



Comparative Study on Load Balancing Algorithm for Multiprocessor Interconnection Networks

Nazir Ahmad¹, Hatim M. Elhassan Ibrahim Dafallaa², Mohammed Burhanur Rehman³, Iqrar Ahmad⁴

¹King khalid university, Kingdom of Saudi Arabia, naanhma@kku.edu.sa

²King khalid university, Kingdom of Saudi Arabia, hibrahem@kku.edu.sa

³King khalid university, Kingdom of Saudi Arabia, malrhman@kku.edu.sa

⁴King khalid university, Kingdom of Saudi Arabia, iaazad@kku.edu.sa

ABSTRACT

To achieve a high performance network and effective sharing of computing resources, it is important to distribute the load evenly among different nodes. Therefore, efficient scheduling strategy is required to map the load onto the set of nodes. The main problem faced in the design of scheduling algorithm is the lack of information about the network load distribution and hence the task execution time estimation. Distributed System is important to distributing the work load on processors. Distributed system can be viewed as a collection of computing and communication resources shared by active users. When the demand for computing power increases, Load Balancing problem become important. The purpose of load balancing is to improve the performance of system through an application load. Load balancing is the method of distributing the load among the various nodes of an interconnection network to improve both imbalance load and job execution time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. In this paper, study of existing load balancing algorithm is brief discussed along with their objective, method, advantages and future work. In simulation study, our previous works DLBS, ITSLB and LBSM is compared on FCC, MC and X-Torus interconnection networks and experimental results have been reported accordingly.

Key words: Interconnection Network, Heterogeneous System, Level of load Balance, LIF,

1. INTRODUCTION

Distributed System (DS) is mainly used because of its capability to demand for high computation power which solved complex problem, desirability of resource sharing and thus it increases efficiency, rapid growing of network infrastructure and gets immediate feedback too. Any type of DS may have a general goal like to solve a large complex

computational issue, the user receives the group of autonomous processors as a unit alternatively, each computer may have its own user with individual needs etc. The principle of the DS is to manage the use of shared resources (i.e., memory) or provide the communication services for all users. The distributed system can also be defined as the set of processors connected by the use of a high speed network for completing the required task. A distributed system is described as collection of either homogeneous system or heterogeneous system [1]. Homogeneous Distributed System (HoDS) is that distributed system where collections of identical processors are linked to a high speed network for completing some tasks. To make a HoDS the following criteria must be fulfilled such as the hardware of each processor is guaranteed the storage illustration and the outputs for operations on floating point numbers (FPN). The benefit of HoDS is that the communication and computation cost are constant in any type of task scheduling algorithm. The use of maximum utilization of processors for all presented resources is to require dividing the all tasks equally in HoDS. Heterogeneous distributed system (HeDS) is that distributed system when all processors of different computational speed, complexity, cache size, frequencies and function etc., are connected with different speed links in order to completing those tasks or solving problems which needs different non identical processors. So, for implementing HeDS is to very difficult as compared to HoDS. In HeDS, all the processors generally incorporate specialized processing capacities and thus tasks are allots to them accordingly. The applications of HeDS are in the commercial companies, scientific research fields as well as job scheduling where, robustness real-time security, and high performance computing, are the main priorities [2,3]. Distributed System is important to distributing the work load on the processors [4]. Load balancing algorithm is plays a important role in homogeneous and heterogeneous distributed system in order to distribution of the tasks with better performance in terms of minimizing LIF, execution time, migration time and maximizing speedup etc. There are many scopes of load balancing for scheduling algorithm in each and every field to schedule the task. Since, heterogeneous distributed system can be defined as a range of different system resources, which can be local or globally distributed that are utilized to execute computationally intensive

applications. The effectiveness of HeDSs significantly depends on the methods used to schedule the tasks of a parallel application. The aim of HeDS system is to minimize the overall finish time or make span. DS network is mostly heterogeneous in nature. The network topology, processing nodes, communication medium, catch size, operating system, and speedup are dissimilar in different type of interconnection networks which are broadly distributed over the globe. The benefits of LB techniques are improving the overall performance of the system by each processor, it decreases the ideal time of job, small jobs have no need to suffer from long starvation, it maximizes the optimal resource, minimizes the response time, higher throughput, it reduces the cost with high gain and last but not least extensibility and incremental growth.

