Volume 3, Issue 12, December 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Different Approaches for Resource Allocation in Grid Environment

Charu Sharma, Tanu Gupta
Department of Computer Science & Engineering,
Punjab Technical University, Punjab, India

Abstract: ***Grids are systems that involve resource sharing and problem solving in heterogeneous dynamic grid environments. A distributed system is composed of such computers which are separately located and connected with each other through a network. Resource allocation is a problem that arises whenever different resources, which have preferences in participating, have to be distributed among multiple autonomous entities.Since it involves using resources from different places, from different ownerships and with different individual qualities, it requires a complex resource management process for its proper functioning This paper provides a brief overview on grid computing and different approaches for resource allocation algorithm for resource allocation in grid.***

Keywords -- ***Grid computing, Resource allocation, Static allocation, Dynamic allocation, Auction based allocation.***

I. Introduction

Grid computing is concerned with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organization. Grid computing is an issue resolving environment for utilizing the unused resources and maximizing the resource capability, also it is an innovative approach that leverages existing IT infrastructure to optimize compute resources and manage data and computing workloads. Grid computing is next step to distributed computing or networking. Grid is a network of geographically distributed resources including computers, peripherals, switches, instruments, and data. Grid computing is a term referring to the combination of computer resources from multiple administrative domains to reach a common goal. It enables sharing of resources. Grid computing is an integrated computer network linking large geographically distributed and heterogeneous computer systems and resources, which eliminates the need for dedicated servers for job computations but uses distributive resources collectively to enhance computational power. Resource allocation is one of the important system services that have to be available to achieve the objectives from a grid computing. A common problem arising in grid computing is to select the most efficient resource to run a particular program. In this case, the resources are scattered over a wide span of area belonging to different individuals or companies, which are used to solve different segments of one single problem whose solution is gotten only after combining all the individual solutions of the segments computed by the different allocated resources. Also users are require to reserve in advance the resources needed to run their programs on the grid. The execution time of a program can be estimated by using aspects of static analysis, analytical bench marking and compiler based approach. In this paper we review the algorithm of static allocation, Dynamic allocation and auction based algorithm.

II. Related Work

The resource allocation process in grid environments consists of three main phases; the first phase is resources discovery, the most important issues when dealing with resource discovery is how to publish the resource information by providers and how those resources can be discovered by grid clients *[1-2]*. This information is handled and supported by a resource registry *[1]*. The second phase is resource selection which regarding with the process of selecting the best resources to execute a certain task *[3]*. The last phase of resource allocation process is resource usage which is concerning with running the task on the selected resources and monitoring the execution *[1]*.
The term resource management in grid computing can be defined as those operations that control the way that grid resources and services are made available for use by entities like users, applications and services *[5]* to ensure efficient utilization of computer resources and for optimization performance of specific tasks. Due to the complexity, heterogeneity and dynamic nature of grid computing environments, resource management is faced with challenges making it a complex task to match the capabilities of available resources to the needs of the entities listed above *[5-6]*.
Krauter et al *[4]* presented a taxonomy of grid resource management systems; this taxonomy mainly focused on categorizing grid systems and the whole process of resource management in grid. However, this study does not focus on resource allocation in details.
Darshan Kanzariya, Sanjay Patel *[7]* presented the various grid resource allocations strategies to select the most efficient resource to run a particular program.

III. Static Based

A static based resource allocation constitutes fixed accounting scheme such as a fixed access to a computer node. Based on this approach, main objective is to assign applications processes to compute servers that can present the required Quality of Service as well as execute the processes in a cost-efficient method. They presented a protocol of identifying compute servers that can execute the application with minimal cost as well as provide the required Quality of Service for the application. Each process has specific resource requirements, such as memory, network Band width, and number of processors and so on. The discovery of resources and the assignment process are being modeled as tree and the execution of a process took place through a search of a solution tree. The authors also came up with a protocol that allocated processes to the computing servers. In a related development, by Somasundaram and Radhakrishnan, Incoming jobs from different users are collected and stored in job list and available resources are stored in resource list. According to Swift Scheduler, jobs allocations as well as resource selection process are executed using heuristic searching algorithm on Shortest Job First, which minimizes the average time waiting of jobs. Therefore, the turnaround time is minimized and resource utilization is greater than before. They carry out the Swift Scheduler test in GridSim through various number of jobs, number of resources against total processing time, resource utilization and cost. To be noted, for resource allocation to be more efficient using this method to achieve a high throughput and high resource utilization, they should consider to hybrid the Swift Scheduler with an evolutionary algorithm such as Particle Swarm Optimization technique (PSO) which is a computational method that optimizes a problem by iteratively trying to enhance a candidate solution with respect to a given measure of high quality. PSO optimizes a problem by having a population of prospect solutions, here named particles and moving these particles around at the search-space based on simple mathematical formulae over the particle’s position and velocity. Each particle's mobility is inspired by its local best known position and is also guided toward the best identified positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm when it comes to the best solutions.[8][9].

