An Efficient Local Hierarchical Load Balancing Algorithm (ELHLBA) in Distributed Computing

Rafiqul Zaman Khan, Md Firoj Ali
Department of Computer Science, Aligarh Muslim University, Aligarh, India

***Abstract*—Load balancing algorithm can efficiently improve the performance of a distributed computing system than the system without load balancing algorithm. Dynamic load balancing algorithm is accountable for balancing load among the nodes depending upon the system state at any instant of moment. In centralized approach the information is collected by a specially designated central node and in distributed approach each node has the autonomy to collect the information about the load of the system. For a large global distributed system centralized approach of the load balancing algorithm is not efficient due to the contention problem. In distributed approach either a sender or a receiver may poll all the nodes in a network for load balancing causing huge overheads. Hierarchical load balancing approach imbibes the merits of both centralized and decentralized approaches by removing the disadvantages of centralized and decentralized approaches. In this paper we have proposed a hierarchical load balancing algorithm ELHLBA in which we considered the parents of leaf nodes as a front end nodes. We compared our algorithm with other existing algorithm ILHLBA and LHLBA. The simulation results show that our algorithm produces better result than the existing algorithms ELHLBA and LHLBA in respect of response time and throughput against system utilization.**

***Keywords*— Distributed System, Hierarchical, Load Balancing, Under loaded, Overloaded**

I. INTRODUCTION

A distributed computing system (DCS) is a set of autonomous computing systems (nodes) connected through any network. DCS is heterogeneous in nature due to the several aspects like computing power, memory storage, and network bandwidth and arrival pattern of tasks to the different nodes. The prime objective of the DCS is to share the resources in a better way available in the network. So one of the best way to utilize and to share the network resources like files, printers and CPUs is through the load balancing among the processors to maximize resource utilization. In general certain processors may have more numbers of tasks and some others may have less numbers of tasks or even idle causing disparity of job distribution on different processors in DCS resulting low system performance. Most of the time it is noticed in networks that at the peak time the fastest nodes are idle or lightly loaded and the slower nodes are heavily loaded [1], [2]. It is also observed that two third nodes remain idle in the peak hour of computing. The jobs that are fed with the overloaded nodes, will have to wait longer to get processor's attention as there must be more numbers of preceding jobs. Thus overloaded nodes would produce high average response time. Even an application of simple load balancing algorithm produces better performance than no load balancing algorithm applied in a network [3], [4], [5]. So the load balancing concept comes into existence by transferring excess load from the heavily loaded nodes to the lightly loaded nodes so that the load on each node becomes approximately the same. The load balancing algorithms improve the overall performance of the system by exploiting the maximum power of the processors and minimizing the average response time and hence maximize the resource utilization.

The dynamic load balancing approach has three most important policies: transfer policy, information policy and location policy. Transfer policy decides depending upon some predefined value whether a job would be executed locally or remotely. In selection policy the load balancing node selects a suitable node for transferring the selected job depending upon the information collected by state information policy. Both sender initiated and receiver initiated approaches fall under the location policy.

There are two fundamental approaches to the load balancing algorithm design. In static load balancing design approach the tasks are assigned on the basis of a priori knowledge of the system and once the tasks are allocated on the nodes do not change [1], [2]. The performance of the static load balancing algorithms depends on the prior information about the tasks and the system. The decision to transfer the tasks does not depend on the system state change. So this approach is best suited for homogeneous distributed computing system. But the dynamic load balancing algorithms take the decision to transfer the tasks depending on the current state of the system. The tasks are transferred from heavily loaded node to the lightly loaded node [1], [6]. So the quality of dynamic load balancing algorithms depends on the collection of information on load on different nodes in the system. So this approach is best suited for heterogeneous distributed computing system.

In dynamic load balancing the information may be collected either by centralized or distributed approach. In centralized approach the information is collected by a specially designed central node and in distributed approach each node has the autonomy to collect the information about the load of the system. It has been reported that the collection of information by centralized approach about the system state does not cause any performance degradation for a reasonably large distributed computing systems [7].The drawback of this approach is that the performance of a globally distributed system would be very poor and the cost of state information collection would be too much and maintaining a huge information by a single node will surely cause a performance degradation. In the distributed information collection policy the information is collected either by sender initiative or receiver initiative algorithm. In sender initiative approach the heavily loaded nodes search

for lightly loaded nodes for transferring extra load and the receiver initiative approach is the converse of sender initiated approach. In this approach either a sender or a receiver may poll all the nodes in a network for load balancing causing huge overheads. To reduce the overheads the sender or receiver nodes poll a selected number of nodes like nearest neighbors [4], [8]. Another performance problem with this approach is associated with the inter-arrival times and service times.

