executing the consumers' tasks on the providers' resource(s).

JaWs uses the double auction method to match resource providers and resource consumers and charges the resource consumers according to their use of resources. JaWs reduces the effort required by the programmer in developing applications.

JaWs can support many architectures and operating systems and the JaWs framework can be easily extended to support modern distributing computing paradigms and to introduce additional functionality (e.g. checkpoint mechanisms for fault tolerance).

The Jaws system, however, only supports a single market server to accommodate producers and consumers. Having a single market server in such a system may introduce a bottleneck as the number of producers and consumers increases.

Although, web-based computing systems such as POPCORN, Java Market and JaWs follow computation economy strategies, they each support only a single economic model (i.e. POPCORN supports only sealed-bid auction mechanism, Java Market supports only its cost-benefit framework and JaWs supports only the double auction mechanism). In addition, to use these systems, user applications have to be written using Java (for Java Market and JaWs) and also according to the POPCORN programming standards (for POPCORN), which is a major limitation.

2.4.4 GRid Architecture for Computational Economy (GRACE)

Buyya et al. [10, 12] introduced a novel grid architecture called GRACE (Grid Architecture for Computational Economy) to support the use of various economic paradigms for resource management and scheduling in grids. GRACE can employ different economic models for establishing resource service pricing strategies. The main components of the GRACE architecture are [10, 12]:

- Grid Users: A grid user interacts with the grid using grid resource brokers (user software agents). The users in the grid environment have their own brokers who aid them in finding appropriate resources based on user requirements in terms of budget and deadline.

- Grid Resource Brokers (user-level middleware services and tools): Grid resource brokers work as mediators between the grid users and grid resource providers by making use of grid middleware services. The grid resource brokers hide the complexity of resource management and scheduling from the users.

- Grid Middleware Services: Grid middleware services provide services such as authentication, resource information, storage and access, etc. which are needed to make use of distributed resources. GRACE uses some of the Globus [17] components to offer its middleware services for matching grid users and grid resources via grid resource brokers. In addition, the GRACE middleware infrastructure provides trading protocols to specify trading regulations (for example, grid economic models).

- Grid Resource Providers: Grid resource providers provide key components like trade servers (to negotiate with users/resource brokers), pricing strategies (to establish resource prices) and resource accounting/charging (to record resource usage and charge users according to the negotiated prices).

While supporting multiple economic models, GRACE does not provide an effective solution for dealing with variation in supply-and-demand for resources over time which is likely to be very important in commercializing grid computing.

2.4.5 Compute Power Market

Metacomputing systems such as Globus and Legion concentrate on high-end resources such as supercomputers and clusters to perform large scale applications but do not consider harnessing less powerful but plentiful lower-end machines such as personal computers and workstations. Although, the GRACE architecture was designed with high-end systems in mind, GRACE can also be adapted to support low-end machines given certain changes in the infrastructure and middleware services [14].

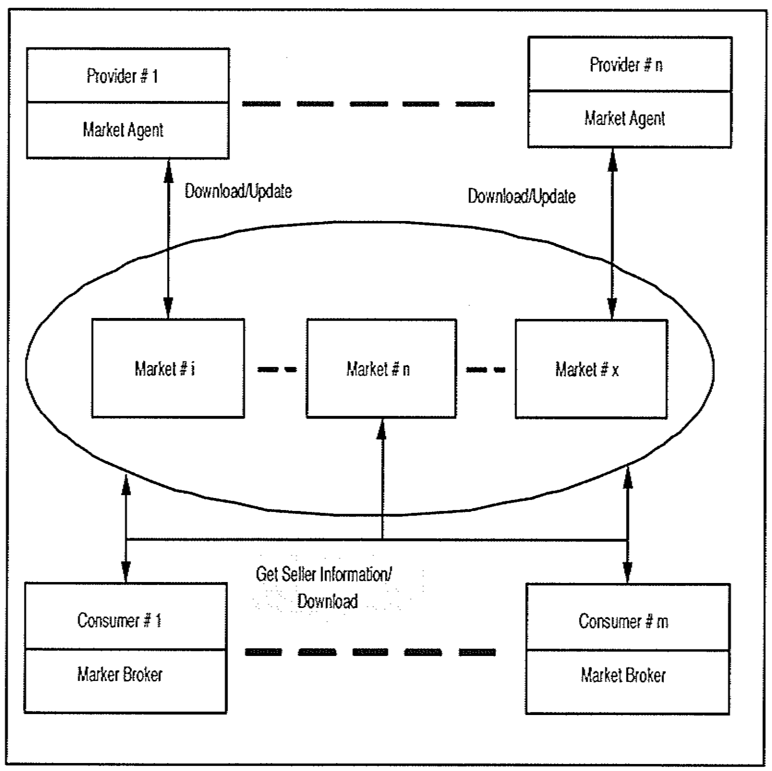

Figure 2.2: The Compute Power Market, adapted from [14].

Buyya and Vazhkudai [14] designed a market-based resource management and scheduling system called the Compute Power Market (CPM) which is mainly targeted at low-end machines. The CPM consists of markets, resource providers, and resource consumers (as shown in Figure 2.2). A market acts as a mediator between consumers and providers. As in other systems, consumers and providers register with the market to specify resource requirements and availability respectively. Once providers register with the market, a "Market Resource Agent" is deployed from the market to the provider's resource. The market resource agent aids the providers in updating the market about the provider's resources. Market resource agents are also responsible for accepting, deploying and launching jobs on resources. Similarly, once consumers register with the market, a "Market Resource Broker" is deployed from the market to the consumer's workstation. A market resource broker aids the consumer in choosing the appropriate resource provider(s) using the information provided by the market. CPM supports three economic models (the commodity market model, contract/tendering model, and auction model, as discussed earlier).

Existing economic models used in grid systems (such as GRACE and CPM) do not provide a way to hedge against price increases/decreases for resources. Being able to do so will be of interest to large scale users of grid systems who have repeated and relatively predictable future need for computing resources over relatively large time periods. In addition, the existing economic models used in CPM, like those in GRACE, do not provide an effective solution for dealing with variation in supply-and-demand for resources over time. Accordingly, derivatives (such as options and futures) may be interesting to consider as an economic model for grid resource management.

2.5 Derivatives

A derivative [20, 37, 36] is a financial instrument whose value depends on the price of an underlying asset. Futures and options are the most common types of derivatives and are actively traded on many financial exchanges.

Both "forward contracts" and "future contracts" define agreements between two traders to buy or sell an asset at a certain future time for a certain price. The main difference is that forward contracts are traded in "over-the-counter markets" while future contracts are traded only via exchanges. In the context of grid computing, the terms forward contracts and future contracts will be considered to be interchangeable.

An option [20, 37, 36] is a contract which gives the right (but not the obligation) to buy (a "call option") or sell (a "put option") an underlying asset within a specified time for a specified price. The specified price in the option contract is called the "exercise price" (or the "strike price") and the specified date in the option contract is called the "expiration date" (or "maturity date") of the contract.

It is important to note that an option contract gives the holder the right but not the obligation to buy/sell an asset whereas with forward/future contracts, the holder is obligated to buy/sell an asset. Also there is a cost associated with acquiring an option contract (called the premium price) whereas forward/future contracts do not have any cost to enter into them.

Investors will purchase a "call option" when they believe that the commodity price will rise in the future well above the strike price, to obtain the profits (which must be greater than the premium price paid to acquire the option). Investors will purchase a "put option" when they beleive that the commodity price will fall in the future well below the strike price, to obtain profits which, again, must exceed the premium price.

There are two major types of options based on exercise rights. They are American-style options and European-style options. American-style options give the holder the choice to exercise an option any time before the expiration date while holders of European-style options are restricted to exercise the options only at the time of expiration.

For example, consider a three month European style put option with strike price of \$20 on 10,000 barrels of oil (at \$20 per barrel) with some premium value, \$50,000. A put option gives the holder the right, but not the obligation to sell the issuer 10,000 barrels of oil for a price of \$20 at a specified date (three months from the date of purchasing the option in this example). The option holder will not sell the oil if the price is higher than the strike price in an open market. However, the option holder still loses the \$50,000 premium paid in acquiring the put option. On the other hand, the issuer should buy the oil if the holder is willing to sell the oil at the specified price if it is lower than the strike price in the open market.

Also, consider a three month European style call option with strike price of \$20 on 10,000 barrels of oil (at \$20 per barrel) with some premium value \$50,000. A call option gives the holder the right, but not the obligation to buy 10,000 barrels of oil (from the issuer) for a price of \$20 at a specified date (three months in the future in this example). The option holder will not buy the oil if the price is less than the strike price in an open market. However, the option holder still loses the \$50,000 (premium price) paid in acquiring the put option. On the other hand, the issuer should sell the oil if the holder is willing to buy the oil at the specified price if it is higher than the strike price in the open market.

Investors and businesses currently depend heavily on options for their financial risk management. Options help them to hedge against price increases/decreases for commodities. Options on stocks, commodities and foreign currencies, etc. are very common in financial markets. With the increase in the commercialization of grid computing, introducing an option-like capability may prove useful in grid environments as well.

Chapter 3

Motivation and Problem Description

In a financial market (or in a grid market place), the price of goods (resources) will naturally tend to increase when there is a shortage in supply and the price of goods will decrease when there is excess supply. Also, many high performance computing applications (for example, in drug discovery and high energy physics, etc.) require very long term computations and have generally predictable future computational needs in terms of time and amount. Such applications will benefit from economic models which not only provide flexible resource pricing but also offer effective risk tolerance in terms of pricing.

Some existing grid systems do support grid economy models but ignore resource reservation. Other existing grid systems support resource reservation but ignore grid economies, still others support neither grid economy models nor resource reservation. Moreover, existing economic models [13] used in grid systems do not provide any means to hedge against price increases/decreases for resources. It seems reasonable to expect that both resource consumers and resource providers will be interested in using an economic model in grid computing which not only provides some techniques to hedge against price increases/decreases for resources but also offer flexible resource pricing and supports resource reservation in advance.

A grid economy model that supports option-like capabilities will offer risk-tolerance to users during periods of variation in supply-and-demand for resources over time. Also, with an option contract model, service providers can maximize their resource utilization and profit even when there is an excess supply of resources which is not possible with existing economic models. Moreover, similar to other economic models, an option contract model will provide flexible resource pricing and assure resource availability within anticipated time frames and for specific resource requirements

For example, consider a scenario where a user wants certain resources in 6 weeks. He intends to reserve the resources which are valued at \$100,000 for 3 weeks and is prepared to pay some premium value (e.g. \$1000) to do so. The user must somehow discover those resource providers that can both provide resources after 6 weeks and who also will charge \$100,000 or less for resource utilization for the period of 3 weeks while requiring a premium value of \$1000 or less. Once discovered, the user can interact with the selected resource providers and negotiate a deal that gives an option to use the resources in 6 weeks for a price of \$100,000. The user pays \$1000 to the service provider (as a premium value) for acquiring the option contract. In the grid context, paying the premium value will also serve to make an advance resource reservation.

Subsequently, there are two situations that might arise. These are illustrated in the following two examples:

- The user discovers that the value of the same resources is now $200,000 in an open market. He may choose to immediately exercise the contract and use the resources within the available resource duration (when the work is ready). In this case, the user saves $99,000 ($200,000-$100,000-$1,000) compared to the current market price. Alternatively the user could sell the resources to other users (if he no longer needs them or his work is not ready to use the resources) to make a profit at the open market value that can be spent later to acquire later services.

- The user discovers that the value of the same resources is now $50,000 in an open market. Then the user is not under obligation to go through with the contract. However, the user still loses the $1,000 option premium. In this case, the service provider will benefit from the premium value of $1,000 paid by the user. However, the service provider still loses the potential of $49,000. Given that the user can now acquire the same resources at a price of $50,000 instead of $200,000, the loss of the $1,000 premium is insignificant.

While, to the best of my knowledge, no systems have been built that incorporate options into grid economy models, in [9], Buyya suggests that there is a need for different economic models to support futures markets in grids and in [13], Buyya et al. state that derivatives such as options and futures may be interesting to consider as economic models for a grid economy. In addition, Newhouse and Darlington [29] suggest that speculative models (e.g. derivatives such as options and futures) can be included in a computational economy for trading resources. Further, according to Buyya et al. [11], reserving resources in advance helps users to get the best value for their money. Finally, Kenyon and Cheliotis [22, 23] state that forwards/futures and options in grid environments will play a vital role in providing resource reservation and price certainty. All these arguments support the contention that options are worth investing for grid resource management.

