Research Journal of Applied Sciences 7 (2): 119-125, 2012
ISSN: 1815-932X

# Fuzzy Based Optimal Allocation of Resources for Grid Scheduling

G. Murugesan and C. Chellappan
Anna University, Chennai, Tamil Nadu, India

**Abstract:** Grid computing is becoming a popular way of providing high performance computing for many process intensive, scientific and business applications. Grid computing consists of large sets of diverse, geographically distributed resources that are collected into a virtual computer for high performance computation. Resource allocation in grid computing is the challenging task due the heterogeneity and dynamic nature of resources. Many of the researches have been done to develop a better resource allocation model. But in most of the work they follow either the time as decision parameter or the cost as decision parameter. A very few work had been focused for both the parameter also they are trying to improve the grid resource performance. In this study, researchers developed a mathematical model as an economic model with the aim to benefit the grid user. The Resource Allocation Model in grid computing consists of two components; resource selection and resource sequencing. Researchers used the fuzzy logic technique to perform the resource selection and resource sequencing. The performance of the mathematical model is evaluated with an extensive set of input values to demonstrate the practical use of the model. The result shows that the approach outperform other models by minimizing the time and cost of the application of the grid user.

**Key words:** Resource Allocation Model, grid scheduling, fuzzy logic, economic model, divisible load theory

## INTRODUCTION

Grid computing is already a mainstream paradigm for resource-intensive scientific applications but it also becomes the useful model for enterprise applications. The grid enables resource sharing and dynamic allocation of computational resources thus increasing access to distributed data, promoting operational flexibility and collaboration and allowing service providers to scale efficiently to meet variable demands. Grid computing requires an effective allocation for the better utilization of the dynamic resources. The execution of user processes must simultaneously satisfy both job execution constraints and system usage policies. Although, many scheduling techniques for various computing system exist, traditional scheduling systems are inappropriate for scheduling tasks into grid resources.

Grid computing consists of large sets of diverse, geographically distributed resources that are collected into a virtual computer for high performance computation. Thus, it is a heterogeneous computing platform that contains different network link and processor capacities. Furthermore, the grid system is not a dedicated system which leads to load variations in links and processors. Due to dynamic nature of system, this introduces many difficulties including resource allocation problem. The grid system contains two actors; the resource provider and the resource user. The resource users are the grid users they have applications to execute in the grid environment and the resource providers are the grid owners willing to offer their resources or computing services to get regular payment from the grid users.

There are two formal approaches for resource allocation in grid computing environments; performance based and economic based. The first approach is based on the system response parameters such as bandwidth usage, memory usage and the other parameters like computing speed and so on. However due to the dynamic and heterogeneous nature of resources developing a performance based model become intangible. Lot of work has been proposed for improving system performance based on divisible load theory. The second economic based approach is attractive for the concept of efficiency or cost effectiveness as a valuable asset. It is necessary to consider the scheduling problem in the grid computing system as an economic model.

One reason is that for a computer utility that charges customers for access to distributed and networked computing resources. The grid user always interested to utilize the cheapest processor to compute their loads at the same time the load has to be processed within the deadline assigned for each load. Under the network environment because the resources provider and the resource user are different in the target, strategy and the supply-demand mode, particularly in the geography distribute characteristic, the resources management style

**Corresponding Author:** G. Murugesan, Anna University, Chennai, Tamil Nadu, India

of traditional concentrated control is no longer valid. But the present Network Resource Management Scheduling System adopts the conservative of strategy: the parts responsible for scheduling decide the operations how to be assigned to the processor according to the cost function driven by system core parameter. The strategy doesn't consider the resource access. Under the network environment, resource scheduling can be considered as an optimization problem under limited conditions. In this study, researchers present an economical model for allocating resources effectively in grid environment with respect to the user quality of service requirements such as deadline and budget constraints and analyze the performance of the algorithm for its effectiveness.