IV. Dynamic Based

A dynamic based resource allocation is a process whereby dynamic mechanisms adapt their participation conditions according to the change of available resource quantities. Based on this model, the method here has been used by a combining best fit algorithm and process migration. According to this approach, a resource reservation is decided by an administration based on monitoring outcome specified by the system at a given time. In addition, applications requirements may dynamically be transformed at runtime. Moreover, they presume a global grid network *[10]* where resources are distributed all over the globe. Users put forward applications to their local area network scheduler. The scheduler afterward allocates resources to each application taking into consideration the application's Service Level Agreement without an administrator intrusion. The scheduler select resources related to the application requirements and allocate them to the requesting application. The resource manager links a separate thread for each registered grid application. While a resource observer daemon runs on each host to gather information regarding resources as well as to send them to the recorder database.

A proposed a co-allocation architecture*[11]* which creates flexible set of basic co-allocation mechanisms that can be used to build a wide range Operating System (OS) application specific co-allocation tactics, which addresses the challenges of Grid environments. Whereas providing an application with a single concept for monitoring and controlling those resources. These mechanisms permit a dynamic construction, configuration and have power over the set of independently administered and controlled resources. Furthermore, they inspect the co-allocation problem, as well presented a mechanisms with the aim of allowing every single application to channel the resource‟ selection throughout the period of co-allocation process. Therefore these mechanisms address all the issues relating to allocation, control, and configuration and monitoring of distributed computations. The authors implement a construction of two co-allocators: transaction co-allocator Globus Resource Allocation Broker (GRAB) and an interactive transaction co-allocator-Dynamically Updated Resource Online Co-allocator (DUROC). Both GRAB and DUROC are implemented as a collection of libraries which is designed to be connected with application codes. Co-allocation requests are expressed in terms of an extensible resource allocation language, or RSL. The Globus toolkit formed a central part of DUROC and has seen wide use both directly within tools such as MPICH-G in applications. Despite the fact that the co-allocation strategies is being offered, they do not address the problem ensuring a given co-allocation request will succeed or not. It is to be noted that advance reservations co-allocation with the use of some mechanisms, (e.g., computers and bandwidth) managed by various existing schedulers using Plus, which provides advance reservation capability coordinating with a few queuing systems, such as TORQUE (provides control over batch jobs and distributed computing resources) and Sun Grid Engine (manages and schedules the allocation of distributed resources for instance processors, memory, disk space). That will enables the construction of application-level of co-reservation and support a dynamic discovery and independently controlled and administered to be sure that the co-allocation will succeed. Main goal is to allow an integrated management of the communication Infrastructure which is being required for the grid operation. Based on this architecture the network administrators, in each administrative domain that composes the grid, define translation rules in order to control how to create the network policies based on the grid requirements. Furthermore, they implemented a Web based prototype, in order to support the proposed architecture, where the grid administrator is allowed to specify the grid policies and the network administrators in each domain can specify the translation rules. The presented solution characterize a translation mechanism where network policies are being created by translation rules being use as input data information retrieved from the grid policies. Moreover, the network policies being generated through the translation mechanism which are translated to the network configuration actions executed through the Policy Decision Points *[12]* of a regular policy-based management system. Each and every domain possess network administrator that describe local translation rules, which are different compared to the rules distinct by other administrator, a single grid policy is translated to different network policies in each administrative domain. Even though the translation rules are flexible, this forces the network administrators to gain knowledge of a new language to define more adequate translations. It is to be noted, that the bandwidth consumption investigation is required, although performance and scalability observations of the translation

