CHAPTER 4

ECONOMIC PRICING BASED JOB SCHEDULING ALGORITHM

4.1 INTRODUCTION

The resource consumer needs a model to specify their resource requirements and to choose their preferences among different available resources. A broker has to be opted to support resource discovery and strategies for the application of scheduling the distributed resources dynamically at runtime depending on their availability, capability, and cost along with user-defined requirements as defined by Kyle Chard et al (2008).

The producers (job executers) and consumers (job owners) have different goals, objectives, strategies, and supply-and-demand patterns. More importantly both resources and end-users are geographically distributed over different time zones. In managing such complex environments, traditional approaches to resource management, that attempt to optimize system-wide measure of performance, cannot be successfully employed. The traditional approaches use centralized policies that need complete state information and a common fabric management policy, or a decentralized consensus based policy. Due to the complexity in constructing a grid environment, it is impossible to define an acceptable system-wide performance matrix and common fabric management policy.

With the proposed algorithm, it is proposed and explored the use of an incentive framework, called Economic Pricing Based Job Scheduling Algorithm (EP), for managing producer and consumer problems in the grid scheduling environments. The incentive approach provides a fair basis in successfully managing the decentralization and heterogeneity existing in the grid environment. The competitive economic models provide algorithms/policies and tools for resource sharing or allocation in grid systems. These models can be based on exchanging resources or prices. In the Demand Pricing Model, all participants need to own resources and trade resources depending on the price of demanding resource. In the Best Bid Pricing Model, the resources have a price, based on the negotiation mechanism.

They aim at enhancing the system throughput, utilization, and complete execution at the earliest possible time rather than improving the utility of application processing. They do not take resource access cost (price) into consideration, which means that the value of processing applications at any time is treated to be the same.

The end user does not want to pay the highest price but wants to negotiate a particular price based on the demand, value, priority, and available budget. In an economic approach, the scheduling decisions are made dynamically at runtime and they are driven and directed by the end-users requirements whereas a conventional cost model often deals with software and hardware costs for running applications, the economic model primarily charges the end user for services that they consume based on the value they derive from it. Pricing based on the demand of users and the supply of resources is the main driver in the competitive, economic market model. Therefore, in the grid environment, a user is in competition with other users and a resource owner with other resource owners.

The main contribution of this chapter is to provide a decentralized computational economy for grids along with system architecture and policies for resource management for different incentive models. Currently, the user community and the technology are still rather new and not well accepted and established in commercial settings. However, it is believed that the grid can become established in such settings by providing incentive to both consumers and resource owners for being part of the grid.

Since the grid uses the Internet as a carrier for providing remote services, it is well positioned to create a cooperative problem solving environment and means for sharing computational and data resources in a seamless manner. An incentive approach to the grid computing introduces a number of new issues like resource trading and QoS based scheduling in addition to those such as Site Autonomy, Heterogeneous Substrate, Policy Extensibility, Online Control which are already addressed by existing grid systems. To address these new issues, the economy based Grid systems need to support the following aspects.

- An Information and Market directory for publicizing grid entities
- Models for establishing the value of resources
- Resource pricing schemes and publishing mechanisms
- Economic models and negotiation protocols
- Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling.
- Accounting, Billing, and Payment Mechanisms

In this chapter the requirements of users (resource providers and consumers) are identified in the grid economy and various resource management issues that need to be addressed in realizing such a grid system. Brief discussion of the popular economic models for resource trading and the present related work that employs pricing economy in resource management is made Amir Danak et al (2011). It is proposed that scalable architecture and new services for the grid that provide mechanisms for addressing user requirements.

The resource providers need:

- Tools and mechanisms that support price specification and generation schemes to increase system utilization.
- Protocols that support service publication, trading, and accounting.

For the market to be wealthy, coordination mechanisms are required to help to reach a reasonable market price at which the supply of the service equals the quantity demanded.

The EP (Producer and Consumer) based scheduling mechanism is achieved by two models namely.

- Demand Pricing Model
- Best Bid Pricing Model

4.2 COMPUTATIONAL ECONOMY AND ITS BENEFITS

The current research and investment into computational grids is motivated by an assumption that coordinated access to diverse and geographically distributed resources is valuable. In this paradigm, mechanisms are needed to allow such coordinated access, but also sustainable, scalable models and policies that promote precious grid resource sharing. Based on the success of economic institutions in the real world as a sustainable model for exchanging and regulating resources, goods and services, a computational economy framework is needed. Among other things, this framework provides a mechanism to indicate which users should receive priority.

Like all systems involving goals, resources and actions, computations can be viewed in economic terms. With the proliferation of networks, high-end computing systems architecture has moved from the centralized toward the decentralized models of control and action; the use of economic driven market mechanisms would be a natural extension of this development. The ability of trade and price mechanisms to combine local decisions by diverse entities into globally effective characteristics, imply their value for organizing computations in large systems such as Internet-scale computational grids.

The need for an economy pricing resource management and scheduling system comes from the answers to the following questions:

- What comprises the grid of and who owns its resources?
- What motivates resource owners to contribute their resource to the Grid?
- Is it possible to have access to all resources in the Grid by contributing our resource?
- If not, how to access all the grid resources?

- If resources are accessed through collaboration, can they be used for any other purpose?
- Do resource owner charge consumer with the same price or different prices
- Is the access cost same for peak hours?
- How can resource owners maximize their profit?
- How can the users solve their problems with a minimum cost?
- How can a user get high priority over others?

If the user relaxes the deadline by which results are derived, can solution cost be reduced? The grid resource management system must dynamically trade for the best resources based on the metric of the price and performance available and schedule computations on these resources such that they meet users' requirements. The grid middleware needs to offer services that help resource brokers and resource owners to trade for resource access.

The benefits of economic-based resource management include the following:

- It helps in building a large-scale grid as it offers incentive for resource owners by contributing their (idle) resources to others and earns profit from it.
- It helps in regulating the supply and demand for resources.
- It offers an economic incentive for users to back off when solving low priority problems and thus encourages the solution of time critical problems first.

- It removes the need for a central coordinator (during negotiation).
- It offers a uniform treatment of all resources. That is, it allows trading of everything including computational power, memory, storage, network bandwidth/latency , data, and devices or instruments.
- It allows users to express their requirements and objectives.
- It helps in developing scheduling policies that are user-centric rather than system-centric.
- It offers an efficient mechanism for allocation and management of resources.
- It helps in building a highly scalable system as the decision-making process is distributed across all users and resource owners.
- It supports a simple and effective basis for offering differentiated services for different applications at different times.
- Finally, it provides an advantage for the both resource owners and users so that they can make their own decisions to maximize the utility and profit.

4.3 REQUIREMENTS FOR ECONOMIC PRICING BASED GRID SYSTEMS

To offer greater value to users than traditional systems, the economic-based resource management system needs to provide mechanisms and tools that allow resource consumers and providers to express their requirements and facilitate the realization of their goals. They need

- The means to express their requirements, valuations, and objectives [value expression].
- Scheduling policies to translate them to resource allocations [value translation].
- Mechanisms to enforce selection and allocation of differential services, and dynamic adaptation to changes in their availability at runtime [value enforcement].