2. RELATED WORK

Scheduling is a technique that allocates the tasks for the compilation at a specific time. The major goal of scheduling, the all jobs are completed with minimum make span and maximum reliability onto the processors. Scheduling is the part of our routine in life for instance in the banks maintained a queue for depositing and withdrawing the money in front of window. Scheduling can be of two types that are static as well dynamic. In static scheduling having all information of tasks is known before execution such as execution time of each task, capacity of task, weight of task, precedence constraints and such scheduling cannot takes place during run time, whereas in dynamic scheduling decisions are done during run time, which may change parameters during run time. But this type of scheduling takes place in run time and also it is much more flexible and faster than static scheduling because scheduling decisions in dynamic nature that's why it can change even during run time [5]. A scheduling algorithm is describes how much CPU time is allotted to processes with try to reduce the make span, execution time and

maximization of speedup. The static and dynamic scheduling algorithm can be applied on both types of multiprocessors systems that are HoDS as well as HeDS. The task scheduling is the procedure of scheduling the tasks towards the available processors in order to get efficient execution of parallel applications on the distributed computing system. In task scheduling, we divided the tasks into smaller subtasks for an optimal solution. Today, tasks scheduling is the major issue to be considered for the researchers and various tasks scheduling method are needed in heterogeneous distributed system. This tasks scheduling can be done both the DS that is HoDS and HeDS. It can be categorized into dependent (i.e., Directed Acyclic Graph (DAG)) and independent task scheduling. DAG is directed graph which includes vertices or the node weight (i.e., shows the expected time of tasks) and directed edges weight (i.e., shows the precedence constraints, communication time and communication cost between the tasks). Thus DAG $(G) = (V, E)$, where V is the set of vertices and E is the set of edges. In DAG, source node of an edge is called parent node or entry node, whereas sink node is called child node or exit node. DAG represents data dependencies between the subtasks in task scheduling. In independent tasks scheduling, scheduling of tasks can be run without any dependencies of all other tasks, here tasks can be processed whether other tasks finished or not [6]. There are various independent task scheduling algorithms such as Min-Min [12],[14], Max-Min [6],[8],[15],[16], Opportunistic Load Balancing (OLB) [9], [7], Minimum Execution Time (MET) [7],[10] Minimum Completion Time (MCT) [10], Suffer age [11], Longest Job in the Fastest Resource-Shortest Job in the Fastest Resource (LJFR-SJFR) [12], Dynamic Load Balancing Strategy (DLBS) [1], Independent Task Scheduling with Load Balancing (ITSLB) and Load Balancing Strategy with Migration cost (LBSM) Algorithm.

A Comparison of aspects influencing the heterogeneous environment has been taken for the consideration and different load balancing algorithms are discussed as shown Table 1.

Table 1: Comparison of various Load balancing algorithm

| Algorithms | Objectives | Method | Advantages | Future Scope |
|------------|--|--|---|--|
| Min-Min | Completion of all task to suitable processors and estimates execution time | Assigning task with MECT to the suitable resource/processor | Easiest method to assignment of tasks to the resources | Improving make span by assigning tasks also with largest completion time |
| Max-Min | Completion of all task to suitable processors and estimates execution time | Assigning the task with maximum expected completion time to the suitable processor | Give approach of mapping task on the faster processor and better load balancing and make span | To Improve execution time |
| OLB | To get optimal utilization of given resources | Assigning the tasks to the processors without knowing its expected execution time | Keeping all the machines as busy as possible | To Improve make span |

| | | | | |
|------------|--|--|---|--|
| MET | Allocation of tasks | Assigning the task with minimum expected execution time for execution of tasks | System provides better performance due to allocating the tasks to the fastest processor | Provide support for heterogeneous environment |
| MCT | Assigning the task with MECT to available resources | Assigning task with MECT and task with MCT | Optimal utilization of given resources and provides better performance | Assigning task with MET |
| Suffer age | Determines the task with maximum suffer age | Work into two stages firstly calculate the suffer age value then assigns the with maximum suffer age value to the suitable resource with MCT | Assigns only suitable task with maximum suffer age value to resource with MCT | To improve performance of the system |
| LJFR-SJFR | To get overall minimum completion time as well maximum completion time | Longest and shortest tasks are assigns alternatively to the fastest resources | Better performance | To improve performance of the task mapping |
| DLBS | Improving LIF and execution time | Used for homogeneous distributed system | Minimization of LIF and execution time while completing large number of tasks | Work also for heterogeneous distributed system |
| ITSLB | Minimize LIF | Maximum Overloaded to under loaded processors | Maximize Speedup | Context Aware Load Balancing |
| LBSM | Maximize LLB | Maximum Overloaded to under loaded processors with migration cost | Maximize Resource Utilization | Context Aware Load Balancing |

