ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

Performance Evaluation of Diffusion Method for load balancing in Distributed Environment

Javed Hussain[1], Durgesh Kumar Mishra[2]

Computer Centre, Vikram University, Ujjain, India and Research Scholar, Mewar University, Chittorgarh, India[1]

Department of Computer Science and Engineering, Shri Aurobindo Institute of Technology, Indore, India[2]

Abstract —In this paper, we study about diffusion load balancing algorithms with their implementation. Our analysis is based on different topologies in simulated environment. The purpose of load balancing algorithm is to distribute the excess load of processor to lightly loaded processor. The objective of this analysis is to find out best stable network amongst chain, 2D and 3D Mesh networks in this diffusion method approach. We analyzed load balancing algorithm based on the diffusion method which tends to balance a processor and its neighbors to achieve stability in a network. Simulation result has shown better performed network among chain, 2D, 3D networks.

Index term —Diffusion, Load Balancing, Chain, 2D, 3D Network, Topology, Algorithm.

I. INTRODUCTION

Low cost processor and interconnect technology has increase interest in locally distributed computers. Distribution is termed as load balancing. In distribution environment balancing of work load among processor is big issue. Processing power does not mean only the processing speed of processor; it includes the overall configuration of node. Two major categories of load balancing have been – static load balancing and dynamic load balancing. Static load balancing uses only information about the average system behaviour at the initial stage, i.e., done at compile time while dynamic load balancing uses current load information for allocation of tasks [13]. In case of dynamic load balancing, initiation strategies exist within rebalancing criterion. At some point of time, initiation is required to check load status of workstations and rebalance the system, if necessary. There are mainly two types of strategies, which several researchers used – period (time slice) - initiated and event-initiated. In period initiated strategy, the system is monitored regularly at fixed time intervals [8]. If nodes are found overloaded then loads are shifted to under-loaded or idle nodes to balance the system. Event initiated strategy may be either receiver-initiated or sender-initiated. In receiver-initiated strategy, an under-loaded or idle node generates a request and search for overloaded nodes to receive some load. Inversely, in sender-initiated strategy, an overloaded node generates a request to shift some of its load to an under-loaded or idle node. Overall Load Balancing based on the idea of migration of excess load from heavily loaded node to lightly loaded ones. It is require in the efficient system that each processor has to reach on average load that is each processor should be instructed to give or take some amount of load with their directly connected neighbour processors. The actual workload migration is follow some predefined instruction to achieve the load balancing in the system. Here we analyse load balancing in a network through diffusion method. This approach is distributed, iterative and parameterized by diffusion parameter (alpha). The execution of diffusion approach is divided into sequence of iterative steps.

II. RELATED LITERATRE AND OBJECTIVE

The load balancing process can be divided in four phases [5]. These four phases are the following: a) *processor load evaluation:* A load value is estimated for each processor in the system in order to use the concept of load unit in the second and third phases. b) *Load balancing profitability determination:* This phase evaluates the imbalance load factor and decides whether or not load balancing is profitable at that time. c) *Task migration strategy:* Sources and destinations for load migration are determined in this phase. d) *Task selection strategy:* The source processors select the most suitable tasks for efficient and effective load balancing and send them to the appropriate destinations. Centralized schemes [4] tend to be more accurate because they use global domains where the load information of all the processors in the system is collected. Diffusive load balancing technique[7] were compare with three other techniques, static load balancing, round robin balancing and shortest queue load balancing. In each simulation run, the same behaviour of clients is used in order to compare responses of the load balancing techniques. The iterative methods used for the solution of diffusion problems. Using Diffusive Load Balancing technique, a client requests object creation by sending a message to a round robin router, which forwards the message to one of the servers. After a method

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

on the object is evoked for the first time, a search for the granting server takes place in the *G (V, E)* representing the network topology. Request is moved from one server to its neighbouring server, if the difference in workload between the server and its neighbours is higher than the propagation differential, a threshold value, which determines whether to forward a server's request to another server, or not. The workload of a server is approximated using the number of client tasks in the ready queue. A simulation result [7] shows that diffusive load balancing is better than round robin and static load balancing in a dynamic environment. However, in a static environment where the number of clients can be known in advance, static load balancing does very well due to its low overhead. Static load balancing is a good choice in such an environment since it is relatively simple to implement. [9] Demonstrate the strength of the technique by analysing diffusion continuous and discrete load balancing algorithms for several network models. A new dynamic load balancing algorithm is proposed [15] based on diffusion approach for homogeneous systems where the processing capacities of all nodes in the system are equal. The proposed algorithm works iteratively to balance the load among the nodes in a system. The dynamic distributed diffusion algorithm has been developed for coarse and large granularity applications, where the load shall be treated as an Integer quantity. The functioning of the proposed algorithm is demonstrated by using a random graph and simulation has shown the proposed algorithm performs better in terms of time taken to balance the load, minimizing the load variance among the nodes and maximizing the throughput. In this paper we will compare the chain, 2D, 3D topology for local area network environment using simulator for diffusion method.

