Static Vs Dynamic Load Balancing Algorithms in Parallel Computing Environment

Raj Kumar[1] Prof. (Dr.) Vijay Dhir[2]
[1] Research Scholar, IKG PTU, Jalandhar, Punjab India,
[2] Professor CSE, Punjab, India

ABSTRACT

In this digital Era, parallel computing and distributed computing is playing a very vital role in multiprocessor environment to meet out the growing resource needs of users day by day. When several processes are being executed in multiprocessor environment, there are possibilities that some of the processors are being idle while others are being heavily loaded. In other words, we can say that there is total load imbalance in the system. The algorithm which is used to balance the load among processors is known as the load balancing algorithm. The load balancing algorithms can be divided into types : static and dynamic. In static load balancing algorithms, the information regarding allocation of processes to the different resources is based on the estimation of expected execution or completion time of the processes at compile time itself. Where in case of dynamic load balancing algorithms, this information is not available to algorithm prior to run time. So dynamic load balancing algorithm balances the load according to the currently prevailing information at runtime. Aim of this research paper is to find out some qualitative factors for comparison of both type of algorithms.

Keywords:- *Parallel Computing, Makespan, Latency, Throughput, Load Balancing Algorithms, Resource Utilization, Task Scheduling.*

I. INTRODUCTION

As we know that there is load imbalance problem in multiprocessor systems due to which some processors are heavily loaded while other processes are light reloaded that leads to results in poor utilization of resources available within the system. So load balancing algorithm which can remove the load imbalance from the system and its performance can be enhanced by maximizing the resource utilization, minimizing the completion time of processes, increasing its reliability etc. In other words, a load balancing algorithm is said to be most suited for multiprocessing environment which can provide us optimal results in load balancing scenario. The queue length of a processor can be taken as the measure of load on that processor. For example, heavily loaded processors will have high school length, lightly loaded processors will have medium queue length and idle once will be having zero queue length [1] [2] [4].

II. REVIEW OF THE LITERATURE

Over past several years, many researchers had developed many heuristic load balancing algorithms which are suitable for different computing environments. The system for which load balancing algorithm is to be used may be loosely coupled or tightly coupled [3]. In case of loosely coupled system, the computers or workstations are connected to each other through high speed network and communication between these computers take place through this network by passing messages to one another. And for tightly coupled system, the computers or nodes are connected to each other through a bus to share memory and secondary storage[5]. Here computers are having their its own local memories and global shared memory. Further, resources to be shared maybe heterogeneous or homogenous. In case of homogenous system, all the computers are of similar type in terms of processing power and storage capacity. In heterogeneous system, the computers are of different processing and storage capacities [6] [7]. Now the algorithm which is used to balance the load in such environment may require prior information at compile time is known as static or may require more information at runtime i.e. dynamic. In order to balance the load, the job transfer request may be initiated by the lightly loaded processor which is known as the serve initiated algorithm or it may be initiated by the heavily loaded processor which is known as the source initiative algorithm [8] [9]. In this paper we will discuss about static and dynamic load balancing algorithms which are explained in next two sections.

III. STATIC LOAD BALANCING ALGORITHMS (SLB)

In static load balancing algorithms, the allocation of processes or jobs to the resources is based on the expected execution / completion time of the processes, their inter-processor communication requirements etc. This information is estimated at the compile time of the algorithm. That is why, it is known as static algorithm. Further, it can be classified into two types: optimal and suboptimal. Optimal static load balancing algorithm have complete information about the probable execution times of processes on different resources very well so that optimal assignment of these processes to different resources of the system can be made on the basis of some criterion function. This ensures the optimal results at the end. Since in case of suboptimal static load

balancing algorithm, this complete information is not available which can result in optimal solution to the given load balancing problem. Here, in this method of load balancing relies on some Thumb Rule and heuristic approach is used to solve the specific load balancing problem[10] [11] [12].

IV. DYNAMIC LOAD BALANCING ALGORITHMS (DLB)

Because of fact that the static load balancing algorithms assume so much information in advance about the execution or completion time of the processes which may not be correct or even if it is correct, then lot of computations are to be carried out to find some optimal schedule. This limitation of static load balancing algorithms is taken care of by the dynamic load balancing algorithms which makes the decision of load transfer on this basis of current load conditions i.e. actual execution times of the processes. The workload is not assigned statically rather dynamically. So the workload can be reallocated depending upon the need if there are some heavily loaded processors and others are lightly loaded or even idle. So dynamic load balancing algorithm continuously monitor the performance of execution of the processes and if it notices that load imbalance has reached at some threshold level, it will reallocate or redistribute the load from heavily loaded nodes to the lightly loaded nodes so that optimum results can be obtained with minimum makespan. However this job migration for information exchange incur additional overhead to increase execution time of processes [9] [13] [14].

