Journal of Computer Science 5 (10): 760-763, 2009
ISSN 1549-3636

Load Allocation Model for Scheduling Divisible Data Grid Applications

Monir Abdullah, Mohamed Othman, Hamidah Ibrahim and Shamala Subramaniam
Department of Communication Technology and Network,
University Putra Malaysia, 43400 UPM, Serdang, Selangor DE, Malaysia

Abstract: Problem statement: In many data grid applications, data can be decomposed into multiple independent sub-datasets and distributed for parallel execution and analysis. **Approach:** This property had been successfully employed by using Divisible Load Theory (DLT), which had been proved as a powerful tool for modeling divisible load problems in data-intensive grid. **Results:** There were some scheduling models had been studied but no optimal solution has been reached due to the heterogeneity of the grids. This study proposed a new optimal load allocation based on DLT model recursive numerical closed form solutions are derived to find the optimal workload assigned to the processing nodes. **Conclusion/Recommendations:** Experimental results showed that the proposed model obtained better solution than other models (almost optimal) in terms of Makespan.

Key words: Data grid, scheduling, divisible load theory

INTRODUCTION

In the last decade, data grids have increasingly become popular for a wide range of scientific and commercial applications[1,2]. Load balancing and scheduling play a critical role in achieving high utilization of resources in such environments[3]. Scheduling an application is significantly complicated and challenging because of the heterogeneous nature of a grid system. Grid scheduling is defined as the process of making scheduling decisions involving allocating jobs to resources over multiple administrative domains[7]. Most of the scheduling strategies try to reduce the Makespan or the maximum completion time of the task which is defined as the difference between the time when the job was submitted to a computational resource and the time it completed. Makespan also includes the time taken to transfer the data to the point of computation if that is allowed by the scheduling strategy[7].

In other hand, in many data intensive grid applications, data can be decomposed into multiple independent sub datasets and distributed for parallel execution and analysis. High Energy Physics (HEP) experiments fall into this category[1]. HEP data are characterized by independent events and therefore this characteristic can be exploited when parallelizing the analysis of data across multiple sites. The DLT paradigm[11] has emerged as a powerful tool for modeling data-intensive computational problems incorporating communication and computations issues[4]. An example of this direction is the work by[5] where the DLT is applied to model the grid scheduling problem involving multiple sources to multiple sinks. In that model, they did not consider the communication time. Whereas, the scheduling in grid applications must consider communication and computation simultaneously to achieve high performance.

Relevant materials to the problem addressed in this study are in[6,8,9] where Constraint DLT (CDLT), Adaptive DLT (ADLT), A^2DLT and Adaptive Task Data Present (ATDP) models are proposed. These models are proposed for scheduling divisible load data-intensive grid applications. In CDLT model, they stated that the scheduler targets an application model wherein a large dataset is split into multiple smaller datasets[5]. Then, these datasets processed in parallel on multiple virtual sites, where a virtual site is considered to be a collection of computing resources and data servers. However, in CDLT, the communication time for transferring load is not considered. In addition, ADLT and A^2DLT models are proposed in considering communication time as well as computation time in[8,9], respectively. The proposed TDP model is proposed without considering input transfer time.

Our objective is to design a load distribution model by taking into account the communication time and computation time in such a way that the entire processing time of the load is a minimum. The main contribution of this study is the closed form solutions

Corresponding Author: Monir Abdullah, Department of Communication Technology and Network, University Putra Malaysia, 43400 UPM, Serdang, Selangor DE, Malaysia Tel: +603-89466535 Fax: +603-89466577

for the minimum completion time and the optimal data allocation for each processing nodes are obtained. We validate the model through mathematical proof and comprehensive simulations.

A generic data grid computing system infrastructure considered here comprises a network of supercomputers and/or clusters of computers connected by Wide Area Network (WAN), having different computational and communication capabilities. We consider the problem of scheduling large-volume loads (divisible loads) within in multiple sites. Communication is assumed to be predominant between such cluster nodes and is assumed to be negligible within a cluster node[6,8-10].

