# Review of Parallel Computing using Load Balancing for Sheared Memory

**Sanjeev Shukla[1] Nisha Patil[2] Narendra Joshi[3]**
[1,2,3]Department of Computer Engineering
[1,2,3]Sandip University, Nashik, Maharashtra, India

*Abstract*— In order to expand the execution of a distributed application in any variety of distributed architecture, the algorithm of Load balancing was developed. These algorithms are used in parallel systems to increase performance of the parallel systems which ensures all the computing cores are reserved and at no single time the core will be overloaded even as the other cores remain idle. The concept of balancing load on servers using different types of load balancing techniques is the advancement of load balancing algorithm. It is used to enhance the performance of system through a redistribution of load between processor or server machines. It will be practically possible to execute the applications on any number of machines of worldwide distributed systems. Static or dynamic load balancing are categorized in Load balancing algorithms at compile time the static load balancing algorithms distributes the tasks to the processing elements, while dynamic algorithms bind tasks to processing elements at run time. In a distributed computing background this thesis describes the concepts of a number of load balancing techniques. It also describes a variety of types of load balancing strategies, their qualities, demerits and comparison.

***Keywords:*** Parallel Computing, Load Understanding Techniques

## I. INTRODUCTION

The computing power of sequential computers have increased through parallel and distributed computing. The distributed systems used more than one processor to execute the program. Workload for a processor is the period for processing time which takes to perform all processes assigned to that processor.

Load balancing involves the allotment of jobs throughout an amount of processors, thus improves performance by increasing throughput exclusive of need to obtain additional or faster computer hardware. Load balancing is to make sure that every processor in the system do the similar amount of work at any point of time.

To complete the processes in shortest possible time there is problem to complete a balance in load distribution between processors. A Server can be handled with certain number of clients because a Server procedure has partial resources which involve memory, hard disk, and CPU speed.

A server will be loaded and that may lead to slow down the performance, hang and crash issues, when the number of clients is increased. By keeping the copies of servers and distribute the load among them, we can balance the load on a server.

In distributed computing systems to improve the quality of service, Load balancing is very essential. To avoid resource blocks as well as to better utilize available resources, the incoming requests are optimally spread among existing system resources.

In order to address increased loads, Load balancing provides horizontal scaling e.g., adding computing resources. Load balancing increases availability, improves performance by increasing reliability, increases throughput, maintains, stability, optimizes resource utilization and provides fault tolerant capability.

The speedup is achieved as well as the performance of the parallel system through the load balancing of an application

Redistribution of balanced work-load by means of tasks and minimizing the inter process communication needs with optimal resource utilization and job response time are the primary optimization objective of load balancing.

By equalizing the workloads of processing elements is the aim of load balancing, improves the performance of parallel computers

The main goals of a load equalization algorithmic rules are as follows:

*1) Performance Improvement:*

Achieve a bigger overall improvement in system performance at an affordable value, e.g., scale back task latent period whereas keeping acceptable delays.

*2) Job Equality:*

All jobs within the system ought to treat equally.

*3) Fault Tolerance:*

To own performance strength underneath restricted failure within the system.

*4) Modifiability:*

Have the potential to change itself in accordance with any changes or enlarge within the distributed system configuration.

*5) System stability:*

The capability to reason for backup things like sudden flow of arrivals in order that system performance doesn't decline on the far side a precise threshold whereas preventing nodes of the distributed system from disbursal an excessive amount of time passing up jobs among themselves rather than death penalty of these jobs.

## II. TYPES OF LOAD BALANCING STRATEGIES

Various ways and algorithms are planned, enforced and classified in a very range of studies. Broadly, load leveling may be a reasonably programming optimization downside.