engine seems to be more critical, primarily concerning the number of grid rules, levels of rule nesting and the number of translation rules defined by the network administrator.
A study *[13]* show method analytical hierarchy process is been introduced to resolve resource allocation problem. There are two techniques being used in this area called static and dynamic methods. For static method, the information of tasks and performing platform thought to be fixed and defined in advance. Here, as soon as a task is sent to grid, the scheduler must decide on what resource the task may possibly be applied to. This decision advancement is a Multi-Criteria Decision Making (MCDM) problem. There are various forms of clarification for this area of problems. Analytic Hierarchy Process (AHP) is one of the popular and extensively used techniques in MCDM problems. Complex problems are being converted to simple ones by assessing their mutual effects using these techniques. Based on ARA, users put forward their tasks to grid scheduler then the scheduler name an MCDM problem by using the task properties and dynamic resource characteristics. To choose a resource, the schedulers depart through AHP and at last send the ID of chosen one to the user. After that, the scheduler recalls resource information from Grid Information System (GIS) to produce the general preference matrices of resources for each resource characteristic. The scheduler in represented technique gets the resource information from GIS and uses it to appropriately share out the system load between grid resources. They use GridSim as there simulator to assess the performance of the technique. The results proved that their method improves the value of task successful execution rate as well as reduces the mean waiting time of tasks.

V. Auction Based

The 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). The auctioneer sets the rules of auction, acceptable for the consumers and the providers. Auctions basically use market forces to negotiate a clearing price for the service.
The main participants in the auction model are: User Brokers (UB), Grid Service Providers (GSP) and Local Markets for Auctions (LMA). In the following we present each of these participants and describe their role in the model and their characteristics.

A. User broker (UB):

Each grid user has a User Broker. The User Broker is responsible for auction (resource) discovery, auction analysis and selection, bid submission, sending user jobs to resources, collecting the results and providing the user with a uniform view of grid resources. There are four components of the user broker:

1) ***Job Management Agent***: It is responsible for user interaction, job creation, submission and monitorization. It also coordinates the mechanism analysis and selection, resource discovery and the bidding process. When the jobs complete it collects the results of the computation
2) ***Resource (Auction) Discovery Agent***: It is responsible for resource/auction discovery. It sends a request for resources/auctions to the Local Market for Auctions. The Local Market for Auctions sends back the information on the auctions that match the request.
3) ***Auction Analysis and Selection Agent****:* It is responsible for analyzing the auction information submitted by the Local Market for Auctions. Based on the user requirements and on the properties of the auctions it selects an auction in which the user will participate.
4) ***Bidding Agent****:* It is responsible for choosing and submitting the bid to the selected GSP auctioneer agent or to the selected External Auctioneer (EA). If it is a successful bid the Job Control agent sends the user jobs for execution to the corresponding GSP.

B. Grid Service Providers (GSP): GSPs contribute their resources to the Grid and charge the users for services. Depending on the type of auction chosen by the GSP we have two scenarios. The first scenario is when a GSP decides to participate in a one-sided auction protocol (e.g. First Price, Vickrey). In this case GSPs create auctions mechanisms that are posted on the Local Market for Auctions. Different GSPs can deploy different auction types. The Auctioneer Agent is responsible for posting the GSP"s auction mechanism on the LMA. It also runs the auction allocation mechanism, collects the bids from users and determines the winning users. The winning users are the users that send jobs for execution. Once the winning users are determined it informs the users of the result of the auction by sending success or reject messages. It also coordinates the admission control and resource scheduling.
C. Local Market for Auctions (LMA): It provides support for GSPs to deploy auction mechanisms, and enables the users to find the right auctions that match their requirements and preferences. It also provides a set of External Auctioneers (EA) which will be responsible for running two-sided auctions (e.g. double auction). LMA takes a request from a user specified in an appropriate language and returns the auctions that match the request. It also accepts ask prices from GSPs in case some of them decided to participate in two-sided auctions.

VI. Classification of Auction Based Allocation

Auction based allocation can be classified into five types:
A. English Auction (first-price open cry)
B. First-price sealed-bid Auction
C. Vickrey (second-price sealed-bid) Auction
D. Dutch Auction
E. Double Auction (continuous)

A. English Auction (first-price open cry):

All bidders are free to increase their bids exceeding other offers. When none of the bidders are willing to raise the price anymore, the auction ends, and the highest bidder wins the item at the price of his bid. In this model, the key issue is how GRBs decide how much to bid. Those not interested in bidding anymore can openly declare so (open-exit) without the possibility of re-entry.

B. First-price sealed-bid Auction:

Each bidder submits one bid without knowing the others' bids. The highest bidder wins the item at then price of his bid. In this case a broker bid strategy is a function of the private value and the prior beliefs of other bidders' valuations. The best strategy is to bid less than the true valuation and one might still win the bid, but it all depends on what the others bid.