Similar requirements are raised (Chun 2001) for market-based systems in a single administrative domain environment such as clusters and they are limited to co-operative economic models since they aim for social welfare. The grids need to use competitive economic models as different resource providers and resource consumers have different goals, objectives, strategies, and requirements that vary with time.

Essentially, resource consumers need a utility model to allow them to specify resource requirements and constraints. For example, the Globus allows the users to specify the deadline and budget constraints along with optimization parameters such as optimization of time [value expression]. They need brokers that provide strategies for choosing appropriate resources [value translation] and dynamically adapt to changes in resource availability at runtime to meet user requirements [value enforcement].

Some research systems support resource reservation in advance (e.g., reserving a slot from time t1 to t2 using the Globus GARA and bind a job to it) as proposed by Foster et al (2000) and allocate resources during reserved time [value enforcement]. A number of research systems have explored QoS based resource (e.g., CPU time and network bandwidth) as suggested by Lazar and Semret (1997) allocation in operating systems and queuing systems, but the inclusion of QoS into mainstream systems has been slow paced (e.g., Internet mostly uses the best effort allocation policy McKnight and Boroumand (2000), but this is changing with IPv6).

4.4 GRID RESOURCE ARCHITECTURE FOR ECONOMY PRICING

A Grid Resource Architecture for Economic Pricing (GRAEP) is shown in Figure 4.1. This architecture is generic enough to accommodate different economic models used for resource trading and determine the service access cost. The key components of the grid include,

- Grid User with Applications (sequential, parametric, parallel, or collaborative applications)
- User-Level Middleware Higher Level Services and Tools
- Programming Environments
- Grid Resource Brokers
- Core Grid Middleware (services resource trading and coupling distributed wide area resources)
- Grid Service Providers

Figure 4.1 A Generic Grid Resource Architecture For Economic Pricing

The two key players in the market oriented computational grids are resource providers (**hereafter referred as GSPs—Grid Service Providers**) and resource consumers (**hereafter referred as GRBs— Grid Resource Broker)**

4.4.1 Grid Resource Broker

GRB acts as a consumer's representative or software agent. Both have their own expectations and strategies for being part of the grid. In the grid economy, resource consumers adopt the strategy of solving their problems within a required timeframe and budget.

Resource providers adopt the strategy of obtaining best possible return on their investment. The resource owners try to maximize their resource utilization by offering a competitive service access cost in order to attract consumers. The users (resource consumers) have an option of choosing the providers who offer the service that best meet their requirements. If resource providers have local users, they will try to recoup the best possible return on "idle/leftover" resources. In order to achieve this, the grid systems need to offer tools and mechanisms that allow both resource providers and consumers to express their requirements.

The grid resource consumers interact with brokers to express their requirements such as the budget that they are willing to invest for solving a given problem and a deadline, a timeframe by which they need results. They also need capability to trade between these two requirements and steer the computations accordingly. The GSPs need tools for expressing their pricing policies and mechanisms that help them to maximize the profit and resource utilization. The various economic models, ranging from commodity market to auction-based models can be adopted for resource trading in the grid computing environments.

The GRAEP provides services that help both resource owners and users maximize their objective functions. The resource providers can contribute their resources to the grid and charge for services. They can use GRAEP mechanisms to define their charging and access policies and the GRAEP resource trader works according to those policies. The users interact with the Grid by defining their requirements through high-level tools such as resource brokers (also known as grid schedulers). The resource brokers work for the consumers and attempt to maximize user utility. They can use GRAEP services for resource trading and identifying GSPs that meets its requirements.

As mentioned earlier, the goal is to realize this grid economy architecture by leveraging existing technologies such as Globus and Legion and develop new services that are particularly missing in them. Therefore, focus is mainly on two things: first, to develop middleware services for resource trading using different economic models; second to use these services along with other middleware services in developing advanced user-centric grid resource brokers. The remainder of this section presents how to realize the grid economy vision and show co-existence of our modules with other systems.

4.4.2 The Resource Broker

It acts as a mediator between the user and grid resources using middleware services. It is responsible for resource discovery, resource selection, binding of software, data, and hardware resources, initiating computations, adapting to the changes in the grid resources and presenting the grid to the user as a single, unified resource. The resource broker consists of the following components:

4.4.3 Job Control Agent

Job Control Agent (JCA) is a persistent control engine responsible for shepherding a job through the system. It coordinates with schedule advisor for schedule generation, handles actual creation of jobs, maintenance of job status, interacting with clients/users, schedule advisor, and dispatcher.

4.4.4 Schedule Advisor

It is responsible for resource discovery (using the grid explorer), resource selection and job assignment (schedule generation) to ensure that the user requirements are met with.

4.4.5 Grid Explorer

It is responsible for resource discovery by interacting with the grid information server and identifying the list of authorized machines, and keeping track of resource status information.

4.4.6 Trade Manager

It works under the direction of resource selection algorithm (the schedule advisor) to identify resource access costs. It uses market directory services and GRAEP negotiation services for trading with grid service providers (i.e., their representative trade servers).

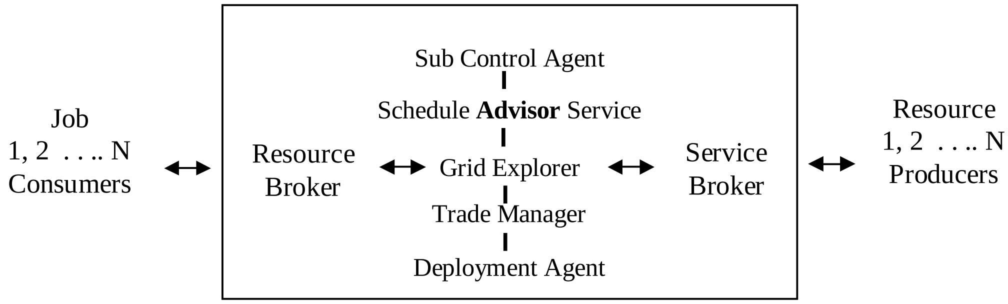

Figure 4.2 Components of Economic Model

4.4.7 Deployment Agent

It is responsible for activating task execution on the selected resource as per the scheduler's instruction and periodically updates the status of task execution to JCA.

The Nimrod-G resource broker follows this architecture and offers functionalities that are expected from economic-based grid scheduling systems. It allows the users to submit their application created using its parameter specification language; and express their requirements and objectives in the form of deadline, budget with time or cost as the optimization parameter [value expression]. The broker uses scheduling algorithms to select resources dynamically at runtime depending on their availability, capability, and cost to meet user requirements [value translation]. It continuously adapts to changes in resource availability conditions by performance profiling (establishing job completion rate) and reschedules jobs appropriately to ensure that users' requirements are met with [value enforcement].

