Survey on Load Balancing Algorithms

Prof. Mrs. K.H.Wanjale[#1], Ashutosh Atre[#2], Tarun Kulchandra[#3], Ankit Singhania[#4], Shashikant Borude[#5]

[#]*Computer Department, Pune University*
Vishwakarma Institute of Information Technology
Pune, Maharashtra, (India)
[2]atre.ashutosh@gmail.com
[3]kulchandra.tarun@gmail.com

Abstract— **A distributed system is composed of a collection of autonomous computers which are connected through a network and distribution middleware and it enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. The load balancing problem becomes important when the demand for computing power increases. This problem of task scheduling and load balancing in distributed and parallel system are most challenging and important area of research in computer engineering. Load balancing is a method of distributing a workload across two or more resources so as to improve the performance of a parallel and distributed system. In this paper we present the performance analysis of various load balancing algorithms based on different parameters, considering two typical load balancing approaches static and dynamic. The analysis indicates that static and dynamic both types of algorithm can have advancements as well as weaknesses over each other. Deciding type of algorithm to be implemented will be based on type of parallel applications to solve. The main purpose of this paper is to help in design of new algorithms in future by studying the behavior of various existing algorithms.**

Keywords- **Load balancing (LB), workload, distributed systems, Static Load balancing, Dynamic Load Balancing**

I. INTRODUCTION

The study of distributed computing has grown tremendously to include large range of applications due to advents of micro-electronic technology. It has resulted in the availability of fast, inexpensive processors along with high speed network connections.

In parallel and distributed systems, workload or processing time is divided into tens or hundreds of interconnected computers across network. It has many advantages over traditional standalone computer system. Distributed systems provide resource sharing, which can be shared memory or I/O devices or time slice of workload. Major issue in distributed computing is to develop an effective technique for distributing workload. Technique must also take into consideration various other factors such as throughput of system, stability, fault-tolerance and resource utilization. Distribution techniques are performed at local as well as global level. Local scheduling is performed by operating system to time slices of the processor. On the other hand, global scheduling decides where to execute a process in multiprocessor distributed system. Global scheduling is carried out by single master processor while local scheduling is performed by each computer in the network. Global scheduling can be static or dynamic. Load sharing and load balancing are classified under dynamic global scheduling. Load sharing try to avoid the unshared state in processors which remain idle while tasks compete for service at some other processor whereas Load balancing also do the same but it goes one step ahead of load sharing by trying to maintain equal loads at all processors. Load balancing ensures that every processor in the system does approximately equal amount of work at any point of time. Processes can be moved from one node to other in the middle of its execution to achieve equal load at all nodes in system. Main assumption on which load balancing algorithms rely is that on hand information at each node is accurate. Load balancing can be centralized in one processor or distributed among all nodes that participate in load balancing.

II. PAPER ORGANIZATION

In our paper, we have tried to analyse various different load balancing algorithms. In first section Introduction is given, and then in III detailed introduction of Load balancing techniques, IV gives various comparison parameters and V discuss conclusion and future work.

III. LOAD BALANCING ALGORITHMS

Autonomy of the processors and the interprocessor communication overhead is the main challenge in the load balancing on multi computers. The best choice for running distributed and parallel program applications is distributed computing environment. In such applications, a large process is partitioned and then distributed among multiple nodes for parallel computation. It has been observed that in a distributed system the probability of host remaining idle while other host has multiple jobs queued up can be very high. Load balancing can be improved in such situations. Performance

© 2013 JCT JOURNALS. ALL RIGHTS RESERVED

can be improved by either transferring jobs from the currently nodes having heavy workload to the nodes having light workload or distributing load evenly among the nodes. Such algorithms which help to achieve the above said goals are known as load balancing algorithms.

They are classified according to workload in their queues as heavily loaded, lightly loaded and idle processor. Workload at any processor is based upon queue length of CPU.

