AN ANALYTICAL MODEL FOR DYNAMIC LOAD BALANCING IN 2D MESH

Md Firoj Ali*

Rafiqul Zaman Khan*

ABSTRACT

An efficient load balancing algorithm may reduce the communication overheads among the nodes in a network .In this paper we propose an improvement over the dynamic load balancing in the 16 processor 2D mess. In decentralized approach both the dimension exchange method and the diffusion method are widely applied for the load balancing. But along with the dimension exchange method we use four nodes as leader nodes to reduce the communication over heads. We made an analytical model to show the improvement.

KEYWORDS

Load balancing, threshold load, communication overheads

* Department of Computer Science, Aligarh Muslim University, Aligarh, UP, India

1. Introduction

Parallel computer architectures comprise of several computational elements (processor) in parallel to solve a big scientific problem to achieve better performance which is not possible by a single processor. The problem must be broken down into small tasks. The tasks should be distributed in such a way that all the processing elements may have equal number of tasks (load) to finish the computation nearly about the same time by all the processing elements.

The distribution of loads to the processing elements is simply called the load balancing problem. In a system with multiple nodes there is a very high chance that some nodes will be idle while the other will be over loaded. The goal of the load balancing algorithms is to maintain the load to each processing element such that all the processing elements become neither overloaded nor idle that means each processing element ideally has equal load at any moment during execution. So the proper design of a load balancing algorithm may significantly improve the performance of the system [1, 3, 5, 7].

The load balancing policies may be static, dynamic, centralized or decentralized. In static load balancing policies, all information about the system is known before any execution. Once the processes are assigned, no change or reassignment is possible at the run time. Static policies are very simple, since they do not collect system state's change information, they provide small performance improvement. The dynamic (adaptive) policies use to collect the information about the system state's change regularly or periodically at fixed interval of time. There is a gradual significant performance improvement but very complex to design and implement and hence costly.

In centralized load balancing approach a single processor is responsible for collecting the global load information and it takes all the load balancing decision while in decentralized load, each processor discloses its load to the others and updates its own load information for balancing the load locally to achieve substantial global load balancing. Centralized systems impose lower overheads than decentralized one. But decentralized approach is more reliable than centralized approaches [1].

Transfer policy and location policy are the two important factors in dynamic load balancing algorithms. The transfer policy firstly decides whether a task will be executed locally or remotely.

Then the location policy will decides the location (processor) where a task, selected for remote execution, should be sent.

The load balancing decision will be taken either by heavily loaded node (sender initiated approach) or by lightly loaded node (receiver initiated approach) [3]. In sender initiated approach, the overloaded nodes will search for lightly loaded nodes to transfer their extra overloaded loads. Similarly, in receiver initiated approach, the lightly loaded nodes search for overloaded nodes from which load may be transferred.

2. OUR MODEL

Numerous papers have been published in dynamic load balancing in 2D mess [2, 4, 5, 6]. Mainly decentralized approach has been followed. Nearest neighbor algorithms are mainly applied to measure the performance improvement in this case. In dimension exchange method, any processor which invokes the load balancing operation balances its load successively with its neighbors. At each step, processor balances its workload with one of its neighbors, and uses the new result for the subsequent balancing. In this way all the processors may reach to the balanced state. In diffusion method, any processor which invokes a load balancing operation compares its workload with those of its nearest neighbors, and then gives away or takes in certain amount of workload with respect to each nearest neighbor.

In 16 processor 2D mess we have considered four nodes as leader nodes which have the degree four. Here P0, P1, P2, P3 are the leader nodes Fig 1 (b). We have also assumed sender initiated dimension exchange load balancing algorithm in which the members in a group apply this algorithm within the group only.

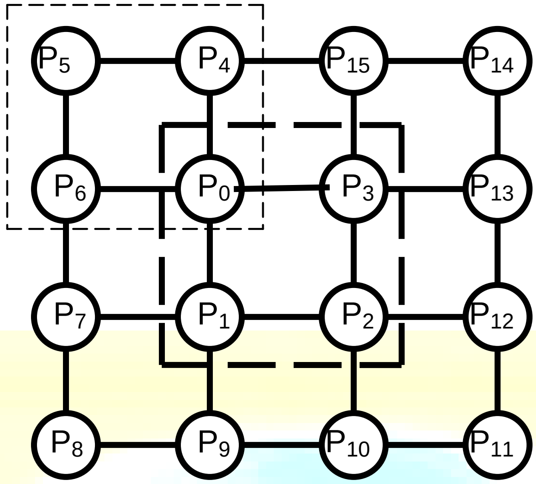

Figure1. Sixteen Processor 2D Mess (a) dashed rectangle represents a group of nodes (b) bold dashed rectangle represents the leader nodes of four groups.