3. SIMULATION STUDY

The simulation studies were concluded to monitor the allocation of the batch of independent tasks on the X-Torus, MC and FCC interconnection network. The tasks are randomly generated between 0-1000 to 0-100000 etc., and size of task's between 0.0-500.0 msec in the simulation. The batch of independent tasks is scheduled on the X-Torus, MC and FCC interconnection as per the DLBS [1], ITSLB [13] and LBSM [14] algorithms. The simulation evaluation is carried out to compare the performance of the algorithm on the performance metrics such as LIF, and level of load balance. In Figure 1 show which is representing the variation of LIF while fixed the number of the processors is eight but varying the batch of tasks. In Figure 2 show which is representing the variation of Level of Load Balance while fixed the batch of tasks are 25000 but varying the number of

processors 4, 16 and 64. The LIF is varying when keeping the number of processors fixed while the number of tasks is increased for all load balancing strategy viz. DLBS, ITSLB(Max-Max), ITSLB(Min-Max), LBSM(STS) and LBSM(LSTS) as predictable in such a case. At any situation, LBSM (LSTS) performs best LIF on FCC network from other strategies as shown in figure 1. As can be seen in figure 2, the Level of Load Balancing goes on increasing for all load balancing strategies. Both strategies are performing better on MC interconnection network on account of LLB for all considered batch size. In both strategies, LLB is lesser for lesser number of processor on X-Torus, FCC and MC interconnection networks. However, So, MC provides better load balancing as compared to X-Torus and FCC interconnection network. LBSM (STS) is showing better performance on MC interconnection network.