4.5 GRAEP FRAMEWORK

The grid middleware offers services that help in coupling a grid user and remote resources through a resource broker or Grid enabled application. It offers core services such as remote process management, coallocation of resources, storage access, directory information, security, authentication, and QoS such as resource reservation for guaranteed availability and trading for minimizing computational cost. Many of these services are already offered by Globus components and they include,

- Resource allocation and process management.
- Resource Co-allocation services.
- Unicast and Multicast communications services.
- Authentication and related security services.
- Distributed access to structure and state information.
- Status and Health Monitoring components.
- Remote access to data via sequential and parallel interfaces.
- Construction, caching, and location of executables.
- Advanced resource reservation.

A layered architecture for the realization of the GRAEP framework is shown in Figure 4.1. It offers a grid economy infrastructure that co-exists with or built on top of the existing middleware such as Globus:

- Applications
- Problem solving
- Various resource trading protocols

- A mediator for negotiating between users and grid service providers (Grid Market Directory)
- A deal template for specifying resource requirements and service offers
- A trade server
- A pricing policy specification
- Accounting (e.g., Query Bank (Jackson 2000))and payment management (Grid Bank)

The proposed middleware services are designed to offer low-level services that co-exist with Pricing Policies

4.5.1 Pricing Policies

These define the prices that resource owners would like to charge users. The resource owners may follow various policies to maximize their profit and resource utilization and the price they charge may vary with time and one user to another user. The pricing can also be driven by demand and supply like in the real market environment. That is, in this commodity market model, pricing is essentially determined by objective functions of service providers and users.

The pricing policy can also be based on auction. In this auction based economic model, pricing is driven by how much users value the service and the highest bidder wins the access to Grid services. Resource Accounting and Charging components (such as Grid Bank along with Query Bank) are responsible for recording resource usage and bill the user as per the usage agreement between resource broker (TM, a user agent) and trade server (resource owner agent). The service providers publish their services through the Grid Market Directory (GMD). They use the grid trading services' declarative language for defining cost specification and their objectives such as access price for various users for different times and durations, along with possibilities of offering discounts to attract users during peak off hours.

The Grid Trading Server (GTS) can employ different economic models in providing the services. The simplest model would be a commodity model wherein the resource owners define pricing strategies including those driven by the demand and resource availability. The GTS can act as auctioneer if the Auction-based model is used in deciding the service access price or else an external auctioneer service can be used.

4.5.2 Grid Open Trading Protocols

The resource trading protocols define the rules and format for exchanging commands between a GRAEP client (Trade Manager), which is part of the Grid broker and a Trade Server, which is part of the Grid service providers. Figure 4.3 shows a sample multilevel negotiation protocol that both client and server need to follow while trading for the cost of resource access designed by Buyya et al (2000). The wire-level (low-level) details of these protocols are skipped, as they are obvious.

A finite state machine representation of multilevel negotiation protocols that both client and server need to follow for the bargaining/tender model is shown in Figure 4.4. In this model, the broker's Trade Manager (TMs) contacts the resource owner's Trade Server (TS) with a request for a quote. The TM specifies resource requirements in a Deal Template (DT), which can be represented by a simple structure with its fields corresponding to deal items or by a "Deal Template Specification Language", similar to the

ClassAds mechanism employed by the Condor system which developed by Basney and Livny (2009).

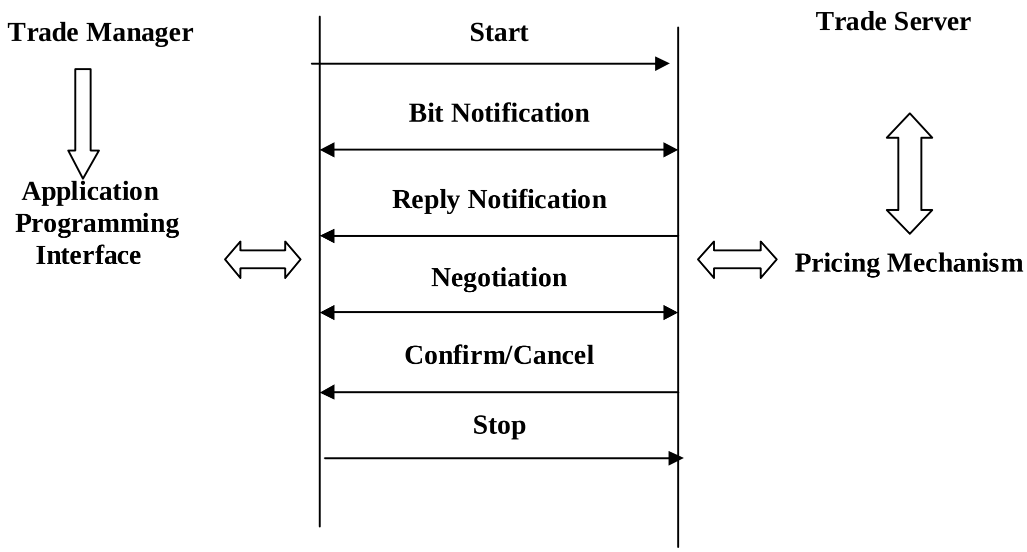

Figure 4.3 Trading Protocols

The contents of DT include expected start time, usage duration, memory, and storage requirements along with its initial offer. The TM looks into DT and updates its contents and sends back to TS. This negotiation between TM and TS continues until one of them indicates that its offer is final. Following this, the other party decides whether to accept or reject the deal. If accepted, then both work as per the agreement mentioned in the deal. The overhead introduced by the multilevel point-to-point protocol can be reduced when resource access prices are announced through Grid information services (e.g., MDS) or market directory.

A number of interaction protocols for a business negotiation on the Internet have been presented in the reference Kumar and Feldman (2008). It highlights some commonalities in the structure of different price negotiation mechanisms such as fixed price sales, auctions, and brokerages. These business negotiation models and protocols are also applicable for the resource trading and such models and protocols have been explored in the resource management and scheduling system.