Another type of load balancing exists which is widely known as hierarchical load balancing approach. This approach syndicates the merits of both centralized and decentralized approaches .In this approach the disadvantages of centralized approach and decentralized approach are minimized.

II. Hierarchical Load Balancing Strategy

In decentralized load balancing approach the sender or receiver nodes poll a number of nodes to send/receive the load while in centralized approach a single node is responsible for maintaining the state information of all the nodes in a network and every node either sender/receiver sends/gets load after consulting the central node. Decentralized load balancing strategy is successful for large distributed computing but centralized load balancing approach is the best strategy for a small network but it would very cost over a globally distributed network [7]. It has been shown that the centralized strategy is best suited for a cluster of nodes in a large distributed system [9]. Thus the concept of hierarchical load balancing strategy brought a remarkable improvement in load balancing strategies in distributed computing. In this approach it is very easy to implement both the advantages of centralized and decentralized strategies together due to its hierarchical structure.

In hierarchical approach a set of specially designated nodes is responsible for maintaining the state information of a set of nodes in below of their hierarchy. The system information is maintained by all sub tree nodes and as we move towards the root node more information would be maintained by the corresponding sub tree root nodes and the root node would maintain the global state information. Fig1 illustrate the hierarchical organization of eight nodes.

The hierarchical topology is chosen for load balancing for the following advantages: Hierarchical network is easier to expand; It is easy to manage and maintain the network because the whole network is divided into small clusters (segments) and error detection and correction is also easy and if one cluster is damaged, other cluster will continue to work.

In general the nodes are considered as only router except the leaf nodes in a hierarchical distributed network system [10], [11]. The leaf nodes are considered to be the computing processor and rest are as manager (coordinator) nodes responsible for load balancing only. But intermediate nodes with finite buffer capacity also have been utilized as front-end node which is responsible for both load balancing and computing purpose [12].

III. Proposed Hierarchicalalgorithm

We have already proposed an improved version of local hierarchical load balancing algorithm (ILHLBA) [13] which minimizes the problems encountered in LHLBA. In ILHLBA approach the sub tree node only maintains the overall status of a sub tree 0, +1 or -1 as shown in Fig 2. For a lightly loaded system if the left sub tree does not have any sender, the request for a job by a receiver will reach up to the root and at the same time if the right sub tree also does not have any sender, a lot of requests will reach to the root at the same time and thus will cause a huge gathering of messages to the root resulting low performance due to the congestion on the links R0R1 and R0R2 and also due to the “no job” message to each receiver. These two problems have been minimized by exchanging the latest status of R1 and R2 by R0. Suppose that R1 and R2 are in -1 state and this status is being exchanged between them by R0 whenever there is a change of status. Thus R1 will not send any job request message to R0 and hence R0 will not have to manage all the requests coming from both sub trees.

In local hierarchical load balancing (LHLBA) algorithm [10] leaf nodes are considered as computing nodes and rests are considered as the co-operating nodes. But co-operating nodes are also can be used as front-end node [12]. Front end nodes act as a co-operating as well as computing nodes. We proposed that the parents of leaf nodes (POLN) only would behave like a front-end node. In this approach POLN will execute the tasks if the leaf nodes are overloaded and if POLN becomes overloaded, it will follow the ILHLBA [13]. In a highly loaded system when there would be very few numbers of receivers, the role of POLN would be very effective.