As grids are now becoming more mature and their use is increasing, it seems to be an appropriate time to explore the use of an option model in grid resource management. Further, interest in commercial "pay-for-use" grids is now growing with the deployment of such systems as those provided by United Devices [41]. The large-scale commercialization of grid computing will eventually require something like options to provide a means of competitive and flexible resource pricing.

To be able to use option-like mechanisms to obtain these benefits, an extended framework based on the general ideas introduced in GRACE, CPM etc. must be created. Additionally, different option based policies for grid resource management must be defined and assessed to determine their effectiveness in providing price certainty, risk tolerance, etc. Finally, real systems will have to be built to allow for testing in large-scale, real-world settings and to permit assessment of system performance, robustness, etc.

My first research contribution will be to introduce a framework for supporting options in grid resource management systems. My additional contributions in this research will be to provide a basic system implementation based on the framework and initial policies for resource management in grid environments using an option contract model and apply them in a simulation study to do a preliminary assessment of the benefits of options in grid resource management.

Chapter 4

Solution Strategy

In existing commodity/stock markets, if a user executes an option, then the user will become the owner of the corresponding asset. Of course, grid resources cannot be bought permanently like commodities in financial markets. Resource providers will only provide their resources to resource consumers for a limited time to solve computational problems. Therefore, my option contract framework for a grid economy model will be designed to allow resources to be purchased/sold for a limited time period.

I have chosen to use an existing economic model, the posted price model (as discussed earlier), to establish resource pricing (both the initial premium value to enter into an option contract and the underlying resource price). The bargaining model, tender/contract model or auction model could also be used to establish prices but the posted price model avoids the overhead of additional negotiations inherent in the other models. These overheads are mainly due to repeated bidding for resources by consumers or due to bargaining to determine the final price by resource providers and resource consumers. Avoiding repetitive bidding is an important efficiency consideration in a distributed/grid system. On the other hand, using the posted price model, resource consumers will not have an opportunity to negotiate for resources. Considering the pros and cons of many existing economics models, the posted price model seems to be both suitable and efficient for use in my framework. Additionally, it is the simplest of the models discussed and this will be beneficial when first introducing option concepts to grid users who are likely to be intolerant of complex systems.

The steps/activities involved in using my option contract based framework will be:

1. Users will approach a grid resource broker [1] to buy resources. They will specify one or more resource requirements such as number of CPUs, memory capacity, storage space and the anticipated resource usage duration, desired/maximum premium value, desired/maximum service price, etc. to the grid resource broker who will negotiate with service providers and will eventually return a set of resources that meet the user's needs or will indicate failure to do so.

2. Grid Service Providers (GSPs) will publish their available resource bundles [2], the initial premium value to acquire an option contract for the bundle, as

[1] A grid resource broker hides the complexity, in terms of messaging protocols, etc., of resource acquisition from the users.

[2] A resource bundle will consist of a number of CPUs, their memory capacity and storage space, etc.. Such bundles provide a convenient unit of resource allocation which is coarser than individual machines and which suits the physical hardware (since, in many cases, a single resource "target" will consist of multiple processors – e.g. a cluster). Negotiating for a few such bundles will also incur less overhead than negotiating for many individual resources.

well as the resource price and available duration for the bundle in a grid market place built around a Grid Market Directory (GMD). Like conventional market places, the grid market place embodies resource providers representing producers and resource brokers/users representing consumers and the GMD which facilitates resource trading between resource providers and resource consumers. The GMD will typically not just be one machine but will instead be a collection of machines with replicated information. This approach will provide enhanced availability, reliability and performance.

My GMD will be similar to Yu et al.'s [40] which offers grid service providers a mechanism to register themselves as resource providers and to publish their service descriptions along with their service prices so that grid resource brokers can discover resources that fit their users' requirements.

3. By querying the GMD, each grid resource broker will identify one or more GSPs who meet their users' requirements. If no GSPs are found at that moment, then the grid resource broker will periodically look again for GSPs that meet the users' requirements. In some cases, users may specify large resource requirements which will require the allocation of multiple resource bundles. Also, in some instances resource brokers may want to negotiate for part of a resource bundle (i.e. for less time or for less processes, etc.).

4. Once suitable resources have been located, the broker (or, if necessary, the user) will write an option contract [20] with the selected GSP(s) to use their resources at an agreed upon future time for an agreed upon price, premium, etc.

5. The broker (or user) will then pay the negotiated premium value to the GSP(s) to acquire the requested option contract and the GSP(s) will reserve the required resources. The payment mechanism can be implemented using any grid accounting system (e.g. GridBank [7]) Discussion of the details of such payment mechanisms is beyond the scope of this thesis.

6. Later, the user may use the resources and must then pay (using GridBank [7] or a similar facility) the GSP as per the contract value.

7. If the user no longer needs the resources (before the contract period), then the user can sell the reserved resources to other users. In this scenario, the user would then act as a resource provider, as described above, most likely through his/her broker.

8. If the user does not use the resources within the contract period and does not sell the resources to other users then the user will lose the premium value paid to the GSP and the resources will be left idle until the expiration of the reserved resource period. Accordingly, the premium value likely will have to be reasonably high so that the resource providers can get at least some minimum benefit from reserved resources if resource consumers do not use or re-sell the resources to other users. This will provide strong motivation for users to re-sell options they will not exercise.

9. Finally, if a resource provider over commits its resources, it can enter into the grid market to buy resources from another GSP to handle its over commitment (in this scenario, GSP's will act as resource consumers).

The usefulness of market-based models/mechanisms can be evaluated using one or more of the following criterion [33].

- Social Welfare: A market-based model results in *Social Welfare* if it provide benefits (in terms of payoffs or utilities) to all participating members.
- Pareto Efficiency: A market-based model is said to be *Pareto Efficient* if it makes one individual better off in terms of their own objectives without making other individuals worse off.
- Individual Rationality: A market-based model is said to be *Individually Rational* if it provides better individual payoffs to all participating members. In other words, each participating member should get better payoffs that are not less than what those members would get by not participating in the negotiation.
- Stability: Market-based model protocols should not be manipulatable by participating members (i.e. protocols should behave in the desired manner).
- Computational Efficiency: A market-based model should consume little computation time to ensure that as many resources as possible are left for executing user application.
- Distribution and Communication Efficiency: A market-based model should not create excessive communication overhead to avoid interfering with running applications.

My framework will provide desirable characteristics when used in grid environments. Notably, my framework is capable of supporting:

- Social Welfare: An option contract framework for grids provides benefits to all participating members (i.e. users and resource providers can hedge against price increases/decreases for resources respectively).

  For example, with options, users will have certainty of resource availability within the anticipated timeframe and price constraints when there is low supply of resources and resource providers will have certainty of selling resources when there is excess supply of resources.

- Pareto Efficiency: An option contract framework for grids provides better payoff to all participating members without making one individual better off and other individual worse off.

  For example, with options, users will not have any obligation to use the resources and can avoid paying the total price for resources bought on an option contract when the user's work is not ready to use the resources. At the same time, resource providers will benefit from the additional premium values paid by the users.

- Individual Rationality: Users and resource providers will get better payoff using an option model compared with what they would get by not using an option model.

  For example, users will always have the choice to re-sell resources to make profit (during high demand for resources) and resource providers can maximize their resource utilization and profit when there is an excess supply of resources.

Recall that, in existing financial markets, there are two types of options (call and put). In a grid environment, a put option does not support global welfare and pareto efficiency. For example, with a put option, the provider is not obligated to sell the required resources when the consumer needs them and thus consumer computations may be interrupted causing negative consequences. Thus, using an options-like mechanism in grid computing, it is reasonable to consider only call options rather than put options. This illustrates that the "options,, in my framework are not like conventional options in financial markets even though they share some of the underlying principles.

Chapter 5

Implementation

The Grid Market Directory (GMD) provides the core "matchmaking" service that pairs user requests with collections of one or more bundles offered by service providers, subject to constraints such as price and duration of resource usage. This chapter describes the data model (i.e. database structure) and operations (in terms of SQL queries) processed against the data stored by the GMD. This corresponds to the core code implemented by my prototype framework. The effectiveness of the prototype is assessed using simulation (integrating the prototype code), as described in the following chapters.

5.1 Components

For my research, I have designed and implemented the following components from scratch:

- A non-replicated Grid Market Directory to store resource consumers' requirements and resource providers' services and prices.

- Strategies to advertise resource consumers' needs and resource providers' services and prices.

- Strategies to match resource consumers with (groups of) resource providers.

Figure 5.1 shows a high-level architectural overview of my framework.

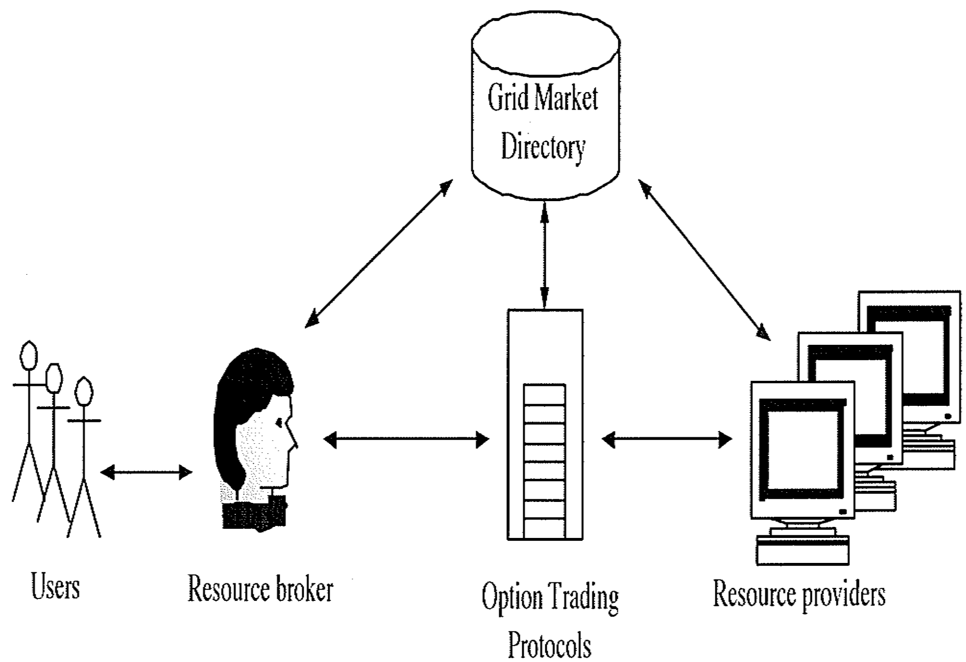

Figure 5.1: Architectural Overview

For my research, I used MS-Access to implement a Grid Market Directory (GMD) repository to store resource consumers' requirements, resource providers' published services and prices as well as all service agreements made between resource providers and consumers. I used Java and JDBC (Java Database Connectivity) to implement the matching strategies and to interact with the MS-Access database, respectively.

5.1.1 Data Structures

To implement a Grid Market Directory for the three different scenarios considered in this thesis (resource trading for immediate use, resource trading involving advance resource reservation and resource trading involving options), three similar, but slightly different data models were designed. Figure 5.2, Figure 5.3 and Figure 5.4 show the database tables and the relationships among them, for each of the three different test scenarios. The "Grid Resource Providers" table (in each figure) stores the resource providers' services and corresponding asking prices. The "Grid Resource Consumers" table (in each figure) stores each resource consumers' requirements. The "Grid Service Contract" table (in each figure) stores the service agreements (based on successful matchings) between resource consumers and resource providers.

The grid market directory implemented in my prototype is not replicated. Replicated strategies for GMD implementation are not needed to assess the effectiveness of using an options-like mechanism for resource allocation in grid systems. As with any distributed, replicated structure, the challenge in implementation will be ensuring effective, low-overhead, consistency, maintenance between the replicas.

