International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 16 (2014), pp. 1727-1732

http://www. irphouse.com

Load Balancing and Scheduling of Tasks in Parallel Processing Environment

Vivek Kumar Prasad

*Computer Science and Engineering Department,
Nirma University, Ahmedabad*

Abstract

This proposed model applies to scheduling and doing load balancing of the tasks in the parallel processing applications which can be divided into parts of arbitrary sizes, which in turn can be processed independently on remote computers. The objective is to come up with the optimal divisible load scheduling and balancing model to solve a computational problem in a minimal amount of time (with the reduction in delay occur due to the communication) for the processors located in the networks. I have proposed a model where by using the Hidden Markov Model (HMM) and A Virtual Node Based Network Distance Prediction Mechanism we can schedule as well as can do better load balancing in the parallel processing. The resultant model can assign the correct task to the correct processes in the distributed system and thus reduce the communication delay while doing scheduling or load balancing. The above logic can be placed at the virtual node which acts as an intermediate node, which is placed between the clusters.

Keywords: Scheduling, Load Balancing, Tasks, Parallel processing, HMM, Virtual Node Based Network Distance Prediction.

I. Introduction

A parallel programs are represented by a directed acyclic task graph [1] in which each node, represents a task. The amount of computation required in a task is called the computation cost. The edges in the parallel program graph correspond to the communication messages and precedence constraints among the tasks. A number is associated with each edge to denote the amount of communication data from a task to another. This number is called the communication cost. The source node and the destination node of an edge is called the parent node and the child node, respectively. A node which does not have parent node is called an entry node whereas a node which does not have child node is called an exit node. These tasks can be implemented in scheduling which ultimately helps in load balancing too. The scheduling problem arises in many circumstances and more especially in the design of multiprocessor operating systems in distributed systems. In most cases, the problem can be formulated as follows: a set T *(Ti,. . . . . . , Tj)* of *m* tasks with their precedence constraints are to be scheduled on a system of *n* identical processors so that the overall execution time is minimum. This problem was largely studied as an optimization problem for which the optimization algorithm is to construct an optimal schedule and load balancing. There might be other approaches to achieving efficiency of parallel computer systems. Some of them concentrates on a particular elements of the system, other try to optimize the system as a whole. The example can be allocation, load balancing, routing etc. Which differs very slightly from the classical scheduling. The rest of the paper is organized as follow. Section II gives the details about the problem statements, Section III illustrates the proposed solution which is followed by the conclusion.

II. Problem Definition.

In the simplest formulation of the problem, processors are assumed to be connected by a complete network were processors can simultaneously compute, send, and receive messages. On the completion of a task its results are immediately broadcast to all the successors as well as other publications incorporated some models of the interconnection but without introducing too many details, the issues that will be encountered here are as the execution of the tasks in a distributed system causes communication delays if the predecessor and the successor tasks are executed on different processors, communication delay depends on sender, receiver, and the location of processors and hierarchical structure. The parallel computing system consists of connected clusters of processors. Within the local cluster communication is faster [2]. Between the clusters communication is slower. The challenges are assigning the correct task to the correct processes in the distributed system and reduce the communication delay while doing the scheduling or load balancing.

III. Proposed Solution

We must remember that certain assumption are to be considered w. r. t the present paper, as the tasks were non preemptive in nature, static, processors are uniform, arbitrary interconnected network, the number of hop will give the path from one processor to another processor, to which we will be discussing shortly and the amount of available memory at each processor is sufficiently big to hold all the intermediate results and the received messages. Now Let us consider some of the existing algorithms, which will categorize the task into some levels.

Hu Algorithm [3] uses the concept of a task level. A task level is the length of the path from the task to the root of the in - tree. The task (s) with the highest level are selected to be executed on available processor (s) first.

Hence, it is highest level first (HLF) or level scheduling algorithm.

Algorithm HLFET (Highest Levels First with Estimated Times) [4] gives priority to the tasks with the highest level.