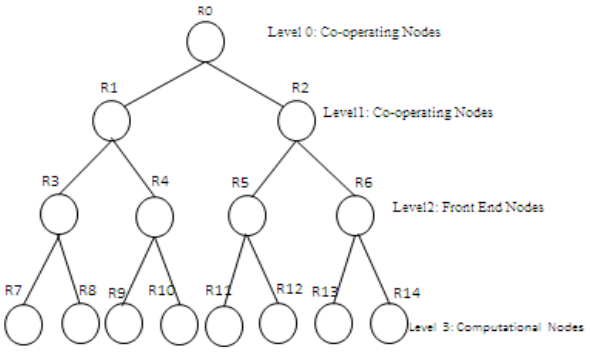

Fig. 1 A sample hierarchical topology with eight omputing nodes and four front-end nodes

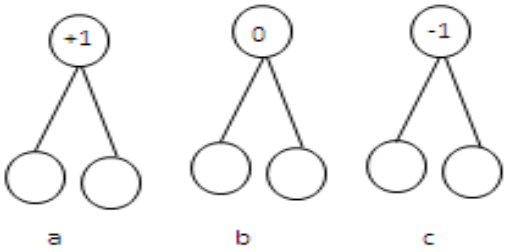

Fig. 2 after local load balancing a) both nodes either over loaded or one is over loaded and another is moderate loaded, b) both nodes are moderate loaded c) both are under loaded or one is under loaded and other is moderate loaded

```
begin:
  for a leaf node
        if the load is greater than threshold load
          send status  +1 to parent node;
.
    for parent node
        compare the load of leaf node and adjust the load;
          if  the load of both the leaf nodes is in OL state
            parent executes the task from any leaf  node;
              if one leaf  node is in +1 state and another is in -1 state
                balance the load;
                  if one node is still in +1 state
                    parent node execute the extra load from +1 state  node;
        if parent node is overloaded
        wait for receiver.
    end for
  end for
end.
```

Fig. 3 Algorithm for ELHLBA

IV. Simulation And Result

The proposed load-balancing algorithm ELHLBA is compared with ILHLBA and LHLBA. The simulation framework has been designed with the following setup. We considered a hierarchical topology with the number of computing nodes 8. The connectivity of all the nodes is ensured. The comparison has been carried out in terms of response time and throughput for the varying loads. Load Balancing Simulator in Java is used for this purpose for testing the behavior of different load balancing algorithms under the same conditions and evaluates the behavior of the algorithms for different arrival patterns.

Fig 4 and Fig 5 represent the response time for different system utilization for 25% and 50% of sneding nodes present in the system respectively. Both Fig 4 and Fig 5 also compare the response time among LHLBA and ILHLBA. Our proposed algorithm ELHLBA is better than ILHLBA and LHLBA algorithm in a overloaded system.

Fig 6 and Fig 7 represent the troughput for different system utilization for 25% and 50% of sending nodes present in the system respectively. Both Fig 6 and Fig 7 also compare the throughput among ELHLBA, LHLBA and ILHLBA. Our proposed algorithm ELHLBA is better than ILHLBA and LHLBA algorithm for highly loaded system.

