Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECOND PRICE AUCTIONING FOR EFFECTIVE UTILISATION OF GRID RESOURCES

[1] THIRUNAVUKKARASU, [2] G.K.S. KARUPPASAMI
[1] Jairuppa College Of Engineering, Tiruppur, Tamil Nadu 641604, India
E-mail: [1] thirunavukkarasu465@gmail.com, [2]*karuppusami.gks@gmail.com*

ABSTRACT

In dynamic environments, if the number of users gets increased, then the workload increases thereby making the management of resources a tedious task. This demands the utilization of economical grid computing model for effective allocation of resources. In the proposed work, a second price auction mechanism is considered to employ a pricing mechanism that is incentive compatible. Such mechanisms ensure that the users' self-interests are maximized when they honestly reveal their preferences to the allocation mechanism. It can further provide powerful guarantees about optimal user behavior and can deliver highly efficient allocations. The method is very efficient and also it elicits honest information from the participants thereby successfully managing the resource allocation in the Grid environment. A Gossip based mechanism is also considered in the proposed work in order to reduce the overheads caused by the waiting queues

Keywords: *Auction, Grid, Economic Model, Resource Scheduling, Gossip.*

1. INTRODUCTION

A Grid is said to possess decentralized control where most of it's time and energy is utilized towards concentrating the resource coordination. Among the cardinal functions of a grid, resource discovery, resource maintenance, resource negotiation and dispute solving becomes the key responsibilities.

The resources in the grid environment are geographically distributed and each of them is owned by a different organization. Each of them has its own resource management mechanisms [1] and policies. Each mechanism charge different prices for different users demanding the need for the support of computational economy in resource management.

The economy model offers resource owner's better "incentive" for contributing their resources and helps recover cost they incur while serving grid users or finance services that they offer to users and also make some profit. For enhancing/expanding computational services and upgrading resources, return-on-investment mechanism helps. Usually, in a computational market environment, resource users want to minimize their expenses (the price they pay) and owners want to maximize their return-on-investment. This demands a grid resource management system to grant equal gratification.

Due to tremendous growth in wirelessly connected mobile devices, creating an enormous collective unexploited potential for resource utilization is essential. Since the mobile devices are varied like laptops, PDAs, mobile phones, etc., and they vary in computational capabilities, power, hardware and software functions, the nodes with higher computational capabilities and power can share the resources with devices of lesser capabilities. Integrating grid resource aggregation model with mobile ad hoc platform can be instantly constructed anytime and anywhere.

Opportunistic networks are a special form of mobile ad hoc networks in which an end-to-end path never exist between the sender and receiver. The link performance is of varied in nature. Opportunistic network considers mobility property as an advantage and hence ensures that the sender can send data even to unsynchronized receiver. Opportunistic network grows from a seed which is either a set of nodes employed together at the time of the initial opportunistic network deployment or a single node at unavoidable circumstances. Seed has the responsibility of expanding the network. Foreign devices, node clusters, networks and other systems can join the opportunistic network upon receiving the invitation from seed as helper. Any full-fledged opportunistic network member may be allowed to invite external nodes [2].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The issue to be solved is about seed's decision of extending the invitation to external nodes. Helpers after knowing OPPNET's goal can collaborate and execute all jobs. But the main aim of its design may be entirely different. OPPNETs having a large number of mobile nodes connected to each other can provide service at different levels to other devices. Since server's reachability is not guaranteed, user cannot rely on one particular device for a certain service.

2. RELATED WORK

2.1 Grid Computing

Different grid based economic models have been proposed over time by grid researchers [3]. Wang et. al., [4] have proposed a mobile agent based approach for building computation grid over MANET. Mobile agent is used to distribute computations and aggregate resources. The mobile agent searches for resources and executes the computations on the node that is willing to accept it and is responsible for negotiation of resource provisioning for running the computation job.

Reverse dutch auction [5] is an important milestone in grid pricing models. In this approach, the more higher the price, lesser the rounds of auction. This leads to a higher closed price which is more advantageous. However, on the contrary, the lesser the price increments, the more the rounds of auction, which makes the communication requirement, increase.

In a grid environment, the resources can be managed using a virtualization server. A power grid dispatching automation system based on virtualization is employed to design uniform resources management platform [6]. The issue is that while the communication of demands is uncertain, allocation of grid resources becomes more complex. To solve this Sundaram [7] has proposed a hardware called IPDT- FUZZY which considers the stress of grid applications with such uncertainties. Here both communication and computational demands are expressed as fuzzy numbers.

2.2 Resource Allocation

The Future Internet Era has wide use of wireless access which leads to the need for increased efficiency in resource provisioning. Panagiotis et. al., [8] have proposed a solution by exploiting the potentialities of opportunistic network. Umair et. al., [9] have proposed a novel algorithm for service composition in opportunistic networks. With the proposed algorithm, mobile users can benefit from a larger set of services available locally in an environment. Proposed service composition makes efficient service selections from devices that are in close proximity.

The availability of the service in a node is broadcast to all one hop neighbors. A mobile ad hoc service grid can also be used which combines concepts of grid on to ad hoc networks [10]. If the service provider is not within one hop, then there is a chance for resource discovery to fail. Since each node maintains the resource look up table, devices with less storage capabilities gets affected. Resource allocation issues can be well handled using auctioning models because they provide a decentralized structure. Those models are easier to implement than other economic models [11].

Michael presents a resource allocation protocol called First-Price sealed auction [12]. This auction protocol is actually an offline problem because the organizer of an auction makes decisions after receiving all the bids. Because of dynamic characteristics of computing resources in the computational grid environment, supply and demand change dynamically and therefore online auction models are more advisable.

Economy based models are emerging over recent past for more efficient management of grid resources [4]. Shared distributed computing environments having challenges in resource allocation can use models which applies well-known and proven economy mechanisms to solve them. Auctions have been particularly preferred by many such projects – for example, Tycoon [10] and Bellagio [13] – as they provide a decentralized structure and they are easy to implement. They even provide immense flexibility to participants to specify their valuations and are considered as the most efficient among current market management systems [14].

2.3 Application Of Resource Management in OPPNETs

Howie et. al. [15], have proposed a job brokering service, called HA-JES, which is built upon the notion of computational economy. HA-JES fosters the balanced resource consumption by dynamically and transparently virtualizing underlying underutilized, and often under-priced, resources to satisfy QoS requirements from users.

