An Economic Allocation of Resources for Multiple Grid Applications

G. Murugesan, Dr.C. Chellappan

Abstract—**Resource allocation is an important component of a Grid Computing infrastructure. In this paper, we proposed a mathematical model for resource allocation for multiple Grid Applications with multiple processors (sinks). Most of the researches are based on the Divisible Load Theory mechanism; the processors which are participated in the processing have to select a divisible job and also all the processor has to complete the process at the same time. But in our work, initially considering the entire load from sources are divided into equally and select a set of processor from the available processors. The numbers of processors are selected with the help of its processing capacity. Equal allocation of load is attractive in multiple processor systems when real time information on processor and link capacity that is necessary for optimal scheduling is not available. A new mathematical model for minimizes the computing cost with equal allocation of divisible computation and communication load is developed. A cost optimal processor sequencing result is found which involves assigning load to processors in order of the cost per load characteristic of each processor**

Index Terms — **Grid Scheduling, Integer Programming, Multiple Sources, Resource Allocation**

I. INTRODUCTION

Until very recently, grid computing was popular only in the research arena. However recent business reports suggest that enterprises are working towards implementing enterprise wide grids to share and utilize their vastly distributed computing resources. The aim of Grid computing is the utilization of underutilized resources to achieve faster job execution time. Gird can also provide access to software, computers, data, and other computing resources. Grid computing reduces the cost by connecting different machines like PCs and workstations to behave as a larger computational machine rather than purchasing special expensive machines to execute the complicated jobs. This can be sometimes showed as dividing the cost of the resources participating in the Grid between the users who use the Grid.

The management is an essential part in any system in the world. It comprises both hardware and software management. It determines if the system succeeds or fails. Furthermore, the Grid's resources are scattered and geographically distributed, therefore more care is needed when they are being managed.

An important problem that arises in the area of grid computing is one of the optimally assigning jobs to resources to achieve business objectives. In Grid Computing environment Scheduling, performance prediction and resource management are important but challenging tasks. Grid scheduling consists of finding a suitable assignment between a computational workload and computational resources which are participated in the Computation. Reasonably allocating the resources to the tasks can effectively improve the concurrency ability of the distributed system. Over the past 19 years a good deal of research has been conducted on scheduling and load distribution with divisible loads. A divisible load is a data parallel load that can be arbitrarily partitioned among links and processors to gain the advantage of parallel processing. However most of this research has involved load distribution from a single source. That is, load originates from a single node in a larger grid or network. Multi-source load scheduling has received less attention but is a logical next step for research in this area.

Task scheduling is an integrated part of parallel and distributed computing. The Grid scheduling is responsible for resource discovery, resources selection, job assignment and aggregation of group of resources over a decentralized heterogeneous system; the resources belong to multiple administrative domains. The resources are requested by a Grid application, which use to computing, data and network resources etc. However, Scheduling an applications of a Grid system is absolutely more complex than scheduling an applications of a single computer. Because to get the resources information of single computer and scheduling is easy, such as CPU frequency, number of CPU's in a machine, memory size, memory configuration and network bandwidth and other resources connected in the system. But Grid environment is dynamic resources sharing and distributing. Then an application is hard to get resources information, such as CPU load, available memory, available network capacity etc. And Grid environment also hard to classify jobs characteristic, that run in Grid. There are basically two approaches to solve this problems, the first is based on jobs characteristic and second is based on a distributed resources discovery and allocation system. It should optimize the allocation of a job allowing the execution on the optimization of resources. The scheduling in Grid environment has to satisfy a number of constraints on different problems.

This paper is organized as follows: section II presents the related works, grid resource allocation is discussed in

---

Manuscript received June 18, 2009.

G. Murugesan, Research Scholar, Department of Computer Science and Engineering, Anna University, Chennai – 600 025 (Phone: 044-2266 0142; E-mail: murugesh02@gmail.com).