4.5.3 Grid Open Trading APIs

The GRAEP infrastructure supports generic Application Programming Interfaces (APIs) that can be used by the Grid tools and application programmers to develop software supporting the computational economy. The following trading APIs are C-like functions (high level view of trading protocols) that GRAEP clients/brokers can use to communicate with trading servers:

- grid_trade_connect(resource_id, tid)
- grid_request_quote(tid, DT)
- grid_trade_negotiate (tid, DT)
- grid_trade_confirm(tid, DT)
- grid_trade_cancel(tid, DT)
- grid_trade_change(tid, DT)
- grid_trade_reconnect(tid, resource_id)
- grid_trade_disconnect(tid)

where,

tid = Trade Identification code DT = Deal Template

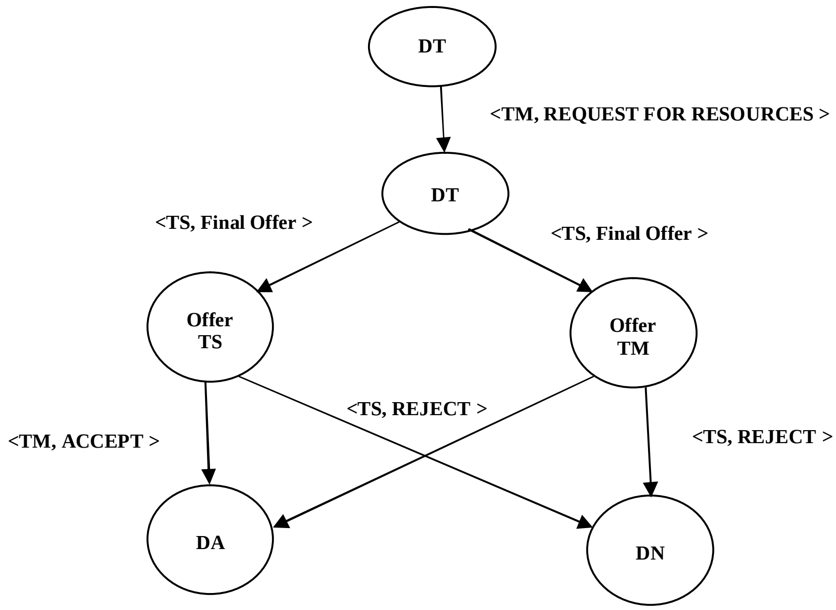

Figure 4.4 Finite State of Trading

DT -Deal Template TM-Trade Manager TS-Trade Server DA-Deal Accepted DN-Deal Not accepted

4.5.4 Pricing, Accounting and Payment Mechanisms

In a computational Grid economy environment, both resource owners and users want to maximize their benefits. As there are many GSPs offering similar services, they need to have a competitive pricing structure in order to attract users, efficiently utilize resources, and maximize profit. The resources consumed by the user applications need to be accounted for and charged. Various payment mechanisms need to be supported. The users can purchase resource access credits in advance or pay-after-usage. Each GSP can maintain this by using systems like QBank or there can be global Grid-wide bank called GridBank (GBank) that mediates payment for services accessed by the user. Figure 4.5 shows various components at a GSP node and their interactions during resource trading, consumption, metering (measuring), billing, and payment handling.

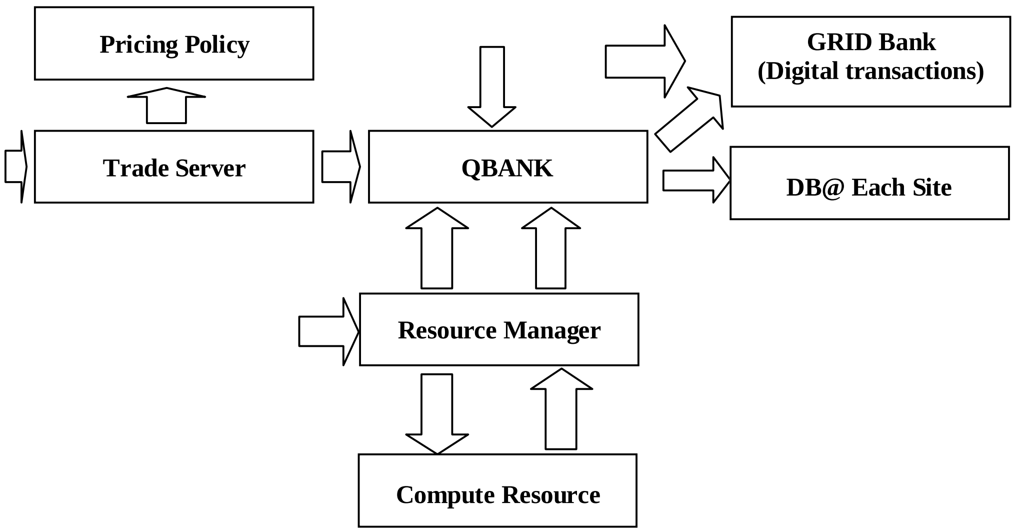

Figure 4.5 Interactions between Resource Management Components

4.5.5 Determining the Price

A simple pricing scheme is a fixed price model, but this does not work when the users place demands that vary with applications and time. In the context of software agents, many researchers have investigated pricing schemes based on the supply and demand for resources and the quality requirements. The pricing schemes based on different parameters include

- Competitive economic models (e.g., auctions and contract-net)
- Usage timing (Peak On, Peak Off, lunch time like pricing telephone services)
- Usage period and duration (short/long)
- Demand and supply (e.g., Smale model)

- Foresight-based (i.e., an ability to model and predict responses by competitors)
- Loyalty of Customers (like Airlines favoring frequent flyers!)
- Historical data
- Advance agreement/contract with service provides
- Calendar based
- Bulk Purchase
- Voting in which trade unions decide pricing structure
- Resource capability as benchmarked in the capital market

Five different pricing strategies, ranging widely from one that requires perfect knowledge and unlimited computational power to one that requires very little knowledge or computational capability, are employed in two different buyer populations, namely quality-sensitive and price sensitive buyers. The resulting collective dynamics have been investigated using a combination of analysis and simulation. In the population of quality-sensitive buyers, all pricing strategies lead to a price equilibrium predicted by a game-theoretic analysis. However, in a population of price-sensitive buyers, most pricing strategies lead to large-amplitude cyclical price wars. These pricing strategies and issues are also applicable to the Grid and strategies need to be designed such that the resource provider's benefit through efficient resource utilization and consumers will have the ability to trade-off between cost and timeframe in the grid marketplace.

4.5.6 Service Items to be Charged and Accounted

User applications have different resource requirements depending on computations performed and algorithms used in solving problems. Some applications can be CPU intensive while others can be I/O intensive or a combination. For example, in CPU intensive applications it may be sufficient to charge only for CPU time whilst offering free I/O operations. This scheme cannot be applied for I/O intensive applications. Therefore, the consumption of the following resources needs to be accounted and charged:

- CPU -User time (consumed by user App.) and System time (consumed while serving user App.)
- Memory
- Maximum resident set size -page size
- Amount of memory used
- Page faults
- Storage used
- Network bandwidth consumption
- Signals received, context switches

Software and Libraries accessed (particularly required for the emerging Active Server Pages). Access to each of these entities can be charged individually or in combination. Combined pricing schemes need to have a costing matrix that takes a request for multiple resources in pricing which is a method of grid resource allocation. The grid commerce is one such framework investigating the use and enhancing the Smale model for devising pricing strategies in the context of allocating resources for Grid users. By simulating hypothetical resource consumers and resource producers, they measure the efficiency of resource allocation under two different market conditions:

1. Demand based Pricing Model
2. Best Bid Pricing Model

By comparing the results of both market strategies in terms of price stability, market equilibrium, consumer efficiency, and producer efficiency, the G-commerce concludes that commodity market is a better choice for controlling Grid resources than the existing auction strategies.