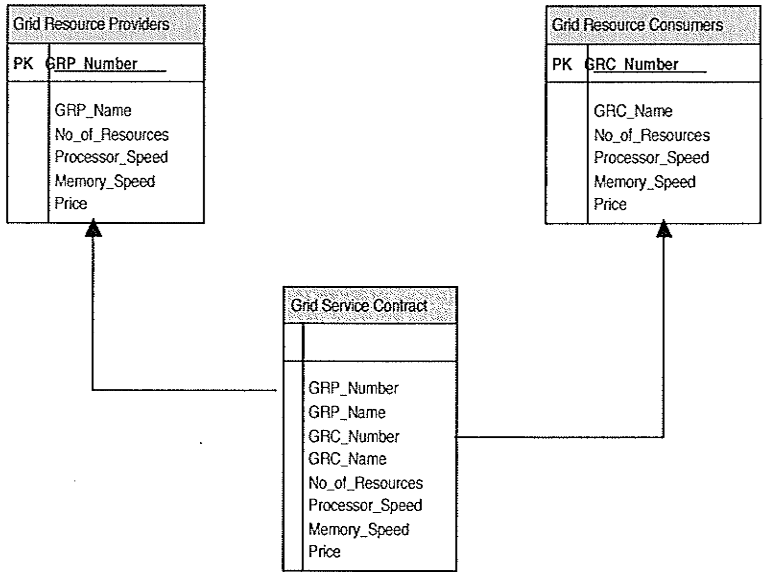

Figure 5.2: Database Model to support Resource Trading for Immediate Resource use by consumers

5.1.2 Advertising and Matching Capabilities and Requirements

Each provider must advertise the "capabilities" provided by the resources that he/she makes available for purchase and each user must advertise the "requirements" needed by his/her computation for which resources must be purchased. Such advertised information is, ultimately, stored in the GMD and, subsequently, matched by the GMD code. The format of advertisements and the mechanism(s) by which advertisements are delivered to the GMD (e.g. socket calls vs. RPC vs. RMI) do not affect the assessment. Further, the specific information that needs to be stored to do matching in two different implementations of the framework may vary. To illustrate the effectiveness of the options technique, only a simple model for

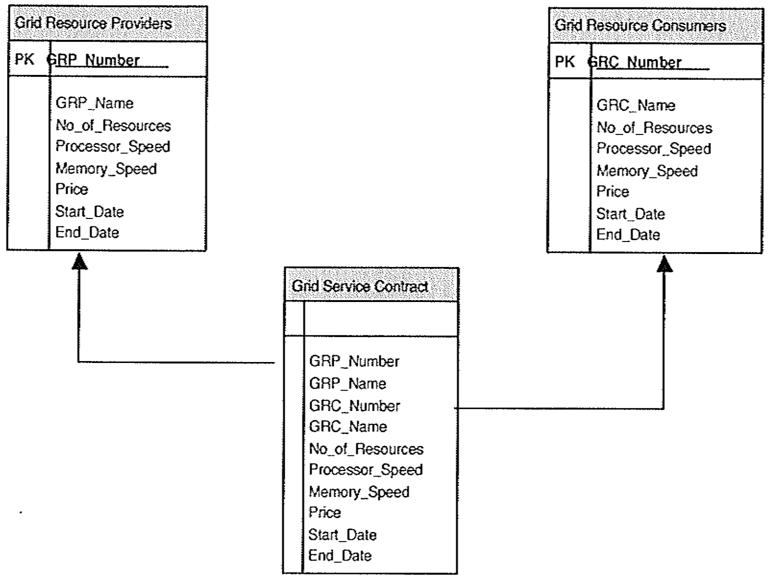

Figure 5.3: Database Model to support Resource Trading Involving Advance Resource Reservation

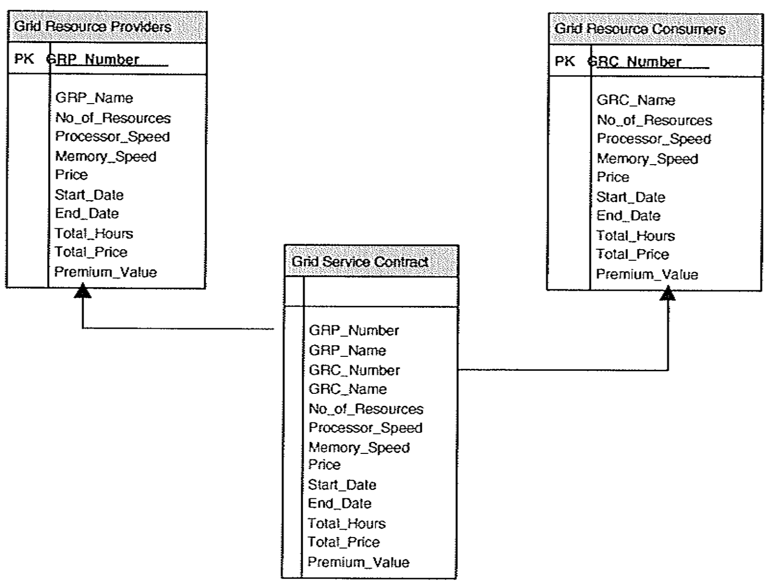

Figure 5.4: Database Model to support Resource Trading using Options

describing resources is used in my prototype. It considers the number of resources (CPU bundles), processor and memory speeds and the resource "rental period" and cost. Naturally, other criteria could be considered and different algorithms plugged into the framework to meet specific requirements.

The strategies used to match resource consumers' requests with (groups of) resource providers' resources in the prototype are implemented using the following queries. Query-1 is used in resource trading for immediate use. Query-2 is used in resource trading involving advance resource reservation. Query-3 is used in resource trading involving options.

Query-1:

SELECT Resource Providers

FROM Grid Resource Providers

WHERE (No_of_Resources >= Resource Consumer.No_of_Resources)

AND (Price <= Resource Consumer.Price)

AND (Processor_Speed >= Resource Consumer.Processor_Speed)

AND (Memory_Speed >= Resource Consumer.Memory_Speed)

ORDER BY Price ASC

Query-2:

SELECT Resource Providers

FROM Grid Resource Providers

WHERE (

(Start_Date <= Resource Consumer.Start_Date)

AND (End_Date >= Resource Consumer.Start_Date)

)

```
AND (No_of_Resources >= Resource Consumer.No_of_Resources)
AND (Price <= Resource Consumer.Price)
AND (Processor_Speed >= Resource Consumer.Processor_Speed)
AND (Memory_Speed >= Resource Consumer.Memory_Speed)
ORDER BY Price ASC
```

Query-3:

```
SELECT Resource Providers
FROM Grid Resource Providers
WHERE (
(Start_Date <= Resource Consumer.Start_Date)
AND (End_Date >= Resource Consumer.Start_Date)
)
AND (No_of_Resources >= Resource Consumer.No_of_Resources)
AND (Price <= Resource Consumer.Price)
AND (Processor_Speed >= Resource Consumer.Processor_Speed)
AND (Memory_Speed >= Resource Consumer.Memory_Speed)
AND (Premium_Value <= Resource Consumer.Premium_Value)
ORDER BY Price ASC
```

The focus of this thesis is on the discovery of available resources based on users' resource requirements and on the creation of option contracts not on the later use of the discovered resources. Accordingly, protocols supporting resource use are not discussed.

Chapter 6

Simulation

Since economy based resource management in grid settings is still in the very early stages of development [21], it is not possible to use real empirical data (based on actual demand and supply of resources) to evaluate novel grid economic models. Hwang et al. [21] suggest that the use of simulated markets will help in designing optimal market architectures and the results of the new market mechanisms could then be used to estimate the effects in real market settings when grids are commercialized. It is this approach that I take in this thesis.

There are also some extensive grid simulators such as MicroGrid [38] which provide "virtual grid infrastructure" that can be used to explore and evaluate the efficiency of grid environments. However, MicroGrid does not support economic models for managing computational resources. Some economic based grid resource management frameworks, such as GRACE and CPM, could conceivably be used to incorporate options and evaluate their benefits, but accessing a large scale grid test bed is impractical. For these reasons, I chose to simulate my framework to facilitate assessing the efficiency of a simple grid resource management system based on options. In particular, I used SSJ (Stochastic Simulation in Java) [26] to simulate my framework. SSJ is a software tool implemented using the Java programming language that facilitates easy simulation programming. Additionally, SSJ made it easy to integrate my implemented algorithms (which were written in Java) into the simulation system.

One of the goals of my framework is to simulate (using the implemented components) and evaluate the efficiency of a simple grid resource management system both with and without an options like capability. In such a scenario, efficiency can be defined as the total number of successful resource trading events between resource consumers and resource providers which is determined by how well resource consumers are finding the resources they require. Resource allocation efficiency was estimated in my simulations for the following three different assumed environments:

- Resource trading for immediate use by consumers.
- Resource trading to reserve resources for use at anticipated future times (i.e. resource trading involving advance resource reservation).
- Resource trading using options.

The above three different scenarios were measured and compared to each other for the following situations:

- High demand but limited supply of resources – where the number of resource providers was kept constant and the number of resource consumers was varied.

- High supply but limited demand for resources – where the number of resource consumers was kept constant and the number of resource providers was varied.

6.1 Simulation Structure and Components

To simulate my framework, I created three major modules that were integrated with the implemented components discussed in Chapter 5. First, a module was created for publishing resource consumers' needs in the Grid Market Directory according to a Laplacian distribution. The Laplace distribution (also known as the double exponential distribution) is widely used for modeling in the areas of finance and economics [24]. As my thesis shares some of the underlying structures of options in financial markets, I chose to use the Laplace distribution in my simulation study.

Second, a module was created for publishing resource providers' services and prices (also according to a Laplacian distribution) and finally, a module was created for matching resource consumers with (possibly groups of) resource providers using the specific strategies discussed Section 5.1.2. The resulting service agreements from this processing are stored in the Grid Market Directory.

6.1.1 Simulation Parameters

- Resource Providers: The total number of resource providers is given as input parameter.

- Resource Consumers: The total numbers of resource consumers is given as input parameter.

- Number of Resources: The total number of available resources (at resource providers) or requested resources (by resource consumers) is also an input parameter.

- Processor speed of each resource: In any given simulation, this parameter is generated using a Laplacian distribution.

- Memory capacity of each resource: In any given simulation, this parameter is generated using a Laplacian distribution.

- Resource Usage Period (specified by a Start Date and End Date)::The period of time during which a resource is available for use. The resource usage period is the time during which the resource providers are willing to sell their resources and during which the resource providers will buy access to the resources. In any given simulation, start date and end date are also generated using a Laplacian distribution.

- Resource Price (per hour) of each resource: The resource price is either fixed or varied (according to a Laplacian distribution) for different scenarios.

- Premium Price: The price paid by resource consumers to resource providers in acquiring an option contract. The premium price is computed based on the total resource value (i.e. as a function of resource usage period and resource price). In my simulation, I assume the premium price is 25 % of the total resource value.

The following subsections describe the simulation parameters in greater detail for each simulated scenario.

Parameters	No of Resource Providers	No of Resource Consumers	No of Resources	Processor Speed	Memory Speed	Resource Price
Fixed	1000 2000 3000 *in each run*	N/A	N/A	N/A	N/A	\$1.0/*Hr* – *Resource Providers Price*
Varied	N/A	1,000 *to* 20,000	*Between* 1 *and* 5000	*One in* (500, 1000, 1500, 2000, 2500, 3000)	*One in* (128, 256, 384, 512, 1024, 2048)	*One in* (\$0.5/*Hr*, \$0.75/*hr*, \$1.0/*hr*) – *Resource Consumers Price*

Table 6.1: Parameters used for Simulating Resource Trading for Immediate Use during Periods of High Demand but Limited Supply of Resources.

6.1.1.1 High Demand but Limited Supply of Resources

In these experiments, the number of resource providers will be held constant and the number of resource consumers will be varied. As the number of resource consumers increases, the market demand increases but the supply of grid resources is unchanged. Such situations may occur when many resource consumers concurrently try to meet strict deadlines to accomplish anticipated tasks (e.g. quarterly/annually targeted goals) or when there is some sort of "crisis computing". For example, in an academic setting, many researchers may require huge number of resources in a short period of time to carry out their experiments so that they can meet a publication deadline.

Parameters	No of Resource Providers	No of Resource Consumers	No of Resources	Processor Speed	Memory Speed	Resource Price	Resource Usage
Fixed	1000 2000 3000 *in each run*	N/A	N/A	N/A	N/A	\$1.0/*Hr* – *Resource Providers Price*	N/A
Varied	N/A	1,000 *to* 20,000	*Between* 1 *and* 5000	*One in* (500, 1000, 1500, 2000, 2500, 3000)	*One in* (128, 256, 384, 512, 1024, 2048)	*One in* (\$0.5/*Hr*, \$0.75/*hr*, \$1.0/*hr*) – *Resource Consumers Price*	*Between* 1 *and* 24 *Months*

Table 6.2: Parameters used for Simulating Resource Trading for Advanced Resource Reservation during Periods of High Demand but Limited Supply of Resources.