The system architecture of job scheduling [16] can be categorized as centralized scheduling, hierarchical scheduling and decentralized scheduling. In centralized scheduling, central scheduler collects the information of network resources. If the sizes of the grids increase, it will

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

not be scalable and can also lead to node failure. In Hierarchical scheduling, it does not depend on a single scheduler. A different policy is used for local and global scheduling. In decentralized scheduling the distributed schedulers interact with each other and schedule jobs at remote nodes for execution. Central job pool can also be used along with direct communication by local schedulers to submit jobs. Fault tolerance and reliability can be well handled using this type of job scheduling.

An effective resource management can use gossip mechanism which helps to allocate resources with respect to the load changes and scalability. Gossip protocols will be executing dynamically and acquire synchronization. This helps to be robust against node failures. Sathiyaseelan et. al. [17], has proposed a gossip protocol which lies between distributed architecture and inner key elements.

The proposed work is based on the decentralized – second price auction based job scheduling method. The main objective is to achieve the highest possible price for the seller by motivating bidders to bid truthfully. Furthermore, second price mechanism reduces the likelihood that a bidder will overpay for an item and also increases the likelihood that the seller will get the most for an item. The comparison of existing methods is shown in table 1.

3 SECOND PRICE AUCTION MODEL

The overall block diagram shown in figure 1 describes about forming a reliable or a minimum delay service provisioning paradigm.

A simple architecture is considered in which nodes can opportunistically request service executions to the peers they can opportunistically communicate with, and collect results after the execution (on those peers) has been completed. Request is referred to as a request for the execution of a given service, and the term results denote output results of the service execution. Since the nodes are often mobile and service provisioning takes one or more contact events to complete, service provisioning in such a highly dynamic environment is a challenging task. This may increase the service delay and also leads to frequent network disconnectivity. The main focus of the proposed work is to reduce the challenges caused due to mobility. This is done by selecting relatively a stable service provider to execute the request for service such that service provisioning takes place within the contact time itself thereby reducing the delay of the overall service provisioning paradigm.

4 SERVICE PROVISIONING

In oppnets, we first deploy a seed oppnet, which may be viewed as a pretty typical ad hoc network. Various communication media like Bluetooth, wired Internet, WiFi, ham radio, RFID, satellite, etc., is used by the seed oppnet to detect foreign devices or systems. Detected systems are identified and evaluated for their usefulness and dependability as candidate nodes (which are ready to share their resources) for joining the oppnet. Best candidates are invited into the expanded oppnet.

Since mobility of the nodes is application dependent, choosing the correct application domain to model the mobility behavior of the nodes is a very important criterion to be considered. To develop a mobility model is a very difficult task since the mobile nodes possess randomness in their movement. Hence an application domain is considered where the regularity in the movement of the nodes is modeled. Typical application domains of opportunistic network are pervasive healthcare, intelligent transportation system and crisis management.

A mobility model that records regular movements is used to construct a movement pattern which consists of a series of stable positions and associated time duration to efficiently manage mobility. A very simple architecture (service provisioning paradigm) as shown in figure 2 is considered in which encountered peers can be opportunistically requested for service executions by the nodes. Then the nodes collect results after the execution (on those peers) has been completed. The members of service provisioning environment are the nodes in the network which are classified based on the computational capability and power. Node which is ready to provide service with higher computational capability is called as service provider node (SPN) and the node which requests for service is called Service requesting node (SRN).

The SRN will initiate to form an initial service provisioning environment by sending the initiation message to all the service providers. After getting the initiation message, the service providers will respond with the service response to the corresponding service requester to whom it wishes to provide the services thereafter.

After getting responses from the service providers, a cache will be formed at the service requesting side which consists of a list of service providers intended for that service requester. When a request has been generated at the service requester, it sees into its cache the corresponding service provider node which will be free to execute

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the request. Then the service request will be forwarded to the service provider which has been selected for the service execution.

Since the nodes are often mobile, service provisioning in such a highly dynamic environment takes one or more contact events to complete. Hence it is a challenging task to provide service provisioning in such a mobile environment due to the frequent mobility of the nodes which may increase the service delay and also leads to performance degradation of the network. The main focus is to reduce the delay by selecting the stable service provider node for service execution such that service provisioning takes place within the contact time itself.

5 GRID BASED RESOURCE MANAGEMENT

The second part of figure 1 describes about resource coordination of service provider in an effective manner using Grid computing. Continuous resource reallocation is needed because resource availability is subject to change quickly. It helps to provide graceful degradation during overloads or quality-of-service (QoS) improvements when more resources become available. An opportunistic network that is designed for adaptive and autonomic reaction to failures and overloads should take advantage of the flexibility of the service it provides. Best-effort connections [20] can tolerate any changes in their allocation, but real-time flows might require a fixed allocation. If not, the so far accrued utility will be lost. The flows in the network can be regularly adapted to achieve optimized QoS if every service is associated with multiple levels of acceptable quality.

We consider the environment which consists of a large number of specialized as well as multipurpose devices. Many of them are portable and are linked through wireless connections, with fluctuating link availability. Ideally, such pervasive networks can enable a broad range of distributed applications that need exchange of information between multiple devices.

A node willing to provide service with higher computational capability and power is called as a Grid service provider node (GSP) and the node which requests for the service is called as a Grid service requester node (GSR). The GSPs and GSRs are the members of the grid. The nodes that are willing to share their resources specify a cost for their resources. The service requester node accepts a service based on the cost, service time, etc. This leads to a negotiation between SRN and SPN. Since opportunistic network is an infrastructure-less network, there is no centralized authority to keep track of the negotiation between SRN and SPN. In order to form a grid and to keep track of the negotiation, we have an SPN that volunteers to act as a grid head node (GHN). The block diagram for economic based grid computing is depicted in the figure 3.

