DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5831

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A SIMULATION MODEL FOR THE PERFORMANCE EVALUATION OF HYBRID INTERCONNECTION ARCHITECTURES

Savita Gautam
University Women's Polytechnic
Aligarh Muslim University
Aligarh, India

Nitin Bansal
Department of Computer Science,
Doon Valley Institute of Engineering and Technology
Karnal, India

Abdus Samad
University Women's Polytechnic
Aligarh Muslim University
Aligarh, India

Abstract: Efficient scheduling of tasks in multiprocessor interconnection networks is of primary importance for parallel execution. In this paper a new scheduling algorithm that schedules the tasks on different networks of processors is proposed and evaluated. The objective is the effective utilization of resource (processors) to execute tasks of an application. Processor networks have been used in the form of different topologies. We consider a new class of architecture known as linearly extensible multiprocessor networks which are best suited with lesser number of processors. Performance of different multiprocessor networks with the proposed algorithm as well as with other standard scheduling algorithms is evaluated in terms of performance parameters namely load balance error and execution time. Simulation results show that the linear architectures achieve better performance with the proposed scheduling scheme as compared to other considered scheduling algorithms.

Keywords: Linear architecture, scheduling algorithm, load imbalance, execution time, level wise scheduling, tasks.

I. INTRODUCTION

The problem of scheduling different tasks in multiprocessor systems is a well advanced research area. Network of processors has emerged as a promising solution for high-performance execution of programs. The selection of topology in such systems has a vital role in the overall performance of parallel systems.

Cube-based topologies have gained so much attention due to their coveted properties. Variety of hypercube network has been available as topologies to map parallel applications [1]. In order to improve the performance of these systems further architectural innovations are being carried out to make the network simple or less complex. One such class of network is linearly extensible network which employ the desirable properties of cube-based architectures with linear extension of processors [2[, [3]. These architectures remove the drawbacks of cube-based topologies such as complex routing model, under utilization of nodes and reliability. Although a lot of work has been carried out on regular topologies such as arrays, tree, meshes or hypercube still there is scope to devise task scheduling for hybrid type of architectures. The scheduling of tasks on irregular architectures has not been rigorously studied. The evaluation of these architectures with appropriate scheduling algorithm is the topic of our interest in this paper.

Effective evaluation techniques are required to compare performance of various architectures. The main contribution of this paper can be summarized as:

- Various networks which are based on linearly extensible architectures or hybrid architecture are used for analysis and performance evaluation. Therefore, the results could be useful for the network designer.
- A novel approach for scheduling the tasks is presented and applied on the considered topologies. This may lead us to have a better approach for efficient utilization of nodes in the networks.
- The results obtained from simulation model may further be utilized for detailed study of linearly extensible multiprocessor networks with different scheduling algorithms.

The complete paper is organized in six sections. Section 1 is introduction. Section 2 describes the related work and discusses the scope of present work. In section 3 the existing algorithm as well as proposed algorithm for solving the problem is described. Simulation results are evaluated and discussed in section 4. Curves are drawn to carry out comparative study of the results obtained by implementing the proposed algorithm. Section 5 concludes the paper followed by a list of references given in the last section.

II. RELATED WORK

The task of mapping application to parallel systems and balancing the load on the processing elements (nodes) is considered as the research problem in itself. It is not easy to design an algorithm for a system when processing elements share resources and work cooperatively. Chained Cubic Tree (CCT) is considered as hybrid architecture [4]. The author suggested a task scheduling for CCT network and claimed optimal or near optimal solution in scheduling and executing the tasks. Similarly, a multi source divisible load scheduling is proposed for arbitrary network that consists of heterogeneous processing nodes [5]. The algorithm utilizes the concept of nearest node means that provide small communication delay. However, this concept could not be applied in networks where nodes are further and having largest communication. A similar approach is discussed in Minimum Distance Scheduling (MDS) in which the concept of minimum distance is applied while implementing the algorithm [6].

To cover the further node in the network another scheduling algorithm known as Two Round Scheduling (TRS) has been reported which were originally designed for both cube-based systems as well as for linear systems [7]. The TRS scheme is an impressive technique in terms of load balancing error until the nodes have regular or symmetrical connectivity. In many topologies like tree type architectures nodes may be present at remote areas, in that case TRS may result large execution time.