The target data intensive application model can be decomposed into multiple independent subtasks and executed in parallel across multiple sites without any interaction among sub tasks. For example, let's consider job decomposition by decomposing input data objects into multiple smaller data objects of arbitrary size and processing them on multiple virtual sites. High Energy Physic (HEP) jobs are arbitrarily divisible at event granularity and intermediate data product processing granularity[2]. In this research, assuming that a job requires a very large logical input Data set (D) consists of N physical datasets and each physical dataset (of size L_k) resides at a data source (DS_k, for all k = 1,2…,N) of a particular site. Figure 1 shows how the logical input Data (D) is decomposed onto networks and their computing resources.

The scheduling problem is to decompose D into datasets (D_i for all i = 1, 2,...,M) across N virtual sites in a Virtual Organization (VO) given its initial physical decomposition. We assume that the decomposed data can be analyzed on any site.

For the notations, definitions that used in this research are stated in Table 1.

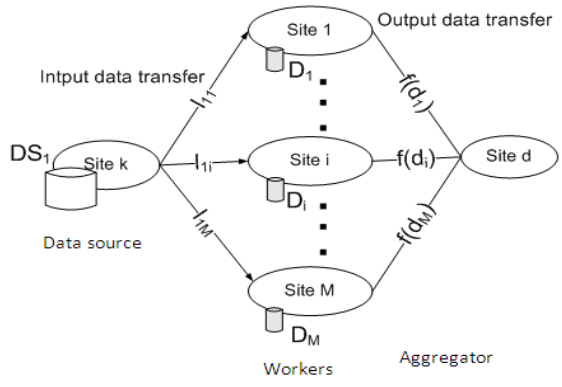

Fig. 1: Data decomposition and their processing

The execution time of a subtask allocated to the site i (T_i) and the turn around time of a job J (Tturn_around_time) can be expressed as follows:

$$T_i = T_{input\_cm}(i) + T_{cp}(i) + T_{output\_cm}(i,d)$$

$$T_{turnaround\_time} = \max_{i=1,...,M}\{T_i\}$$

The cost (T_i) includes input data transfer ($T_{input\_cm}(i)$), computation ($T_{cp}(i)$) and output data transfer to the client at the destination site d ($T_{output\_cm}(i,d)$):

$$T_{input\_cm}(i) = \max_{k=1..m}\{\alpha_{ki} \cdot \frac{1}{z_{ki}}\}$$

$$T_{cp}(i) = d_i \cdot w_i$$

$$T_{output\_cm}(i,d) = f(d_i) \cdot z_{id}$$

We assume that data from multiple data sources can be transferred to a site i concurrently in the wide area environment and computation starts only after the assigned data set is totally transferred to the site. Hence, the problem of scheduling a divisible job onto n sites can be stated as deciding the portion of original workload (D) to be allocated to each site, that is, finding a distribution of distribution of $\{\alpha_{ki}\}$ which minimizes the turn-around time of a job. The proposed SA approach uses this cost model when evaluating solutions at each generation.

In all the literature related to the divisible load scheduling domain so far, an optimality criterion[11] is used to derive an optimal solution is as follows. It states that in order to obtain an optimal processing time, it is necessary and sufficient that all the sites that participate in the computation must stop at the same time. Otherwise, load could be redistributed to improve the processing time. The timing diagram for this distributed system in optimal case is depicted in Fig. 2. In this timing diagram, communication time appears above the axis and computation time appears below the axis.

Table 1: Terminology, definitions and notations

N	The total number of data files in the system
M	The total number of nodes in the system
L_i	The loads in data file i
L_{ij}	The loads that node i will receive from data file j
L	The sum of loads in the system, where $L = \sum_{i=1}^{N} L_i$
α_{ij}	The fraction of L that node i will receive from all data file j
w_i	The inverse of the computing speed of node i
Z_{ij}	The link between node i and data source j
T (i)	The processing time in node i