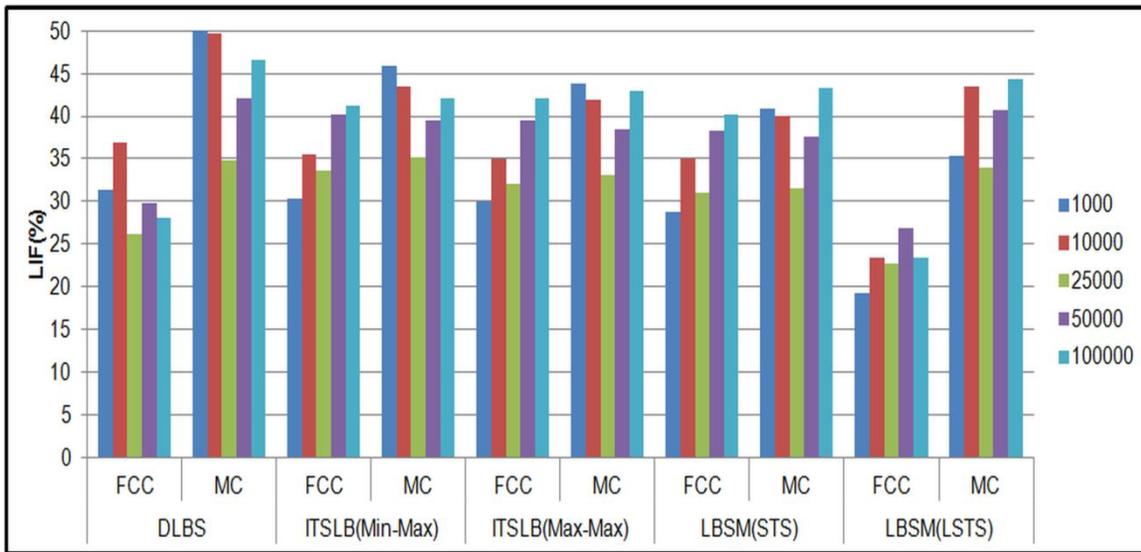


Figure 1: LIF v/s independent batch of tasks

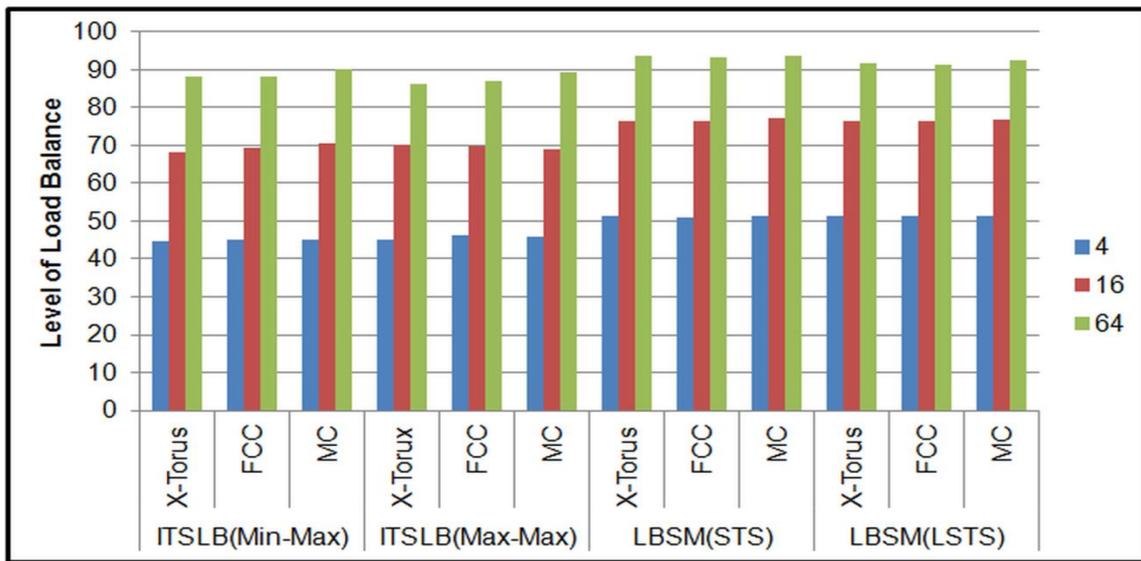


Figure 2: Level of load balance v/s independent batch of tasks

4. PROPOSED HYBRID APPROACH INCLUDING MIGRATION COST AND LOAD BALANCING

It is necessary to achieve systems related and user related objectives in heterogeneous environment. The proposed model is constructed for efficient resource utilization from user’s perspective by calculating level of load balancing of computational node. The main aim of this model is to achieve “Efficient resource utilization and load balancing on computational node in heterogeneous distributed environment”. In order to perform optimum utilization of node resources, make span time should be minimized that leads to scheduler assigning the task an efficient way to the resources. The proposed approach is designed in two phase. First phase will be selection of strategies such as Max-Max,

Min-Max etc., so that each task has minimum make span. Second phase will be migration of task from MOL to MUL so that each computational node has maximum load balance. Apart from load balance, make span is also considered which response time from task submission to task completion is. Proposed model performs scheduling of n batch of tasks to m computational node which is NP complete problem. The Load balancing algorithm will be implemented by any heuristic techniques and AI approach. The scheduling algorithm will be designed in a manner so that it properly utilizes the resources. The proposed model is a novel and hybrid approach which is specifically designed to combined the advantages of minimum completion time and efficient resource utilization. Proposed work will be done using fuzzy rule based method and load balancing among nodes in heterogeneous distributed

environment. In this work main challenging work is after calculating completion time of task, further it performs load balancing among resources to make resource utilization more effective. One or combination of intelligent methods with heuristics may also be context for load balancing purpose.

5. CONCLUSION AND FUTURE WORK

Heterogeneous distributed system plays a vital role in resource sharing and load balancing among resources. It has become most common and widely adapted technique to solve computational problem occurred in scientific and technical areas. Load balancing and job scheduling is an essential need of today's grid environment. A scheduling algorithm is not good until it doesn't provide us good and efficient timings. In this paper illustrated various load balancing algorithms for heterogeneous distributed system. Furthermore, a compared approach between these algorithms likes their objective, method, advantages, and future work have also been proposed. Moreover, in simulation studies, our previous works DLBS, ITSLB and LBSM is compared on FCC, MC and X-Torus interconnection networks, LBSM (LSTS) outperforms among considered strategies on account of LIF on FCC interconnection network whereas LBSM (STS) performs best for LLB on MC interconnection network. Our future work is to develop a context aware load balancing technique heterogeneous distributed system.