Dr. C. Chellappan, Professor and Head, Department of Computer Science and Engineering, Anna University, Chennai- 600 025 (E-mail: drcc@annauniv.edu).

the section III, section IV shows the resource model, design and analysis of load distribution model described in the section V, section VI presents the mathematical model for resource allocation, section VII shows the experimental results, section VIII and section IX shows the conclusion and references respectively.

II. RELATED WORKS

In this section, we present some of the works that are relevant to the problem addressed in this paper. The problem of minimizing the processing cost of extensive processing loads originating from various sources presents a challenging task that if successfully met could faster a range of new creative applications. Inspired by this challenge, we sought to apply linear programming technique to assign equally divided jobs to the available resources. For divisible loads, research since 1988 has established that optimal allocation /scheduling of a divisible load to processors and links can be solved through the use of very tractable linear model formulation, referred in divisible load theory (DLT). DLT can model a wide variety of approaches with respect to load distribution (sequential or concurrent), communication (store and forward and virtual cut-through switching), and hardware availability (the presence or absence of front-end processors). Front-end processors allow a processor to both communicate and compute simultaneously by assuming communication duties [9]. DLT has been proven to be remarkably flexible. The DLT model allows analytical tractability to derive a rich set of results regarding several important properties of the proposed strategies and to analyze their performance. A 2002 paper on multi-source load distribution combining Markovian queuing theory and divisible load scheduling theory is discussed in Ko and Robertazzi [1]. In 2003 Wong, Yu, Veeravalli, and Robertazzi examined multiple source grid scheduling with buffer and without buffer capacity constraints [2]. Moges, Yu and Robertazzi considered multiple source scheduling using two root processor for small size models via linear programming and closed form solutions in 2004 and 2005, respectively [3], [4]. Marchal, Yang, Casanova, and Robert in 2005 studied the steady-state multi-application scheduling problem use of linear programming to maximize throughput for large grids with multiple loads/sources expresses a notion of fairness between applications [5]. Yu and Robertazzi proposed the use of min cost flow and multi-commodity flow formulations for steady state divisible load scheduling with multiple sources [6]. Viswanathan et al. [7] proposed a distributed algorithm to handle large volumes of computationally intensive arbitrarily divisible loads submitted for processing at cluster/ grid systems involving multiple sources and sinks. The different scheduling algorithms on heterogeneous platforms for divisible workloads are proposed by Beaumont et al [8]. This paper is significant for proposing some new optimization approaches for the multiple source scheduling problems in grids. Scheduling is one of the most studied topics in distributed systems. In this paper we dealt with multiple sources with multiple resources part of the work is based on optimal allocation of loads [10]. Each source having their jobs and the entire job of the each source is divided and each portion of the job has to be submitted from a set of sources. Initially we are selecting a set of best resources from the available resources. Each sources job has to be equally divided in to sub jobs and each sub job was assigned to a resource from the selected group of the resource.

III. GRID RESOURCE ALLOCATION

A generic grid computing system infrastructure considered hare comprises a network of supercomputers and/or a cluster of computers connected by local area networks, as shown in Figure. 1 having different computational and communication capabilities. We consider the problem of scheduling large-volume loads (divisible loads) within a cluster system, which is part of a grid infrastructure. We envisage this cluster system as a cluster node comprising a set of computing nodes. Communication is assumed to be predominant between such cluster nodes and is assumed to be negligible within a cluster node. The underlying computing system within a cluster can be modeled as a fully connected bipartite graph comprising sources, which have computationally intensive loads to be processed (very many operations are performed on them) and computing elements called sinks, for processing loads (data). To organize and coordinate distributed resources participating in the grid computing environment, we utilize the resource model which contain more than one sources and resources, based on which the uniform and scalable and allocating computational resources are identified. The contribution of this paper is that we propose equal division of load from a resource with the help of generating a random number and the divisible load is assign to a set of resources from the resources participated in the scheduling process. Now, we shall formally define the problem that we address.