5.1 Resource discovery

The most important step in resource allocation is resource discovery. Since some resource information (e.g., CPU load or available storage) changes dynamically, the resource discovery process becomes complicated. Resource attribute and status information are maintained in a distributed database when we use any resource discovery technique. Each technique differs in the way they update, organize, or maintain the distributed database. Hence we need to devise fault tolerant and highly scalable discovery techniques. The following trade-offs are best measures to find the good grid resource discovery mechanism: (i) identifying resources in shortest time span, (ii) identifying resources which shall be utilized to the fullest extent and (iii) identifying resources at minimum investment

The GHN which is willing to provide service will initiate the action of forming the grid by sending a grid_hello_message as call for proposal (CFP). The nodes that are willing to be a member of a grid respond to the grid_hello_message. The CFP consists of node ID, stability time, position and hop count. The node ID gives the identification of the sending node; and stability time and position are obtained from its trace file. They denote the current position and the associated stability time. Hop count helps to bounds the propagation of the grid_hello_message to a fixed number of hops. This helps to avoid the formation of one large centralized grid, and instead facilitates multiple decentralized grid structures. A node decides whether it wants to become a member or wants to request for service after receiving a grid_hello_message. It sends a response message depending on the decision.

The node joining a grid sends a grid_joining_message. The grid_joining_message consists of SPN ID – ID of the joining node, GHN ID – head ID, Resource parameter indicates the resource parameter that is available with a SPN, service fee, Position and Stability. The service fee indicates at what cost it will service a request.

The GHN (Grid Head Node) forms a grid table after receiving responses from the member nodes as shown in table 2. This table maintains the details

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

like node ID column which lists the identification of the member nodes. The resource parameters columns specify the resources available with that node like computational capability, power, storage etc. Type of service indicates what type of service is needed by a Service Requester (SR). Service fee of a GSP (Grid Service provider) specifies at what cost it will service a SR. Price of a SR specifies at what price it needs a service. Position is the physical location of a node and stability is how much time a node is going to be present at that location. Busy indicates whether a node is providing service. Free indicates that GSP is free to provide service. The head maintains all the details about its members.

5.2 Auction based job scheduling

The system model contains N wireless nodes forming group of nodes. Each group contains a Grid Head Node (GHN), Grid Service Provider (GSP) and Grid Service Requester (GSR). For each GSR, time is assumed to be slotted and divided into time intervals, each consisting of T consecutive time slots. GHN act as an auctioneer in this auction model. At the beginning of each time interval, requests by each client arrive at the GHN. Each client specifies a cost and a delay bound or deadline of t_n time slots. The request of n^{th} client is to be serviced by the nearest SPN which is assigned by the GHN no later than the time slot in each time interval. Otherwise, the request packet expires and is dropped from the system. The performance of a GSR is defined by the long-term average rate that it is served by GSP node, subject to per-packet delay constraints. The utility gained by a GSR is determined by its service rate and service time through its utility function.

Further, to impose a certain degree of fairness and avoid starving of some consumer, the assumption is made, that each consumer requires a certain lower bound on its service rate. Depending on the service delay of each consumer, the network utility factor is characterized. The incoming jobs i.e., the service request by the consumer nodes are listed as per the request arrived. The requests from the nodes which are frequently using grid resources are local request and from nodes which are new to the network may be remote request.

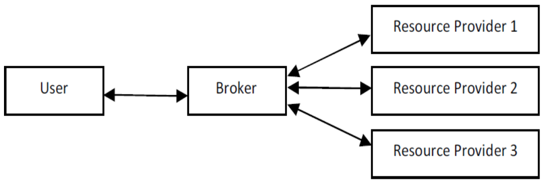

Figure 4: A Simple Grid Job Scheduling Model

Second Price auction model is a type of closed bid or sealed bid auction wherein bidders report their values to the auctioneer (GHN) for their corresponding jobs as shown in Fig 4. Each bidder doesn't know others' bidding values. In practice, utility functions may be known only to their clients. In this auction, clients offer their bids for service in each time interval. The server selects a subset of clients to serve and charges them based on their bids. The decision of selecting clients and charging them is based on second price auction. The advantage of this design is that it reduces the likelihood that a bidder will overpay for an item and it also increases the likelihood that the seller will get the most for an item. This auction is not only truthful but also utility optimal.

5.2.1 Auction algorithm

Initially, the user submits jobs to the broker with the initial price, the budget of the job and the deadline of the auction. The initial price is the highest price that the user can't regret while trading with the first bidder at this price. The budget is the maximum amount of money the user can spend on the job. The broker creates an auction and sets additional parameters of the auction. It posts the auction to itself. The broker informs the bidders that an auction is about to start. Then, the broker creates a call for proposals (CFP) and broadcasts the CFP to all the bidders. Then one auction is begun. In the process of auction, when the broker receives a bid, it will store that bid request in its cache. After three subsequent bid requests, decision is made by the Grid head node. The Grid head node selects the bid request among the bid requests received based on the second price auction model.

5.2.2 Match making

The grid matchmaking process involves three types of agents: consumers (Grid Service requesters), producers (Grid Service Providers), and a matchmaking service (Grid Head Node). A matchmaking service uses a matching algorithm to evaluate a matching function which computes the matching degree. Users describe their applications

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

with necessary attributes and submit them to Matchmaker (Grid Head Node). On the other side, resources publish information to Matchmaker. Matchmaker then matches job requests with available resources based on the attributes such as budget, deadline etc. It provides numerous advanced functionalities such as job arrays and workflows support, check pointing, job migration, rescheduling, fault recovery. It enables users to define resource requirements and rank resources and mechanism for transferring files to/from remote machines. The job scheduling is done based on the stability time and the location of GSP. A grid head node first verifies, whether the service time of a grid service requester is acceptable by GSP. If many GSPs have greater stability time, then the GSP that is nearer to GSR requesting for a service is assigned.

5.2.3 Negotiation establishment

Resource negotiation (i.e. exchange or trading of resources between Grids) appears as an important feature to enable Grids to face sudden, transient spikes of computing requests from users. The broker (Grid Head Node) creates a proposal and chooses one out of a list of resource providers based on factors such as resource price or capability to initiate a negotiation session and submit the proposal. If the proposal is within the deadline, then the proposal is accepted straightaway and then a confirmation message is returned to the service requester.