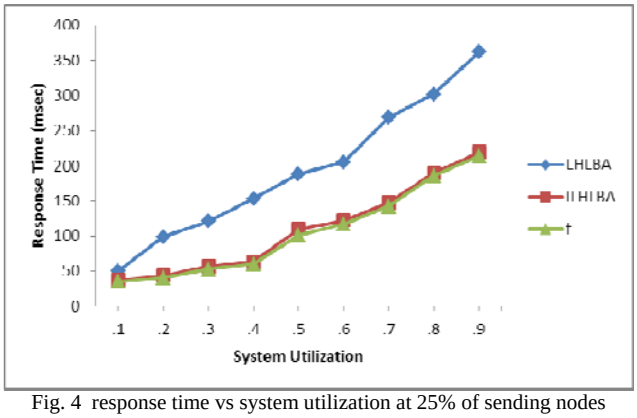

Fig. 4 response time vs system utilization at 25% of sending nodes

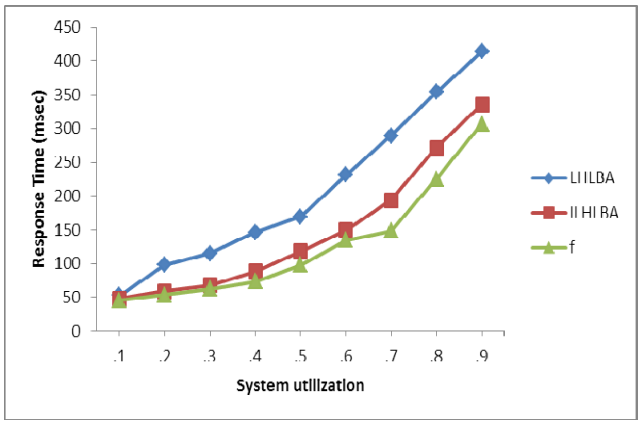

Fig. 5 response time vs system utilization at 50% of sending nodes

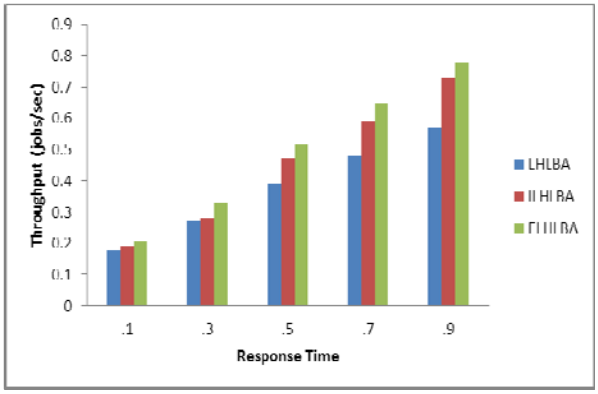

Fig 6. throughput vs system utilization at 25% of sending nodes

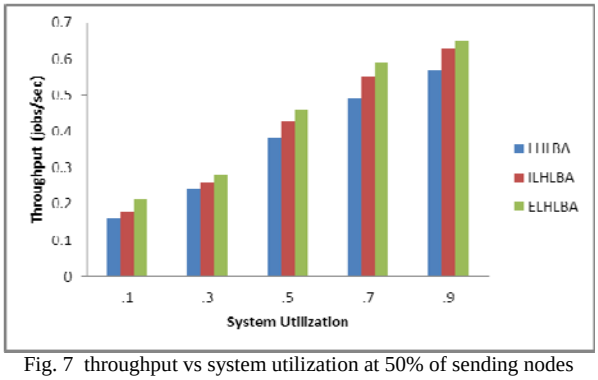

Fig. 7 throughput vs system utilization at 50% of sending nodes

V. CONCLUSIONS

Even a simple load balancing algorithm generates better performance in a distributed system than a system without any load balancing algorithm. Load balancing algorithms allocate the loads from heavily loaded nodes to the lightly loaded nodes to keep all the nodes in a network to be busy in so that the maximum use of the resources can occur to get the better performance. In centralized approach the information is collected by a specially designated central node and in distributed approach each node has the autonomy to collect the information about the load of the system. The drawback of this approach is that the performance of a globally distributed system would be very poor and the cost of state information collection would be too much and maintaining huge information by a single node will surely cause performance degradation. In distributed approach either a sender or a receiver may poll all the nodes in a network for load balancing causing huge over heads. Hierarchical load balancing approach syndicates the merits of both centralized and decentralized approach by removing disadvantages of centralized and decentralized approaches. The hierarchical topology is chosen for load balancing for the following advantages: hierarchical network is easier to expand; it is easy to manage and maintain the network because the whole network is divided into small clusters (segments) and error detection and correction is also easy and if one cluster is damaged, other cluster will continue to work.

In this paper we have taken parents of leaf nodes as the front-end nodes which executes the extra tasks on leaf nodes preventing from remote execution every time and we have also evaluated the performance of ELHLBA, HLBA and ILHLBA and we compared our algorithm ELHLBA with the existing algorithms ILHLBA and LHLBA. The simulation results show that our algorithm produces better result than existing algorithms in respect of response time and throughput against system utilization.

REFERENCES

[1] M. F. Ali and R. Z Khan, The study on load balancing strategies in distributed system, *International Journal of Computer Science & Engineering Survey, Vol.3, No.2,* 2012, 19-30.

[2] G. Bernard, D. Steve and M. Simatic , A survey of load sharing in networks of workstations, the british computer society, *The Institute of Electrical Engineers and IOP Publishing Ltd,* 1993, 75-86.

[3] I. Ahmad, A Gafoor and G. C. Fox, Hierarchical scheduling of dynamic parallel computations on hypercube multicomputers, *Journal of Parallel and Distributed Computing 20,*1994, 317-329.