After discussing the static and dynamic load balancing algorithms, now it is time to identify the different factors based upon which, these two types of algorithms can be compared with each other.

V . FACTORS CONSIDERED FOR STATIC VS DYNAMIC DIFFERENTIATION

1. **Workload** The information pertaining to tasks allocation to different processors or resources is the workload to be handled by load balancing algorithms.

   In case of SLBA, this information is static i.e. resource allocation information is known to us at compile time prior to running of the algorithm.

   In case of DLBA, this information is dynamic i.e. resource allocation information may change as per demand of the load balancing needs at run time.

2. **Completion Time** It is the sum total of waiting time of the process in ready queue or Ready time of the resource for that process and execution time of the process when it is actually submitted to the CPU i.e.

   $$C_{ij}=E_{ij}+R_j$$

   Where C_{ij} is the completion time of task i on resource j, E_{ij} is the executuion time of task i on resource j and is the ready time of resource j for actual execution of task i.

   In case of SLBA, completion time of process is relatively less as there is no job migration from one resource to other.

   In case of DLBA, completion time of processes is higher relatively since there is pre-emption of tasks as per need of the load balancing.

3. **Makespan** It is the difference between earliest start time of the first process to the latest finish time of last process submitted for execution.

   In case of SLBA, makespan of algorithm is lower as processes are pre-allocated to resources and there is no process migration as well.

   However, in case of DLBA, makespan is more due to communication overhead and higher completion time of processes.

4. **Deadline meeting capability** It is the measure which ensures that the given process must be completed within pre-specified time constraint or deadline.

   SLBA is less promising to meet the deadline criterion as expected from it due to low relisbilty.

   DLBA needs to be more complex to meet the deadline constraints.

5. **Energy Consumption** It is the measure of energy or power consumption by load balancing algorithm in order to redistribute processes to resources to gain the optimal results.

SLBA has low energy consumption as there is no inter process communication and as a result no increase in execution or completion time of processes.

DLBA for solving the similar problem of size n may take little bit more makespan and accordingly more energy or power consumption in comparison to SLBA.

6. **Load Imbalance** It is the scenario in which some processors are heavily loaded while others are being idle. This results in poor resource utilization.

   In case of SLBA, Load imbalance occurs frequently because it tries to keep allocating to processors without checking their availability.

   In case of DLBA, load imbalance is tackled with some priority mechanism to schedule the processes on different resources. So, it provides us better resource utilization.

7. **Task Scheduling** It is the allocation of processes to different resources present in network.

   In case of SLBA, it is done at compile time as pre-planned.

   In case of DLBA, it is done at run time according to prevailing conditions to provide optimum results.

8. **Throughput** It's the measure of degree of resources utilization by load balancing algorithm for processing of tasks / jobs in efficient manner.

   In case of SLBA, throughput level provided is on lower side relatively.

   In case of DLBA, throughput level provided is on higher side comparatively.

9. **Latency** It denotes the delay made by scheduler to allocate a process to resource in comparison to its expected time of start of actual execution by CPU. It adds to the overall completion time of the process.
   In case of SLBA, latency occur high due to pre-planned allocation of resources which may be unavailable at run time.

   In case of DLBA, latency is not a big problem as algorithm keeps on trying to balance the load among processors.

10. **Communication Cost** It is the overhead incurred in message passing to share information among different nodes in network.

    In case of SLBA, communication cost is less due to pre allocation of resources.

    In case of DLBA, communication cost is more as this algorithm works dynamically to detect the load imbalance to make the efficient usage of resources.

11. **Job Migration** It's the switching of jobs from heavily loaded resources to idle ones to balance the load.

    In case of SLBA, job migration is not possible.

    In case of DLBA, job migration is frequent here algorithm works dynamically to detect the load imbalance to make the efficient usage of resources

12. **Estimation of times** Here we are talking about estimation process of expected execution / completion times of processes on different resources.

    In case of SLBA, expected execution / completion times of processes on different resources are predicted in advance at compile time itself.

    In case of DLBA, such prediction is not required at all.

13. **Complexity** It signifies the ease of implementation of the load balancing algorithm in simulation or real-time environment.

    In case of the SLBA, its implementation is easy due to its simplicity.