IV. RESOURCE MODEL

Our model organizes sources and the resources with a structure of a hybrid hierarchical tree Figure 1. In the tree first level nodes called root nodes represent the sources which contains the divisible loads, the second level specifies the scheduler, which get the total workload (divisible load) from each sources, the third level specifies the equal division of loads of each sources, and the fourth level nodes (leaf nodes) are represented as resources or processors to perform the processes. In a dynamic Grid environment, nodes may join or leave frequently, and nodes status may change dynamically. In our model, we are considering the environment as static; i.e. once the node joined in the scheduling process the entire node has to participate in the processing. But in divisible load theory all the nodes has to involve or process all sources portion of jobs. In this model we are not compelling all the resources to process all the sources loads, instead only the selected set of resources.

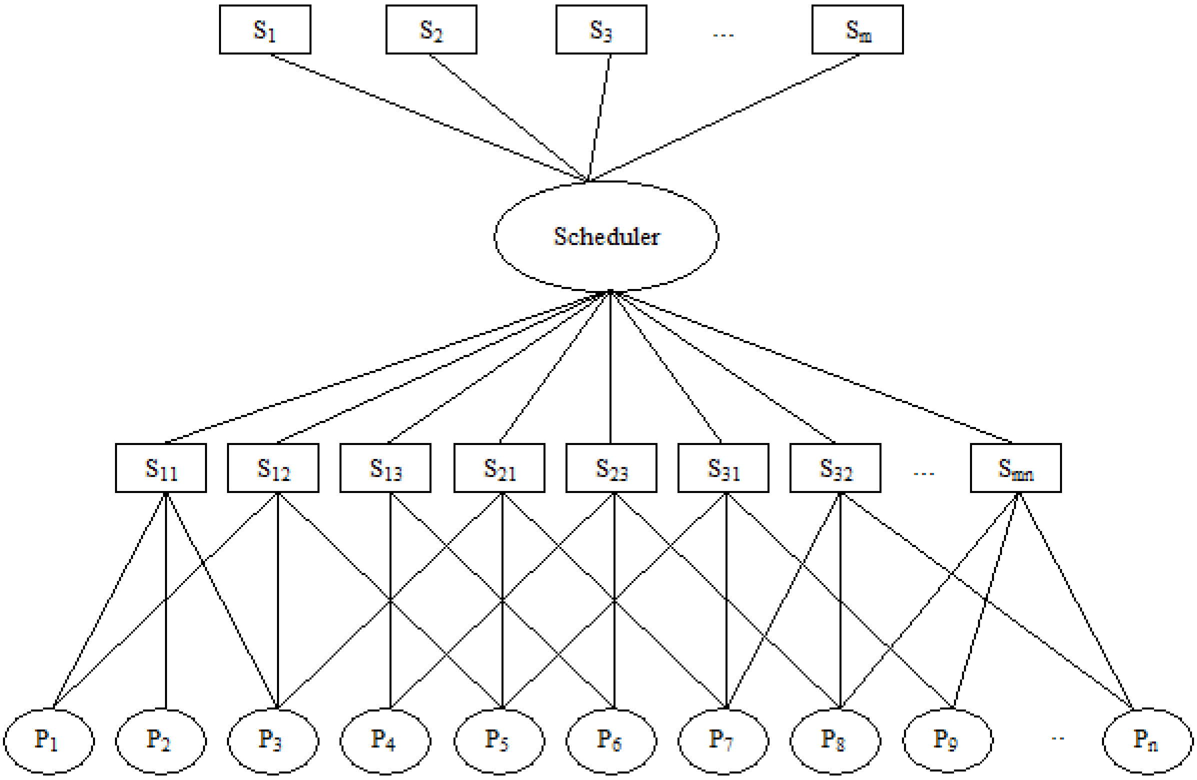

Figure 1. Abstract view of a cluster comprising sources and sinks with a scheduler