III. THEORETICAL DESCRIPTION AND METHODOLOGY

This paper uses a simulation model to show the load balancing method and algorithm used. Here three different topology is used (i) Chain (ii) Mesh 2D (iii) Mesh 3D for Diffusive load balancing method.

The diffusion method:

Description of the diffusion method and algorithms given by Lau are used in the simulator[16]. The diffusion method is an iterative algorithm. In every step, a fixed fraction of the load difference between two neighbouring processors is exchanged. When such local operations are used, the load distribution should converge to the globally equal, flat load distribution. The efficiency of the diffusion method depends on a diffusion parameter α which determines the size of the transferred load fraction.

α- diffusion parameter, N- number of processor, *i* - processor number, *i*=1...*N*, *R* - vector of current load processors, R= [L_1, L_2 , L_N], L_i - current load of processor i, $\Delta(i)$ - a set of directly neighbouring processors, $\delta_i(j)$- a load transferred from processor i to j.

In every step, a load $\delta_i(j)$ is transferred from processor *i* to every neighbour $j \epsilon \Delta(i)$.

$$\delta_i(j) = \alpha[L_i - L_j], \text{ with } \alpha \epsilon (0,1) \quad - (1)$$

For $\delta_i(j) < 0$, the load is transferred in the inverse direction i.e. from processor j to i.

Every change of state of the processor load L_i by synchronous load balancing can be described by the following transition equation.

$$L_i(t+1) = L_i(t) - \sum_{j \epsilon \Delta(i)} \delta_i(j) \quad - (2)$$

Where *t* - the current step in time, t +1 - the next time moment.

IV. RESULT

Assumption:

Above approaches have strict locality of the communication and the control. Assuming that -tasks are indivisible, identical and task numbers are integers. Due to the assumption load balancing process can be simulate for many tasks. No. of Processor = 12, Initial load on at every processor is fixed for each values of alpha.

A. *Topology = Chain*

TABLE I

α	Time Sec.)
0.01	20
0.10	12
0.20	10
0.30	18
0.40	18
0.50	25
0.60	30
0.70	34
0.80	39
0.90	50
0.99	55

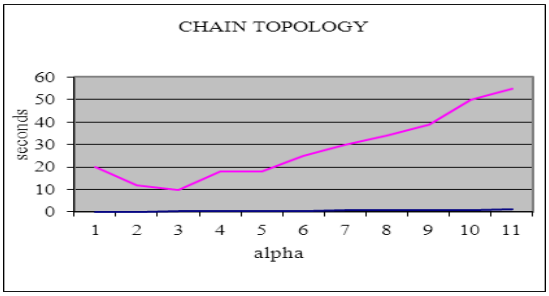

Fig. 1

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

B. Topology = 2D Mesh

TABLE II

α	Time(Sec.)
0.01	67
0.10	40
0.20	40
0.30	50
0.40	35
0.50	20
0.60	14
0.70	10
0.80	8
0.90	6
0.99	5

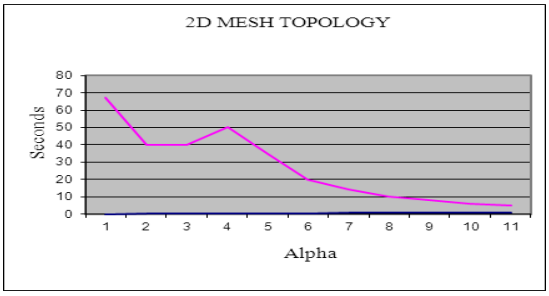

Fig. 2

C. Topology = 3D Mesh

TABLE III

α	Time(Sec.)
0.01	21
0.10	14
0.20	12
0.30	11
0.40	9
0.50	6
0.60	5
0.70	4
0.80	4
0.90	4
0.99	3