REFERENCES

- [1] Alam M, Varshney AK. A New Approach of Dynamic Load Balancing Scheduling Algorithm for Homogeneous Multiprocessor System. *International Journal of Applied Evolutionary Computation (IJAEC)*. 2016; 7(2): 61-75.
<https://doi.org/10.4018/IJAEC.2016040104>
- [2] Daoud MI, Kharma N. A high performance algorithm for static task scheduling in heterogeneous distributed computing systems. *Journal of Parallel and distributed computing*. 2008; 68(4): 399-409.
<https://doi.org/10.1016/j.jpdc.2007.05.015>
- [3] Jiang Y. A survey of task allocation and load balancing in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*. 2016 Feb 1;27(2):585-99.
<https://doi.org/10.1109/TPDS.2015.2407900>
- [4] Peleg D. Distributed computing: a locality-sensitive approach. *Society for Industrial and Applied Mathematics*; 2000 Jan 1.
<https://doi.org/10.1137/1.9780898719772>
- [5] Singh K, Alam M, and Sharma S. A survey of static scheduling algorithm for distributed computing system. *International Journal of Computer Applications*. 2015 Nov; 129(2):25-30.
<https://doi.org/10.5120/ijca2015906828>
- [6] Rafsanjani MK, Bardsiri AK. A new heuristic approach for scheduling independent tasks on heterogeneous computing systems. *International Journal of Machine Learning and Computing*. 2012 Aug 1;2(4):371.
<https://doi.org/10.7763/IJMLC.2012.V2.147>
- [7] Sang A, Wang X, Madihian M, Gitlin RD. Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems. *Wireless Networks*. 2008 Jan 1;14(1):103-20.
<https://doi.org/10.1007/s11276-006-8533-7>
- [8] Kokilavani T, Amalarethinam DG. Load balanced min-min algorithm for static meta-task scheduling in grid computing. *International Journal of Computer Applications*. 2011 Apr;20(2):43-9.
<https://doi.org/10.5120/2403-3197>
- [9] Al Nuaimi K, Mohamed N, Al Nuaimi M, Al-Jaroodi J. A survey of load balancing in cloud computing: Challenges and algorithms. In *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on 2012 Dec 3* (pp. 137-142). IEEE.
<https://doi.org/10.1109/NCCA.2012.29>
- [10] Freund RF, Gherrity M, Ambrosius S, Campbell M, Halderman M, Hensgen D, Keith E, Kidd T, Kussow M, Lima JD, Mirabile F. Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. In *Heterogeneous Computing Workshop, 1998.(HCW 98) Proceedings. 1998 Seventh 1998 Mar 30* (pp. 184-199). IEEE.
- [11] Armstrong R, Hensgen D, Kidd T. The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions. In *Heterogeneous Computing Workshop, 1998.(HCW 98) Proceedings. 1998 Seventh 1998 Mar 30* (pp. 79-87). IEEE.
- [12] Mukhopadhyay R, Ghosh D, Mukherjee N. A study on the application of existing load balancing algorithms for large, dynamic, heterogeneous distributed systems. In *Proc. 9th WSEAS Int. Conf. Software Engineering, Parallel and Distributed Systems (SEPADS) 2010 Feb 20* (pp. 238-243).
- [13] Khan Z, Alam M, and Haidri R A. Effective Load Balance Scheduling Schemes for Heterogeneous Distributed System. *International Journal of Electrical & Computer Engineering*. Oct 2017; 7(5). (In Press)
<https://doi.org/10.11591/ijece.v7i5.pp2757-2765>
- [14] Alam M, and Shahid M. A Load Balancing Strategy with Migration Cost for Independent Batch of Tasks (BoT) on Heterogeneous Multiprocessor. *Interconnection Journal of Applied Evolutionary Computation*. 2017, Vol. 8, Issues 3, pp. 74-92.
<https://doi.org/10.4018/IJAEC.2017070104>
- [15] K.Gowsic et. al. Design of a Hybrid Logic Based A daboost Decision Tree Model for Identifying Web Attacks. *International Journal of Advanced Trends in Computer Science and Engineering 2019, Vol. 8, Issues 1.2*, pp. 52-58.
- [16] Danish Ahmad et. al. A Review and Analysis of Big Data and MapReduce *International Journal of Advanced Trends in Computer Science and Engineering 2019, Vol. 8, No. 1*, pp. 1-3.
<https://doi.org/10.30534/ijatcse/2019/01812019>