Analogously to the nonpreemptive, the algorithm of Muntz and Coffman [5]. The algorithm for these two problems is based on two concepts: task *level* and *processing capability*

Multiclass Fixed Partitioning (MFP) and Multiclass Adaptive Partitioning (MAP) [6] Here the tasks are divided into three classes of small, medium, and large tasks according to the amount of work on a certain number of processors. FIFO queues are maintained for waiting tasks of each class. A small task arriving to the system is allotted processors from all partitions, medium tasks from medium and large partitions, large task from large partitions only.

Reservation and Priority [7]. It combines reservation and priority based on aging with allocation algorithm. When a new task arrives in the system, a free sub mesh is searched for. If the allocation fails the algorithm tries to make a reservation for the task. If the reservation is successful, then the task joins reservation queue. Otherwise, it joins waiting queue. When some task departs from the system, the sub mesh it was using is not only assigned to the tasks that had reservation there, but also allocating other tasks from the reservation and waiting queues is attempted.

Linear Clustering (LC) [8] The LC algorithm greedily clusters the tasks constituting the longest path in the task graph.

The Algorithm by Guinand and Trystram [9], where the entire task graph is divided into the sub tress and the sub tree are assigned to the various processors.

We have many more algorithms which has been implemented and study of all the algorithms are beyond the scope of this paper, The conclusion that we can draw from the above algorithms are that every tasks has some priority over the other and unless some particular tasks has been completed, other can’t event start. So as a result the tasks are divided into three classes of level - 1, level - 2 and level - 3 tasks according to the order of the execution and the level - 1. High priority tasks are placed at level - 1, level - 2 task which can run only if the level - 1 task will be completed and the remaining (level - 3 tasks).

A Virtual Node Based Network Distance Prediction Mechanism [10], Based on partial measurement results, network coordinates system embeds the network distance space into a geometric space and assigns each node a coordinate. Network distance between any two nodes can then be calculated from their coordinates with a distance function. For a network with N nodes H = {H1, H2, …, HN}, suppose distance between any two nodes Hi and Hj is represented as Dij, then all pair - wise network distances compose an N×N non - negative distance matrix D, in which all values except Dii is larger than 0. The basic operation in network coordinate system is designing an M - dimension geometric space and embedding the N nodes into that space with the constraint that errors between computed distances and measured distances are minimized. to describe the hierarchical Internet structure with a single metric space such as Euclidean one, which loses lots of network features during space embedding and prevents the further improvement of distance prediction accuracy. In order to solve this problem, we design a virtual node based network distance prediction mechanism VNetPharos [10], which divides Internet into different prediction regions according to the hierarchical Internet structure, and uses virtual nodes to represent joints between different prediction regions.

Hidden Markov Model [11], An HMM is a double embedded stochastic process with hierarchy levels. It can be used to model much more complicated stochastic processes [12] as compared to a traditional Markov model. With the help of virtual node based network distance prediction mechanism we can have an array of distances between the node to any other nodes, from this we can create four clusters, Cluster 1: Contains all the local nodes (minimum distances nodes) of any cluster let us name it as A, Cluster 2: The nodes with are located near to cluster A (hop count will be average), Cluster 3:The node which are available far away from the cluster A (hop count will be more) and The node which are located at the dead end of the cluster A (Max. Hop count available). Based upon the task we can prioritize (We have already discussed above about prioritizing the tasks) the dependency mechanism or the tasks, now the tasks which is closely coupled will be scheduled or given (load balancing) to the local cluster A or minimum hop distance node and again if task are a bit less dependent or loosely coupled, then the task can be scheduled or load balancing can be done to the average distance nodes and the third and the last, if the tasks are not interdependent then the task can be done on the larger hop distance node/machines or the cluster 3 and the node which are located at the dead end of the cluster A (Max. Hop count available) are forced to be left free (ideal processors) to accommodate future burst arrivals of the tasks. By doing so we can solve the problem of delay which can be encountered because of the execution of the tasks in a distributed system, like if the predecessor and the successor tasks are executed on different processors on different machine having large communication distances. (Communication delay depends on sender, receiver, and the location of processors). The resultant model can assign the correct task to the correct processes in the distributed system and thus reduce the communication delay while doing scheduling or load balancing. The above logic can be placed at the virtual node which acts as an intermediate node, which is placed between the clusters.