The message consists of GSR ID, GHN ID, GSP ID, Job ID, cost, position and stability. The GSR ID is the ID of the node requesting service, GHN ID is the ID of the node sending the message and GSP ID is the ID of the node that has been assigned to provide service. The job ID is a unique ID assigned by GHN to identify the communication between the grid service requester and grid service provider. Position indicates the physical position of the grid service provider that has been assigned to GSR. On receiving this message GSR starts communicating with GSP for its service. The position of GSP is available in the message; hence GSR can easily communicate with GSP using the routing protocol in the network layer.

5.3 Service Monitoring

After getting the service, the grid service requester sends an acknowledgement about its completion of the service to the Grid Head Node. The acknowledgement consists of Service completion field which indicates that the service is completed; Job ID is sent so that the Grid Head Node can understand which service was completed. Similarly the grid service provider sends a service_completion_message to the Grid Head Node after completing the service for a grid service requester. The completion message consists of GSP ID, GHN ID, job ID, WtoC, URP and service fee. To identify the job that has been completed, job ID can be used. If the SPN is willing to continue (WtoC) in a grid it sends the willingness as well as the updated resources parameters (URP) to the GHN. Using this information the GHN will know that the service has been successfully completed and updates the resource parameters of the grid service provider in its table.

5.4 Dispute Solving

After submission of request, if there is a chance of transient failure in the grid then the job has to be resubmitted again in the next attempt. At this point grid broker negotiates with the grid user to compromise for the loss of time and if grid user accepts then the jobs are executed by the available resources. The proposed dispute solver component service accepts the failure notice from the current Grid head node and submits the request to the adjacent grid head node. By this the number of requests can be reduced at the job polls in the grid head node.

5.5 Gossip Mechanism

While using gossip message, two main issues are considered, namely update spreading and error correction. A single gossip message may get multiplied when many fresh updates are given. In grid environment which has large geographically distributed data centers, the use of gossip alters the way of handling the system. The effect of using gossip mechanism is compared and evaluated with non-gossip based system.

6 RESULTS

Table 3: Parameters for Simulation

Parameters	*Values*
No of nodes	50
Simulation Time	2000 seconds
Terrain Dimensions	(1000,1000) meters
Mobility	Mobility Trace Model
MAC Protocol	802.11
Routing Protocol	DSR
Radio-Tx-Range	15 dBm

Simulation studies (table 3) have been carried out to evaluate the proposed Economic based grid computing model. A grid computing environment

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

has been simulated in this set up using 12 Service providers. We analyzed the performance using various parameters like average time a customer has to wait in queue, node position, Service Requester's service rate, average delay per customer, average number of control packets.

- Average Service Delay - Average time taken by a Grid Service Provider to service the request of Grid Service Requester.
- Number of control packets - Number of control packets involved in routing
- Average Waiting Time - Duration of time a request has to wait in a queue
- Job Success Rate - Average number of successfully completed requests

6.1 Comparative Evaluation Between Opportunistic Computing And Economy Based Grid Model

6.1.1 Number of requests in pending state

It determines the total number of requests waiting in the queue because of the busy state of the grid service provider. As the number of service requesters increases, the average number of pending requests also increases. This is because as the number of requests increase, the availability of encountering free service providers is minimum and hence more number of requests is put in the pending list. The number of requests in the pending state of both opportunistic computing and economic based grid computing models are considered.

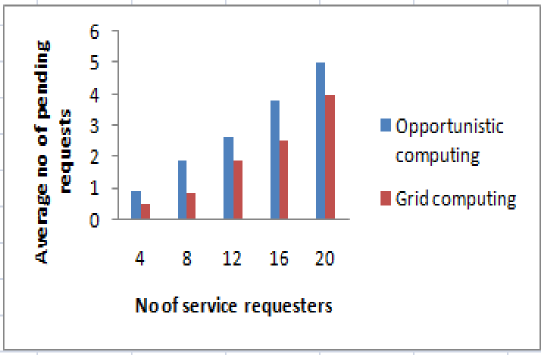

Figure 5: Average number of requests in pending state

The average number of requests in the pending state are more in Opportunistic computing compared to Economic based grid computing model. Since more number of service providers are available at the Grid head node and at the same time the Grid head node can effectively allocate the request by using an auction based mechanism, the pending requests are less in Economic based Grid computing. Figure 5 shows the results of comparison.

6.1.2 Average Waiting Time

It determines the average time a service requester needs to wait in the queue. As the number of service requesters increases, the waiting time also increases. This is because as the number of requests increase, the availability of encountering free service providers is minimum and hence the arriving request needs to wait in the queue for a long time. The average waiting time of both opportunistic computing and economic based grid computing models are considered and evaluated. It is shown in figure 6.

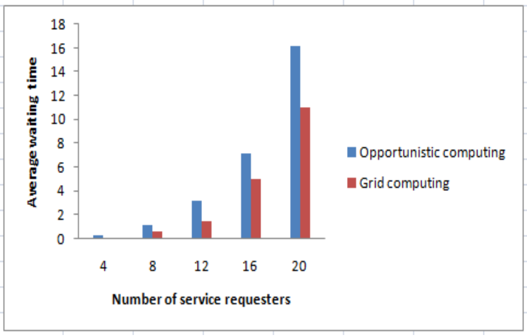

Figure 6: Average waiting time

Average waiting time in opportunistic computing is more compared to Grid computing because in opportunistic computing, the probability of encountering service providers for an arriving request is less and hence the arriving request needs longer time to wait in the queue. But in grid computing, the probability of encountering grid service providers for an arriving request is more since many service providers are available one at a time in the Grid environment. Hence an arriving request doesn't need to wait in the queue for a long time.

6.1.3 Job Completion Rate

It determines the total number of successfully completed requests. As the number of service requesters increases, the job completion rate decreases. The reason is as the number of service requesters increases, more number of requests is generated. But the available free service providers will be minimum and hence the job completion rate decreases as the number of service requesters increases.

Job completion rate in grid computing is more compared to opportunistic computing because there will be a high probability in provider availability in

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

grid computing compared to opportunistic computing. Figure 7 shows the results of comparison.