Most of the current researches proposed on economic approach are based on different type of Auction Theory or Tender/Contract-Net Model. However, the Auction Theory based approaches are not feasible in the development of an enterprise wide grid. For example Nimrod-G, a resource broker developed by Buyya using three optimizing criteria such as cost minimization, time minimization and conservative cost and time minimization. But this model fails to focus on an explicit objective also, it is an *Ad-hoc* Method, only concern with deadline and budget and it does not provide cost effectiveness in either way. The other one in an economic approach is TeraGrid which is dedicated for research purposes and based on deadline of the grid application, the availability of grid resources and the description of the resources required to execute the grid application. Because TeraGrid and the most of the other economic approach models are not suitable for profit seeking firms, the cost of the running jobs is not taken into account to meet the optimizing criteria. Instead, all the proposed solutions are trying to match the application with the available resources with the objective of minimizing the makespan or maximizing the overall system performances.

## RELATED WORKS

As economic models are introduced into grid computing, new research opportunities arise. Because the economic cost and profit are considered by grid users and resource providers, respectively new objective functions and scheduling algorithms optimizing them are proposed. The economic problem exists only when the resources and the participants, namely consumers and producers, maximize their utility by deciding among needs that cannot concurrently satisfied and lead to different utility levels (Nakai, 2002). This study is based on the previous research by Murugesan and Chellappan (2011), there researchers are not considered the resource sequencing mechanism. There are various economic models for Grid scheduling has been proposed with different techniques and objectives such as Commodity Market Model, Bargaining Model, Tender/Contract-Net Model, Auction Model, Community Model, Monopoly/Oligopoly and so on proposed by Buyya *et al.* (2002). In (Takefusa *et al*., 2001; Viswanathan *et al*., 2007), a deadline scheduling algorithm that supports load correction and fallback mechanisms to improve the algorithm's performance in an environment that is based on Client-Server Model is proposed. It submits each job to a resource that can finish it in time less than or equal than the time specified for completing the job.

The aim of this research is to decrease the number of jobs that don't meet their deadlines. The resources are priced according to their performance. It uses bricks simulator which is written in Java after extending it to assess the performance of the implemented deadline algorithm. In several algorithms called Deadline and Budget Constrained (DBC) scheduling algorithms are presented which consider the cost and makespan of a job simultaneously. These algorithms implement different strategies. For example, guarantee the deadline and minimize the cost or guarantee the budget and minimize the completion time. The difficulties to optimize these two parameters in an algorithm lie in the fact that the units for cost and time are different and these two goals usually have conflicts (for example, high performance resources are usually expensive). Researchers can also find such examples (Venugopal and Buyya, 2005). The most straight forward method of an early grid and matching price model is the Sun Grid. The Sun Grid offers a pool of pre-installed applications as well as options for users to deploy their own applications on Sun's resources. The Pricing Model for this service is straight forward: $1/CPU-hour. The effective workload allocation model with single source has been proposed (Ger and De, 2008) for data grid system (Abdullah *et al.*, 2009). The time and cost trade-off has been proposed (Garg *et al.*, 2009) with two meta-scheduling heuristics algorithms that minimize and manage the execution cost and time of user applications. Also they have presented a cost metric to manage the trade-off between the execution cost and time.

## THE PROBLEM AND MODEL DESCRIPTION

In a credit card company every month they need to process around 30 million accounts and they have 100 processing elements to process its customer accounts so, each of the processing elements has to process minimum of 300,000 records. So if they divide the entire 30 million records into equal parts then to complete the process take long time due to the variations of computation speed and communication link speed. So, dividing the job into equal is not advisable for this type of applications. Divisible load theory provides the mathematical machinery to perform time optimal processing. In this problem the number of instruction to be processed by each PE is constant because the process for each customer becomes same. Divisible load theory is a methodology involving the linear and continuous modeling of partitionable computation and communication loads for parallel processing. It adequately represents an important class of problems with applications in parallel and distributed system scheduling, various types of data processing, scientific and engineering computation and sensor networks.

The main focus is to find out the number of records to be processed by each of the PEs in the Grid System. Whenever they want to process the customers' record, submit to the Grid System. So, they need to spend lot of money to process the records. Here researchers have developed a resource allocation mechanism for the benefit of the credit card company to minimize their usage cost. The model takes the input from the company as the total number of records to be processed, the completion time of the entire records and the budget allotted to the process. The grid system check the available PEs and select the required number of PEs. Once the PEs is selected, they have to be arranged in a way such that which PE get workload portion first and next and so on. In this research we propose a fuzzy logic based mechanism to sequences the PEs instead of arbitrarily choosing the resources for job allocation.