V. Design And Analysis of Load Distribution Model

In all the literature within the divisible load scheduling domain so far, an optimality criterion that is used to derive an optimal solution is as follows. It states that in order to obtain an optimal processing time, or an optimal make-span it is necessary and sufficient that all the sinks that participate in the computation must stop at the same time instant from a source. In multiple source environment all the resources has to take a portion of load from each sources as well as they has to be finish the work at the same time. In real time environment we cannot enforce all the processor to complete its processing at the same time. In our model we are not enforcing its load processing of all the processors at the same time as well as not all the processor has to take a portion of loads from all the sources. Otherwise, load could be redistributed to improve the processing time. We use this optimality principle in the design of our distribution strategy. The scheduling strategy involves the portioning and distribution of the processing loads originated from S_1, S_2, … S_m. The total loads of the each source are equally divided in to portion of loads and each portion is allotted into a separate processor. Before job scheduling, the scheduler identifies the number of resources required to participate the process with respect to the numbers portions.

We consider a tightly coupled, bipartite multiprocessor system. In the Grid system, we assume that there are M sources denoted as S_1, S_2, … S_m and N resources denoted as P_1, P_2, . . . P_n For each source, there is a direct link to all the resources. Each source Si has a load, and which is equally divided into portion of loads, denoted by S_{11}, S_{12}, . . . S_{nm}. Without loss of generality, we assume that all sources can send their loads to all the selected resource simultaneously. Similarly, we also assume that all the resource can receive load portions from all sources at the same time instant. The objective in this study is to schedule all the M loads among N resources such that the processing time, defined as the time instant when all loads have finished being processed by all the N resource, is minimal. The scheduling strategy is such that the scheduler will first obtain the information about the size of the loads that other sources have in their local memory. The scheduler will then calculate and notify each source of the optimum amount of load that each source has to give to each sink. This information can be easily communicated via any means of standard or customized communication protocol and it would not incur any significant communication overhead. The resources will then start computing the loads immediately after they receive their respective loads. It may be noted that we assume that each resource has adequate memory/buffer space to accommodate and process all the loads it receives from all the sources. We also assume that communication time is faster than computation time so no processor starves for load.

VI. NOTATIONS USED IN THE MODEL

N	No. of Processor required completing the job
M	No. of Sources involved in the scheduling process
C_k	Amount to spend to utilize the k^{th} resource
T_k	Time required to process a unit load by the k^{th} resource
s_{ij}	j^{th} portion of load from i^{th} source
b_i	Budget of the i^{th} source
T_k	Total available time of k^{th} resource
x_{ijk}	$= 1$; if i^{th} source j^{th} portion of load processed by k^{th} resource $= 0$; Otherwise

Minimize

$$\sum_{i=1}^{M}\sum_{j=1}^{m_i}\sum_{k=1}^{N} C_k t_k S_{ij} X_{ijk}$$

Subject to

$$\sum_{k=1}^{N} X_{ijk} = 1 \qquad ; i = 1, 2, \ldots M \qquad ; j = 1, 2, \ldots m_i$$

$$\sum_{i=1}^{M}\sum_{j=1}^{mi} t^k S_{ij} X_{ijk} = T_k \;\; ; k = 1, 2, \ldots N$$

$$\sum_{i=1}^{M}\sum_{j=1}^{m_i}\sum_{k=1}^{N} C_k t_k S_{ij} X_{ijk} = b_i$$

$$X_{ijk} = \{ 0 , 1 \}$$

$$S_{ij} >= 0$$

$$t_k >= 0$$

VII. EXPERIMENTAL RESULT

Let us assume that the Grid system consists of five processors (resources) namely P_1, P_2, P_3, P_4, and P_5 with four sources S_1, S_2, S_3 and S_4 trying to utilize the grid system to execute their workloads. Also assuming that the workloads of the four sources are equally divided and the divisions for S_1 and S_2 become three and for S_3 and S_4 are two respectively. The total workload and the divisions of workloads of each source are shown in table I. Also the processing time, processing cost per unit time and the total available time of each processor are described in table II.

When solving the model using the LINDO software package, the processor assignments are shown in table III. The total cost of the execution of all the workload is Rs.138.