Genetic algorithms have also been used in the recent past for solving task scheduling problem; however, they are based on chromosomal representation [8], [9], [10]. The authors in [9] utilize single fitness function to improve the performance. The given algorithm has been implemented for heterogeneous networks. Multistage interconnection network is another class of networks that has been studied for both evaluations of architectures as well as for handling of task scheduling [11]. Simulator is the key to evaluate the simulation performance of an interconnection network. Different simulators like Functional Simulator of Interconnection Network (FSIN) and a simulation framework known as INSEE have been developed and further extended for specific functionality such as evaluation of novel topologies and parallel algorithms [12], [13], [14], [15].

Real implementation suffers from high cost as well as greater overhead. The scope of this paper is to propose a simulation model by analyzing the performance of existing scheduling algorithm as well as proposed algorithm on different types of interconnection networks. The proposed simulation model consists of task assignment through scheduler on a set of interconnected nodes. We have considered three categories of architectures namely tree type, linearly extensible type and hybrid type of networks.

III. PROPOSED ALGORITHM

In order to estimate the performance of the scheduler we considered different networks with same number of nodes for each class of processor network. The proposed algorithm is designed based on some modification in Two Round Scheduling (TRS) scheme. Therefore, the concept behind TRS is discussed first before presenting the proposed design.

TRS schemes

The two round scheduling is actually based on the principle of Minimum Distance Scheduling (MDS) with an enhancement of next level of connectivity. Instead of migrating load to only directly connected processors, the nodes connected through one intermediate node are also considered for task migration. During the load balancing operation the donors distribute their excess load to acceptors nodes at second stage links available in the network and hence improved the load balancing error. However, this process though improves the load imbalance but at the cost of greater execution time. Any cutting time technique will enhance the performance of the algorithm.

To address this issue the criterion of selecting the node for migration is reviewed and a new algorithm is proposed that eliminates the problem of over utilization and under utilization of nodes which is due to the limitation of connectivity.

Proposed Algorithm

The proposed scheduler identifies different set of nodes to migrate tasks at particular level. Tasks migration is restricted to the identified set until all the nodes are not balanced. Since a particular node may have multiple communication paths with other nodes, a fundamental issue is which path should be selected for optimal performance. The proposed scheduling creates a set of nodes before establishing communication. The migration of tasks will take place only to the selected nodes in a set. This basically gives the answer 'from which route' the communication is. The complete algorithm is described in Fig. 1.

1. Consider the set of processors
2. Load is generated at each processor.
3. For main processor find the set of processor.
4. Calculate the total load on all processors.
5. Find the IdealLoad that is desired on each processor.
6. Shift the load to other processor such that load on main processor should be equal to IdealLoad.
7. If Load on main processor is greater than IdealLoad transfer the load from main processor to other connected processors.
8. If Load on main processor is less than IdealLoad than transfer the load from connected processors to main processor.
9. Obtain the set of second level processors for each of the processor that are connected to main processor.
10. Perform the migration of load to these second level processors.
11. Calculate load imbalance error in terms of load imbalance factor.
12. Calculate Execution Time to balance the network.

Figure 1. Proposed Algorithm

To evaluate the performance of the proposed system through simulation, we define several performance parameters.

- The load imbalance error is calculated as Load Imbalance Factor (LIF) which defines the deviation of load after a load balancing action.
- Ideal Load (IL) is the desirable load which is supposed to be available at each node after the algorithm is applied on a network.
- Execution time is equal to the time taken by the scheduler to complete task scheduling and balancing.

IV. SIMULATOIN RESULTS

In this section, we discuss the performance of the proposed algorithm. The simulation results consist of generating task structure on different networks having equal number of nodes. The tests have been run for same pattern of task structures for measuring LIF (%) and execution time (msec.). The latest C++ standards have allowed the measurement of execution time up to nanoseconds, therefore, the results could be compared fairly. The results obtained are compared with the TRS scheme.

In the proposed model three distinct types of architectures tree, linearly extensible networks and hybrid networks are considered for simulation purpose. In tree type the considered topologies are six nodes Binary Tree (BTree) and Ternary Tree (TTree) shown in Fig. 2 and Fig. 3.

In linear type of architectures we considered Linearly Extensible Tree (LET), Linearly Extensible Triangle (LEΔ) [16] and in hybrid architectures Linearly Extensible Cube (LEC) [17] and Linear Cross Cube (LCQ) [18] with six nodes are taken into consideration. Each network consists of six nodes whose connectivity is incorporated through adjacency matrix. Moreover, the number of tasks arrived at a particular point of time is specified.

