Conclusions and Future Works
Conclusions

First of all, the main objective was to develop dynamic load balancing algorithms that drives any initial load distribution in distributed system into the even load distribution by treating load as indivisible independent units. It has been focused in this thesis to develop fully distributed load balancing algorithm because this approach is more practical than centralized approach. The exhaustive analysis of the state of the art in load balancing algorithms led to the study of parallel and distributed computing, load balancing algorithm taxonomy, simulation tools, algorithm design aspects and algorithm implementations.

In this thesis, we presented four new algorithms: An Efficient Diffusion Load Balancing Algorithm in Distributed Computing System, An Improved Local Hierarchical Load Balancing Algorithm (ILHLBA) in Distributed Computing, An Efficient Local Hierarchical Load Balancing Algorithm (ELHLBA) in Distributed Computing and A New Distributed Load Balancing Algorithm.

In the first algorithm: “An Efficient Diffusion Load Balancing Algorithm in Distributed Computing System”, we have shown that the overheads depend on the number of OLN present in the domain of ULN and vice-versa. Our simulation results show that the numbers of overheads communicated at a particular moment are fairly less than KN and our algorithm complexity merges with KN when the number of ULD is equal to N . The simulation results also show that 31.4% and 31.9% less over heads is produced in optimal case as compared to worst case in respect of 4X4 and 16X16 mess topology respectively. Most important thing is that at 50% of ULN, the overheads becomes fairly less than expected due to the fall in the value of Ki. The value of Ki falls because there is a great probability to have more under loaded neighbors under the domain of an ULN and Ki=0 when there will be no over loaded node in the network.

In the second algorithm: “An Improved Local Hierarchical Load Balancing Algorithm (ILHLBA) in Distributed Computing” is based on hierarchical network. 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
algorithm, we have evaluated the performance of LHLBA and ILHLBA and we
compared our algorithm ILHLBA with the existing algorithm LHLBA. The simulation
results show that our algorithm produces better result than existing one in respect of both
response time and throughput against system utilization.

Our proposed algorithm ILHLBA algorithm is better than LHLBA in a lightly loaded
system having 25% receiving nodes by 29.8% and for 50% receiving nodes by 25.7%
in respect of response time vs system utilization respectively. And ILHLBA algorithm is
to LHLBA in a lightly loaded system having 25% receiving nodes by 17.8% and
for 50% receiving nodes by 14.1% in respect of throughput vs system utilization
respectively.

In the third algorithm: “An Efficient Local Hierarchical Load Balancing Algorithm
(ELHLBA) in Distributed Computing”, 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. Our proposed algorithm ELHLBA is better than ILHLBA and LHLBA by
16.3% and 43.7% in 25% of receiving node system and ELHLBA is better than ILHLBA
and LHLBA by 16.9% and 25.7% in 50% of receiving node system in respect of
response time vs system utilization.

Our proposed algorithm ELHLBA is better than ILHLBA and LHLBA algorithm by
10.4% and 26.1% in 25% receiving node system and ELHLBA is better than ILHLBA
and LHLBA algorithm by 13.3% and 33.5% in 50% receiving node system in respect of throughput vs system utilization.

In the last algorithm: “A New Distributed Load Balancing Algorithm”, each node maintains a load table holding the current load situation of its neighbors. A new concept in threshold load in each node avoids unexpected over loading and also ensures more job allocation towards more powerful nodes. As the nodes inform their load situation on status change, the over loaded or under loaded nodes do not need to poll for transfer of jobs or to invite the jobs from the neighbors and thus causes low congestion in the network. Response time and overheads are measured by applying this algorithm in mesh topology of sixteen nodes and compared the new algorithm with Ni’s drafting algorithm. In our algorithm, the response time is better than Ni’s drafting algorithm by 12.1% and overheads is better than Ni’s drafting algorithm by 10.3% with respect to the arrival rate.

**Future Works**

Many works on designing effective load balancing algorithms have been done but no load balancing algorithm is universal in nature for distributed computing. Every load balancing algorithm has some limitations.

Tasks have been assumed equal in size and independent in nature for all the proposed algorithms in this thesis. Therefore, one of the future works is to assume the tasks with variable size and some of the tasks are partially or fully dependent to each other.

All the proposed algorithms in this thesis have been simulated for mesh topology and hierarchical topology. The proposed algorithms may be simulated also for other network topologies in future.

The quality of load balancing algorithm depends on how quickly a node can come to know the load on the other nodes in a network. Thus, if a node can come to know the load situation in advance or at the right time, the performance of the system would be highly improved. So, Artificial Intelligence may be applied to collect the load information by a node either in advance or in right time to perform the load balancing.