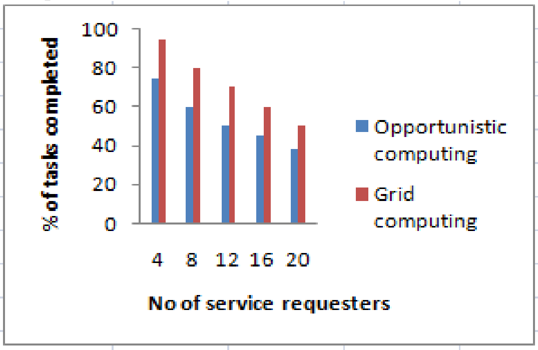

Figure 7: Job Completion Rate

6.1.4 Job Execution time

This determines the time taken by the service provider to service the request of the service requester. As the number of service requesters increases, the job execution time increases. This is because as the number of service requesters increases, more number of requests is generated. And hence it takes more time for the service provider to complete all the requests. The job execution time of both opportunistic computing and economic based grid computing models are considered. Job execution time in grid computing is more compared to opportunistic computing. This is because to reveal true utility values by each node during auction takes more time in grid computing compared to opportunistic computing. Figure 8 shows the results of comparison.

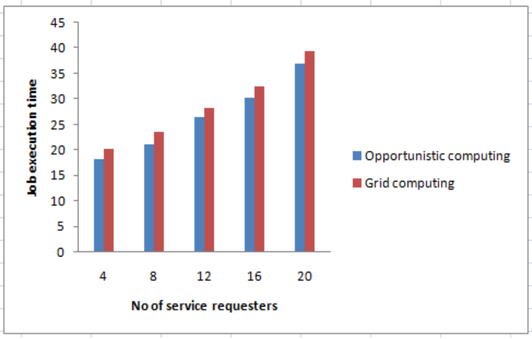

Figure 8: Job execution Rate

6.1.5 Average Routing Delay

This determines the delay involved in routing the control packets. As the number of service requesters increases, the routing delay also increases. This is because as the number of service requesters increases, more number of control packets are generated and hence the average routing delay also increases. The average routing delay of both opportunistic computing and economic based grid computing models are considered. Average routing delay in grid computing is more compared to opportunistic computing because average routing delay considers the delay in routing the control packets at the network layer. Since more number of control packets is involved in grid computing, the average routing delay for grid computing will be more compared to opportunistic computing. Figure 9 shows the results of comparison.

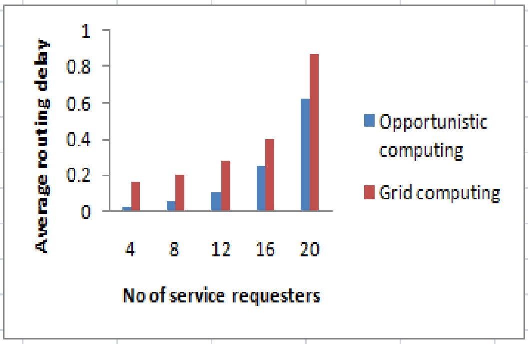

Figure 9: Average routing delay

6.1.6 Control Packets overhead

This determines the total number of control packets involved in routing. As the number of service requesters increases, the average number of control packets also increases. This is because as the number of service requesters' increase, more number of requests is generated and hence more number of packets is involved in routing. The number of control packets of both opportunistic computing and economic based grid computing models are considered. The average number of control packets are more in Economic based grid computing model compared to Opportunistic computing. This is because in Economic based grid computing model, the service requester can both submit their request to the service provider as well as get their responses from the service provider through Grid head node. Hence more number of control packets are involved in routing in Economic based grid computing compared to Opportunistic computing model. Figure 10 shows the results of comparison.

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

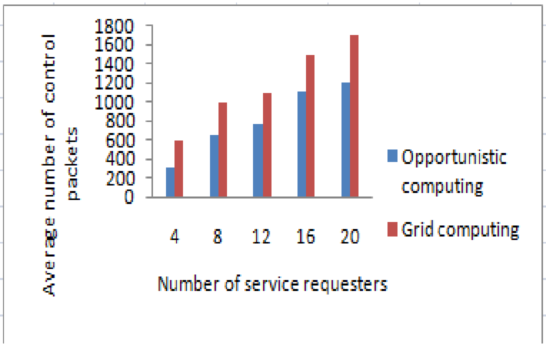

Figure 10: Control packets overhead

6.2 Comparative Evaluation Between Gossip Based Mechanism And Non-Gossip Based Mechanism

6.2.1 Number of requests in pending state

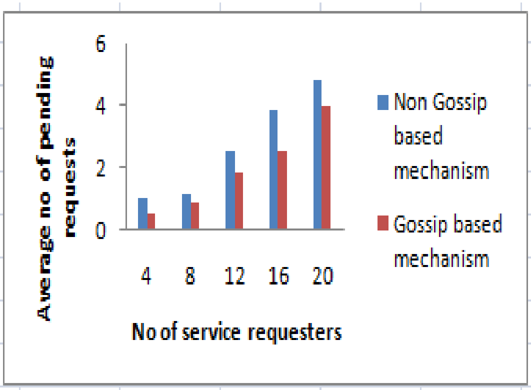

Figure 11: Average number of pending requests

This determines the number of requests waiting in the queue of the Grid head node because of the busy state of the grid service provider. The number of requests in the pending state of both Gossip based mechanism and Non-Gossip based mechanism are considered. The average numbers of requests in the pending state are slightly more in Non-Gossip based mechanism compared to Gossip based mechanism. This is because for an arriving request, if there are no enough service providers to service the request, then that request can be forwarded to the nearest grid head node. Hence by using this Gossip based mechanism, a substantial amount of pending requests can be minimized. Figure 11 shows the results of comparison.

6.2.2 Job Completion rate

This determines the total number of successfully completed requests. The job success rate of both Gossip based mechanism and Non-Gossip based mechanism are considered. There will be a slight increase in the job completion rate in gossip based mechanism compared to non-gossip based mechanism because the service provider availability may vary in different grids. Hence in Gossip mechanism, if suppose one grid doesn't have enough providers to execute the task, the corresponding task can be forwarded to the nearest grid head node to execute the task. Hence Job completion rate is more in gossip based mechanism compared to non-gossip based mechanism. Figure 12 shows the results of comparison.