The resource allocation process consists of two components; resource selection and resource sequencing. Resource sequencing plays a vital role in resource allocation due to heterogeneity of resources in the Grid System. Researchers used the Fuzzy Logic (FL) Technique to sequence the resource. FL was conceived as a better method for sorting and handling data but has proven to be a excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning. To use the fuzzy technique there is a need to define the membership function. The membership function gives the degree of membership lies between 0 and 1. We used the gamma function as a membership function to get the degree of membership from the crisp set. The gamma function is defined as follows:

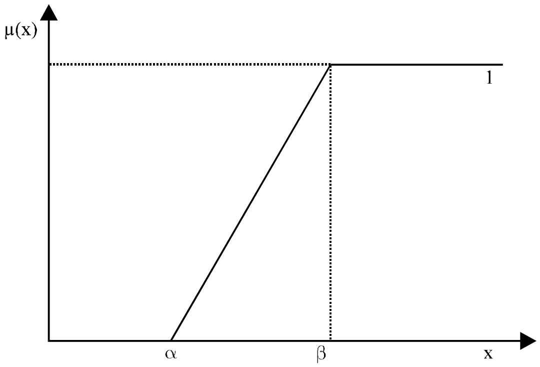

Fig. 1: Gamma function

$$\Gamma(x,\alpha,\beta)=\begin{cases}0 & ;x<\alpha\\ \dfrac{x-\alpha}{\beta-\alpha} & ;\alpha\leq x<\beta\\ 1 & ;x\geq\beta\end{cases}$$

and it is shown in Fig. 1. The variables α and β represent the limit of the gamma function. μ (x) be the degree of membership and lies between 0 and 1.

## RESOURCE AND WORKLOAD DISTRIBUTION MODEL

Normally the resource in the Grid System becomes a collection of Processing Elements (PEs) in the form of cluster with a cluster head shown in Fig. 2. All the PEs in the Grid System can be a normal desktop PC connected into the network. The PEs in the resource can be homogeneous or heterogeneous nature with respect to processing speed, memory capacity and its architecture, etc. If the PEs is homogeneous then one can easily allocate the workloads into the PEs with minimum allocation overheads. But in the case of heterogeneous PEs there are lot of allocation overheads due to different platform and the varying processing speed and memory capacity. Here we considered the PEs is in heterogeneous in nature.

The originator receives the workload it divides the entire workload into task (fraction of workload) and each task is assigned to a processor. Researchers used the term fraction of workload and task interchangeably. Here we assume that all the processors will start computing the task as they start receiving them and that the communication delay between the originator and the processors is negligibly smaller than the computation time

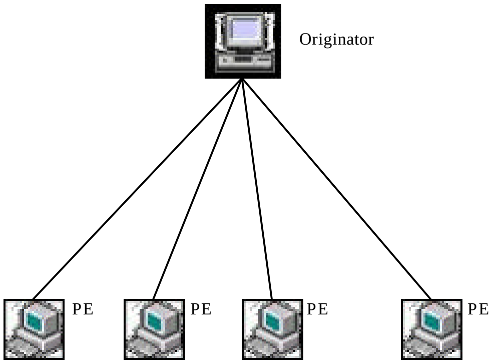

Fig. 2: Load Distribution Model

owing to high speed links so that no processor starve for load. In order to derive an optimal solution it is assumed that the processors involved in the computation process will available in the entire processing period. Also we assume that each processor has adequate memory/buffer space to accommodate and process the loads received from the originator.

The divisible load distribution model considered here goes through the following process. The load to be processed arrives at the node, called the originator or the cluster head. The originator distributes the workload fraction into the selected PEs. PE can be dedicated one or non-dedicated one which depends on the resource architecture under considered in the allocation process. All the PEs is connected to the originator with a communication link. The role of the cluster head is whenever it receives the load from the grid broker, it divides the load into m fractions called task and each task could be assigned to a PEs. If the PEs are homogeneous, i.e., all the processing elements are equipped with same operating system, memory capacity, processing speed and so on, then the cluster head divides the load into equal division. Otherwise unequal fraction of load is considered (Fig. 2).