 

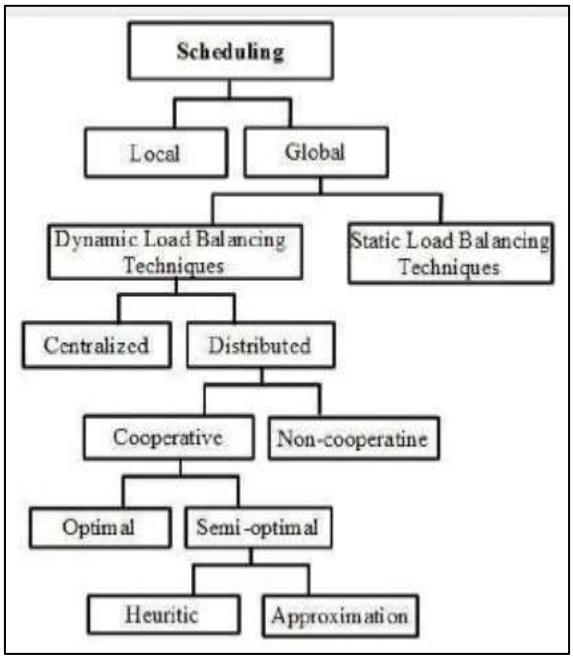


Fig. 1: Different types of load balancing techniques

The load equalization strategy is also determined by scrutiny, like with an oblong lattice of grid points split into smaller rectangles, in order that the load equalization drawback is solved before the program is written. looking on the data utilized in load equalization call, it is often divided into 2 broad classes i.e. international or native policies. In international policies, the load balancer uses the performance profiles of all on the market workstations. In native policies workstations area unit divided into totally different teams. The profit in a very native theme is that performance profile data is just changed at intervals the native cluster. The selection of a world or native policy depends on the behavior associate degree application can exhibit. Looking on the time to bind the tasks to process components, load equalization algorithms are often any categorized as static or dynamic. The non-trivial static load equalization algorithms distribute the tasks to process components at compile time, whereas dynamic algorithms bind tasks to process components at run time. Static load equalization algorithms trust the estimate execution times of the tasks and inter-process communication demand. it's not satisfactory for parallel programs that square measure of the dynamic and/or unpredictable kind. Consequently, in dynamic load reconciliation, tasks square measure generated and destroyed while not a pattern at run time. Further, counting on the situation wherever the load reconciliation call is administrated i.e. the resident of the load balancer, these is any classified either as centralized or distributed load reconciliation. The case once the load balancer resides at the master node is named centralized load reconciliation policy, otherwise if constant resides in the slightest degree the workstations into consideration is named the distributed load reconciliation policy.

Further, in quasi-dynamic, the circumstances crucial the optimum balance modification throughout program execution, however discretely and often. as a result of the modification is distinct, the load balance downside and thence its answer stay an equivalent till following modification. If these changes square measure sporadic enough, any savings created within the succeeding computation compose for the time spent determination the load leveling downside. The distinction between this and therefore the static case is that the load leveling should be applied in parallel to forestall a sequent bottleneck. The scope of this paper is restricted to dynamic load leveling solely.

## III. Dynamic Load equalization Techniques

Short Discussion the circumstances crucial the best balance amendment oft or ceaselessly throughout execution, so the value of the load equalization calculation once every amendment ought to be reduced additionally to optimizing the cacophonic of the particular calculation. This implies that there should be a choice created each therefore typically to make a decision if load equalization is important, and the way abundant time to pay on that. Dynamic (or adaptive) policies, on the opposite hand, deem recent system state data and confirm the task assignments to processors at run time.

Hence, they're a lot of attractive from a performance purpose of read within the dynamic approach, the load settlement choices are supported the present state of the system; tasks are allowed to move dynamically from associate overloaded node to associate under-loaded node to receive quicker service. This ability to react to changes within the system is that the main advantage of the dynamic approach to load reconciliation. A dynamic load understanding formula consists of 4 parts, Load measure rule, associate data Exchange rule, associate Initiation rule and a Load understanding Operation.

## IV. Policies in Load equalization Algorithms

Load equalization algorithms are often outlined by their implementation of the subsequent policies

1) Information policy: specifies what work info to be collected, once it's to be collected and from wherever.
2) Triggering policy: determines the acceptable amount to begin a load equalization operation.
3) Resource kind policy: classifies a resource as server or receiver of tasks in line with its availableness standing.
4) Location policy: uses the results of the resource kind policy to search out an acceptable partner for a server or receiver.
5) Selection policy: defines the tasks that ought to be migrated from full resources (source) to most idle resources.
6) Problems in Performance analysis the most objective of load equalization ways is to hurry up the execution of applications on resources whose employment varies at run time in unpredictable manner.