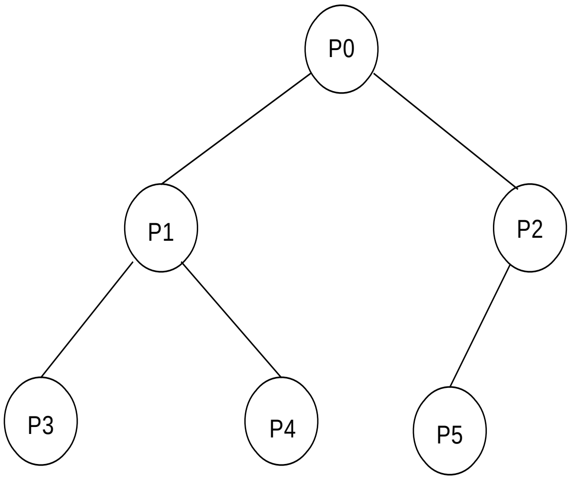

Figure 2. Six nodes BTree network.

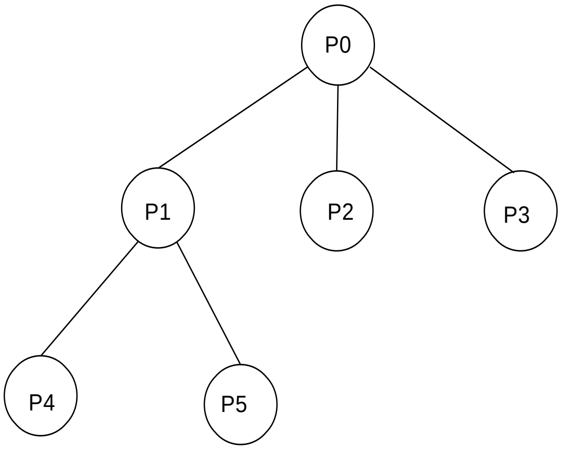

Figure 3. Six nodes TTree network.

Performance of proposed algorithm when evaluated on tree type networks i.e. Binary Tree (BTree) and Ternary Tree (TTree) there is improvement in LIF both at the higher value as well as in the pattern of reduction in values. However, the effect is not that much significant as in LEC, LCQ, LET and LEΔ networks. This is due to the selection of nodes at different levels (Fig. 4 and Fig. 5). It might be possible that set of processors of a particular root are selected for load balancing at a particular level. This possibility is rare in cube-based architectures as there is no concept of child nodes in such networks.

The same algorithm when applied on other two classes of networks produces similar results. The linearly extensible networks possess uniform extension of nodes at higher levels of networks and therefore able to manage the load distribution efficiently with the proposed algorithm.

Similarly, hybrid networks also have similar property as well as having good connectivity and hence producing even better results than linear type architectures. The comparative study of the simulation results obtained for different interconnection networks with both the algorithms is carried out in terms of maximum and minimum values of LIF and demonstrated in Table 1.

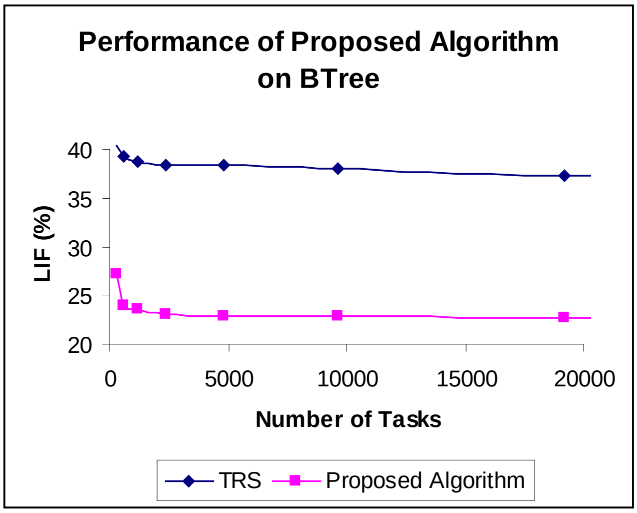

Figure 4. Performance of proposed algorithm on BTree network.

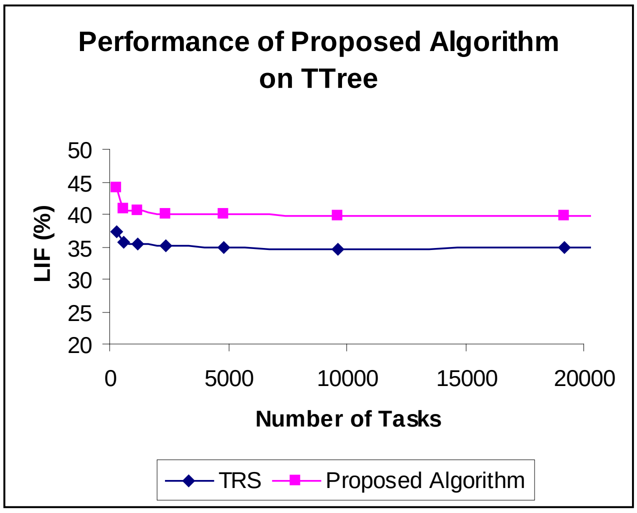

Figure 5. Performance of proposed algorithm on TTree network.