4.6 PRICING MODELS

4.6.1 Demand based Pricing Model

In the commodity market model, resource owners specify their service price and charge users according to the amount of resource they consume. The pricing policy can be derived from various parameters and can be flat or variable depending on the resource supply and demand. In general, services are priced in such a way that supply and demand equilibrium is maintained. In the flat price model, once pricing is fixed for a certain period, it remains the same irrespective of service quality. It is not significantly influenced by the demand, whereas in a supply and demand model, prices change very often based on supply and demand changes. In principle, when the demand increases or supply decreases, prices are increased until there exists equilibrium between supply and demand. Pricing schemes in a Commodity Market Model can be based on:

- Flat Fee
- Usage Duration (Time)
- Subscription

Demand and Supply model based has been presented by McKnight and Boroumand (2000) The resource owners publish their prices and rules in the Grid Market Directory (GMD) service as shown in figure 4.6 similar to publishing through yellow pages. This is accomplished by defining the price specification that Grid Trade Server (GTS) can use for publishing service access price in the market directory. A simple price specification may contain the following parameters.

- consumer_id, which is as same Grid-ID
- peak_time_price (say, between 9am-6pm: office hours on working days)
- lunch_time_price , offpeak_time_price
- discount_when_lightly_loaded (i.e., if the load is less than 50% at any time)
- raise_price_high_demand (i.e., raise in price if the average load is above 50%)
- price_holiday_time (i.e., during holidays and weekends)

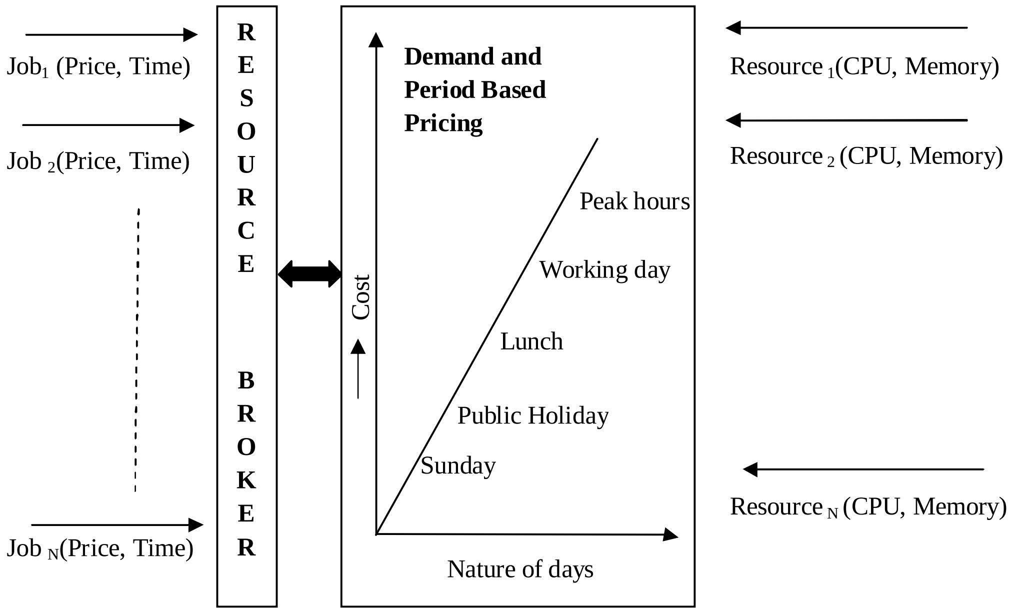

Figure 4.6 Demand Pricing Model

Traditionally, computational services are priced based on their production cost and desired profit margin. However, the consumers' perception of value is based on parameters such as supply and demand for resources, priority and service quality requirements. Therefore, the resource value in Grid economy needs to be defined as a function of many parameters as follow:

Resource Value = Function (Resource strength, Cost of physical resources, Service overhead, Demand, Value perceived by the user, Preferences);

The last three parameters are difficult to capture from consumers unless they see any benefit in disclosing them as they vary with time and application. However, there are consumers who prefer regular access to resources during a particular period of the day. For example, those involved in making regular decisions on supply chain management of goods shipping from inventory to the departmental stores prefer calendar-based guaranteed access, and stable but competitive pricing to resources unlike spot-market based access to services (Norskog 2005).

In this case, demand and preferences are clear, pricing policy can be easily negotiated in advance in a competitive and reasonable manner and QoS of resource can be guaranteed through reservation during the required period as agreed in advance.

Consumers can be charged for access to various resources including CPU cycles, storage, software, and network. The users compose their application using higher-level grid programming languages. For example, in the Nimrod problem-solving environment, a declarative programming language is provided for composing parameter sweep applications and defining application and user requirements such as deadline and budget.

4.6.2 Best Bid Pricing Model

The Best Bid (BB) auction model supports one-to-many negotiation, between a service provider (seller) and many consumers (buyers), and reduces negotiation to a single value (i.e., price).

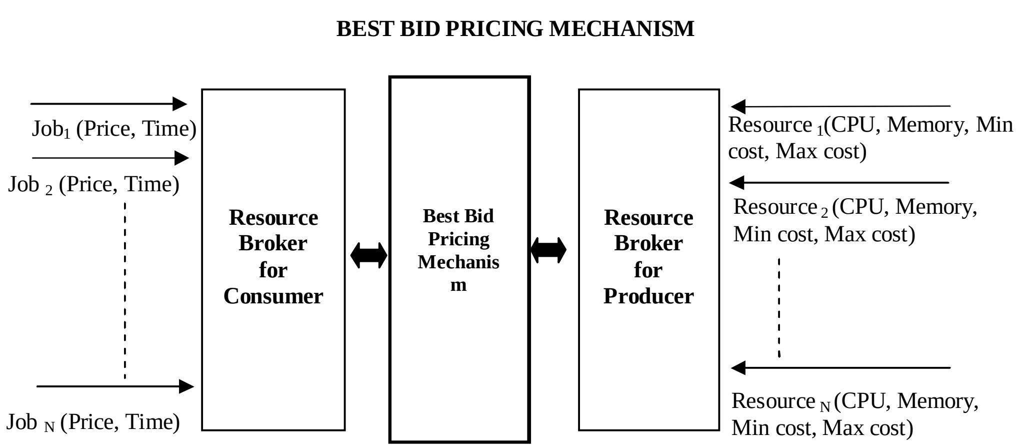

Figure 4.7 Best Bid Pricing Model

The auctioneer sets the rules of auction, acceptable for the consumers and the providers. The auctions basically use market forces to negotiate a clearing price for the service.

In the real world, auctions are used extensively, particularly for selling goods/items within the set duration. The three key players involved in auctions are: resource owners, auctioneers (mediators), and buyers (Figure 4.7). Many e-commerce portals such as Amazon.com and eBay.com are serving as mediators (auctioneers). Both buyers' and sellers' roles can also be automated. In the grid environment, providers can use an auction protocol for deciding service value/price (Figure 4.7). The steps involved in the auction process are:

a. GSPs announce their services and invite bids.

b. Brokers offer their bids (and they can see what other consumers offer if they like -depending on open/closed).

c. Step (b) goes on until no one is willing to bid higher price or auctioneer stops if the minimum price line is not met.

d. GSP offers service to the one who wins.

e. Consumer uses the resource.