Based on the considerable amount of work done over the past few years on load balancing the general problem can be studied in different types of computing environments, using different techniques and at different levels. The system can be classified as loosely coupled or tightly coupled. Loosely coupled multiprocessor system are based on multiple standalone single or dual processor commodity computers interconnected via high speed communication system, whereas tightly coupled multiprocessor system contains multiple CPU that are connected at the bus level , these CPU's may have access to the central shared memory or they may participate in a memory hierarchy with both local and shared memory . If the resources to be shared are of the same type and capacity then it is called as a homogeneous system, otherwise it is referred as heterogeneous system. The algorithms used for load balancing which require no information, or only information about individual jobs then it is called as static algorithm or the algorithms which makes the decisions based on the current load situation is known as dynamic algorithm. The transfer of a job may be initiated by the originating host (source-initiative algorithm), or by the target host (server-initiative algorithm). The execution unit that is to be transferred may range from complete jobs submitted by the users, or individual processes, or even smaller program modules. These units may also be the components of parallel computations with specific communication requirements. the transfer of jobs can be done before the start of execution (initial job placement) or it may also be allowed during its execution (process migration)

The scope of our paper is to analyze the performance of various load balancing technique and identification of the qualitative parameters to compare both forms of load balancing algorithms (static and dynamic) in the distributed computing system.

Depending on the current state of the system, load balancing algorithms can be divided into 2 basic categories - static and dynamic.

A. Static load balancing

Static load balancing schemes takes into consideration priori knowledge of the applications along with the statistical information about the system. In this method the performance of the processors is determined at the beginning of execution. And the work load is assigned by the master processor depending upon their performance. The functionality of slave processors is to calculate the allocated work and submit their result to the master. static load balancing methods are always non-preemptive. The goal of static load balancing method is to reduce the execution time, minimizing the communication delays. A general disadvantage of all static schemes is that the final selection of a host for process allocation is made when the process is created and cannot be changed during process execution to make changes in the system load.

Static load balancing can be classified into two categories – optimal and sub-optimal.

1) Optimal SLB: In this type of algorithm when all the information regarding the state of the system as well as the resource needs is known an optimal assignment can always be preferred based on some criterion function. Examples of such optimization measures are minimizing total process completion time, maximizing utilization of resources in the system, or maximizing system throughput. For example simulated Annealing (SA) and genetic algorithms (GA's) are optimization techniques.

2) Sub-Optimal SLB: Sub-optimal methods can be applied for some computations where optimal solution does not exist; it relies mainly on the rules-of thumb and heuristics to guide a scheduling process. List scheduling is the most popular technique despite of poor performance in high communication delay situations. Lot of static algorithms, taking into account their optimal and

© 2013 JCT JOURNALS. ALL RIGHTS RESERVED

sub-optimal nature, has been suggested by researchers so far. This includes approximate algorithms like Solution space Enumeration and search, Graph theoretic approach, Mathematical programming and queuing theoretic. Some others are round-robin algorithm, recursive-bisection algorithm, heuristic algorithms and randomized algorithms.

B. Dynamic load balancing

In this type of algorithm the workload is distributed at runtime amongst the processors. The main drawback with SLB algorithms is that they assume too much job information which may not be available in advance and even if it is available, complex computation may needed to obtain the optimal schedule possible. Due to this drawback more research is done in DLB than SLB algorithms. DLB algorithms take decisions based on current load condition at execution time. So here workload is not determined statically but can be redistributed dynamically depending on current workload.

Dynamic Load Balancing algorithms involve continuous monitoring of workload on all the processors. When load imbalance reaches a particular level (predefined level) then workload is redistributed. Continuous monitoring needs extra CPU cycles, so care must be taken to invoke it only when it is necessary. Redistribution of workload creates extra overhead at execution time.

Dynamic load balancing algorithm takes into considerations following issues:

1) Priority assignment policy: It is used to determine the priority of execution of local and remote processes at any particular node.

2) Load estimation policy: It is responsible for determining the estimation of the workload of a particular node of the system.

3) Migration limiting policy: It deals with determination of the total number of times a process, can migrate from one node to another.

4) Process transfer policy: It aims to determine whether to execute a process locally or remotely.

5) State information exchange policy: It determines exchange of the system load information among the nodes.

6) Location Policy: The part of the load balancing algorithm which selects a destination node for a transferred task is referred to as location policy or Location strategy.

IV. QUALITATIVE PARAMETERS

A. Overload Rejection

If Load balancing algorithm cannot take additional overhead then it should be rejected. When overload situation ends rejection methods must be stopped.

B. Reliability:

This factor considers reliability of algorithms during machine failure situations. Static load balancing algorithms are less stable than dynamic algorithms as no run time decisions are taken in static algorithms.

C. Adaptability

This factor checks whether the algorithm is adaptive to dynamic situations. Static load balancing algorithms are not adaptive as this method fails in varying nature problems while dynamic load balancing algorithms are adaptive towards every situation.

D. Stability

It can be characterized in terms of the delays in the transfer of information between processors and algorithms. Static load balancing algorithms are considered as stable as no information regarding present workload state is passed.