C. Vickrey Auction Protocol [14] :

The auction considered as the basis for this protocol is the Vickrey auction, also called the second-price auction. In this type of auction bidders don't know the bid values of other bidders. The highest bidder wins and pays the price equal to the second highest bid. This protocol is executed by GSP.After GSP has posted the auction description on LMA the users decided to participate in auction at GSP.

D. Dutch Auction:

It is similar to a first-price sealed-bid auction because in both cases the bid matters only if it is the highest, and no Relevant information is revealed during the auction process. From the broker's bidding strategic point of view, a Dutch auction is similar to an English auction (first-price sealed-bid auction). The key difference between them is that in English auction the bids start with a low opening and increase progressively until demand falls, whereas in a Dutch auction the bids start with a high opening price and decrease progressively until demand rises to match supply.

E. Double Auction Protocol [15]

The auction considered as the basis for this protocol is the double auction [18]. In this type of auction the users submit bids and GSPs submit asks to an External Auctioneer. The equilibrium price is determined by matching asks (starting from the lowest price to the highest) with demand bids (starting from the highest price to the lowest). This protocol is executed by EA which is part of LMA. Once a set of GSPs decided to participate in a double auction, EA posts the auction description on LMA. We assume that only GSPs having resources of the same type participate in one double auction protocol.

1) Two auction based auction algorithm[17]

The basic design of both the Bellagio and Mirage systems is that of a repeated, sealed-bid combinatorial auction1, which allocates resources to competing users over time. Given that the winner-determination problem in a combinatorial auction is known to be NP-hard , we adopt a greedy algorithm to ensure that the auctions clear quickly irrespective of the number of size of user bids. For this purpose, we adopt a simple heuristic that greedily allocates (also called a "first-fit" schedule) bids in order of a bids value density, which is the bid value divided by its size and length. We repeat these allocation algorithm k times, with a different ordering of the top k bid value densities, and select the best allocation among these k attempts. In this setting, the auction is run periodically. Once bids are received the auction must determine winning bids and the associated resource allocation. A user in our systems submits bids to the auction using a two-phase process (Figure 1). First, she adopts a resource discovery service to find candidate resources that meet her needs. Second, using the concrete resources identified from the first step, such a user can place bids using a system-specific bidding language.

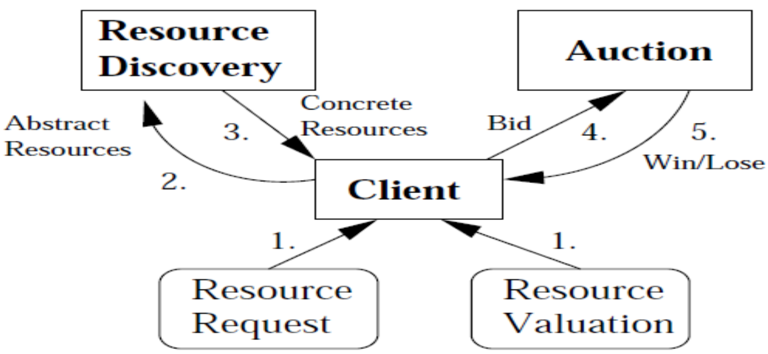

Fig 1 Bidding and Acquiring Resource

2) Greedy Double auction algorithm[16]

A greedy double auction mechanism (GDAM) is proposed which does not focus on how to maximize the total market value but how to improve the resources utilization and benefit the majority of the market participants

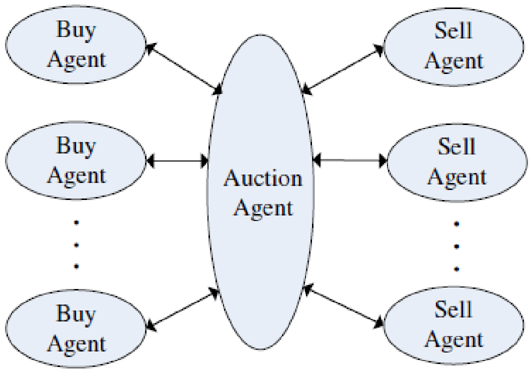

Fig 2. Greedy Auction Algorithm

Under GDAM, the auction market provides trading service for the participants, using trading price and trading amount as two QoS parameters. As for a seller *j* he requires the selling service with *sell request 1* and *sell request 2* as his QoS requirements; as for a buyer *1* he requires the buying service with *broker request 1* and broker request 2 as his QoS requirements. A seller's asking price is his expected profit plus the cost of providing per unit of resource, while a buyer gets his bidding price by subtracting the cost from the value created by consuming per unit of the resource. By doing this, the profits of the successful participants can be guaranteed. With trading at the expected price, no agents will complain about the unfairness. And our mechanism works in three different cases.

