Optimal Load Balancing Algorithm in Distributed Systems

Sandeep Agarwalla[*]
Computer Science Dept. IIMC, Khairatabad,
Hyderabad, India

Abstract--The basic function of a load balancing algorithm is to transfer load(task) from heavily loaded computers to idle or lightly loaded computers. This accelerates the performance of the task. The algorithms that are used for load balancing are does not give optimal solution to the distributed environment. In today's world it is important that task execution should be done in a very fast pace. Clients do not want to wait for the responses from the server for a small duration even; so it is important optimize the load balancing strategies to give High performance.
The main characteristic is to derive a formula by combining the basic load balancing algorithm and get the best optimal performance solutions for it. Though there are more than 7 algorithm, only three are combined to produce the solution.

Key words: Distributed Tasks, Weighted Algorithm, Least Used Algorithm, Least Latency Algorithm

I.   INTRODUCTION

Scheduling for distributed systems is significantly more complex than for single processor systems. A distributed scheduling policy for a general purpose system can be divided into two components: a local scheduling discipline determines how the CPU resources at a single node is allocate among its resident process, while a load distribution strategy distributes the system work load among the nodes through process migration.

User performance expectation generally center on the quality of service provided to the process that initiate. In addition to the average quality of service received, fairness is important concern. Two users simultaneously initiating equivalent process expect to receive about the same quality of service. Similarly, a user submitting the same job several times, under equivalent workload, expects each to receive about the same quality of service. To ensure fairness the variance in quality of service under a given workload should be acceptably low.

II.   OBJECTIVES & METHODOLOGY
1. Accelerating the task scheduled to be processed in an optimal way.
2. Users in the network would yield better result from servers in the distributed environment.

Methodology
The present study focuses on the secondary sources of data from Books, Journals, Articles, and interviews with eminent personalities from various departments of the University.

Existing Algorithms
There are a number of algorithms used for balancing the tasks; they yield a result that is good but the optimal solutions cannot be yielded from them. Lets discuss some of the existing algorithms and discuss about their performance.

Weighted Balance
Assign more traffic to a faster link or less traffic to a connection with a bandwidth cap.Set a weight on the scale for each connection and outgoing traffic will be proportionally distributed according to the specified ratio. Thereby increasing the overall performance.
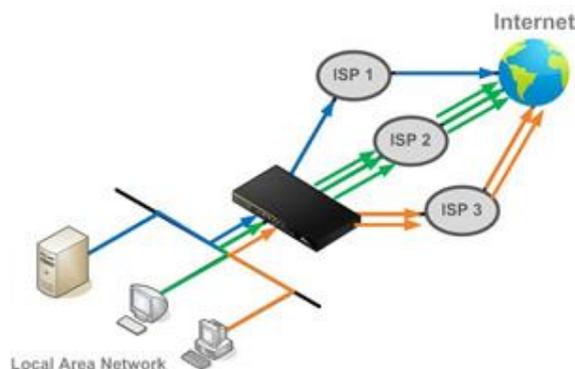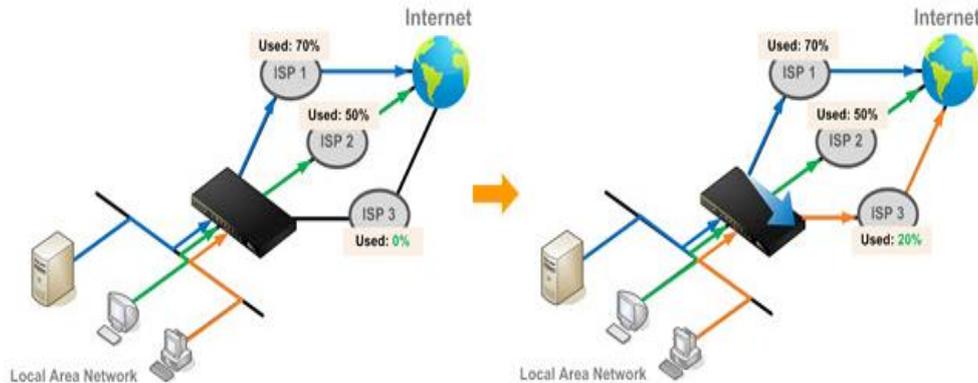
Problems:
1. Latency time is not taking into consideration while the task have been distributed among the various ISPs.
2. Band width of the channels are not taken into consideration while allocation of the task to the ISPs.

Least Used
Help you choose the better connection with more free bandwidth or a node with higher rate of idleness.Traffic will be directed to the link with the most available bandwidth among the selected connections, load will proportionately distributed; increasing the performance and balancing the load.
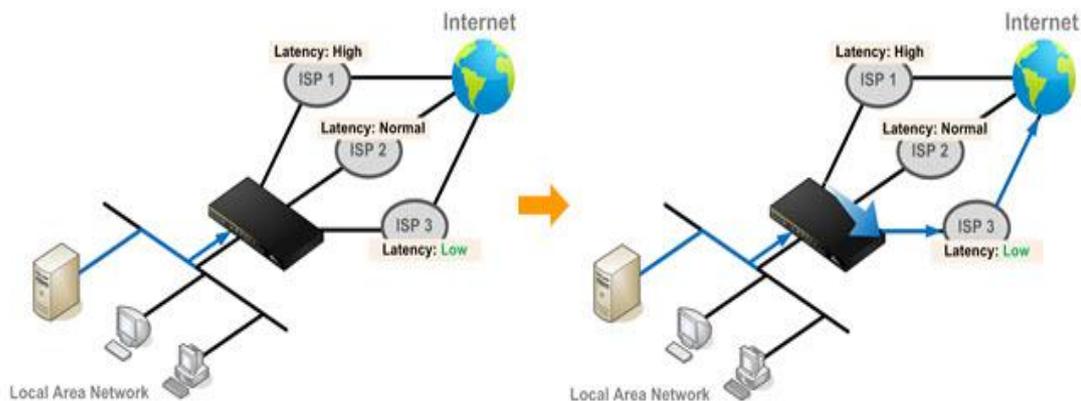
Problems:
Though the methodology in distribution of the task into various ISPs, provide a good yielding result yet the latency time is never consider in allocation of the task to the ISPs.