Parameters	No of R.P	No of R.C	No of Resources	Processor Speed	Memory Speed	Resource Price	Resource Usage	Premium Value
Fixed	1000 2000 3000 *in each run*	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Varied	N/A	1,000 *to* 20,000	*Between* 1 *and* 5000	*One in* (500, 1000, 1500, 2000, 2500, 3000)	*One in* (128, 256, 384, 512, 1024, 2048)	*One in* (\$0.5/*Hr*, \$0.75/*hr*, \$1.0/*hr*) –*R.P/ R.C Price*	*Between* 1 *and* 24 *Months*	25% *total resource value*

Table 6.3: Parameters used for Simulating Resource Trading with Options during Periods of High Demand but Limited Supply of Resources.

In Table 6.1 and Table 6.2, I set the resource provider's resource bundle price to be high (\$1.0/hr) since there is high demand for resources and I set the resource consumer's request price depending on various factors (including the number of resources, processor speed, etc). This helps to evaluate the behavior of resource consumers when there is no opportunity to hedge against high resource price situations (during high demand periods).

In real financial markets, option value is not easy to determine. The overall value of an option is determined by many factors such as strike price, current market price of the underlying asset, expiration date etc. Moreover, investors will choose to buy/sell options with different strike prices to limit losses while maximizing profits.

In Table 6.3, I set the resource provider's resource bundle price to various values (e.g. \$0.5/Hr, \$0.75/hr and \$1.0/hr). As options are primarily used to hedge against price variations, resource prices will tend to be somewhere between the high and low prices at a value that maximizes the participants' benefits. The actual price paid for a resource will fluctuate over time depending on user demand (among other things). For evaluation purposes, assessing the effects of having low, moderate and high prices should be sufficient to characterize behavior. Resource consumers' resource request prices are, again, set according to various factors (including the number of resources, processor speed, etc). This helps to evaluate how effective options are in hedging against price variations during high demand periods.

Parameters	No of Resource Consumers	No of Resource Providers	No of Resources	Processor Speed	Memory Speed	Resource Price
Fixed	1000 2000 3000 *in each run*	N/A	N/A	N/A	N/A	\$0.5/*Hr* −*Resource Providers Price*
Varied	N/A	1, 000 *to* 20, 000	*Between* 1 *and* 5000	*One in* (500, 1000, 1500, 2000, 2500, 3000)	*One in* (128, 256, 384, 512, 1024, 2048)	*One in* (\$0.5/*Hr*, \$0.75/*hr*, \$1.0/*hr*) −*Resource Consumers Price*

Table 6.4: Parameters used for Simulating Resource Trading for Immediate use by Consumers during Periods of High Supply but Limited Demand for Resources.

6.1.1.2 High Supply but Limited Demand for Resources

In these experiments, the number of resource consumers will be held constant and the number of resource providers will be varied. As the number of resource providers increases, the supply of grid resources increases but the market demand remains unchanged. Such situations may occur when resource consumers have little work to do so resource providers must try to sell resources at reduced prices to achieve some profit (e.g. to meet expected quarterly/annually profit levels).

In Table 6.4 and Table 6.5, I set the resource provider's resource bundle price to be low (\$0.5/hr) since there is limited demand for resources and I set the resource

Parameters	No of Resource Consumers	No of Resource Providers	No of Resources	Processor Speed	Memory Speed	Resource Price	Resource Usage
Fixed	1000 2000 3000 *in each run*	N/A	N/A	N/A	N/A	\$0.5/*Hr* *– Resource Providers Price*	N/A
Varied	N/A	1,000 *to* 20,000	*Between* 1 *and* 5000	*One in* (500, 1000, 1500, 2000, 2500, 3000)	*One in* (128, 256, 384, 512, 1024, 2048)	*One in* (\$0.5/*Hr*, \$0.75/*hr*, \$1.0/*hr*) *– Resource Consumers Price*	*Between* 1 *and* 24 *Months*

Table 6.5: Parameters used for Simulating Resource Trading with Advanced Reservation during Periods of High Supply but Limited Demand for Resources.

Parameters	No of R.C	No of R.P	No of Resources	Processor Speed	Memory Speed	Resource Price	Resource Usage	Premium Value
Fixed	1000 2000 3000 *in each run*	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Varied	N/A	1,000 *to* 20,000	*Between* 1 *and* 5000	*One in* (500, 1000, 1500, 2000, 2500, 3000)	*One in* (128, 256, 384, 512, 1024, 2048)	*One in* (\$0.5/*Hr*, \$0.75/*hr*, \$1.0/*hr*) *– R.P/ R.C Price*	*Between* 1 *and* 24 *Months*	25% *total resource value*

Table 6.6: Parameters used for Simulating Resource Trading with Options during Periods of High Supply but Limited Demand for Resources.

consumer's request price depending to various factors (including the number of resources, processor speed, etc). This helps to evaluate the behavior of resource providers when there is no opportunity to hedge against low resource prices when there is an excess supply of resources.

In Table 6.6, I set the resource provider's resource bundle prices between high and low price to various values (e.g. \$0.5/Hr, \$0.75/hr and \$1.0/hr) (similar to the high demand but limited supply situations as explained earlier). Resource consumers' resource request prices are, again, set according to various factors (including the number of resources, processor speed, etc). This helps to evaluate how resource providers can hedge against selling resources for low prices when there is limited demand for resources.

In any given simulation, each resource consumer makes only a single request (but each request may specify many resources). In the simulations, it does not matter whether ten requests come from one resource consumer or one request comes from ten resource consumers since the demand on the resource providers is the same and the effect on pricing is also the same.

Chapter 7

Results

This chapter presents my experimental results and briefly discusses their statistical significance.

7.1 Experimental Results

The primary goal of my research is to do a preliminary benefit assessment of the use of options in grid resource management. Recall that in the simulation experiments, resource allocation efficiency will be assessed in terms of the total number of successful resource trading events between resource consumers and resource providers. This makes sense since when the total number of successful resource allocations increases it mean that the resource consumers are finding the resources they need within their allocation constraints (i.e. price, capabilities and duration). Multiple experiments were repeated for each scenario (i.e. immediate use, advanced reservation and options) and the resulting successful allocation results were carefully analysed to avoid inaccurate or statistically insignificant results.

7.1.1 Efficiency of Simple Resource Allocation

The results presented in this section consider only resource providers and consumers that are entirely independent of one another. This assumption will be relaxed in later sections.

7.1.1.1 Resource Allocation Efficiency when there is High Demand but Limited Supply of Resources

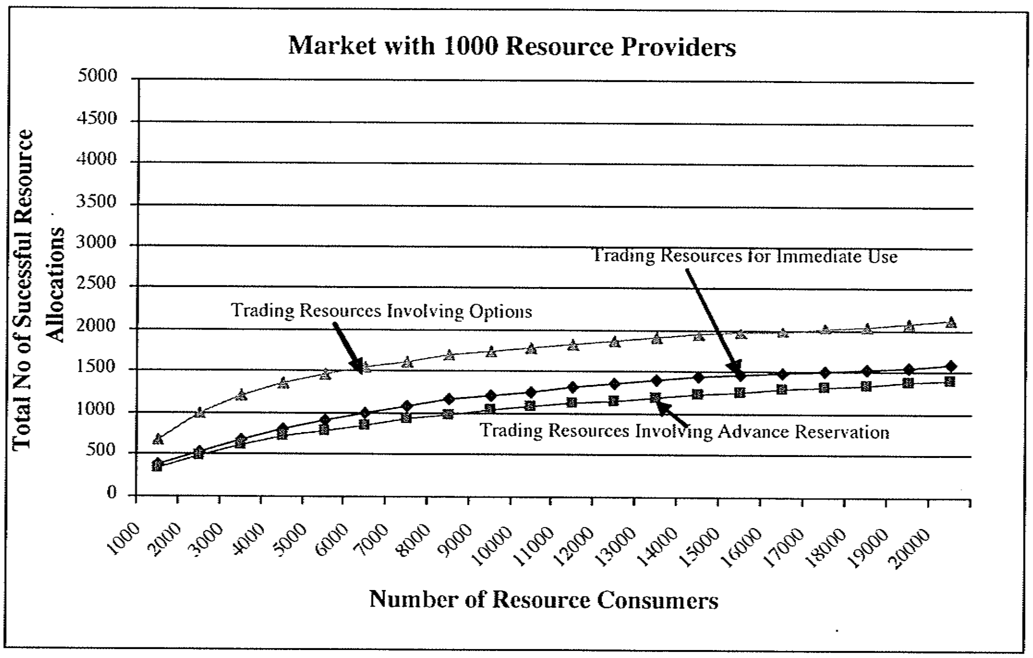

Figure 7.1: Resource Allocation Efficiency for the three Scenarios with 1000 Resource Providers.

Figures 7.1, 7.2 and 7.3 illustrate the behavior of the market with high demand but limited supply of resources (i.e. with more resource consumers and less resource providers) for each of the fundamental scenarios with 1000, 2000, and 3000 resource providers, respectively.

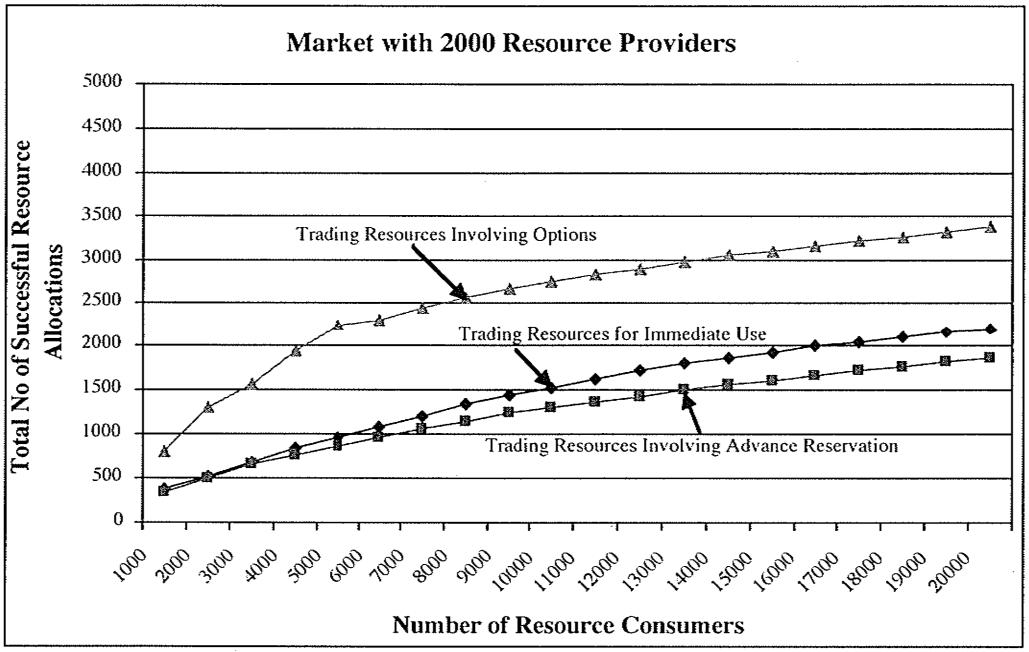

Figure 7.2: Resource Allocation Efficiency for the three Scenarios with 2000 Resource Providers.

In Figure 7.1, with less resource providers and more resource consumers, the number of successful resource allocations is reasonable for all three scenarios. With an increase in resource providers (see Figures 7.2 and 7.3), the number of successful resource allocations increases as more resource providers are available to accommodate more resource consumer requests. This, naturally, reflects the availability of more resources.

No of Resource Providers	No of Resource Consumers	Trading Resources for Immediate Use				Trading Resources with Resource Reservation				Trading Resources Using Options			
		Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources		
			0.5	0.75	1		0.5	0.75	1		0.5	0.75	1
1000	1000	385	274	0	111	339	243	0	96	667	371	283	13
1000	2000	524	302	0	222	483	281	0	202	992	452	493	41
1000	3000	665	324	0	341	618	309	0	309	1210	512	637	61
1000	4000	801	351	0	450	714	320	0	394	1352	546	725	81
1000	5000	897	357	0	540	784	328	0	456	1448	569	787	92
1000	6000	988	368	0	620	854	335	0	519	1544	599	842	103
1000	7000	1078	379	0	699	923	341	0	582	1612	607	894	111
1000	8000	1156	389	0	767	965	344	0	621	1678	625	928	125
1000	9000	1210	395	0	815	1032	350	0	682	1727	640	955	132
1000	10000	1252	396	0	856	1072	355	0	717	1775	652	986	137
1000	11000	1303	400	0	903	1109	359	0	750	1816	665	1008	143
1000	12000	1357	401	0	956	1147	363	0	784	1852	677	1029	146
1000	13000	1393	404	0	989	1191	369	0	822	1897	699	1059	139
1000	14000	1425	411	0	1014	1225	374	0	851	1934	704	1082	148
1000	15000	1455	417	0	1038	1252	377	0	875	1962	711	1096	155
1000	16000	1480	418	0	1062	1277	379	0	898	1986	714	1113	159
1000	17000	1501	420	0	1081	1309	382	0	927	2018	720	1137	161
1000	18000	1522	422	0	1100	1334	383	0	951	2055	729	1157	169
1000	19000	1548	425	0	1123	1362	388	0	974	2094	746	1176	172
1000	20000	1575	427	0	1148	1401	392	0	1009	2130	751	1204	175

Table 7.1: Price Paid by Resource Consumers during three Scenarios with 1000 Resource Providers.