Table I. Comparative Study of Simulation Results

Scheduling Algorithms	BTree		TTree		LET		LEΔ		LEC		LCQ	
	LIF (%)		*LIF (%)*		*LIF (%)*		*LIF (%)*		*LIF (%)*		*LIF (%)*	
	Max.	***Min.***	***Max.***	***Min.***	***Max.***	***Min.***	***Max.***	***Min.***	***Max.***	***Min.***	***Max.***	***Min.***

TRS	50.40	43.39	49.80	47.62	52.50	48.79	67.80	65.33	57.77	54.25	69.80	54.33
Proposed	40.29	37.26	37.38	34.66	47.88	46.41	57.83	56.00	41.66	38.60	57.80	48.65

The pattern of decreasing the values of LIF at higher stages of task structures is also shown in Fig. 5. It is clear from the figure that the proposed algorithm when implemented on these networks show similar pattern of balancing the nodes. The value of LIF is significantly reduced at higher stages of tasks and therefore the schedular gains a significant speed up.

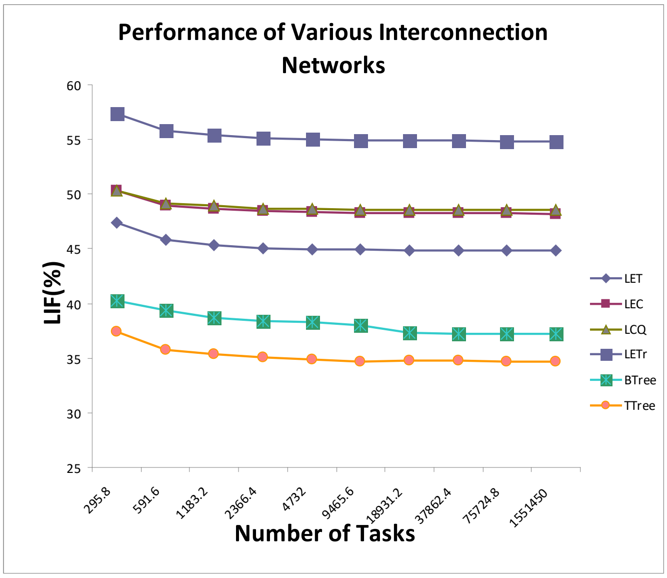

Figure 6. Performance of proposed algorithm on various networks.

Comparing alternative algorithm solution requires the time complexity with optimal solution. The second performance parameter is the the execution time which is the total time to perform a balancing action. We observed that the proposed algorithm takes approximetly equal time for different types of architectures particularly with lesser number of processors (6-10 processors). This trend is depicted in Fig. 7.

The relationship between execution time and number of iterations also depends upon the number of tasks available to schedule. Since the algorithm follows a very simple methodology and handles the migration dynamically the affect of greater number of tasks could be minimized.

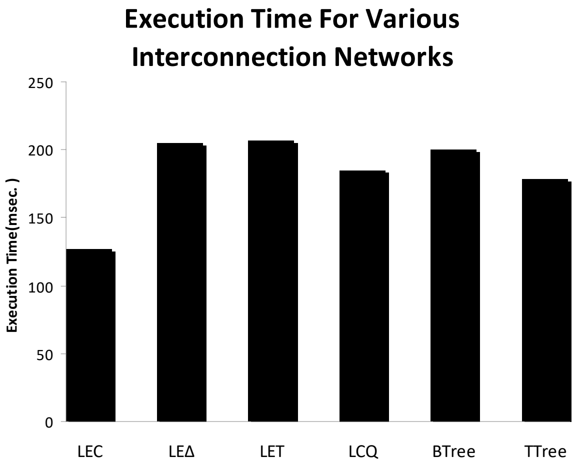

Figure 7. Performance of proposed algorithm on various networks.

V. CONCLUSION

The presented work in this paper is based on the implementation of a new dynamic scheduling algorithm on three different classes of interconnection networks namely tree networks, linearly extensible networks and hybrid networks. The simulation results show that the proposed algorithm improves schedulability by reducing load balancing error and maintaining execution time. The graphical representation demonstrates that the value of LIF in all cases is lesser than that obtained by applying standard scheduling algorithm and further reduced when greater numbers of tasks are applied on the systems.