The topology considered is single level tree topology, the cluster head is the root node and all the other PEs are its child node. The root node is directly connected to all the child nodes to communicate with them and there is no direct communication link between the child nodes. Here we are assuming that there is no communication failure and computation failure. But in real situation it is not possible. To reduce the computation complexity we considered as failure free communication and computation process. In tree topology, the workload distribution can be of random order or sequential order among the child processors or nodes. In sequential order, consider there is m number of child nodes then m! sequences are possible so, it become a NP-hard problem.

To overcome the NP-hard problem the researcher comes up with a solution of optimal sequencing the resources. An optimal sequence is that in which the load distribution follows the order with decrease in link speed. However, from the network designer's perspective, the best way allocate the workload fraction can be arrange the faster processors and links to connect the faster processor to faster link and the next faster processor to the next processor link and so, on then follow the optimal sequence of load distribution. Here we come up with the solution for sequencing the resources using their processing capacity and processing cost based on fuzzy logic technique. The following notations are used to solve the problem:

- $P_0$ is the originator which initially holds the total workload
- There are m child nodes (processing elements) denoted by $p_1, p_2, \ldots, p_m$
- Each child node $P_i$ takes a time $t^c_i$ to execute a unit workload (computation time for a unit task)
- Each child node $P_i$ needs time $t^z_i$ units to receive unit-size workload from the originator (communication time to transfer a unit task)
- Each child processor $P_i$ takes a cost $c^c_i$ as a computation cost to execute a unit workload
- Each link from $P_0$ to $P_i$ utilize the cost to transfer a unit of workload $c^z_i$ as a communication cost
- The total workload is denoted by $W_{total}$
- $P_0$ will allocate $w_i$ unit of task to child node $P_i$
- $T_t$ be the completion time of total workload
- The total budget allotted to complete the workload $W_{total}$ is B
- By definition, child node $P_i$ processes $\alpha_i$ portion of workload and we have:

$$\sum_{i=1}^{m} \alpha_i = W_{total}$$

- Time taken to process $\alpha_i$ portion of workload by each child node $P_i$ is $t^c_i.\ \alpha_i$
- Time taken to transfer $\alpha_i$ portion of workload from originator to $P_i$ is $t^z_i.\ \alpha_i$
- Each child processors are available in $[r_i, d_i]$
- The starting time of $P_i$ is $r_i$
- The ending time of $P_i$ is $d_i$

Normally there are two optimization criteria in resource allocation process in grid scheduling; finish time optimization and computing and communication cost optimization. Using the finish time optimization, one can study or improve the performance of the resource but in the other case aims to reduce the grid usage cost. Sometimes these two optimizing criteria were conflicting.

## DESIGN AND IMPLEMENTATION OF RESOURCE ALLOCATION

Let $W_{total}$ be the total workload received by the originator. The originator select the set of processor to process the entire workload with respect to the budget and the deadline assigned by the grid user. The total workload has to be divided into tasks and each task is assigned to a child node. Researchers used the term child node and processor interchangeably. Researchers are assuming that once the child node is selected in the allocation process, it has to be available during the entire scheduling period. We follow the Divisible Load Theory; state that all the selected child nodes has to participate and complete the computation process at the same time. The resources considered in this study is dynamic resource and they are available only in the period of $[r_i, d_i]$. To summarize, the complete formulation of the model become minimize:

$$Z = \sum_{i=1}^{N} (c_i^c \alpha_i + c_i^z \alpha_i) \tag{1}$$

Subject to:

$$\sum_{i=1}^{N} (c_i^c \alpha_i + c_i^z \alpha_i) \leq B \tag{2}$$

$$\sum_{j=1}^{i} t_j^z \alpha_j + t_i^c \alpha_i \leq T \tag{3}$$

$$\sum_{j=1}^{i} t_j^z \alpha_j + t_i^c \alpha_i \leq D \tag{4}$$

$$r_i + \alpha_i t_i^c \leq d_i;\ \forall i \tag{5}$$

$$r_i + t_i^c \alpha_i \leq T;\ \forall i \tag{6}$$

$$\sum_{i=1}^{N} \alpha_i = W_{total} \tag{7}$$

$$\alpha_i \leq m_i;\ \forall i \tag{8}$$

$$\alpha_i \geq 0;\ \forall i \tag{9}$$

From the earlier model Eq. 1 specifies the objective function, minimize the grid usage cost, it consists of two components; computation cost and communication cost. Equation 2-4 specifies the budget, schedule length and deadline constraints. Equation 5 and 6 makes the available time duration of each resource. The Eq. 7 ensures the sum of fraction of workload is equal to the total workload. Equation 8 and 9 are memory availability and the fraction of workload is a non negative value constraint, respectively.

The proposed model lies in the framework of mixed integer linear programming model even though the structure of linearity is quite simple, this model can be characterized by large scale dimensions (depending on the number of processors to manage) and requires a fast solution since the time horizon of the allocation is quite short.

## COMPUTATIONAL EXPERIMENTS

Since, the quite original nature of the proposed models, the computational experiments have been planned and carried out with the basic aim to access the suitability, reliability and semantic behavior of the same models. To this end, Researchers have considered several test problems designed to face all possible situations that can occur in practice with respect to the intersection of the supply and demand curve. We assumed that transferring a unit of workload is negligibly very small and there is no communication cost. Also we assume that there is sufficient buffer capacity to store and process the workload fraction for each child nodes. Once the resources are identified they have to be arranged in sequence and here the fuzzy logic technique is used. We have generated the fuzzy set which satisfies the gamma function defined in the form of (x, μ (x)) where x is the cost/time value and μ (x) is the degree of membership.

Table 1 shows the fuzzy set operation for computation time and cost of the resources in the Grid System. The first column denote the resource id, the second and third columns shows the computation time

Table 1: Fuzzy set operation for time and cost

| | | | Fuzzy set operation | | |
|---|---|---|---|---|---|
| | Time/Unit | Cost/Unit | | | |
| Resource | load | load | $\mu_A$ (x) | $\mu_B$ (x) | Min. ($\mu_A$(x), $\mu_B$ (x) |
| R1 | 4 | 5 | 0.75 | 0.40 | 0.40 |
| R2 | 3 | 6 | 0.50 | 0.60 | 0.50 |
| R3 | 4 | 5 | 0.75 | 0.40 | 0.40 |
| R4 | 2 | 6 | 0.25 | 0.60 | 0.25 |
| R5 | 4 | 5 | 0.75 | 0.40 | 0.40 |
| R6 | 3 | 5 | 0.50 | 0.40 | 0.40 |
| R7 | 2 | 7 | 0.25 | 0.80 | 0.25 |
| R8 | 3 | 6 | 0.50 | 0.60 | 0.50 |
| R9 | 4 | 4 | 0.75 | 0.20 | 0.20 |

Table 2. Resource availability

| Resource | Time/Unit load | Cost/Unit load | Available time |
|---|---|---|---|
| R1 | 4 | 5 | 40 |
| R2 | 3 | 6 | 40 |
| R3 | 4 | 5 | 40 |
| R4 | 2 | 6 | 40 |
| R5 | 4 | 5 | 30 |
| R6 | 3 | 5 | 30 |
| R7 | 2 | 7 | 50 |
| R8 | 3 | 6 | 40 |
| R9 | 4 | 4 | 40 |

Table 3: Workload allocation for resources

| Resource | Work load allotted | Time taken to process |
|---|---|---|
| R1 | 10 | 40 |
| R2 | 13 | 39 |
| R3 | 10 | 40 |
| R4 | 20 | 40 |
| R5 | 07 | 28 |
| R6 | 10 | 30 |
| R7 | 7 | 14 |
| R8 | 13 | 39 |
| R9 | 10 | 40 |

and cost to compute a single unit of workload fraction by each resources, the fourth and fifth column shows the degree of membership for both computation time and cost, respectively and the last column entries denote the member ship operation performed to compute the set. Arrange the last column entries in non decreasing order which gives the resource sequence.