No of Resource Providers	No of Resource Consumers	Trading Resources for Immediate Use				Trading Resources with Resource Reservation				Trading Resources Using Options			
		Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources		
			0.5	0.75	1		0.5	0.75	1		0.5	0.75	1
2000	1000	388	274	0	114	343	244	0	99	808	465	334	9
2000	2000	531	297	0	234	498	281	0	217	1312	584	694	34
2000	3000	680	318	0	362	655	309	0	346	1565	667	838	60
2000	4000	836	344	0	492	771	321	0	450	1949	713	1145	91
2000	5000	959	351	0	608	870	322	0	548	2246	761	1372	113
2000	6000	1093	364	0	729	971	325	0	646	2311	789	1397	125
2000	7000	1210	375	0	835	1074	331	0	743	2449	814	1483	152
2000	8000	1338	387	0	951	1138	333	0	805	2565	832	1574	159
2000	9000	1445	395	0	1050	1237	344	0	893	2665	857	1628	180
2000	10000	1535	396	0	1139	1300	347	0	953	2747	862	1695	190
2000	11000	1632	400	0	1232	1362	353	0	1009	2826	878	1748	200
2000	12000	1726	401	0	1325	1425	356	0	1069	2899	907	1782	210
2000	13000	1801	404	0	1397	1497	358	0	1139	2971	926	1825	220
2000	14000	1876	409	0	1467	1558	363	0	1195	3043	947	1868	228
2000	15000	1936	413	0	1523	1612	363	0	1249	3098	956	1906	236
2000	16000	1998	414	0	1584	1667	363	0	1304	3156	969	1942	245
2000	17000	2050	416	0	1634	1726	366	0	1360	3207	975	1976	256
2000	18000	2103	418	0	1685	1777	373	0	1404	3260	986	2012	262
2000	19000	2160	421	0	1739	1825	377	0	1448	3316	1004	2046	266
2000	20000	2212	424	0	1788	1876	379	0	1497	3369	1021	2074	274

Table 7.2: Price Paid by Resource Consumers during the three Scenarios with 2000 Resource Providers.

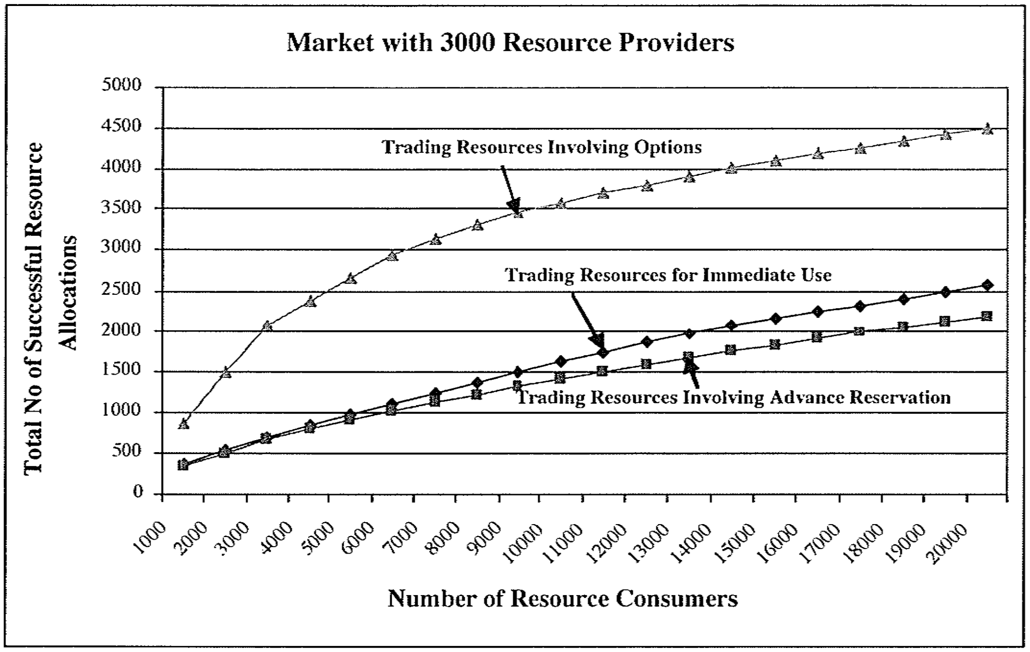

Figure 7.3: Resource Allocation Efficiency for the three Scenarios with 3000 Resource Providers.

Tables 7.1, 7.2 and 7.3 show the prices paid by resource consumers for the three given scenarios (considering 1000, 2000 and 3000 resource providers and the number of resource consumers ranging from 1000 to 20,000). Based on the data in these tables, it is evident that when using options, resource consumers are paying less for resources when compared to trading resources for immediate use or using resource reservation in advance.

Thus, when the demand for resources is high, resource consumers will normally opt for resource trading using options because the resource price will be less. Only

No of Resource Providers	No of Resource Consumers	Trading Resources for Immediate Use				Trading Resources with Resource Reservation				Trading Resources Using Options			
		Total Resource Trading	Prices Paid/Resources 0.5	0.75	1	Total Resource Trading	Prices Paid/Resources 0.5	0.75	1	Total Resource Trading	Prices Paid/Resources 0.5	0.75	1
3000	1000	363	261	0	102	346	244	0	102	861	548	310	3
3000	2000	533	298	0	235	508	280	0	228	1503	718	759	26
3000	3000	684	319	0	365	669	307	0	362	2088	834	1197	57
3000	4000	841	344	0	497	795	321	0	474	2375	888	1399	88
3000	5000	967	349	0	618	905	325	0	580	2660	942	1598	120
3000	6000	1104	360	0	744	1022	328	0	694	2933	996	1785	152
3000	7000	1229	370	0	859	1136	333	0	803	3138	1028	1929	181
3000	8000	1373	382	0	991	1216	339	0	877	3317	1064	2061	192
3000	9000	1503	390	0	1113	1330	345	0	985	3472	1088	2161	223
3000	10000	1616	392	0	1224	1417	348	0	1069	3603	1119	2243	241
3000	11000	1740	395	0	1345	1494	355	0	1139	3720	1146	2318	256
3000	12000	1860	396	0	1464	1578	363	0	1215	3816	1174	2375	267
3000	13000	1967	398	0	1569	1671	364	0	1307	3916	1200	2436	280
3000	14000	2071	404	0	1667	1750	367	0	1383	4026	1226	2510	290
3000	15000	2155	408	0	1747	1826	368	0	1458	4102	1241	2561	300
3000	16000	2241	409	0	1832	1901	372	0	1529	4191	1258	2621	312
3000	17000	2322	411	0	1911	1981	376	0	1605	4269	1270	2673	326
3000	18000	2399	413	0	1986	2053	378	0	1675	4355	1292	2728	335
3000	19000	2486	416	0	2070	2120	385	0	1735	4433	1312	2780	341
3000	20000	2571	419	0	2152	2193	388	0	1805	4511	1338	2816	357

Table 7.3: Price Paid by Resource Consumers during three Scenarios with 3000 Resource Providers.

resource consumers who need resources for immediate use will choose to buy resources for the higher price. Also, few resource consumers are likely to be interested in reserving resources for the current high price. Instead, they will get more benefit if they choose the option scheme for resource trading because they are not obligated to use resources if their work is not ready. Also, they can sell their option (within the anticipated resource usage time) to other users to make profit at the future open market value.

This shows the benefits of using options to resource consumers as they provide a way for consumers to hedge against high resource prices when there is a limited supply of resources. Thus, they can avoid buying resources at greater cost when there is limited supply.

7.1.1.2 Resource Allocation Efficiency when there is High Supply but Limited Demand for Resources

Figures 7.4, 7.5 and 7.6 illustrate the behavior of the market with high supply but limited demand for resources (i.e. with less resource consumers and more resource providers) for the three scenarios and 1000, 2000 and 3000 resource consumers, respectively.

Figure 7.4 shows the results for number of successful resource allocation for the three different scenarios considering 1000 resource consumers and with the number of resource providers ranging from 1,000 to 10,000. As can be observed in the figure, with less resource consumers and more resource providers, the number of successful resource allocations is relatively high. With an increase in the number of resource providers (see Figures 7.5 and 7.6), almost all resource consumers are successfully

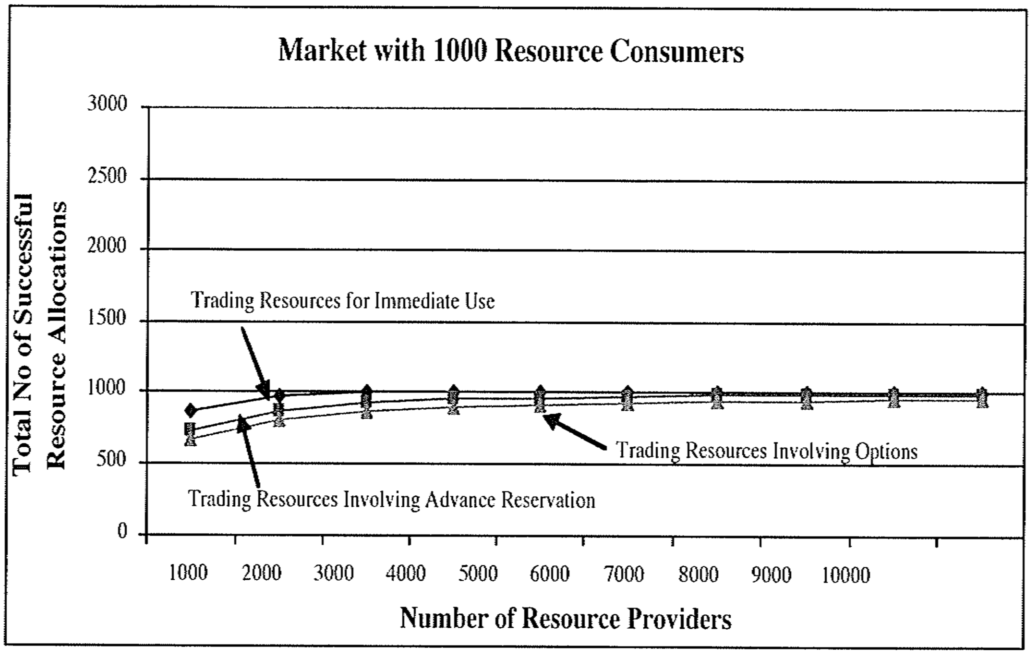

Figure 7.4: Resource Allocation Efficiency for the three Scenarios with 1,000 Resource Consumers.

achieving their resource allocation requests.

Tables 7.4, 7.5 and 7.6 shows the earnings made by resource providers for the three given scenarios (considering 1,000, 2,000 and 3,000 resource consumers and with the number of resource providers ranging from 1,000 to 10,000). From the tables it can be seen that, when trading resource involving options and advance resource reservation, resource consumers are paying less for resources than when trading resources for immediate use. This is because they can lock in lower prices if they know they will need resources in the future.

No of Resource Consumers	No of Resource Providers	Trading Resources for Immediate Use				Trading Resources for Immediate Use				Trading Resources for Immediate Use			
		Total Resource Trading	Earnings made on Resources			Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources		
			0.5	0.75	1		0.5	0.75	1		0.5	0.75	1
1000	1000	858	858	0	0	733	733	0	0	667	371	283	13
1000	2000	970	970	0	0	866	866	0	0	808	460	340	8
1000	3000	997	997	0	0	921	921	0	0	861	539	320	2
1000	4000	1000	1000	0	0	950	951	0	0	898	591	305	2
1000	5000	1000	1000	0	0	961	961	0	0	908	628	280	0
1000	6000	1000	1000	0	0	975	975	0	0	922	662	258	0
1000	7000	1000	1000	0	0	983	983	0	0	938	698	240	0
1000	8000	1000	1000	0	0	985	985	0	0	946	730	216	0
1000	9000	1000	1000	0	0	987	987	0	0	951	750	201	0
1000	10000	1000	1000	0	0	989	989	0	0	955	780	175	0

Table 7.4: Earnings made by Resource Providers during three Scenarios with 1,000 Resource Consumers.