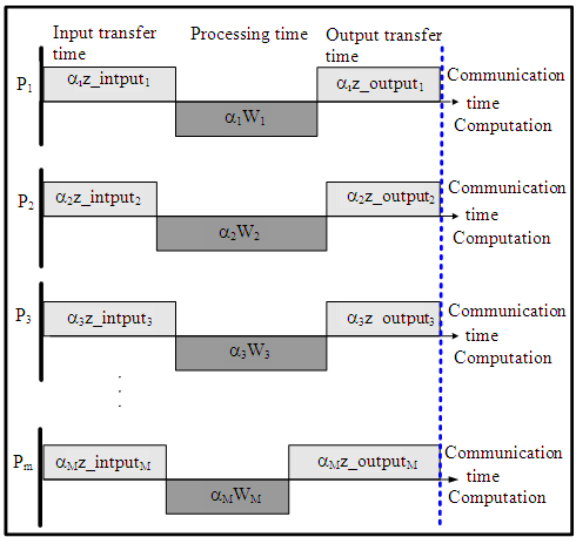

Fig. 2: Optimal case

MATERIALS AND METHODS

The load scheduling problem is to decompose D into datasets (D_i for all i = 1, 2,…,M) across M virtual sites in a Virtual Organization (VO) given its initial physical decomposition. This model includes two steps:

New optimal closed form formulas for scheduling divisible load on large scale data grid system are proposed. Closed-form expressions for the processing time and the fraction of workload for each processing node are derived. Since, the closed form did not consider the communication time[5]. The new model considers computation time as well as communication time.

A new closed form solution is proposed for obtaining the optimal fraction (α_i) as follows:

$$\alpha_1.W_1+\alpha_1.Z_1 = T_1$$

Where:
W_1 = The node speed
Z_1 = Link speed between the source and the node 1.

And similarly:

$$\alpha_2.W_2+\alpha_2.Z_2 = T_2$$

$$\alpha_i.W_i+\alpha_i.Z_i = T_i,\ i = 1,2,\ldots, \quad (1)$$

$$\alpha_M.W_M+\alpha_M.Z_M = T_M \quad (2)$$

In this case:

$$T_1 = T_2 = \ldots = T_M = T \quad (3)$$

$$\alpha_i.W_i+\alpha_i.Z_i = T,\ i = 1,2,\ldots,M \quad (4)$$

$$\alpha_i.(W_i+Z_i) = T \quad (5)$$

$$\alpha_1+\alpha_2+\ldots+\alpha_{M-1}+\alpha_M = 1 \quad (6)$$

From Eq. 5 and 6 we obtain:

$$\alpha_i = \frac{T}{Wi + Zi},\ i = 1,2,\ldots, M \quad (7)$$

$$\frac{T}{W_1 + Z_1} + \frac{T}{W_2 + Z_2} + \ldots + \frac{T}{W_M + Z_M} = 1 \quad (8)$$

Thus, the closed-form expression of processing time (Makespan) is given as:

$$T = \frac{1}{\frac{1}{W_1 + Z_1} + \frac{1}{W_2 + Z_2} + \ldots + \frac{1}{W_M + Z_M}}$$

Moreover, we can add the application type (ccRatio) to the equation as:

$$T = \frac{1}{\frac{1}{W_1.ccRatio + Z_1} + \frac{1}{W_2.ccRatio + Z_2} + \ldots + \frac{1}{W_M.ccRatio + Z_M}}$$

After we get T, we can get α_i by Eq. 7. By calculating the α_i, the optimal time will be calculated.

RESULTS AND DISCUSSION

To measure the performance of the proposed model against CDLT, ADLT and A^2DLT models, randomly generated experimental configurations were used. The estimated expected execution time for processing a unit dataset on each site, the network bandwidth between sites, input data size and the ratio of output data size to input data size were randomly generated with uniform probability over some predefined ranges. The network bandwidth between sites is uniformly distributed between 1 Mbyte sec and 10 Mbps.

The location of m data sources DS_k is randomly selected and each physical dataset size L_k is randomly selected with a uniform distribution in the range of 1 GB - 1 TB. It is assumed that the computing time spent in a site i to process a unit dataset of size 1 MB is uniformly distributed in the range 1-10/r_{cb} seconds, where r_{cb} is the ratio of computation speed to communication speed.

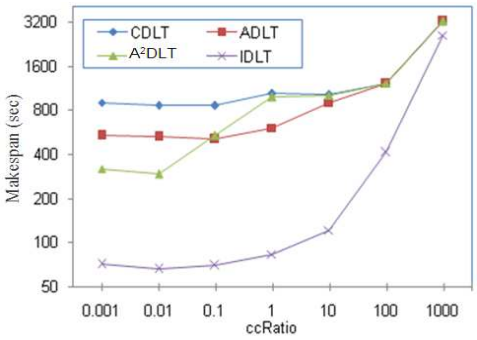

Fig. 3: Makespan Vs ccRatio for CDLT, ADLT, A^2DLT and IDLT (M = 100)

We examined the overall performance of each model by running them under 100 randomly generated Grid configurations. These parameters are varied: ccRatio (0.001-1000), M (20-100), N (20-100), r_{cb} (10-500) and data file size (1 GB-1 TB). To show how these models perform on different type of application (different ccRatio), we created graphs in Fig. 3.