a) ***Supply on Demand***: In this case, there are more supply quantities than demand quantities available in the auction market.
b) ***Demand over Supply****:* In this case, there are more demand quantities than supply quantities available in the auction market.
c) ***Supply equals Demand:*** In this case, the overall supply quantities are equivalent or nearly equivalent to the overall demand quantities and holds.

VII. Conclusion

In this paper grid computing and resource allocation have been studied. Due to the extremely heterogeneous and complex computing environments, availability of resources may possibly fluctuate in Grid environments. Our main objective is to review the various grid resource allocations strategies which will in turn serve as a guide for researchers. In this review we have find different approach about resource allocation. The first price and Second price protocol is limited for small number of agent and resource. Where the auction based allocation is efficient for large number auctioneer and the broker.

References

[1] Schnizler, B., *Resource Allocation in the Grid: A Market Engineering Approach*. 2007: Univ.-Verl. Karlsruhe.
[2] Elmroth, E. and J. Tordsson, *Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions. Future Generation Computer Systems,* 2008. 24(6): p. 585-593.
[3] Malarvizhi, N. and V.R. Uthariaraj. *A Broker-Based Approach to Resource Discovery and Selection in Grid Environments. in Computer and Electrical Engineering*, 2008. ICCEE 2008. International Conference on. 2008.
[4] Krauter, K., R. Buyya, and M. Maheswaran, *A taxonomy and survey of grid resource management systems for distributed computing*. Software: Practice and Experience, 2002. 32(2): p. 135-164.
[5] Khanli, L. M., and Analoui, M. (2008),"*An approach to grid resource selection and fault management based on ECA rules",*Future Generation Computer Systems, 24(4), 296-316.
[6] Yanbing Liu, and Yi Jian. (2006),"*Performance evaluation and modeling in grid resource management*", Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference, 335-338.
[7] Darshan Kanzariya, Sanjay Patel,"*Resource allocation in grid computing*",IJEIT, Volume 2, Issue 8, February 2013**.**
[8] Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.Tibor, G., 2006. "A Resource allocation protocol for providing quality of service in grid computing, using a policy-based approach." AICT-ICIW '06 Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services.
[9] Haruna Ahmed Abba, Nordin B. Zakaria and Nazleeni Haron Grid Resource Allocation: A Review" Research Journal of Information Technology 4(2): 38-55, 2012.
[10] Berman, G.F. Fran and J.G.H. Anthony, 2003b." Grid Computing: Making the Global Infrastructure a Reality. Wiley Series in Communications, Networking and Distributed System,".
[11] Stefan Krawczyk and Kris Bubendorfer "Grid Resource Allocation: Allocation Mechanisms and Utilisation Patterns".

[12] B.Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry and S.Waldbusser, 2001. "Terminology for policy based Management". RFC Editor, United States.

[13] Rouhollah, G. and S.S. Hadi, 2010. "A Grid Resource Allocation Method Based on Analytic Hierarchy Process. 5th International Symposium on Telecommunications (IST'2010)."

[14] W. Vickrey. "*Counter speculation, auctions, and competitive sealed tenders*". Journal of Finance, 16(1):8–37, March 1961.

[15] R. P. McAfee. "*A dominant strategy double auction. Journal of Economic Theory*," 56:434–450, 1992.

[16] Ding Ding, Siwei Luo, and Zhan Gao, "*A Greedy Double Auction Mechanism for Grid Resource Allocation*".

[17] AuYoung, Alvin, Phil Buonadonna, Brent N. Chun, Chaki Ng, David C. Parkes, Jeff Shneidman, Alex C. Snoeren, and Amin Vahdat. "*Two Auction-Based Resource Allocation Environments: Design and Experience. Market Oriented rid and Utility Computing*".

[18] Berman, G.F. Fran and J.G.H. Anthony, 2003." *Grid Computing: Making the Global Infrastructure a Reality. Wiley Series in Communications, Networking and Distributed System*,".