No of Resource Consumers	No of Resource Providers	Trading Resources for Immediate Use				Trading Resources for Immediate Use				Trading Resources for Immediate Use			
		Total Resource Trading	Earnings made on Resources			Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources		
			0.5	0.75	1		0.5	0.75	1		0.5	0.75	1
2000	1000	1243	1243	0	0	1076	1076	0	0	998	461	497	40
2000	2000	1672	1672	0	0	1419	1419	0	0	1322	594	694	34
2000	3000	1863	1863	0	0	1607	1607	0	0	1490	717	746	27
2000	4000	1946	1946	0	0	1725	1725	0	0	1612	800	800	12
2000	5000	1969	1969	0	0	1798	1798	0	0	1680	881	790	9
2000	6000	1988	1988	0	0	1844	1844	0	0	1729	950	773	6
2000	7000	1996	1966	0	0	1878	1878	0	0	1773	1019	750	4
2000	8000	2000	2000	0	0	1897	1897	0	0	1799	1090	706	3
2000	9000	2000	2000	0	0	1917	1917	0	0	1829	1148	679	2
2000	10000	2000	2000	0	0	1932	1932	0	0	1840	1206	633	1

Table 7.5: Earnings made by Resource Providers during three Scenarios with 2,000 Resource Consumers.

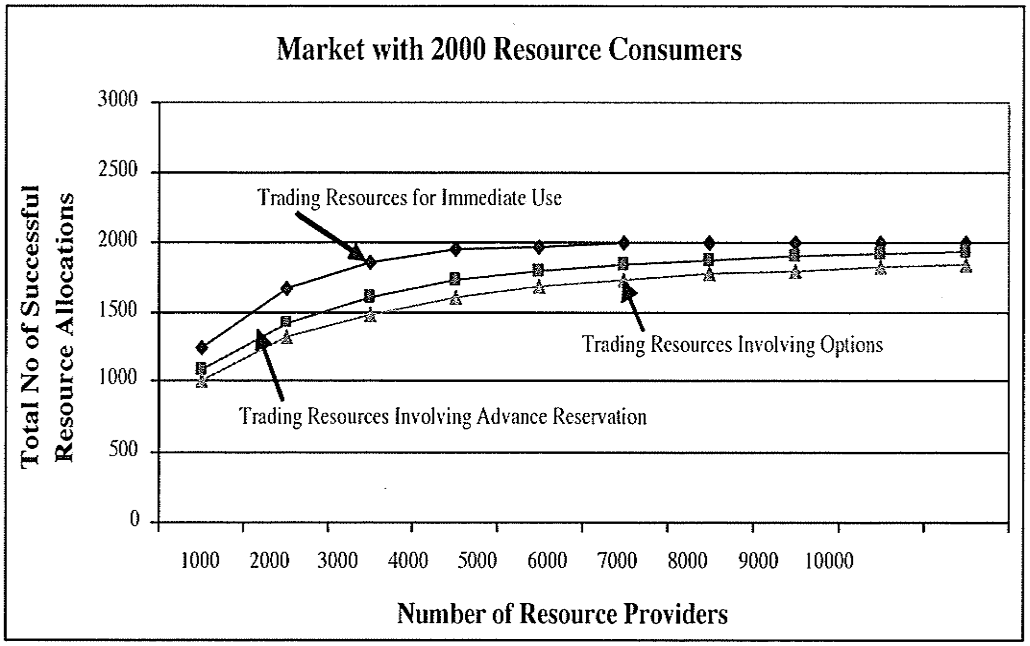

Figure 7.5: Resource Allocation Efficiency for the three Scenarios with 2,000 Resource Consumers.

When there is low demand for and high supply of resources, given the three different choices, many resource consumers will choose to buy resources for immediate use (if their work is ready) as they are finding resource for low prices (from Tables 6.4, 6.5 and 6.6). Also, some resource consumers who anticipate need of resources at a future time will choose to reserve resources at the current market price instead of buying resources using an options scheme (since consumers will getting the price benefit without paying the option premium). Collectively, this makes the option scheme less attractive unless there is a potential for further price decreases.

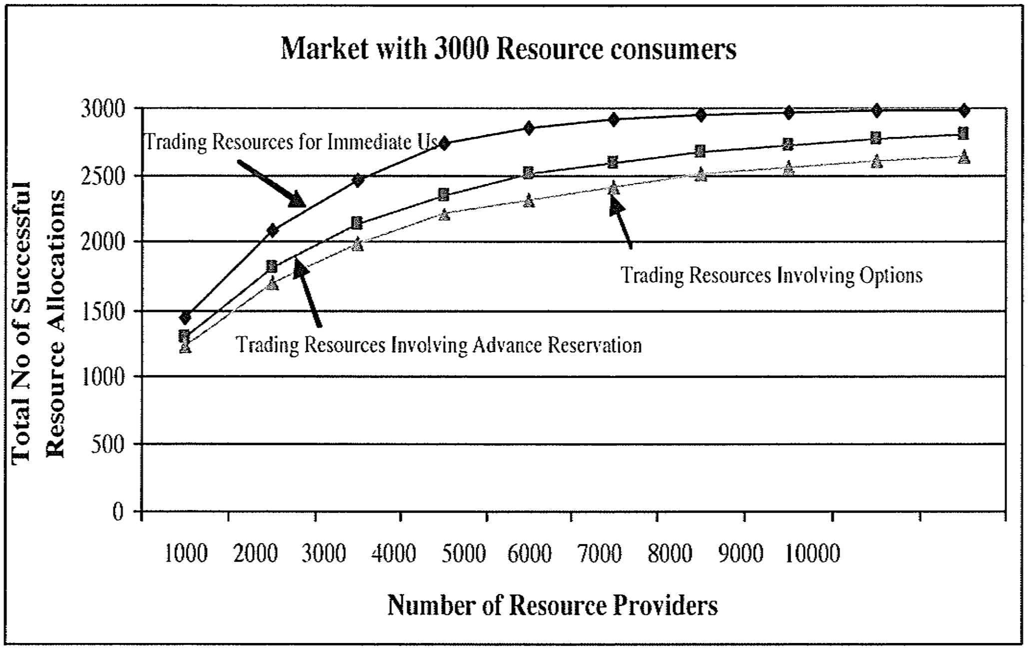

Figure 7.6: Resource Allocation Efficiency for the three Scenarios with 3,000 Resource Consumers.

Some resource consumers may choose to buy an option to use resources since they gain the right but not obligation to use the resources. With an options scheme, resource consumers can reserve (but not use) resources at less cost than purchasing their use directly with the reservation in advance scheme. This is beneficial if their work is not ready when the resources are to be used. Also, they can sell their option (within the anticipated resource usage time) to other users to make a profit at the future open market value.

This shows the benefit that options provide to resource providers by allowing them to hedge against low resource prices when there is an excess supply of resources.

No of Resource Consumers	No of Resource Providers	Trading Resources for Immediate Use				Trading Resources for Immediate Use				Trading Resources for Immediate Use			
		Total Resource Trading	Earnings made on Resources			Total Resource Trading	Prices Paid/Resources			Total Resource Trading	Prices Paid/Resources		
			0.5	0.75	1		0.5	0.75	1		0.5	0.75	1
3000	1000	1442	1442	0	0	1289	1289	0	0	1214	510	645	59
3000	2000	2081	2081	0	0	1811	1811	0	0	1704	676	962	66
3000	3000	2475	2475	0	0	2139	2139	0	0	1995	824	1114	57
3000	4000	2738	2738	0	0	2364	2364	0	0	2214	931	1240	43
3000	5000	2860	2860	0	0	2514	2514	0	0	2334	1034	1265	35
3000	6000	2919	2919	0	0	2605	2605	0	0	2429	1140	1264	25
3000	7000	2945	2945	0	0	2685	2685	0	0	2520	1250	1251	19
3000	8000	2970	2970	0	0	2721	2721	0	0	2572	1327	1228	17
3000	9000	2980	2980	0	0	2774	2774	0	0	2621	1415	1193	13
3000	10000	2986	2986	0	0	2804	2804	0	0	2644	1502	1135	7

Table 7.6: Earnings made by Resource Providers during three Scenarios with 3,000 Resource Consumers.

7.1.2 Resource Allocation Efficiency with Some Consumers Acting as Providers

With an options scheme, resource consumers are not obligated to use resources. This may occur if, for example, the consumer's work is not ready. Also, from Sections 7.1.1.1 and 7.1.1.2, we know that resource consumers are more likely to choose the option scheme for resource trading since they can sell the options to other users to make profit at the future open market value (or simply to avoid loss of premium value if they cannot use the resources themselves). I also conducted experiments where some consumers chose to act as providers by selling their options. This was done to show the additional benefit of options when resource consumers act as a resource providers. Figures 7.7 and 7.8 show the number of successful resource allocation requests when some of the customers holding option contracts on resources act as resource providers to re-sell their options.

7.1.2.1 High Demand but Less Supply of Resources

In Figure 7.7, we can observe that there is an increase in the number of successful resource allocations when some customers are acting as providers. This shows that the reselling of options is helpful in grid environments when resource consumers choose to re-sell the options when there is limited supply of resources from resource suppliers. Not only are more resource requests satisfied in such a "tight market" scenario, but the resource consumers that re-sell also profit by offering unneeded resource access at a (typically) higher open market value.

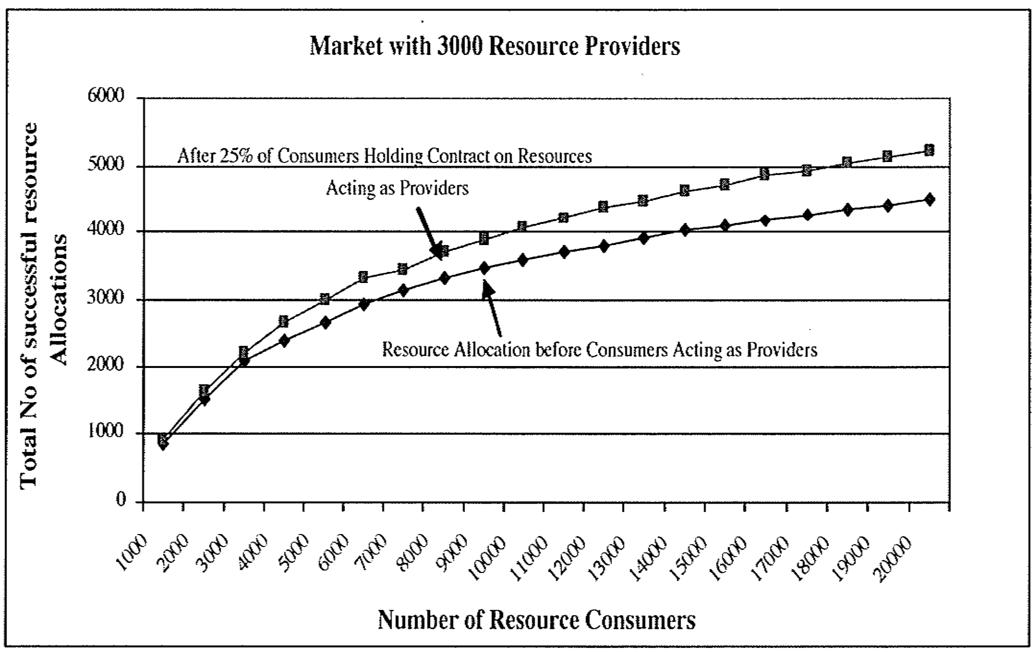

Figure 7.7: Resource Allocation Efficiency with Some Consumers Acting as Providers when there is a Limited Supply of Resources.

7.1.2.2 High Supply but Less Demand for Resources

In Figure 7.8, we can see that there is also a slight increase in successful resource allocation when some customers are acting as providers even when there are many resources available. Because there is less demand for resources, the price of resources might be less then what they brought in the recent past. Thus, fewer resource consumers will be willing to re-sell resources since they will not recoup their original costs. Instead consumers will be motivated to use the resources themselves. However, the resource consumers who no longer need resources or whose work is not ready will still be able to re-sell their option contracts to minimize their potential losses.