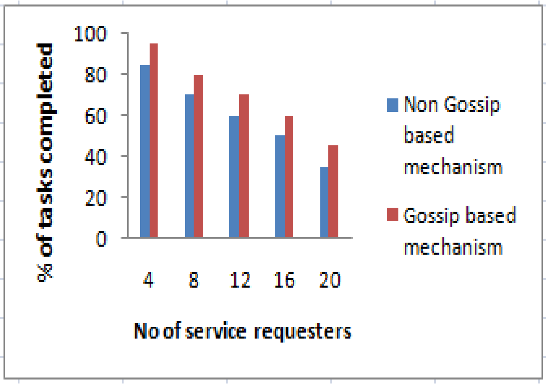

Figure 12: Job completion rate

6.2.3 Average Waiting Time

This determines the average time a service requester needs to wait in the Queue. The average waiting time of both Gossip-based mechanism and Non-Gossip based mechanism are considered. Average waiting time in Gossip based mechanism is less compared to Non-Gossip based mechanism because in Gossip based mechanism, the arriving request doesn't need to wait in the queue even if the service providers are busy, instead it can be forwarded to the next nearest grid head node. Hence an arriving request doesn't need to wait in the queue for a long time in Gossip based mechanism. Figure 13 shows the results of comparison.

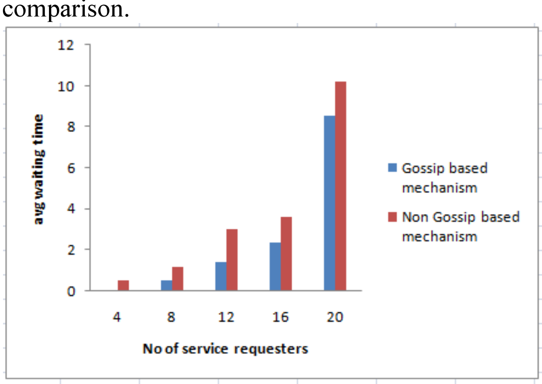

Figure 13: Average waiting time

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6.2.4 Job Execution time

This determines the average time taken by a service provider to service the request of a service requester. The job execution time of both Gossip based mechanism and Non-Gossip based mechanism are considered. Job execution time in Gossip based mechanism is less compared to Non-Gossip based mechanism because in Gossip based mechanism, for each arriving request it need check for the service provider availability. Hence it takes more time in gossip based mechanism compared to non-gossip based mechanism. Figure 14 shows the results of comparison.

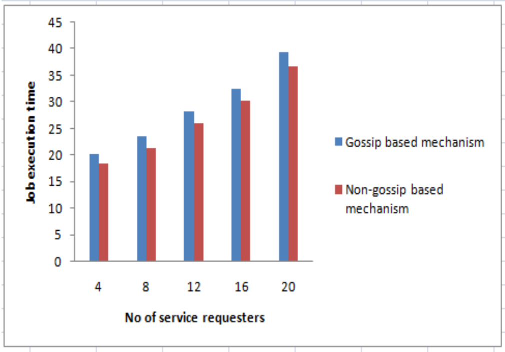

Figure 14: Job execution time

6.2.5 Control Packets overhead

This determines the average time a service requester needs to wait in the Queue. The average number of Control Packets of both Gossip based mechanism and Non-Gossip based mechanism are considered. Average number of Control Packets in Gossip based mechanism is more compared to Non-Gossip based mechanism because in Gossip based mechanism, for each arriving request it needs to perform a check on service provider availability and hence it involves a small increase in the number of control packets in gossip based mechanism compared to non gossip based mechanism. Figure 15 shows the results of comparison.

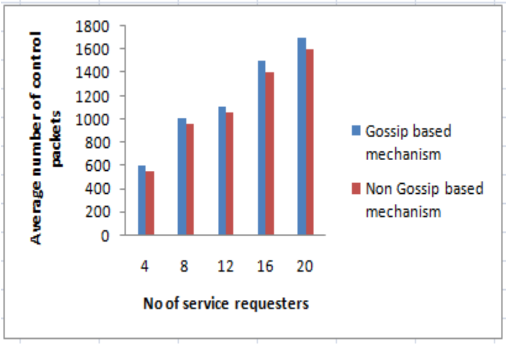

Figure 15: Control packets overhead

7 CONCLUSION

The resources in the Grid are geographically distributed and owned by multiple organizations with different usage and cost policies. They have a large number of self-interested entities with different objectives, priorities and goals that vary from time to time. The management of resources in such a large and distributed environment is a complex task. Hence an economy based grid computing model is considered for the management and regulation of supply-and-demand for resources. In this paper, an auction based scheduling method for assigning the jobs to sites in the Opportunistic network grid environment is considered. Results prove that our proposed economic based grid computing performs better with 20-30% improvement compared to the opportunistic computing method. Extending the resource negotiation and applying game theory for resource management [21] will be of our interest in future.

REFRENCES:

[1] Aburukba R. Ghenniwa & H. Weiming Shen , "Economic-based modelling for resource scheduling in grid computing", 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE, 2012.

[2] Leszek Lilien , Ajay Gupta , Zill-E-Huma Kamal & Zijiang Yang, "Opportunistic resource utilization networks—A new paradigm for specialized ad hoc networks", Elsevier Journal on Computers and Electrical Engineering, Volume 36 Issue 2, March, 2010

[3] Aminul Haque, Saadat M. Alhashmi & Rajendran Parthiban," A survey of economic models in grid computing", Elsevier journal on Future Generation Computer Systems, 2011

[4] Wang, Zhi, Bo Yu, Qi Chen & Chuanshan Gao. "Wireless grid computing over mobile ad-hoc networks with mobile agent." In Semantics, Knowledge and Grid, 2005. SKG'05. First International Conference on, pp. 113-113. IEEE, 2005.

[5] Libin Jhang & Jean Warland, "Stable and utility Maximizing Scheduling for Stochastic Processing Networks", IEEE Forty Seventh Annual Allerton Conference, September 30-October 2, 2010.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] Zhixiang, Ji, Yuan Rongchang, Li Lixin, Di Fangchun, Xie Tao, & Dai Jiao. "Design and Analysis Unified Resource Management Platform of Grid Dispatching System Based on Virtualization Technology." TELKOMNIKA Indonesian Journal of Electrical Engineering 12, no. 3 (2014).