TABLE I. TOTAL WORKLOAD AND ITS DIVISIONS

Sources	Total Workload	No. of Sub-division
S_1	6	3
S_2	9	3
S_3	6	2
S_4	8	2

TABLE II. PROCESSOR CAPACITY

Processors	Processing Time	Processing cost	Availability (time unit)
P_1	2	4	12
P_2	3	3	12
P_3	1	2	12
P_4	2	4	12
P_5	4	2	12

TABLE III. PROCESSOR ALLOTMENT

Sources		Processor Allotted
S_1	S_{11}	P_2
	S_{12}	P_3
	S_{13}	P_4
S_2	S_{21}	P_1
	S_{22}	P_1
	S_{23}	P_3
S_3	S_{31}	P_3
	S_{32}	P_5
S_4	S_{41}	P_3
	S_{42}	P_4

VIII. CONCLUSION

In this paper, we have proposed scheduling strategies for processing multiple divisible loads on grid systems. As in real life the loads will come dynamically and the resources are also in dynamic in nature. But here we considered the both loads and the resources are in static manner. This work can be extended for dynamic nature also. We are in the process of dynamic arrival of loads and the resources. Also here we have used random number to divide the loads from each source equally. This can be extends to divide the load from a source into the resource capacity. Also we are assumed that there is sufficient buffer space is available in all the resources. The experimental result demonstrates the usefulness of this strategy. We need a system to find out the execution time of a task and also the cost of usage of a processor/resource. The execution time is entirely depends upon the processor speed. Also in this model the number of portion of each total work load of a resource is selected with respect to the random number. So we require a separate module for random number generation. In future it can be modified without using random number as well as unequal division of total workload of each source.

REFERENCES

[1] Kwangil Ko and Thomas G. Robertazzi, “Scheduling in an Environment of Multiple Job Submission”, in Proceedings of the 2002 Conference on Information Sciences and Systems, Mar.2002.

[2] Han Min Wong, Dantong Yu, Bharadwaj Veeravalli, Thomas G. Robertazzi, “Data Intensive Grid Scheduling: Multiple Sources with Capacity Constraints”, in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), Nov.2003

[3] M. Moges and T. Robertazzi, “Grid Scheduling Divisible Loads from Multiple Sources via Linear Programming”, in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2005), 2004.

[4] M. Moges and T. Robertazzi, and D. Yu, “Divisible Load Scheduling with Multiple Sources: Closed Form Solutions”, in Proceedings of 2005 Conference on Information Sciences ans Systems, 2005

[5] L. Marchal, Y. Yang, H. Casanova, and Y. Robert, “A Realistic Network Application Model for Scheduling Divisible Loads on Large-Scale Platforms,” in International Parallel and Distributed Processing Symposium IPDPS’2005, IEEE Computer Society, Apr. 2005.

[6] Thomas G. Robertazzi and Dantong Yu, “Multi-Source Grid Scheduling for Divisible Loads”, in 40th Annual Conference on Information Sciences and Systems, Mar.2006

[7] Sivakumar Viswanathan, Bharadwaj Veeravalli, and Thomas G. Robertzzi, Resource-Aware Distributed scheduling Strategies for Large-Scale Computational Cluster/Grid Systems, IEEE Transactions on Parallei and Distributed Systems, Vol.18. No. 10 Oct.2007, pg.1450 -1461

[8] Olivier Beaumont, Arnaud Legrand, and Yves Robert, Scheduling divisible workloads on heterogeneous platforms, Parallel Computing, Vol.29, June2003, pg.1121-1152.

[9] V.Bharadwaj, D.Ghose, V.Mani, and T.G. Robertazzi, Scheduling Divisible Loads in Parallel and Distributed Systems. Computer Society Press, Sept.1996

[10] G.Murugesan, C.Chellappan, An Optimal Allocation of Loads from Multiple Sources for Grid Scheduling, in ICETIC’2009, International Conference on Emerging Trends in Computing, Jan.2009.