Auctions can be conducted as open or closed depending on whether they allow back-and-forth offers and counter offers. The consumer may update the bid and the provider may update the offered sale price.

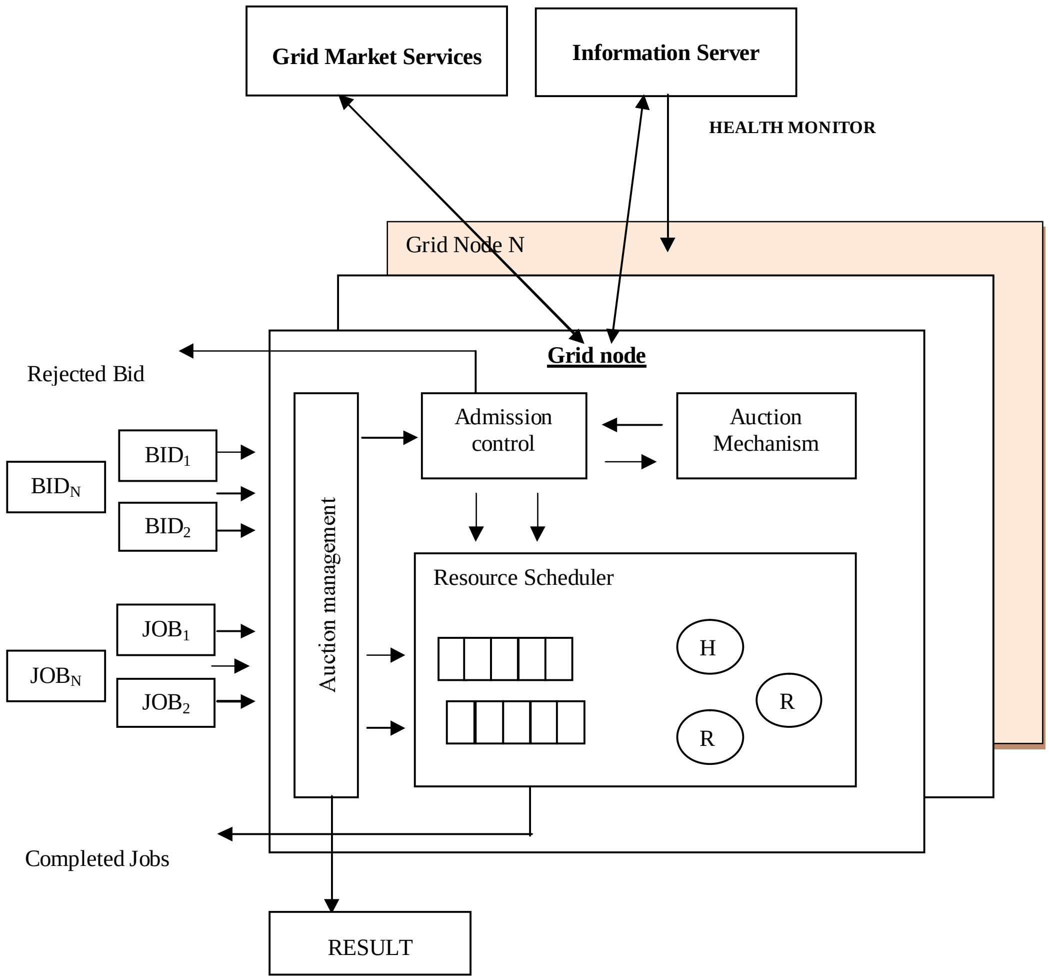

Figure 4.8 Auctions using their own Auctioneer

First the user submits his job with corresponding cost (bid) to the Auction Agent. If the cost is less than the market quote offered by the grid market then concerned job will be rejected as the rejected bid. If the cost is above the market quote, the job is admitted by Admission Control Unit, The jobs and resources are synchronized by the Auction Mechanism Unit and send to Resource Scheduler. The data are processed and dispatched by Resource Scheduler as shown in Figure 4.8.

The double auction model has high potential for Grid computing. The brokers can easily be enabled to issue bids depending on budget, deadline, job complexity, scheduling strategy, and resource characteristics requirements and GSPs can issue asks depending on current load and perceived demand, and price constraints. Both orders can be submitted to GMD agents that provide continuous clearance or matching services. Since bids are cleared continuously, both GRBs and GSPs can make instant decisions with less computational overhead and complexity.

All the above auctions differ in terms of whether they are performed as open or closed auctions and the offer price for the highest bidder. In open auctions, bidding agents can know the bid value of other agents and will have an opportunity to offer competitive bids. In closed auctions, the participants' bids are not disclosed to others. Auctions can suffer from collusion (if bidders coordinate their bid prices so that the bids stay artificially low), deceptive auctioneers in the case of a Vickrey auction (auctioneer may overstate the second highest bid to the highest bidder unless that bidder can vary it), deceptive bidders, counter speculation, etc

4.7 PERFORMANCE AND EVALUATION

Globus is an open source toolkit which allows the people share their databases, applications, computational power and other tools over the Internet in terms of secure and seamless manner. The Globus can be thought as a middleware that includes software services and libraries for resource monitoring, discovery allocation and scheduling management. It is portable so it can be used for any platform. Resource management and Job management are supported by the Globus. It also provides web mechanisms to create distributed system framework. The proposed scheduling framework can easily be implemented with this toolkit for the following experiments.

Experiment 1

The experiments are run twice, once during the Indian peak time, when the US machines are in their peak off times, and again during the US peak on time, when the Indian machine are peak off. The experiments are configured to minimize the cost, within one-hour deadline. This requirement instructs the GLOBUS to use "Cost-Optimization Scheduling" algorithm in scheduling jobs for processing on the Grid.