2.1. Model Assumptions

The following assumptions are made for designing the model.

- Nodes P4, P5 and P6 are under the leadership of P0 Fig1 (a); Similarly, P7, P8 and P9 are under P1; P10, P11 and P12 are under P2 and P13, P14 and P15 are under P3.
- The IDs of the nodes are registered to the corresponding leader node.
- The nodes under a leader node will only participate in load balancing within the concerned group itself including the leader node.
- The leader nodes only responsible for taking decision for balancing node between the groups.
- At a given instant of time t each node has a certain load say W_i (t). So total load in the whole system can be calculated as

$$L = \sum_{i=1}^{N} Wi\,(t)$$

Now the average load per node can be calculated as:

$$\overline{W}_i(t) = L/N$$

where N is the total number of nodes.

- So the aim of the load balancing algorithm is to maintain the average load at each node.
- If L mod N is not zero, R number of tasks must have to be distributed randomly to the R number of nodes each where R<N. So, the load at R number of nodes will be one more than the average load which is considered to be equal to the average load.
- Consider a threshold load TL which is defined as an amount of load to a processor such that when the load to the processor is less than the TL, the load can be accepted by that node or when the load will be greater than TL, the load can be transferred from that node. We consider threshold load at any node is equal to the average load at that moment that is

$$TL = \overline{W}_i(t)$$

- Calculate the load index as

$$Li = TL - W_i\ (t)$$

Three conditions may arise due to the value of L_i at any node in any instant t.

After the load balancing within the group itself, the group either be overloaded, under loaded or moderate loaded.

For L_i = TL, no load will be exchanged which means that the corresponding group is moderate loaded. For L_i < TL, the load can be accepted which means the corresponding group is under loaded. For L_i > TL, the load can be transferred which means that the corresponding group is over loaded.

3. ANALYSIS OF THE MODEL

The group leader will collect the load information from its members. In sender initiated approach, the over loaded group leader will search the under loaded group leader. Once the over loaded group leader determines a under loaded group leader then over loaded group leader will collect the ids of the under loaded nodes and it will direct the over loaded nodes to follow the shortest path algorithm to transfer the load to the under loaded node. Suppose in Fig 1, P_5 is over loaded under P_0 and P_9 is under loaded under P_1 then after exchanging information between P_0 and P_1, P_0

will direct P_5 to follow shortest path to transfer the extra load to P_9 . The following improvements can be done by the above proposed model.

- A group must be either over loaded or under loaded or moderate loaded.
- The load transfer between the groups would be minimized due to the load balancing within the nodes in a group.
- The overloaded node will follow the shortest path to deliver the excess load to the under loaded node.
- Considerable reduce of overheads due to the follow of shortest path

4. Conclusions

This approach can be considered as a simpler and improved version of previously known methods for getting the efficient load balancing model. In this model four nodes are given some leadership power for balancing load between the groups. Most important thing is that a group must be one of the tree situations either in over loaded, under loaded or moderate loaded. So it would be very efficient approach because the group would easily be categorized as over loaded, under loaded or moderate loaded which means that a group can not be under loaded or overloaded at the same time. Another benefit is that the over loaded node follow the shortest path algorithm to transfer the extra load to a under loaded node which would produce very low communication cost.

References

[1] Antonis K., Garofalakis J., Mourtos I. and spirakis P. “A Hierarchical Adaptive Distributed Algorithm for Load Balancing”. Journal of Parallel and Distributed Computing, Elsevier Inc. 2003.

[2] Berenbrink P, Friedetzky T and Hu Z. “A New Analytical Method for Parallel, Diffusion-type Load Balancing”. IEEE, 2006

[3] Dandamudi S.P. “Sensitive Evaluation of Dynamic Load Sharing in Distributed System”. IEEE, 1998.

[4] Marc H. WL and Anthony P. R. "Strategies for Dynamic Load Balancing on Highly Parallel Computers". IEEE, 1993.

[5] Xu C. and Lau F.C.M. "Load Balancing in Parallel Computers: Theory and Practice". Kluwer Academic Press, 1997.

[6] Xu CZ, Monien B, Luling R, Lau FCM. "An Analytical Comparison of Nearest Neighbor Algorithms for Load Balancing in Parallel Computers". IEEE 1995.

[7] Zhong T, Birdwell J. D, Chiasson J, Abdullah C. T and hayat M.M. "A Time delay Model for load Balancing with Processor resource Constraints". 43rd IEEE Conference, 2004.