The proposed technique is especially beneficial to schedule the load on hybrid type of interconnection networks. We have evaluated and presented the results to monitor the performance of these networks with six nodes, however, the proposed technique is equally good for networks having more than six processors.

VI. REFERENCES

[1] A. Samad, J. Siddiqui and Z.A. Khan, "Properties and Performance of Cube-based Multi-processor Architectures", Int. journal of applied evolutionary computation. (IJAEC), vol. 7, no.1, 2016, pp. 67-82.

[2] Z. A. Khan, J. Siddiqui, A. Samad, "Performance Analysis of Massively Parallel Architectures," BVICAM's International Journal of Information technology (BIJIT), ISSN/ISBN NO: 0973-5658, New Delhi, vol. 5. no.1, 2013, pp. 563-568.

[3] B. Parhami, "Challenges in interconnection network design in the era of multiprocessor and massively parallel Microchips," Proc. Int'l Conf. .in Computing, pp. 241-246, June 2000.

[4] B. A. Mahafzah and B. A. Jaradat, "The hybrid dynamic parallel scheduling algorithm for load balancing on Chained-Cubic Tree," Jornal of Supercomputer, vol. 52, 2010, pp. 224-252.

[5] J. Jia, B. Veeravalli and J. Weissman, "Scheduling multisource divisible loads an Arbitray Networks," IEEE Trans. on Parallel and Distributed Systems, vol. 21, no. 4, 2010, pp. 520-530.

[6] Samad A, Rafiq M. Q. and Farooq O. “Dynamic Scheduling Scheme for Linearly Extensible Multiprocessor Systems” presentation in the 2013 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’13), July 22nd – 25th, 2013, Las Vegas, USA.

[7] A. Samad, M. Q. Rafiq, O. Farooq, “Two Round Scheduling (TRS) Scheme for Linearly Extensible Multiprocessor Systems,” Int. J. of Computer Applications, vol. 38, no. 10, 2012, pp. 34-40.

[8] F. A. Omara and M. M. Arafa, “Genetic algorithm for task scheduling problem,” Int. Journal of Parallel and Distributed Computing, vol. 9, 2011, pp. 241-257.

[9] S. K. Nayak, S. K. Padhy and S. P. Panigrahi, “A novel algorithm for dynamic task scheduling,” Future generation Computer System, vol. 28, 2012, pp. 709-717.

[10] D. M. Abdelkader and F. Omara, “Dynamic task scheduling algorithm with load balancing,” Egyptian Informatics Journal, vol. 13, 2012, pp. 135-145.

[11] J. Garofalakis and E. Stergiou, “ An analytical model for the performance evaluation of multistage interconnection networks with two class priorities,” Future Generation Computer Systems, vol. 29, 2013, pp. 114-129.

[12] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner and Simics, “A full system simulation platform,” IEEE Computer vol. 35, no. 2, 2002, pp. 50–58.

[13] J. Navaridas, J. M. Alonso, J. A. Pascual and J. Ridruejo, “Simulating and evaluating interconnection networks with INSEE,” Simulation Modelling Practice and Theory, vol. 9, 2011, pp. 494-515.

[14] J. Miguel-Alonso, J. Navaridas and F.J. Ridruejo, “Interconnection network simulation using traces of MPI applications,” International Journal of Parallel Programming, vol. 37, no. 2, 2009), pp.153–174.

[15] M. Mirza-Aghatabar, S. Koohi, S. Hessabi and M. Pedram, “An empirical investigation of mesh and torus NoC topologies under different routing algorithms and traffic models,” in: Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, Lübeck, Germany, August 29–31, 2007, pp. 19–26.

[16] Manullah, “A Δ-Based Linearly Extensible Multiprocessor Network,” Int. Journal of Computer Science and Information Technologies, vol. 4, no. 5, 2013, pp. 700-707.

[17] A. Samad, J. Siddiqui and Z.A. Khan, “Properties and Performance of Cube-based Multi-processor Architectures”, Int. journal of applied evolutionary computation. (IJAEC), vol. 7, no.1, 2016, pp. 67-82.

[18] Z. A. Khan, J. Siddiqui, A. Samad, “Linear Crossed Cube (LCQ): A new Interconnection Network Topology for Massively Parallel Architectures,” Int. Journal of Computer Network and Information Science (IJCNIS), vol. 7, no. 3, 2015, pp. 18-25.