[7] Sundaram, M. R. "DECISION MAKING FOR GRID RESOURCES DURING UNCERTAIN COMMUNICATION DEMANDS." Journal of Global Research in Computer Science 4, no. 4 (2013): 18-24.

[8] Panagiotis Demestichas, Nikos Koutsouris & Vera Stavroulaki, “Opportunistic Networks and Cognitive Management Systems in the Service of the Future Internet”, Proceedings of the third ACM international conference on Mobile Opportunistic Networks, 2012

[9] Umair Sadiq, Mohan Kumar, Andrea Passarella & Marco Conti, “Modeling and Simulation of Service Composition in Opportunistic Networks”, Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems,2011

[10] Ihsan, Imran, Muhammad Abdul Qadir, & Nadeem Iftikhar. "Mobile Ad-Hoc Service Grid-MASGRID." In WEC (5), pp. 124-127. 2005.

[11] Aksenti Granarov, Bekim Cilku, Igor Miskovski, Sonja Filiposka & Dimitar Trajanov, “Grid Computing Implementation in Ad Hoc Networks”, Advances in Computer and information Sciences and Engineering, Springer, 196-201, 2011.

[12] Michael J.Neely, “Delay Analysis for Max Weight Opportunistic Scheduling in Wireless Systems”, IEEE Transactions on Automatic Control, vol. 54, no . 9, September 2009 .

[13] Shengbin Liao & Wenquing Chen, “Cross Layer optimization Based on Utility Maximization with Delay Constraints in Ad Hoc Wireless Networks”, ISECS International Colloquium on Computing,Communication,Control and Management, IEEE 2009.

[14] Tarik Taleb, Senior Member, IEEE, Nidal Nasser, Member, IEEE, & Markos P. Anastasopoulos, Member, IEEE, “An Auction-Based Pareto-Optimal Strategy for Dynamic and Fair Allotment of Resources in Wireless Mobile Networks”, IEEE Transactions On Vehicular Technology, VOL. 60, NO. 9, November 2011.

[15] Howie H. Huang, Woochul Kang & Andrew Grimshaw, “Achieving high job execution reliability using underutilized resources in a computational economy”, Elsevier journal on Future Generation Computer Systems, 2013.

[16] Hamscher. V, U. Schwiegelshohn, A. Streit, & R. Yahyapour, “Evaluation of job scheduling strategies for grid computing,” Workshop Grid 2000 at 7th International Conference on High Performance Computing , India, pp. 191–202, 2000.

[17] Sathiyaseelan, T., & B. Anantharaj. "Resource Allocation and Management in Cloud Based Networks." The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol. 1, No. 2, May-June 2013

[18] Zhao Lan & Wang Dazhen, “A Resource Mapping Method in Grids Based on Multi-Unit Auction Mechanism”, International Conference on Grid and Cooperative Computing, DECEMBER, 2009.

[19] Preetam Ghosh, Nirmalya Roy & Sajal K Das, “Mobility-Aware Efficient Job Scheduling in Mobile Grids”, Seventh IEEE International Symposium on Cluster Computing and the Grid, 2011.

[20] C. Curescu and S. Nadjm-Tehrani. Price/utility-based Optimized Resource Allocation in Wireless Ad-hoc Networks. IEEE SECON 2005.

[21] Harish, M., Anandavelu, N., Anbalagan, N., Mahalakshmi, G.S., Geetha, T.V.: Design and analysis of a game theoretic model for P2P trust management. In: Janowski, T., Mohanty, H. (eds.) ICDCIT 2007. LNCS, vol. 4882, pp. 110–115. Springer, Heidelberg, 2007.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Comparison Table For The Existing Methods

Title	Algorithm	Concept	Issues
Modeling and Simulation of Service Composition in Opportunistic Networks [9]	Service Composition algorithm	Mobile users can benefit from a larger set of services available locally in an environment. Makes efficient service selections from devices that are in close proximity	Considered devices which are at one-hop only which results in only less number of services to be composed
Achieving high job execution reliability using underutilized resources in a computational economy [15]	Replication in space Algorithm	HA-JES fosters the balanced resource consumption by dynamically and transparently virtualizing underlying underutilized, and often under-priced, resources to satisfy QoS requirements from users.	Doesn't include other factors of QOS assurance mechanism like performance, security, etc.
Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm	Community-aware scheduling algorithm	The CASA is a two-phase solution comprised of an integrated collection of sub-algorithms used to facilitate job scheduling across decentralized distributed nodes	Does not take into account different grid work loads
A survey of economic models in grid computing [3]	-	Different economic models have been proposed over time for the grid. Conducted an extensive survey on these models and presented their strengths and weaknesses in different scenarios as identified by grid researchers. Different models are observed which are suitable for different scenarios and provided a comparison of their performance under these scenarios	-
A Resource Mapping Method in Grids Based on Multi-Unit Auction Mechanism [18]	A Grid Resource Mapping Algorithm Based on Multi-unit Auction	It handles prices announcement, agent creation and resource situation submission, bidding rules, and temporary allocation rules.	More communication overhead
Mobility-Aware Efficient Job Scheduling in Mobile Grids [19]	Cost optimal job allocation algorithm	Proposed the job allocation problem in mobile grid systems considering single class grid jobs, communication delay and node mobility.	Run-time statistics of the nodes are inaccurate, more delay

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Grid resource table

Node ID	Available Resources	Resources Price	Service Delay	Node Position	Stability Time	Bid Value	Job Deadline