IV. Conclusion

The tasks are divided into three classes of levels and these tasks according to the order of the execution has to be allocated the processors in the local cluster, the level - 2 task which can run only if the level - 1 task will be completed and has to be assigned to the second cluster and the remaining (level - 3 tasks) can be run on the third cluster and the fourth cluster processors are made free to accommodate future burst arrivals of the tasks. By doing so we may reduce the delay incurred because of the inter process communication and properly schedule and do load balancing of the tasks. The logic for division and calculating the distances of the hops can be placed in the virtual node separately. So that the other nodes apart from the virtual node will be totally dedicated for the processing of the tasks. The given solution can be worked for the interconnection topology and communication delay models where processors are assumed to be connected by a complete network, processors can simultaneously compute, send, and receive messages and on the completion of a task its results are immediately broadcast to all the successors. Here certain assumption are to be considered as the tasks are non preemptive in nature, static, processors are uniform and the amount of available memory at each processor is sufficiently big to hold all the intermediate results and the received messages.

References

[1] Saeed Rashtbar, Ayas Isazadeh, Leyli Mohammad Khanly, ” *Task Graph Scheduling in Multiprocessor Systems Using a Coarse Grained Genetic Algorithm*” 2nd International Conference on Computer Technology and Development (ICCTD 2010) IEEE 2010

[2] Johnatan E. Pecero, Pascal Bouvry, ” *An Improved Genetic Algorithm for Efficient Scheduling on Distributed Memory Parallel Systems”*

[3] 4]T. C. Hu. *Parallel sequencing and assembly line problems. Operations Research*, 9 (6):841 - 848, 1961

[4] 5]M. Al - Mouhamed and A. Al - Maasarani. *Performance evaluation of scheduling precedenceconstrained computations on message - passing systems.* IEEE Transactions on Parallel and Distributed Systems, 5 (12):1317 - 1322, 1994

[5] R. R. Muntz and E. G. Coffman Jr. *Preemptive scheduling of real - time tasks on multiprocessor systems. Journal of the ACM*, 17 (2):324 - 338, 1970

[6] V. K. Naik, S. K. Setia, and M. S. Squillante. *Processor allocation in multiprogrammed distributed - memory parallel computer systems.* Journal of Parallel and Distributed Computing, 47 (1):28 - 47, 1997

[7] D. D. Sharma and D. K. Pradhan. *Job scheduling in mesh multicomputers. IEEE Transactions on Parallel and Distributed Systems,* 9 (1):57 - 70, 1998.

[8] A. Gerasoulis and T. Yang. *On the granularity and clustering of directed acyclic task graphs. IEEE* Transactions on Parallel and Distributed Systems, 4 (6):686 - 701, 1993

[9] F. Guinand and D. Trystram. *“Scheduling UET trees with*

communication delays on two processors. ” RAIRO Operations Research, 34 (2):131 - 144, 2000

[10] Changyou Xing, Ming Chen*” A Virtual Node Based Network Distance Prediction Mechanism*” IEEE "GLOBECOM" 2008 proceedings

[11] Shailesh S. Dhok,, *”Credit Card Fraud Detection Using Hidden MarkovModel”*. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231 - 2307, Volume - 2, Issue - 1, March 2012

[12] Abhinav Srivastava, Amlan Kundu, Shamik Sural, Arun K. Majumdar, ” Credit Card Fraud Detection Using Hidden Markov Model” IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 5, NO. 1, JANUARY - MARCH 2008