In Fig. 3, the Makespan for the CDLT, ADLT, A^2DLT and the proposed models is plotted against application type (ccRatio). The value of ccRatio is fixed at 1000 and the value of number of nodes M is fixed to be 100. It can be shown from the Fig. 3 that the proposed model is the best for any type of application, as expected, because the proposed model produce the almost optimal solution for scheduling load that is produced from single source.

CONCLUSION

In this study, we have developed an effective Iterative model for optimal workload allocation. The proposed model is proposed for load allocation to processors and links for scheduling divisible data grid applications. The experimental results showed that the proposed model is capable of producing almost optimal solution for single source scheduling. Hence, the proposed model can balance the processing loads efficiently. We are planning to adapt the proposed model to be implemented in multiple sources. With such improvements, the proposed model can be integrated in the existing data grid schedulers in order to improve their performance.

REFERENCES

1. Holtman, K., 2001. CMS requirements for the grid. Proceeding of the International Conference on Computing in High Energy and Nuclear Physics, Sept. 3-7, Science Press, Beijing China, pp: 1-11. http://www.ihep.ac.cn/~chep01/presentation/10-053.pdf
2. Tierney, B., W. Johnston, J. Lee and M. Thompson, 2000. A data intensive distributed computing architecture for grid applications. Future Generat. Comput. Syst., 16: 473-481. http://portal.acm.org/citation.cfm?id=342429.342463
3. Xiao, Q., 2007. Design and analysis of a load balancing strategy in data grids. Future Generat. Comput. Syst. 16: 132-137. http://portal.acm.org/citation.cfm?id=1276047.1276065
4. Tang, M., B.S. Lee, X. Tang and C.K. Yeo, 2006. The impact of data replication on job scheduling performance in the data grid. Future Generat. Comput. Syst., 22: 254-268. http://portal.acm.org/citation.cfm?id=1134243
5. Wong, H.M., D. Yu and B. Veeravalli, 2003. Data intensive grid scheduling: Multiple sources with capacity constraints. Proceeding of the IASTED Conference on Parallel and Distributed Computing and Systems, Nov. 3-5, Marina Del Rey, USA., pp: 7-11. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.8519
6. Kim, S. and J.B. Weissman, 2004. A genetic algorithm based approach for scheduling decomposable data grid applications. Proceeding of the International Conference on Parallel Processing, Aug. 15-18, IEEE Computer Society Press, Washington DC., USA., pp: 406-413. http://portal.acm.org/citation.cfm?id=1020313
7. Venugopal, S., R. Buyya and K. Ramamohanarao, 2006. A taxonomy of data grids for distributed data sharing, management and processing. ACM Comput. Surv., 38: 1-50. http://portal.acm.org/citation.cfm?id=1132952.1132955
8. Othman, M., M. Abdullah, H. Ibrahim and S. Subramaniam, 2007. Adaptive divisible load model for scheduling data-intensive grid applications: Computational science. Lecture Notes Comput. Sci., 4487: 446-453. DOI: 10.1007/978-3-540-72584-8_59
9. Othman, M., M. Abdullah, H. Ibrahim and S. Subramaniam, 2007. A^2DLT: Divisible load balancing model for scheduling communication-intensive grid applications: Computational science. Lecture Notes Comput. Sci., 5101: 246-253. DOI: 10.1007/978-3-540-69384-0_30
10. Viswanathan, S., B. Veeravalli and T.G. Robertazzi, 2007. Resource-aware distributed scheduling strategies for large-scale computational cluster/grid systems. IEEE Trans. Parall. Distribut. Syst., 18: 1450-1461. DOI: 10.1109/TPDS.2007.1073
11. Bharadwaj, V., D. Ghose and T.G. Robertazzi, 2003. Divisible load theory: A new paradigm for load scheduling in distributed systems. Cluster Comput., 6: 7-17. http://portal.acm.org/citation.cfm?id=1017029