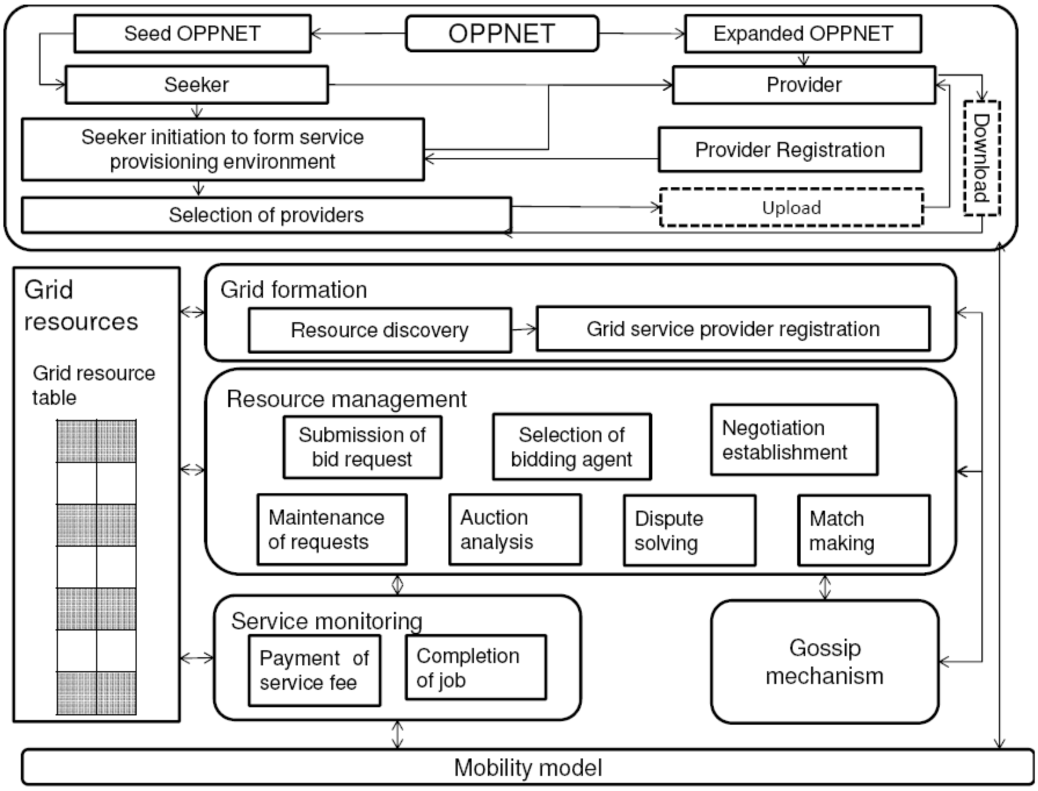

Figure 1: Block Diagram of Second Price Auctioning

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

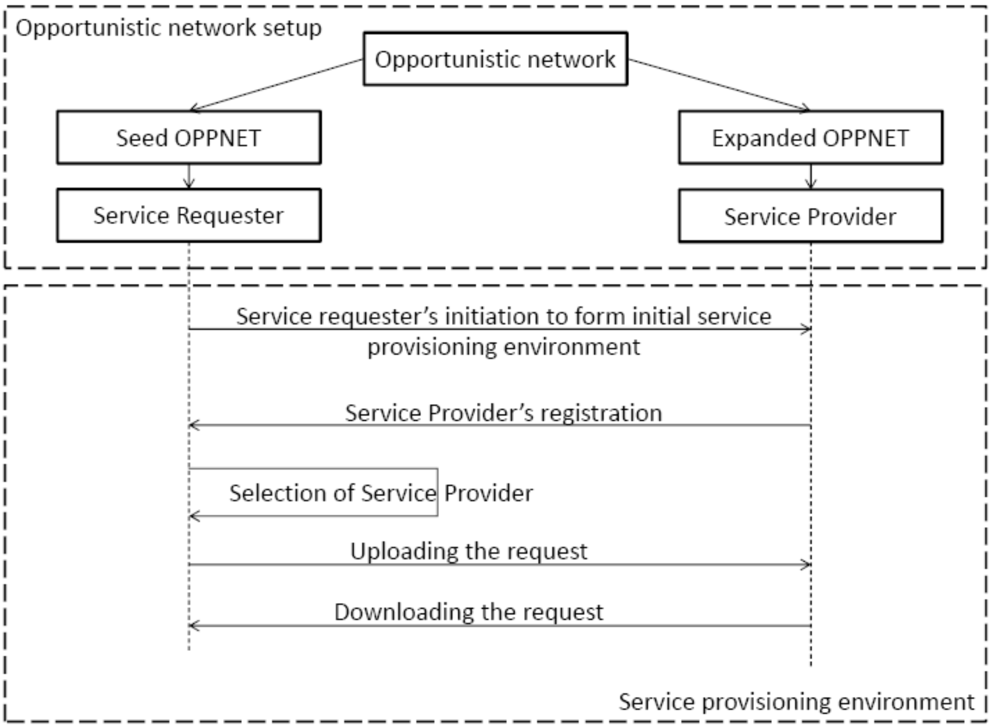

Figure 2: Opportunistic Computing

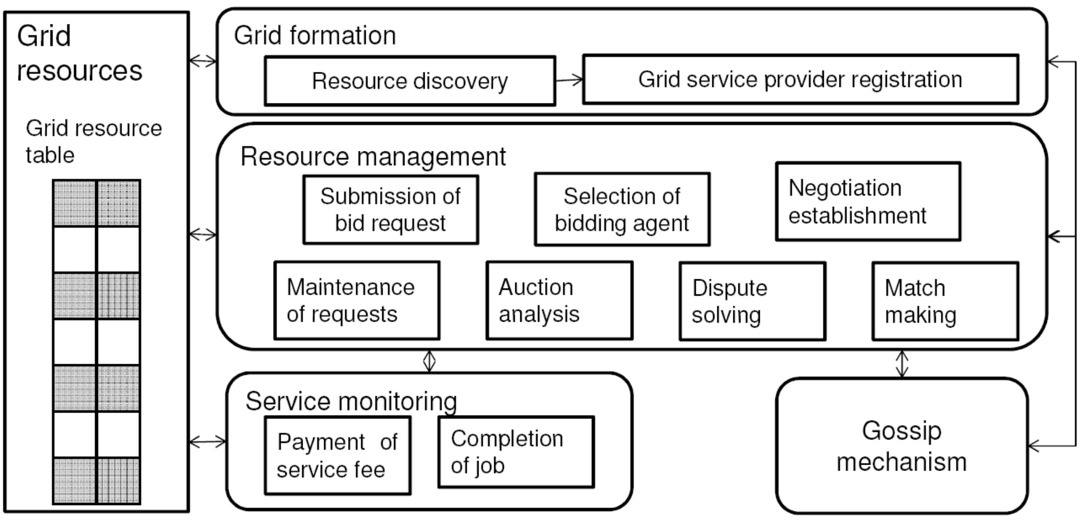

Figure 3: Block Diagram for Auction based Grid Computing