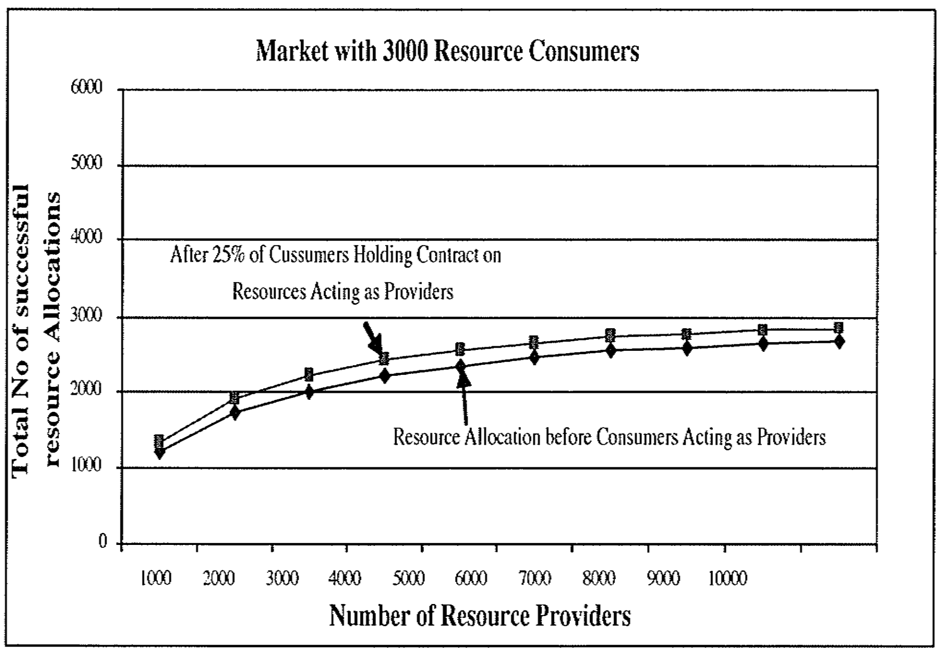

Figure 7.8: Resource Allocation Efficiency with Some Consumers Acting as Providers when there is a Generous Supply of Resources.

7.1.3 Monitoring Resource Allocation Efficiency

Resource providers might be interested to know the grid market trends to assist them in making resource pricing decisions. In essence, resource providers need to monitor resource consumer behaviors to help them decide whether the current scenario has, for example, low or high resource availability. In my experiments I also investigated long term monitoring to detect trends in resource consumer behavior and factored this information into resource provider behaviour. A resource provider can react in one of two ways to the discovery that long term demand is growing: increase resource prices or add more resources. Both of these result in greater profits. Standard market behaviour is to increase prices but this only works until resource saturation occurs (i.e. all resources are fully used). Accordingly, I chose to simulate the effects of adding new resources. Again, I present results for the two extreme cases of high demand but low supply and high supply but low demand.

7.1.3.1 High Demand but Low Supply of Resources

During high resource demand situations, the price for resources is already high according to the market standards and thus resource providers will dedicate more resources to acquire greater market share and increase their profits. Also, from Section 7.1.2.1, we know that resource consumers acting as resource providers will be interested in re-selling their unneeded resources to make a profit on resources which they bought for lesser price.

Figure 7.9 illustrates the behavior of the market with high demand and low supply of resources (i.e. with less resource providers and more resource consumers).

We can observe that there is, again, a small increase in the total number of successful resource allocations after monitoring of consumer trends by resource providers causes them to increase the number of resources available to attract customers.

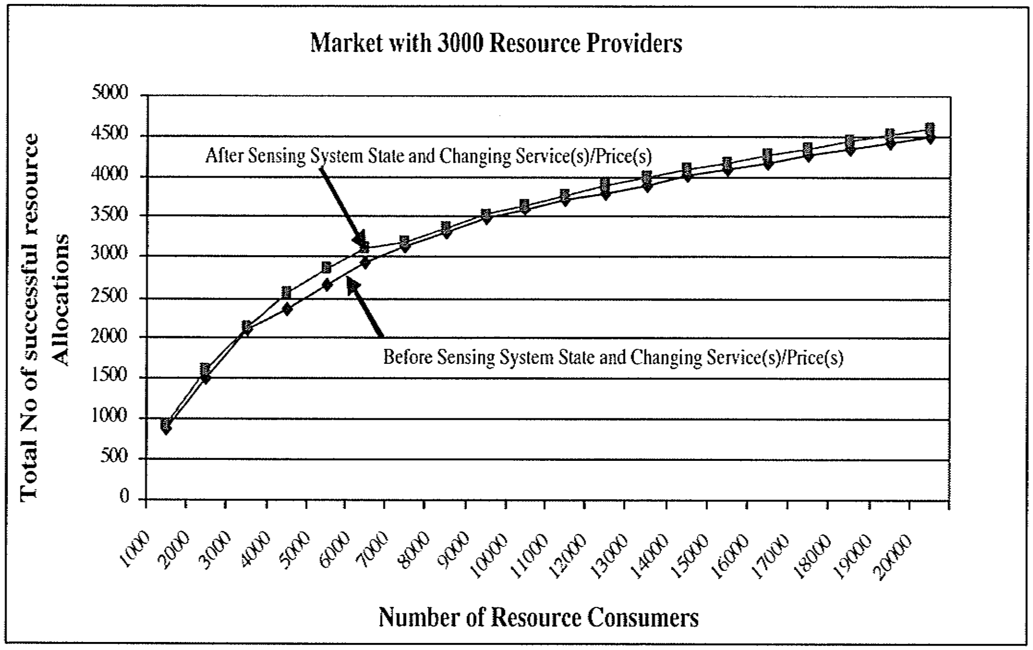

Figure 7.9: Increasing Resources with 3000 Resource Providers.

7.1.3.2 High Supply but Low Demand for Resources

During situations where there is a greater supply of resources, resource providers may want to decrease their resource price to attract more customers. Further, from Section 7.1.2.2, we know that some resource consumers who no longer need resources or whose work is not ready may still be interested in re-selling their resources even though the price of resources is less then what they paid for them.

Figure 7.10 illustrates the behavior of the market with high supply and low demand for resources (i.e. with less resource consumers and more resource providers). We can observe that there is again a small increase in the total number of successful resource allocations when monitored trends are considered by resource providers causing them to lower their prices to attract customers.

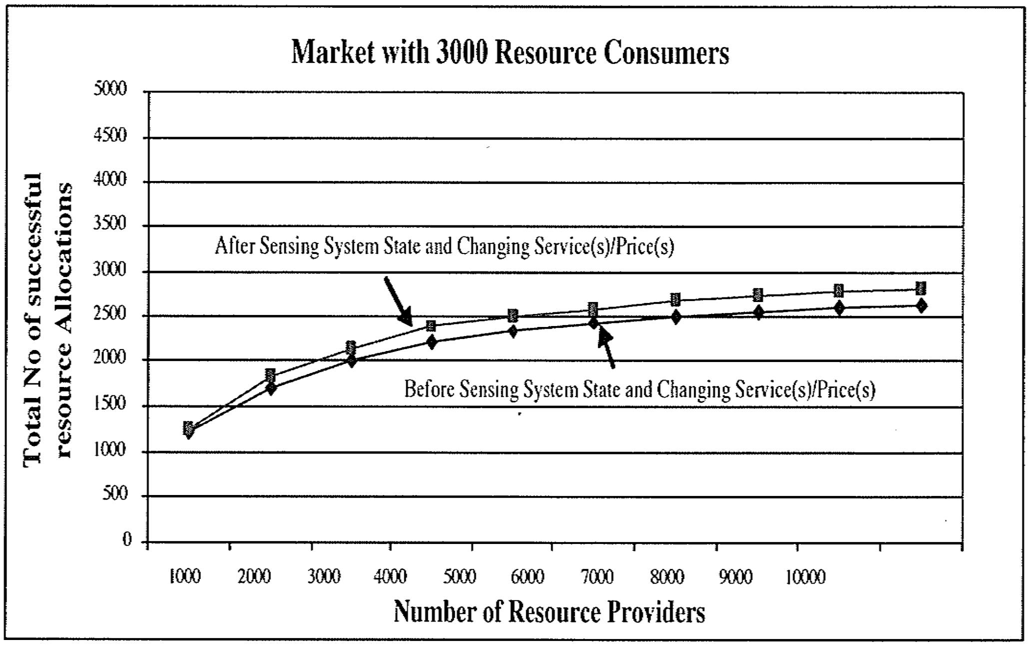

Figure 7.10: Increasing Resources with 3000 Resource Consumers.

7.2 Statistical Significance

All the data from the experimental results were analyzed and validated using Analysis of Variance (ANOVA) tests with multiple comparisons using the least significant difference method.

7.2.1 Resource Allocation Efficiency with High Demand but Low Supply of Resources

Figure 7.11 show the ANOVA test results considering 1000 resource providers and with the number of resource consumers ranging from 1,000 to 20,000. Figure 7.12 shows the ANOVA results considering 2000 resource providers and with the number of resource consumers ranging from 1,000 to 20,000. Figure 7.13 shows the ANOVA results considering 3000 resource providers and with the number of resource consumers ranging from 1,000 to 20,000.

To find the significance (p value) of each experiment, average values of successful resource allocation are considered.

Dependent Variable		p value	Significant
1	2	P < 0.166	NO
	3	P < 0.000	YES
2	1	P < 0.166	NO
	3	P < 0.000	YES
3	1	P < 0.000	YES
	2	P < 0.000	YES

Note:
The mean difference is significant at the .05 level
Experiment 1: Resource Trading for Immediate use
Experiment 2: Resource Trading Involving Advance Resource Reservation
Experiment 3: Resource Trading Involving Options

Figure 7.11: ANOVA Test Results for Resource Allocation Efficiency with 1000 Resource Providers

Dependent Variable		p value	Significant
1	2	P < 0.250	NO
	3	P < 0.000	YES
2	1	P < 0.250	NO
	3	P < 0.000	YES
3	1	P < 0.000	YES
	2	P < 0.000	YES

Note:
The mean difference is significant at the .05 level
Experiment 1: Resource Trading for Immediate use
Experiment 2: Resource Trading Involving Advance Resource Reservation
Experiment 3: Resource Trading Involving Options

Figure 7.12: ANOVA Test Results for Resource Allocation Efficiency with 2000 Resource Providers

The ANOVA test results from the Figures show that there was a statistically significant difference between experiment 1 (trading for immediate use) and experiment 3 (trading with options) in terms of resource allocation efficiency. Experiment 3 and experiment 2 (trading with advance resource reservation) were also found to be significantly different in terms of resource allocation efficiency. No significant difference was detected between experiment 1 and experiment 2. This result suggests that resource allocation efficiency is very similar in experiment 1 and experiment 2.

Dependent Variable		p value	Significant
1	2	P < 0.407	NO
	3	P < 0.000	YES
2	1	P < 0.407	NO
	3	P < 0.000	YES
3	1	P < 0.000	YES
	2	P < 0.000	YES
Note: The mean difference is significant at the .05 level Experiment 1: Resource Trading for Immediate use Experiment 2: Resource Trading Involving Advance Resource Reservation Experiment 3: Resource Trading Involving Options			

Figure 7.13: ANOVA Test Results for Resource Allocation Efficiency with 3000 Resource Providers

7.2.2 Resource Allocation Efficiency with High Supply but Low Demand for Resources

Figure 7.14 shows the ANOVA test results considering 1000 resource consumers and with the number of resource providers ranging from 1,000 to 20,000. Figure 7.15 shows the ANOVA test results considering 2000 resource consumers and with the number of resource providers ranging from 1,000 to 20,000. Figure 7.16 shows the ANOVA test results considering 3000 resource consumers and with the number of resource providers ranging from 1,000 to 20,000.

To find the significance (p value) of each experiment, average values of successful resource allocation are considered.

Dependent Variable		p value	Significant
1	2	P < 0.156	NO
	3	P < 0.007	YES
2	1	P < 0.156	NO
	3	P < 0.152	NO
3	1	P < 0.007	YES
	2	P < 0.152	NO

Note:
The mean difference is significant at the .05 level
Experiment 1: Resource Trading for Immediate use
Experiment 2: Resource Trading Involving Advance Resource Reservation
Experiment 3: Resource Trading Involving Options

Figure 7.14: ANOVA Test Results for Resource Allocation Efficiency with 1000 Resource Consumers

The ANOVA test results from the figures show that there was a statistically significant difference between experiment 1 and experiment 3 in terms of resource allocation efficiency. However, Figure 7.16 shows that there was no significant difference between the three experiments when the number of consumers was increased to 3000.