In case of DLBA, because of dynamically changing requirements to balance load among nodes, more complexity is being faced in its implementation.

14. **Fault Tolerance** In case of failure of some machines, overall performance of the load balancing algorithm is adversely affected.

    In case of SLBA, algorithm is less fault-tolerant.

    In case of DLBA, algorithm is flexible enough to cater the unfinished load of failed or faulty nodes.

15. **Scalability** It measure adaptability of load balancing algorithm with increasing problem size or recourses network to provide optimum results.

    In case of SLBA, algorithm is less scalable w.r.t. problem size.

    In case of DLBA, algorithm tackles well the growing needs of users requests to provide different services and thus more adaptive to scalability issue.

VI. TABULATION OF FACTS

The comparison work in tabular form is shown in Table 1.

TABLE 1: SLBA Vs DLBA

S. No.	Factor	Static Load Balancing Algorithms (SLBA)	Dynamic Load Balancing Algorithms (DLBA)
1	Workload	Known at compile time	As per system state at run time
2	Completion Time	Relatively Less	Relatively More
3	Makespan	Lower	Higher
4	Deadline Meeting Capability	Less Promising	More Promising
5	Energy Consumption	Lower based on Simplicity	Higher based on complexity
6	Load Imbalance	Higher	Lower
7	Task Scheduling	Fixed or pre-planned	Not fixed or no pre-planning at all
8	Throughput	Lesser	Higher
9	Latency	Higher	Lesser
10	Communication Cost	Lesser	Higher
11	Job Migration	Not at all	Possible
12	Estimation	Expected executions times are to be predicted in advance at compile time	No such prediction required
13	Complexity	Simple and easy to implement	More complex to implement
14	Fault Tolerance	May result in inaccurate results in case of machine failure	Take care of machine failures in to account
15	Scalability	Not adaptive to scalability	Scalability is not a problem

VII CONCLUSION & FUTURISTIC SCOPE

From above discussions, it is clear that factors like completion time, makespan, deadline meeting capability, energy consumption are in support of static load balancing algorithms while other factors detailed above are in support of dynamic load balancing algorithms.

In future, focus can be concentrated to quantify these factors for static and dynamic load balancing algorithms after using the proper simulated environment tool by producing the optimum experimental results.

REFERENCES

[1] Mohsen and Hossein Delda, "Balancing Load in a Computational Grid Applying Adaptive, Intelligent Colonies of Ants", Informatica 32 (2008) 327–335.

[2] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma, "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 2008.

[3] http://en.wikipedia.org/wiki/Load_balancing_%28computing%29.

[4] https://devcentral.f5.com/articles/intro-to-loadbalancing-for-developers-ndash-the-algorithms.

[5] S. F. El-Zoghdy and S. Ghoniemy, "A Survey of Load Balancing In High-Performance Distributed Computing Systems", International Journal of Advanced Computing Research, Volume 1, 2014.

[6] Mr. Amitkumar S Manekar, Mr. Mukesh D Poundekar, Prof. Hitesh Gupta and Prof. Malti Nagle, "A Pragmatic Study and Analysis of Load Balancing Techniques In Parallel Computing", International Journal of Engineering Research and Application, Vol. 2, Issue4, July-August 2012, pp.1914-1918.

[7] Daniel Grousa and Anthony T., " Non-Cooperative load balancing in distributed systems", Journal of Parallel and Distributing Computing, 2005.

[8] P. A. Tijare and Dr. P. R. Deshmukh, "Schemes for Dynamic Load Balancing – A Review", International Journal of Advanced Research in Computer Science and Software Engineering 3(6), June - 2013, pp. 688-692

[9] William Leinberger, George Karypis and Vipin Kumar, "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556- 2/00, 2000 IEEE.

[10] Sherihan Abu Elenin, and Masato Kitakami," Performance Analysis of Static Load Balancing in Grid", International Journal of Electrical & Computer Sciences, IJECS-IJENS, Vol: 11, No: 03,june 2011.

[11] http://www.loadbalancerblog.com/blog/2013/06/load-balancing-scheduling-methods-explained.

[12] Suriya and Prashanth, " Review of load balancing in cloud computing", IJCS, vol.10 issue.1, Jan 2013.

[13] Javed hussain and Durgesh kumar mishra, "Analysis and investigation of nearest neighbour algorithm for load balancing", International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 6, November – December 2013.

[14] Sherihan Abu Elenin and Masato Kitakami, "Performance Analysis of Static Load Balancing in Grid", International Journal of Electrical & Computer Sciences, Vol: 11, No: 03, June 2011.