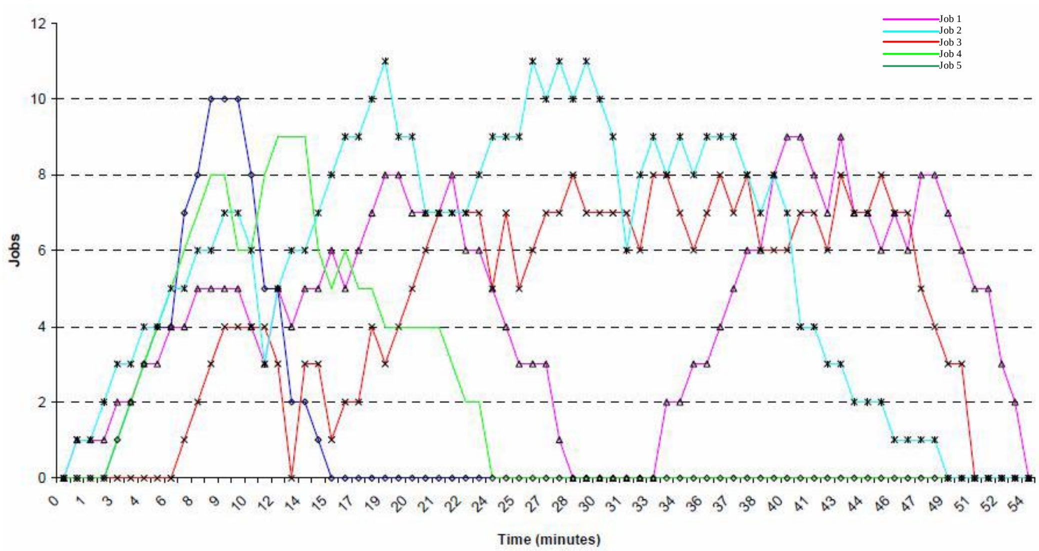

Figure 4.9 Computational Scheduling Indian Peak ON (US Peak OFF)

The number of jobs in execution/queued on resources during the Indian peak on and peak off time scheduling experimentations are shown in Figures 4.9 and 4.10 respectively. The results for the Indian peak on experiment show the expected typical results. After an initial calibration phase, the jobs are distributed to the cheapest machines for the remainder of the experiment. This characteristic of the scheduler is clearly visible in both the experiments. In the Indian peak experiment, after calibration period, the scheduler excludes the usage of Indian resources as they are expensive and the scheduler predicts that it can still meet the deadline using cheaper resources from US resources, which are in peak off time phase.

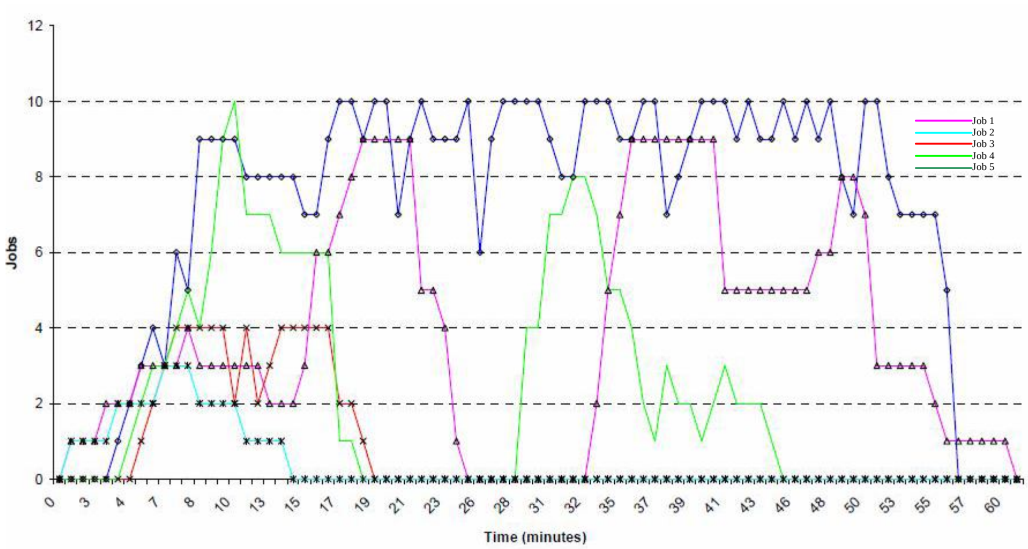

Figure 4.10 Computational Scheduling US Peak ON (Indian Peak OFF)

The number of jobs in execution/queued on resources during the Indian peak on and peak off time scheduling experimentations are shown in Figures 4.9 and 4.10 respectively. The results for the Indian peak on experiment show the expected typical results. After an initial calibration phase, the jobs are distributed to the cheapest machines for the remainder of the experiment. This characteristic of the scheduler is clearly visible in both the experiments. In the Indian peak experiment, after calibration period, the scheduler excludes the usage of Indian resources as they are expensive and the scheduler predicts that it can still meet the deadline using cheaper resources from US resources, which are in peak off time phase. However, in the Indian peak off experiment, the scheduler never excludes the usage of Indian resources and excludes the usage of some of the US resources, as they are expensive comparatively at that time (US in peak on time phase). When the Sun machine becomes temporarily unavailable, the SP2, at the same cost, is also busy; a more expensive SGI is used to keep the experiment on track to complete before the deadline.

The computational economies driven brokering system can be applied to peer-to-peer computing Chao et al (2008)

The Time Optimization scheduling algorithm attempts to complete the experiment as quickly as possible, within the budget available. A description of the core of the algorithm is as follows:

1. For each resource, calculate the next completion time for an assigned job, taking into account previously assigned jobs and job consumption rate.
2. Sort resources by next completion time.
3. Assign one job to the first resource for which the cost per job is less than or equal to the remaining budget per job.
4. Repeat the above steps until all jobs are assigned.

The Cost Optimization scheduling algorithm attempts to complete the experiment as economically as possible within the deadline.

1. Sort resources by increasing cost.
2. For each resource in order, assign as many jobs as possible to the resource, without exceeding the deadline.

Cost Optimization Scheduling: In a competitive commodity-market economy, the resources are priced differently at different times based on the supply and demand. For example, they are priced higher during peak hours and lower during off-peak hours. In this experiment, their impact on the processing cost is applied, by scheduling a resource-intensive parameter-sweep application containing a large number of jobs on the World-Wide Grid resources, during Indian peak on and peak off hours.

The World-Wide Grid test bed resources selected for use in this experiment and their properties is shown in Table 4.1. To test the trading services provided by GTS (Grid Trade Server), experiment is conducted entirely during peak on time and the same experiment entirely during peak off time. It is important to note access price variations during peak on and peak off times and also time difference between Indian and US. The access price is expressed in Grid units (Rs) per CPU second.

Table 4.1 Resource, Organizations and Unit Price

Resource Type and Size (No. of Nodes)	**Organization and Location**	**Grid Services and Fabric**	**Price @ Indian peak time**	**Price @ AU off peak time**
Linux cluster (60 nodes)	Delhi, India	Globus/Condor	20	5
IBM SP2 (80 nodes)	ANL, USA	Globus/LL	5	10
Sun (8 nodes)	ANL, USA	Globus/Fork	5	10
SGI (96 nodes)	ANL, USA	Globus/Condor-G	15	15
SGI (10 nodes)	ISI, USA	Globus/Fork	10	20

Five resources have been selected (vide Table 4.1) from the testbed, each effectively having 10 nodes available for the experiment. Delhi University has a 60-processor Linux cluster running Condor, which is reduced to 10 available processors for the experiment. Similarly, a 96-node SGI at Argonne National Laboratory (ANL) is made to provide 10 nodes by using

Condor glide to add 10 processors to the Condor pool. An 8-node Sun at Argonne and a 10-node SGI at the Information Sciences Institute (ISI).