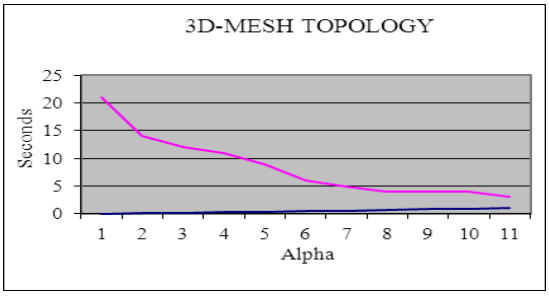

Fig. 3

Analysis:

Fig.1, depicts that as we increases alpha to some extent (alpha=0.3), time to reach at stability is decreases but beyond alpha=0.3 stability time of network is increases. In figure 2, for 2D Mesh network when alpha is between 0.2 to 0.3 network will take equal period of time to become stable and alpha goes from 0.3 to 0.4 stability time increases but beyond alpha=0.4 time for stability is decreases. In figure 3, for 3D Mesh network when we increases alpha, time to reach at stability also decreases almost gradually.

V. CONCLUSION

We have investigated from simulation results that chain topology take more time to become stable as we increases alpha. 2D and 3D mesh network took lesser time to reach in stable state but 3D mesh network is more efficient then 2D mesh network. From simulated result we conclude that 3D mesh topology is better. In future we will analyse nearest neighbour algorithm for similar condition and topologies.

REFERENCES

1. Chou, T.C.K. and Abraham, J.A., *Load balancing in distributed systems.* IEEE Trans. Software Engrg., 8, 1982, pp. 401-412.
2. T.L. Casavant and J.G. Kuhl, *A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems*, IEEE Transactions on Software Engineering, vol 14 No. 2, February 1988, pp 141- 154.
3. G. Cybenko, Load balancing for distributed memory multiprocessors, Journal of Parallel and Distributed Computing, 7:279-301.m1989
4. Hwa-Chun Lin and C.S. Raghavendra, *A Dynamic Load Balancing Policy With a Central Job Dispatcher (LBC)*, IEEE Trans. Software Engrg., vol. 18, No 2, February 1992, pp. 148-158
5. Marc H. Willebeek-LeMair, Anthony P. Reeves, *Strategies for Dynamic Load Balancing on Highly Parallel Computers*, IEEE Transaction on Parallel and Distributed Systems, vol 4, No 9, September 1993, pp.979-1993.
6. Y.F. Hu and R J Blake, An Improved Diffusion Algorithm For Dynamic Load Balancing, Parallel Computing, Elsevier, Vol. 25, Issue 4, April 1999, pp. 417-444.
7. Milan E. Soklic, Simulation of Load Balancing algorithm, A Comparative Study, ACM – SIGCSE Bulletin, December, 2002.
8. Helen D. Karatza, "A Comparison of Load Sharing and Job Scheduling in an Network of Workstations," International Journal of Simulation, Vol. 4, No. 3-4, pp 84-87, 2003.
9. PetraBerenbrink, Tom Friedetzky, Russel Martin, Dynamic Diffusion Load Balancing, Research Report CS RR-402, Department of Computer Science, University of Warwick, July 2004.
10. J. Ghanem, C. T. Abdallah, M. Hayat, S. Dhakal, J.D.Birdwell, J.Chiasson, and Z. Tang, Implementation of Load Balancing Algorithms over a Local Area Network and the Internet, Decision and Control, 2004. CDC. 43rd IEEE Conference on Volume : 4.
11. J.M. Bahi, Contassot-vivier S., R. Couturier, and F. Vernier. A decentralized convergence detection algorithm for Asynchronous parallel iterative algorithm. IEEE Trans. On Para and Dist. Sys., 16(1):4 13, 2005.
12. Emmanuel Jeannot, Flavien Vernier, A Practical Approach of Diffusion Load Balancing Algorithms, INRIA , RR No. 5875, Mar. 2006.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

13. Jawinder Pal Singh, "Dynamic Load Balancing for Cluster Computing," Retrieved from: http://wwwzenger.informatik.tu-uenchen.de/lehre/seminare/clust_comp
14. PetraBerenbrink, Tom Friedetzky, Zengjian Hu, A new analytical method for parallel, diffusion-type load balancing, Journal of Parallel and distributed Computing, 69(2009)54-61.
15. P. Neelakantan, Load Balancing in Distributed Systems using Diffusion Technique, International Journal of Computer Applications (0975 -8887) Volume 39 –No.4, February 2012.
16. [16] Lau, Francis C. M.; Xu, C. Z., Load balancing with the diffusion method on K-ary Ncubes and variants, http://hdl.handle.net/10722/54877,March 1993.