E. Fault Tolerance

It enables an algorithm to continue operating properly in the event of some failure. If the performance of algorithm directly depends on the seriousness of the failure, even a small failure can cause total failure in load balancing.

F. Resource Utilization

Resource utilization includes automatic load balancing. Static load balancing algorithms have lesser resource utilization whereas dynamic load balancing algorithms have relatively better resource utilization as dynamic load balancing take care of the fact that load should be equally distributed to processors so that no processors should sit idle.

© 2013 JCT JOURNALS. ALL RIGHTS RESERVED

G. Process Migration

Process migration parameter provides when does a system decide to export a process? It decides whether to create it locally or create it on a remote processing element. The algorithm is capable to decide that it should make changes of load distribution during execution of process or not.

H. Cooperative

This parameter gives that whether processors share information between them in making the process allocation decision other are not during execution. What this parameter defines is the extent of independence that each processor has in concluding that how should it can use its own resources. In the cooperative situation all processors have the accountability to carry out its own portion of the scheduling task, but all processors work together to achieve a goal of better efficiency.

TABLE I

COMPARISON OF SLB ALGORITHMS

PARAMETERS	ROUND ROBIN	RANDOM	CENTRAL MANAGER	THRESHOLD
Overload Rejection	No	No	No	No
Fault Tolerant	No	No	Yes	No
Forecasting Accuracy	More	More	More	More
Stability	Less	Less	Less	Less
Centralized/ Decentralized	D	D	C	D
Cooperative	No	No	Yes	Yes
Process Migration	No	No	No	No
Resource Utilization	Less	Less	Less	Less

TABLE II

COMPARISON OF DLB ALGORITHMS

PARAMETERS	LOCAL QUEUE	CENTRAL QUEUE
Overload Rejection	Yes	Yes
Fault Tolerant	Yes	Yes
Forecasting Accuracy	Less	Less
Stability	Small	Small
Centralized/ Decentralized	D	C
Cooperative	Yes	Yes
Process Migration	Yes	No
Resource Utilization	More	Less

V. CONCLUSION

In Load balancing algorithms, the workload is either assigned statically (compile time) or dynamically (run time). As it can be seen from the above comparison, static load balancing algorithms are more stable as compared to dynamic load balancing algorithms. Moreover, we can easily predict the behaviour of static load balancing algorithms. This paper highlights the major types of load balancing, their algorithms and a comparative study of them.

ACKNOWLEDGMENT

The authors are grateful to Prof. Mrs. K.H. Wanjale for her discussions and advices.

REFERENCES

[1] Zhong Xu, Rong Huang, "Performance Study of Load Balancing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[2] Y. Wang and R. Morris, "Load balancing in distributed systems," IEEE Trans. Computing. C-34, no. 3, pp. 204-217, Mar. 1985.

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

[4] M. Nikravan and M. H. Kashani, "A Genetic Algorithm for Process Scheduling in Distributed Operating Systems Considering Load balancing", Proceedings 21st European Conference on Modelling and Simulation (ECMS), 2007.

[5] S. Malik, "Dynamic Load Balancing in a Network of Workstation", 95.515 Research Report, 19 November,2000.[8] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[6] Hendra Rahmawan, Yudi Satria Gondokaryono, "The Simulation of Static Load Balancing Algorithms", 2009 International Conference on Electrical Engineering and Informatics, Malaysia.

[7] Daniel Grosua, Anthony T. and Chronopoulosb,"Non-cooperative load balancing in distributed systems", Elsevier, Journal of Parallel and Distributed Computing, 2005.

[8] S.P. Dandamudi, "Sensitivity evaluation of dynamic load sharing in distributed systems", IEEE Concurrency 6 (3) (1998) 62-72.

[9] Hisao Kameda, El-Zoghdy Said Fathyy and Inhwan Ryuz Jie Lix, "A Performance Comparison of Dynamic vs. Static Load Balancing Policies in a Mainframe { Personal Computer Network Model", Proceedings of the 39th IEEE Conference on Decision and Control, 2000.

[10] L. Rudolph, M. Slivkin-Allalouf, E. Upfal. A Simple Load Balancing Scheme for Task Allocation in Parallel Machines. In Proceedings of the 3rd ACM Symposium on Parallel Algorithms and Architectures, pp. 237-245, July 1991.

© 2013 JCT JOURNALS. ALL RIGHTS RESERVED

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

© 2013 JCT JOURNALS. ALL RIGHTS RESERVED