4.7.1 Results and Discussions

The number of computational nodes (CPUs) which are in use at different times during the execution of scheduling experimentation at Indian peak on time is shown in Table 4.2 and Figure 4.11. It can be observed that in the beginning of the experiment (calibration phase), the scheduler has precise information related to job consumption rate for resources, hence it tries to use as many resources as possible to ensure that it can meet deadline. After the Calibration phase, the scheduler predicts that it can meet the deadline with fewer resources and complete the job with more expensive nodes. However, whenever scheduler senses difficulty in meeting the deadline by using the resources currently in use, it includes additional resources. This process continues until deadline is met and at the same time it ensures that the cost of computation is within the given budget.

The total cost of resources (sum of the access price for all resources) in use at different times during the execution of scheduling experimentation at Indian peak on time is shown in Table 4.3 and Figure 4.12. It can be observed that the pattern of variation of cost during the Calibration phase is similar to that of number of resources in use. However, this is not the same as the experiment progresses and in fact the cost of resources will get decreased almost linearly although the number of resources in use does not decline at the same rate. The reason for this behavior is that a large number of resources that the scheduler selected are from peak off time zone (i.e., US is in peak off time when Indian is in peak on hours) as they are cheaper along with Grid Association which has been proposed in the third chapter to enhance the resource utilization of resources. Another reason is that the most of resources used in these experiments belong to US resources compared to Indian resources.

The similar behavior is not found in the scheduling experiments conducted during the Indian peak off time. The variation pattern of total number of resources which are in use and their total cost are similar due to the fact that the large number of US resources are available cheaply. Although the scheduler has used Indian resources throughput the experiment, the scheduler has to depend on US resources for ensuring that the deadline is met even if the resources are expensive.

Table 4.2 Time and Resources in use during Indian Peak ON Time

Time (min)	Resources (No. of CPUs in use)
3	2
6	4
9	10
12	16
15	22
18	30
21	32
24	35
27	34
30	32
33	28
36	25
39	27
42	24
45	21
48	18
51	16

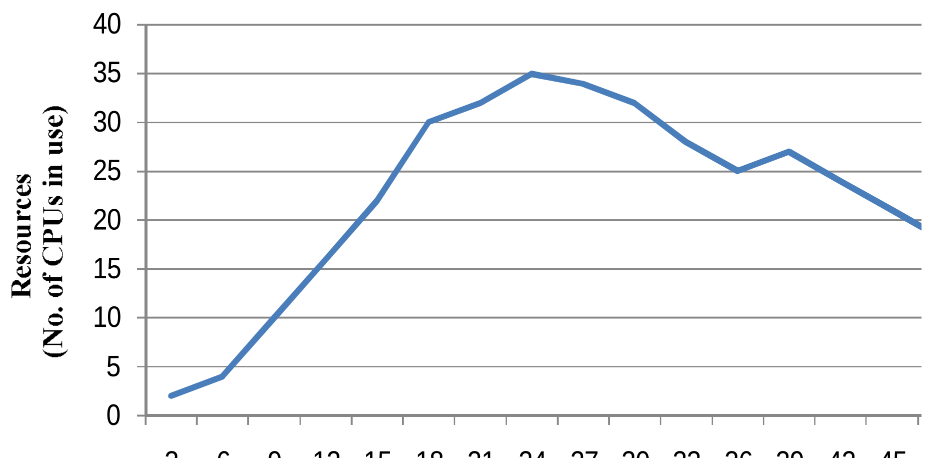

Figure 4.11 Number of Resources during Indian Peak ON Time

Table 4.3 Comparison of Cost of Resources

Time (min)	Cost of Resources in use (Units)	Cost of Resources in use with Association (Units)
3	2	2
6	4	4
9	11	11
12	17	12
15	23	15
18	29	17
21	31	18
24	33	20
27	32	18
30	30	15
33	26	14
36	22	13
39	21	14
42	16	12
45	14	10
48	14	9
51	12	8

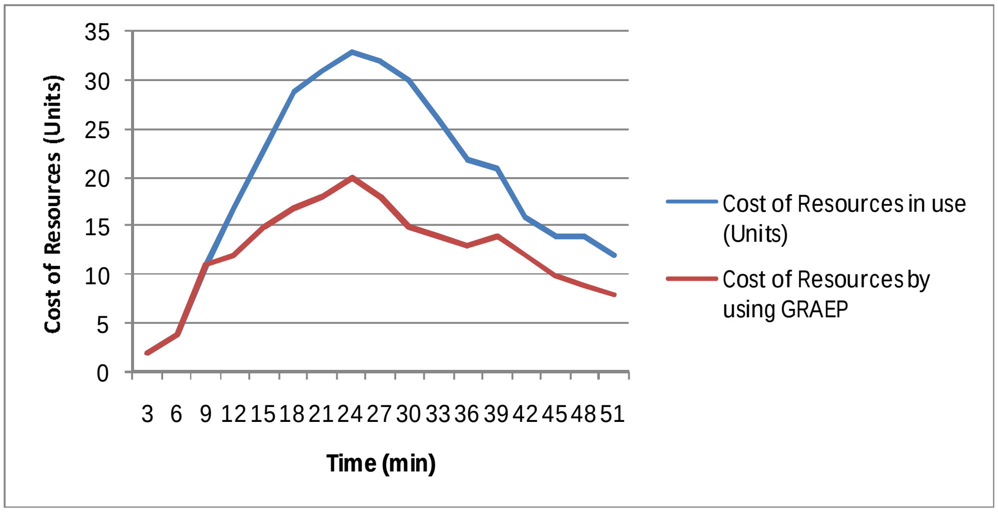

Figure 4.12 Cost of Resource in use at Indian Peak ON

4.8 CONCLUSION

The Resource Management and Scheduling systems of the grid environment need to be adaptive so that they can handle any dynamic changes in the availability for the user requirements. They must provide scalable, controllable, measurable and easily enforceable for the management of different resources. The resource broker supports the deadline and budget constraint algorithms for scheduling the job resource in the large scale distributed system. Thus this methodology is designed in the view of optimizations of cost and thereby indirectly optimizing time for market based deadline and budget constraint scheduling.

The scheduling experiments with different QoS requirements show promising results by exhibiting the effectiveness of the distributed computational economy in the management of different resources and the usefulness in the application scheduling with optimization. The resources are priced differently at different times based on demand and best bid models through the proposed competitive economic pricing scheme. The simulation results of demand based pricing model show that the cost optimization can be achieved with balancing of peak off and peak on time of grid resources. Grid Association methodology and the result shows that there is a linear reduction in the cost of resources as time progress as shown in table 4.3 and results of the Best bit pricing model show that the negotiation mechanism can be established by selecting the fastest and cheapest resources depending upon the user's constraints.

The methodology has been proposed to offer economic incentives to resource providers for sharing their resources with reduced computational cost and relaxed deadline when time frame for earliest result delivering is not too critical. It makes the users trade-off between the deadline and budgets promoting the grid system as a platform for mainstream computing which results in the emergence of a new service oriented computing industry.