The objective is to investigate rate at which the resource users and the resource providers are matched by the system's dynamic allocation mechanism as they emerge. The Table 2 shows the properties of the resources available in the grid system. From the table we are assuming that there are nine resources are in the first column of the table. The second column specifies the amount of time required to compute one unit of workload per unit time with respect to the resources. The third column specifies the cost required to compute a unit of workload in the various resources and the data in the last column shows the resource available time in the schedule.

To perform the experiment we have considered the total number of customer to process is 100 with the dead line of 360 time unit and the budget allotted to complete the process is 600 cost units. We perform the operation of cost minimization. From Table 3 the total of 100 customers is divided into nine tasks and the tasks assigned to the resources are shown in the second column entries in the table. Also in the table the resource R7 is only for 7 customers even it is available in long time due to high cost. The grid system completes the job with the budget of 550 cost unit and it takes 310 time units to complete the process of entire customer.

The experiment can be extended with very large number of resources. IBM offers on demand business service similar to Sun Grid where the customers are well versed in pay as you go service so that the cost incurred towards the excess hardware and software investment and maintenance become minimized. The actual pricing model has not been disclosed as in the case with Sun but is rather on per-customer basis. Beyond the computational grids, research grids such as TeraGrid and SURAgrid exist where the cost is not directly tied to currency but rather to allocation time. Regardless of how one approaches the grid, there is a cost associated with the available services.

## CONCLUSION

In this study, researchers proposed a new scheduling framework for allocating the resources economically. This study is based on divisible workload theory mechanism proposed by Bharadwaj *et al*. (1996). Most of the research based on divisible load theory not considered the resource sequencing as a factor to schedule the resource. A very few work considered the resource sequencing based on link capacity, memory availability and so on. There is no such research which considers both cost and time as a decision parameter to sequence the resource.

Researchers used the fuzzy logic technique to perform the resource selection and resource sequencing. The performance of the mathematical model is evaluated with an extensive set of input values to demonstrate the practical use of the model. The result shows that the approach outperform other models by minimizing the time and cost of the application of the grid user. We tested the model with and without resource sequencing, finds that the solution with sequencing of resource nearly 20% better than without sequencing. We designed the model with only one job at a time and we are proceeding with multiple workloads with both sequential and concurrent process.

## REFERENCES

Abdullah, M., M. Othman, H. Ibrahim and S. Subramaniam, 2009. Load allocation model for scheduling divisible data grid applications. J. Comput. Sci., 5: 760-763.

Bharadwaj, V., D. Ghose, V. Mani and T.G. Robertazzi, 1996. Scheduling divisible loads in parallel and distributed systems. Wiely-IEEE Comput. Soc. Press.

Buyya, R., D. Abramson, J. Giddy and H. Stockinger, 2002. Economic models for resource management and scheduling in grid computing. J. Concurr. Commun. Pract. Exp., 14: 1507-1542.

Garg, S.K., R. Buyya and H.J. Siegel, 2009. Scheduling parallel applications on utility grids: Time and cost Trade-off management. Conf. Res. Pract. Inform. Technol. (CRPIT), Vol. 91.

Ger, K. and B. De, 2008. Resource allocation in grid computing. J. Schedul., 11: 163-173.

Murugesan, G. and C. Chellappan, 2011. An economic allocation of resources for divisible workloads in grid computing paradigm. Eur. J. Sci. Res., 65: 434-443.

Nakai, J., 2002. Reading between the lines and beyond. Technical Report NAS-01-010, NASA Ames Research Center.

Takefusa A., H. Casanova, S. Matsuoka and F. Berman, 2001. A study of deadline scheduling for client-server systems on the computational grid. Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01), August 7-9, 2001, IEEE Computer Society, San Francisco, CA, USA., pp: 406-415.

Venugopal, S. and R. Buyya, 2005. A deadline and budget constrained scheduling algorithm for escience applications on data grids. Proceedings of 6th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP-2005), October 2-3, 2005, Melbourne, Australia, pp: 60-72.

Viswanathan, S., B. Veeravalli and T.G. Robertazzi, 2007. Resource-aware distributed scheduling strategies for large-scale computational cluster/grid systems. IEEE Trans. Parallel Distrib. Syst., 18: 1450-1461.