Lowest Latency
Give you the fastest response time when using applications like online gaming.Traffic will be assigned to the link with the lowest latency time among the selected connections.*Latency time is the taken for bringing in and out processes from one memory to another.*

Problems:
This algorithm is used only when online gaming applications are used by the users in a distributed environment.
However, it never provides better result when users execute other forms of application.

III. OPTIMAL ALGORITHM
The above three algorithm are mostly used while construction a distributed environment. The performance level is good but can be constructed in a way to produce optimal result. The above three are considering only one of the factors while balancing the load. As the first one calculates the weight of individual node and distributing according to that. Second one Finds the idle or least used processor and balance the loading by moving in and out of processes. The third one considering only the latency time. The three factors can be combined together and generate the a result based on the nodes/processor in the system and produce an optimal result and thus gives high performance.
The following results summarize the overall model performance. Here we are simulating the model by using the metric like throughput, number of tasks waiting in the queue with in interval. All jobs are dynamically created and allotted to the processor and processor is selected according to the conditioned as specified in our algorithm. For each processor from 2 to 50 we have taken 100 instances where job are created dynamically and allotted to the processor and then we have plotted a graph of number of processor and number of job completed. And we have seen that throughput increase if the task is I/O intensive related task.
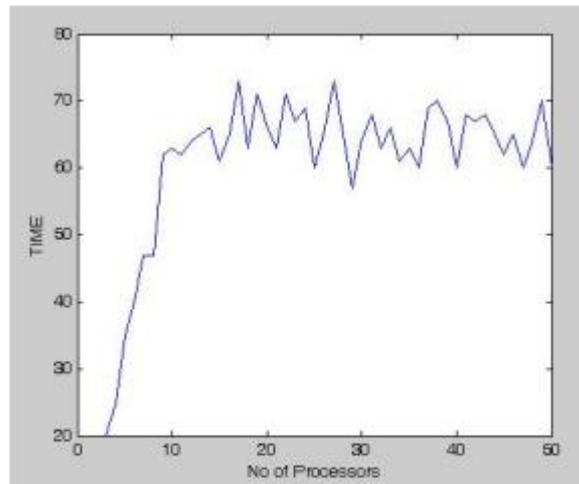
Figure 2: Throughput graph number of processor vs. time

We analyze the system performance and scalability of computing nodes with load balancing. As distributed systems continue to grow in scale, in heterogeneity, and in diverse networking technology, they are presenting challenges that need to be addressed to meet the increasing demands of better performance and services for various distributed application.

IV. CONCLUSION

This paper studies the performance of system under different type of load like I/O as well as CPU, MEMORY based on IOCM dynamic load balancing algorithm in heterogeneous computing system. There are number of different dynamic load balancing techniques for cluster systems; their efficiency depends on topology of the communication networks that connects nodes. This research has developed an efficient load balancing for I/O-, CPU- and MEMORY-intensive tasks. For this we developed a new way to predict and calculate the load of cluster nodes.

This can minimizes the average slow down of all parallel jobs running on a cluster and reduce the average response time of the jobs. Future studies can be performed in following direction. First, we will evaluate the performance of scheme on a large scale of cluster. Second, we have assumed the task is independent, so we will also simulate this scheme for inter-dependent task. Third, in this study we have assumed network communication cost is negligible; therefore we will extend this to balance the load in network resource.

REFERENCES

[1]   Xiao Qin, Performance comparisons of load balancing algorithms for Ointensive workloads on clusters, Journal of Network and computer applications(2006), doi:10.1016/j.jnca.2006.07.001.

[2]   Xiao Qin ,Dynamic Load Balancing for IO-Intensive Tasks on Heterogeneous Clusters, Proceeding of the 2003 International Conferenceon High Performance Computing(HiPCO3).

[3]   Xiao Qin ,Hong Jiang ,Yifeng Zhu ,David R. Swanson ,A Dynamic Load Balancing Scheme for IO-Intensive Applications in Distributed Systems, Proceeding of 2003 international conference on Parallel processing Workshop(ICPP 2003 Workshop).

[4]   Xiao Qin, A feedback control mechanism for balancing I/O-intensive and memory-intensive applications on cluster, parallel and distributed computing practices journal.

[5]   Paul Werstein ,Hailing Situ and Zhiyi Huang , Load balancing in cluster computer, Proceeding of the seventh international conference on Parallel and Distributed Computing, Applications and Technology (PDCAT'06.

[6]   Xiao Qin, H.Jiang, Y.Zhu and D.swanson, Toward load balancing support for I/O intensive parallel jobs in a cluster of workstation, Poc. Of the 5th IEEE international conference cluster Computing(cluster 2003) ,Hong Kong, Dec. 1-4-2003

[7]   Kumar K. Goswami, Murthy Devarakonda and Ravishankar K. Iyer, Prediction–baesd dynamic load-sharing heuristics, IEEE transaction onparallel and distributed systems, VOL.4, No.6, june 1993.

[8]   Mohammed JaveedZaki, Wei Li, SrinivasanParthasarathy, A Review ofCustomized Dynamic Load Balancing for a Network of Workstations.