Hence, it's vital to outline metrics live the resource employment:

1) a way to measure resource workload?
2) What criteria square measure holding to outline this workload?
3) a way to avoid the negative effects of resources dynamicity on the workload, and
4) a way to take into consideration the resources heterogeneousness so as to get an instant average employment representative of the system?

 

## V. COMPARATIVE ANALYSIS OF ASSORTED DYNAMIC

Load equalization Techniques during this section we tend to are about to gift characteristic analysis of the various dynamic load equalization techniques supported the placement of deciding, the knowledge used for the choice creating method, quantify-ability issue, and therefore the overhead of exchanging the profile data.

### *A. Centralized Dynamic Load Understanding Techniques*

During this technique, the responsibility of the Load understanding demand remains with the master node and therefore the data used for the load understanding is gathered from the remaining (slave's) nodes on either on demand basis or once a precise predefined quantity of mounted quantity, or perhaps the knowledge could also be gather only any amendment happens in their operating stage. The noticeable purpose is since the knowledge isn't send at random, the unessential traffic over the interconnection network reduced. Additionally, no unessential profile data is exchange overhead is encountered. But the measurability remains restricted with this system.

### *B. Distributed Non-cooperative Dynamic Load reconciliation Techniques*

In distributed non-cooperative dynamic programming techniques, the responsibility of the load reconciliation techniques distributed over all the operating nodes i.e. workstations rather than the master node. The work load data is gathered supported the on-demand basis i.e. whenever any node changes its current balanced operating state to full state, the precise node mat distribute the load data to schedule to load to be balanced or alike. this system provides moderate measurability over the centralized theme. However, since the load data must distribute over many operating nodes before rescheduling the present overload, this might increate the traffic in interconnection network additionally to the restricted data exchange overhead

### *C. Distributed Cooperative optimum Dynamic Load understanding Techniques.*

In distributed cooperative optimum dynamic load reconciliation techniques, not like distributed non-cooperative dynamic programming techniques the responsibility of load reconciliation call is scattered over all the workstations instead of master node. Further, during this case too load reconciliation data strategy is demand driven not like the case of non-cooperative dynamic programming techniques with the exception of getting average overhead throughout exchange of profile data. This system will offer moderate measurability.

### *D. Distributed Cooperative Semi-Optimal Heuristic Dynamic Load Understanding Techniques*

Its not like the previous 2 techniques, in distributed cooperative semi-optimal heuristic dynamic load reconciliation techniques the responsibility load reconciliation call is allotted over all the workstations conjointly with demand driven data strategy and average profile data exchange overhead and moderate measurability.

### *E. Distributed Cooperative Semi-optimal Approximation Dynamic Load reconciliation Techniques*

during this techniques too, the load reconciliation responsibility, data strategy and measurability remains same not like within the case of distributed cooperative partial best experiential dynamic load understanding techniques i.e. to workstations, demand driven and moderate measurability severally with the exception of intense far more profile data exchange overhead increasing the traffic over the interconnection networks.

## REFERENCES

[1] Frankline Makokha , Okelo-Odongo " A Review of Dynamic Load Balancing Algorithms for Shared Memory Parallel Computing", International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 07 – Issue 01, January 2018
[2] Firoj, A. & Khan, Z. (2012) The Study on Load Balancing Strategies in Distributed computing System: International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.2.
[3] Ros, A. (Ed. (2010)) Parallel and Distributed Computing. Croatia: In-Tech.
[4] LeMair W.H. and Reeves P. A. (1993) Strategies for Dynamic Load Balancing on Highly Parallel computers: IEEE Transactions on Parallel and Distributed Systems. Vol. 4, No. 9.
[5] Pearce, O., Gamblin, T., Supinskiy, B., Schulzy, M., and Amato, M. (2012) Quantifying the Effectiveness of Load Balance Algorithms.US: Department of Energy
[6] Horton, G. (1993) A multi-level diffusion method for dynamic load balancing. Erlangen: Elsevier Science Publishers B.V.
[7] Mr. Sunil Kumar Pandey1, Prof. Rajesh Tiwari2 "The Efficient load balancing in the parallel Computer, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 4, April 2013, ISSN (Print) : 2319-5940 ISSN (Online) : 2278-1021

 