Dependent Variable		p value	Significant
1	2	P < 0.188	NO
	3	P < 0.035	YES
2	1	P < 0.188	NO
	3	P < 0.392	NO
3	1	P < 0.035	YES
	2	P < 0.392	NO

Note:
The mean difference is significant at the .05 level
Experiment 1: Resource Trading for Immediate use
Experiment 2: Resource Trading Involving Advance Resource Reservation
Experiment 3: Resource Trading Involving Options

Figure 7.15: ANOVA Test Results for Resource Allocation Efficiency with 2000 Resource Consumers

Dependent Variable		p value	Significant
1	2	P < 0.230	NO
	3	P < 0.069	NO
2	1	P < 0.230	NO
	3	P < 0.511	NO
3	1	P < 0.069	NO
	2	P < 0.511	NO
Note: Experiment 1: Resource Trading for Immediate use Experiment 2: Resource Trading Involving Advance Resource Reservation Experiment 3: Resource Trading Involving Options			

Figure 7.16: ANOVA Test Results for Resource Allocation Efficiency with 3000 Resource Consumers

Chapter 8

Conclusions and Future Work

As grids are becoming more mature and their use is increasing, it is an appropriate time to explore the use of pay-for-service. Once users are charged for service, however, it becomes important to ensure price certainty or adoption will be unlikely. Supporting an option-like mechanism for grid resource management is one way to do this.

In this thesis I have explored the potential benefits of option contracts for grid resource management systems. The use of such contracts aids both users to hedge against price increases for resources and service providers to hedge against price decreases for resources. In addition, the option contract approach provides an effective solution for dealing with variation in supply-and-demand for resources over time which is very important in commercializing grid computing. Moreover, an option contract model helps to provide flexible resource pricing and assure resource availability within user constraints.

Specifically, I have evaluated the efficiency and practicality of a simple grid resource management system based on an implemented "Grid Market Directory (GMD)". Resource allocation efficiency using the GMD was assessed for three different scenarios (resource trading for immediate use by consumers, resource trading involving advance reservation and resource trading with options). This was done both when there was high demand but low supply of resources and when there was high supply but low demand for resources. (i.e. the extremal situations). Simulation based results show that the use of options in grid environments will provide benefits to resource providers and resource consumers allowing them to hedge against price increases/decreases for resources. The experimental results also show that the use of options is helpful in grid environments for dealing with variation in supply-and-demand for resources.

8.1 Future Work

There are several specific opportunities to extend and, in some cases, improve on the work presented in this thesis.

Shorter term future work might include further assessment of the effects on resource allocation efficiency of using options. For example, the impact on handling large tasks in a resource constrained environment would be interesting to explore. Considering an extension to my framework to support multiple markets would also be useful. In particular, evaluating the performance of multiple grid markets each having constrained numbers of providers to determine if sharing between markets is necessary and could be effectively implemented would be a logical extension to my work.

In the longer term, incorporating my option contract model into an existing framework (such as GRACE or CPM) would be very desirable. Incorporating options into a large scale test bed would help to more clearly validate the benefits of options and would also expose any issues related to charging for use. Finally, assessing variations on the option model itself might lead to improvements. For example, my framework could be extended to support payment of additional premiums as the time of use gets nearer. Such premiums would, of course, be credited against charges for actual use. This might motivate more resource providers to participate in market based grid computing and would also motivate users to re-sell resources which they no longer needed which would help to ensure better, overall resource utilization.

Bibliography

[1] David Abramson, Rajkumar Buyya, and Jonathan Giddy. A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker. *Future Generation Computer Systems*, 18(8):1061–1074, October 2002.

[2] David Abramson, Jonathan Giddy, and Lew Kotler. High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid? In *Proceedings of the 14th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 520–528, Cancun, Mexico, May 2000.

[3] David Abramson, Rok Sosic, Jonathan Giddy, and B. Hall. Nimrod: A Tool for Performing Parametrised Simulations using Distributed Workstations. In *Proceedings of the Fourth IEEE International Symposium on High Performance Distributed Computing*, pages 112–121, Virginia, U.S.A, August 1995.

[4] Yair Amir, Baruch Awerbuch, and R. Sean Borgstrom. A Cost-Benefit Framework for Online Management of a Metacomputing System. In *Proceedings of the First ACM International Conference on Information and Computation*

Economies, pages 140–147, Charleston, South Carolina, United States, October 1998.

[5] Yair Amir, Baruch Awerbuch, and Ryan S. Borgstrom. The Java Market: Transforming the Internet into a Metacomputer. Technical Report CNDS-98-1, Center for Networking and Distributed Systems, Department of Computer Science, Johns Hopkins University, 1998.

[6] Baruch Awerbuch, Yossi Azar, Amos Fiat, and Tom Leighton. Making Commitments in the Face of Uncertainty: How to Pick a Winner Almost Every Time. In *Proceedings of the ACM Symposium on Theory of Computing*, 1996.

[7] Alexander Barmouta and Rajkumar Buyya. GridBank: A grid accounting services architecture (GASA) for distributed systems sharing and integration. In *Proceedings of the 17th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, page 245.a, Nice, France, April 2003.

[8] Rajkumar Buyya. *Economic-Based Distributed Resource Management and Scheduling for Grid Computing.* Ph.d thesis, Monash University, Melbourne, Australia, April 2002.

[9] Rajkumar Buyya. Tutorial/Workshop Notes on the Gridbus Toolkit for Service-Oriented Grid and Utility Computing. Tutorial, Tutorial notes from the 18th International Symposium on High Performance Computing Systems and Applications (HPCS), May 2004.

[10] Rajkumar Buyya, David Abramson, and Jonathan Giddy. Economy Driven Resource Management Architecture for Computational Power Grids. In *In-*

ternational Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), pages 510–518, Las Vegas, USA, 2000.

[11] Rajkumar Buyya, David Abramson, and Jonathan Giddy. Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. In *Proceedings of the Fourth IEEE International Conference on High-Performance Computing in the Asia-Pacific Region*, volume 1, pages 283–289, Beijing, China, May 2000.

[12] Rajkumar Buyya, David Abramson, and Jonathan Giddy. A Case for Economy Grid Architecture for Service Oriented Grid Computing. In *Proceedings of the 10th IEEE International Heterogeneous Computing Workshop (HCW)*, page 20083.1, San Francisco, California, USA, April 2001.

[13] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger. Economic Models for Resource Management and Scheduling in Grid Computing. *Concurrency and Computation: Practice and Experience*, 14(13–15):1507–1542, November/December 2002.

[14] Rajkumar Buyya and Sudharshan Vazhkudai. Compute Power Market: Towards a Market-Oriented Grid. In *Proceedings of the First IEEE International Symposium on Cluster Computing and the Grid*, page 574, Brisbane, Australia, May 2001.

[15] Henri Casanova and Jack Dongarra. NetSolve: A Network Server for Solving Computational Science Problems. Technical Report CS-96-328, University of Tennessee, Knoxville, Knoxville, TN 37996, USA, 1996.

[16] Karl Czajkowski, Ian Foster, Nick Karonis, Carl Kesselman, Stuart Martin, Warren Smith, and Steven Tuecke. A Resource Management Architecture for Metacomputing Systems. In *Proceedings of the Fourth Workshop on Job Scheduling Strategies for Parallel Processing*, pages 62–82. Springer-Verlag, 1998.

[17] Ian Foster and Carl Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, 1997.

[18] Ian Foster and Carl Kesselman. *The Grid : Blueprint for a New Computing Infrastructure.* Morgan Kaufmann Publishers, San Francisco, CA, U.S.A, second edition, 2004.

[19] James Frey, Todd Tannenbaum, Miron Livny, Ian Foster, and Steven Tuecke. Condor-G: A Computation Management Agent for Multi-Institutional Grids. In *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing*, pages 55–63, San Francisco, CA, USA, August 2001.

[20] John C. Hull. *Options, Futures and Other Derivatives.* Prentice Hall Publishers, Upper Saddle River, NJ, USA, fifth edition, 2003.

[21] Junseok Hwang, Choong Hee Lee, Jinwoo Song, and Kyoungmin Pyo. Grid and P2P Economics and Market Models. In *Proceedings of the First IEEE International Workshop on Grid Economics and Business Models*, pages 3–18, Seoul, Korea, April 2004.

[22] Chris Kenyon and Giorgos Cheliotis. Architecture Requirements for Commercializing Grid Resources. In *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, pages 215–224, Edinburgh, Scotland, U.K, July 2002.

[23] Chris Kenyon and Giorgos Cheliotis. Grid resource commercialization: Economic engineering and delivery scenarios. In *Grid Resource Management: State of the Art and Future Trends*, pages 465–478. Kluwer Academic Publishers, 2004.

[24] Kotz, Samuel, Kozubowski, Tomasz, Podgorski, and Krzystof. *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance.* Birkhauser Publishers, 2001.

[25] Spyros Lalis and Alexandros Karipidis. JaWS: An Open Market-Based Framework for Distributed Computing over the Internet. In *Proceedings of the First IEEE/ACM International Workshop on Grid Computing*, pages 36–46, Bangalore, India, December 2000. Springer-Verlag.

[26] Pierre L'Ecuyer, Lakhdar Meliani, and Jean Vaucher. SSJ: A Framework for Stochastic Simulation in Java. In *Proceedings of the 34th Conference on Winter Simulation*, pages 234–242, 2002.

[27] Michael J. Litzkow, Miron Livny, and Matt W. Mutka. Condor-A Hunter of Idle Workstations. In *Proceedings of the Eighth IEEE International Conference on Distributed Computing Systems*, pages 104–111, San Jose, CA, June 1988.

[28] Matt W. Mutka and Miron Livny. Scheduling Remote Processing Capacity in a Workstation-Processor Bank Network. In *Proceedings of the Seventh International Conference on Distributed Computing Systems*, pages 2–9, Berlin, Germany, September 1987.

[29] Steven Newhouse and John Darlington. Computational Communities: A Marketplace for Federated Resources. In *Proceedings of the Ninth ACM International Conference on High-Performance Computing and Networking*, pages 667–674, Europe, June 2001.

[30] Noam Nisan, Shmulik London, Ori Regev, and Noam Camiel. Globally Distributed Computation over the Internet-The POPCORN Project. In *Proceedings of the the 18th IEEE International Conference on Distributed Computing Systems*, pages 592–601, Amsterdam, Netherlands, May 1998.

[31] Rajesh Raman, Miron Livny, and Marvin Solomon. Matchmaking: Distributed Resource Management for High Throughput Computing. In *Proceedings of the The Seventh IEEE International Symposium on High Performance Distributed Computing*, pages 140–146, July 1998.

[32] Ori Regev and Noam Nisan. The POPCORN Market-An Online Market for Computational Resources. In *Proceedings of the First ACM International Conference on Information and Computation Economies*, pages 148–157, Charleston, South Carolina, United States, October 1998.

[33] Tuomas Sandholm. Distributed Rational Decision Making. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. MIT Press, 1999.

[34] Keith Seymour, Hidemoto Nakada, Satoshi Matsuoka, Jack Dongarra, Craig Lee, and Henri Casanova. Overview of GridRPC: A Remote Procedure Call API for Grid Computing. In *Proceedings of the Third International Workshop on Grid Computing*, pages 274–278, Baltimore, MD, USA, November 2002.

[35] Keith Seymour, Asim YarKhan, Sudesh Agrawal, , and Jack Dongarra. *NetSolve: Grid Enabling Scientific Computing Environments.* Elsevier, 2005.

[36] Ruppa Thulasiram, May–August 2004. Advanced Topics in Computer Science (Computational Finance class notes), University of Manitoba.

[37] Paul Wilmott, Jeff Dewynne, and Sam Howison. *Option Pricing: Mathematical Models and Computation.* Oxford Financial Press, 1998.

[38] Huaxia Xia, Holly Dail, Henri Casanova, and Andrew A. Chien. The MicroGrid: Using Online Simulation to Predict Application Performance in Diverse Grid Network Environments. In *Proceedings of the Second IEEE International Workshop on Challenges of Large Applications in Distributed Environments*, pages 52–61, June 2004.

[39] Asim YarKhan, Keith Seymour, Kiran Sagi, Zhiao Shi, , and Jack Dongarra. Recent Developments in GridSolve. *International Journal of High Performance Computing Applications*, 20(1):131–141, 2006.

[40] Jia Yu, Srikumar Venugopal, and Rajkumar Buyya. A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services. Technical Report GRIDS-TR-2003-0, Grid Computing and Distributed Systems (GRIDS) Laboratory, The University of Melbourne, Australia, January 2003.

[41] United Devices. http://www.ud.com/home.htm.