[4] D.l. Eager, E.D. Lazowska and J. Zahorjan, Adaptive load sharing in homogeneous distributed systems, *IEEE Trans. Software Engrg. SE-12,* 1986, 662-675.

[5] J. Xu and K. Hwang, heuristic methods for dynamic load balancing in a message-passing super computer, *Proceeding of Supercomputing '90,* 1990, 888-897.

[6] E. Haddad, Dynamic optimization of load distribution in heterogeneous systems, *IEEE,*1994, 29-34.

[7] M.M. Thiemer and K.A Lantz, Finding idle machines in a workstation-based distributed system, *IEEE Int. Conf. Dist. Computing Systems,* 1988, 112-122.

[8] D.l. Eager, E.D. Lazowska and J. Zahorjan, A comparison of receiver-initiated and sender-initiated adaptive load sharing, *Perfprmance evaluation, Vol.. 6,* 1986, 662-675.

[9] S. Zhou, X. Zheng, J. Wang, and P. Delisle,Utopia: a load sharing facility for large, heterogeneous distributed computer systems, *Software - Practice and Experience,Vol. 23, No. 12,* December 1993,1305-1336.

[10] S. P. Dandamudi and M. Lo, Hierarchical load sharing policies for distributed systems, Technical Report TR- 96-22, Proc. *Int. Conf. Parallel and Distributed Computing Systems,* 1996.

[11] M.H. Willebeek-LeMair and A.P. Reeves, Strategies for dynamic load balancing on highly parallel computers. *IEEE Trans. Parallel and Distributed Systems, Vol. 4, No. 9,* 1993, 979-993.

[12] B. Veeravalli and J. Yao, Divisible load scheduling strategies on distributed multi-level tree networks with communication delays and buffer constraints, *Computer Commun., 27,* 2004, 93-110.

[13] R. Z Khan and M. F. Ali, An Improved Local Hierarchical Load Balancing Algorithm (ILHLBA) in Distributed Computing, *International Journal Of Advance Research In Science And Engineering IJARSE, Vol. No.2,* Issue No.11, November 2013

Dr. Rafiqul Zaman Khan: Dr. Rafiqul Zaman Khan is presently working as an Associate Professor in the Department of Computer Science in Aligarh Muslim University (A.M.U), Aligarh, India. He received his B.Sc. Degree from M.J.P Rohilkhand University, Bareilly, M.Sc and M.C.A from A.M.U. and PhD (Computer Science) from Jamia Hamdard University, New Delhi, India. He has 19 years of Teaching Experience of various reputed International and National Universities viz King Fahad University of Petroleum & Minerals (KFUPM), K.S.A, Ittihad University, U.A.E, Pune University, Jamia Hamdard University and AMU, Aligarh. He worked as a Head of the Department of Computer Science at Poona College, University of Pune. He also worked as a Chairman of the Department of Computer Science, AMU, Aligarh. His Research Interest includes Parallel & Distributed Computing, Gesture Recognition, Expert Systems and Artificial Intelligence. Presently four students are doing PhD under his supervision. He has published about 41 research papers in International Journals/Conferences. Names of some Journals of repute in which recently his articles have been published are International Journal of Computer Applications (ISSN: 0975-8887), U.S.A, Journal of Computer and Information Science (ISSN: 1913-8989), Canada, International Journal of Human Computer Interaction (ISSN: 2180-1347), Malaysia, and Malaysian Journal of Computer Science(ISSN: 0127-9084), Malaysia. He is the Member of Advisory Board of International Journal of Emerging Technology and Advanced Engineering (IJETAE), Editorial Board of International Journal of Advances in Engineering & Technology (IJAET), International Journal of Computer Science Engineering and Technology (IJCSET), International Journal in Foundations of Computer Science & technology (IJFCST) and Journal of Information Technology, and Organizations (JITO).

Mr. Md Firoj Ali: Mr. Md Firoj Ali is presently working as a Research Scholar in the Department of Computer Science in Aligarh Muslim University (A.M.U), Aligarh, India. He received his B.Sc. and MCA Degree from A.M.U. He has been awarded Senior Research Fellowship by UGC, India and also cleared National Eligibility Test conducted by UGC, 2012. His Research Interest includes Load balancing in Distributed Computing System. He has published nine research papers in International Journals/Conferences.