

Research Article

# Load Balancing Dynamically in Distributed and Parallel Computing System with Process Execution Time

Nitin R. Chopde<sup>†\*</sup>

<sup>†</sup>Computer Science & Engineering Department, G.H. Raisoni College of Engg. & Management, SGBAU Amravati, India

Accepted 05 Jan 2015, Available online 01 Feb 2015, Vol.5, No.1 (Feb 2015)

## Abstract

To improve distributed and parallel system performance the load balancing techniques are extensively using that is a process of redistribution of work load among multiple servers in distributed computer systems. In recent research and literature, various approaches, methods and techniques have been proposed to achieve this task. There is an increasing trend towards the use of parallel and distributed systems because of outstanding progress in computer technology and an ever-rising demand for high-speed processing able to support the distributed mode. One of the important instant of any system using parallel computing is the load balancing in which the balance of workload among the system's entire server is designed. In this proposed algorithm, trying to implement a new load balancing algorithm which has new parameter consideration and basis of that centralized nodes(server) balancing workload when it get overloaded will be implement.

**Keywords:** Centralized server, Overloaded, task

## 1. Introduction

In parallel and distributed systems is speedily increasing because of increasing advance in scientific endeavor and the necessity of high-speed processing that may even tend toward the mode of distribution. In many systems a number of servers but may be the probability of a processor being idle in the system and other processors having a queue of tasks at hand is very high. So there is essential of the standardized distribution of workload among these servers.

In such system important factor is speed by which they the processing tasks or incoming request, can becomes one of the issue, but in parallel systems type the capital needed to smoothly drive systems seems logical only on condition that the workload of the system be distributed suitably among the processors. The solution to this issue load balancing algorithm or techniques this aims becomes practical in distributed systems and parallel systems.

To improve system performance by preventing some servers from being overloaded with work while others find no work to do is basic work of load balancing. In this proposed work implementation of algorithm for the balancing load in distributed and parallel systems, the implementation of suitable algorithm which includes various parameters to improve load balancing

## 2. Algorithms works for Load balancing

The most important things of any load balancing algorithm is to is its ability to forward service requests to the most appropriate servers for efficient use of system. The load balancing in distributed and parallel systems is not only reducing execution time of the systems but also maximizing resource utilization (E. Waraich *et al*, 2006).

The Load balancing algorithms are categories on the basis of strategy they used, that is static and dynamic load balancing algorithms. The strategy of static load balancing, basically an judgment of the time required to complete any given task, tasks are assigned. The dynamic load balancing looks at the load balancing problem more realistically by assuming little information is available before any assignment is made. It does not guess any knowledge of where a certain process will finally execute (Ali M. Alakeel *et al*, 2010) .There is no decision at this type regarding a shift of task from one processor to another during the execution time. The various static loads balancing algorithm has been proposed which are

### 2.1 Load balancing through Round Robin algorithm

It is simple algorithm that distributes each new task to next available server. The round robin algorithm can be easy to implement and it is suitable for effective for distribution of the workload among servers subject to

\*Corresponding author: Nitin R. Chopde

the condition that each server has equal processing capacity

### 2.2 Load Balancing approach of weighted round robin with response-time as weight

This is advancement in round robin algorithm. It improves the properties of the round robin algorithm. It finds out which server will allocate the next task. It consistently monitoring response time for servers connected in the system.

### 2.3 Load balancing through fewest connections with limits

This algorithm continuously keeping a record of how many task executing by each server is currently providing then it determines which server gets the next tasks. The overloaded processors are processor which makes huge number of tasks in waiting. The under loaded processors which have a small number of tasks in waiting. Idle processor is one which does not have any task to be done. The DLB algorithms are categorized based on a structure in (A. Osman, H. Ammar *et al*, 2002). When such algorithms are used or when to instigate the work also whether the algorithm is carried out sporadically this classification can be seen in (A. Osman, H. Ammar *et al*, 2002). The server with smaller number of tasks gets the next request(process). In dynamic load balancing algorithm it is distribution takes place at runtime among the processors. The static load balancing differs from dynamic load balancing algorithms in processing. There are various strategies have been proposed for classification of load balancing algorithm. In each of these classified algorithms different definition of load balancing algorithms has been proposed. These are various proposed methods and strategies example can be described in (H. Kuchen *et al* 1990; R. Lüling *et al*, 1992). The dynamic load balancing algorithms (DLB) works with shifts task between the processors decide by the load status at any given moment (A. Grama, *et al* 2003; A. Osman, *et al*, 2002; M. Wu, *et al*, 2002). The accessible load balancing algorithms for parallel system are Random based, Central based and Rendezvous based which can be seen in (C. Fonlupt, *et al* 1995; P. Marquet, *et al*, 1996)]. The basis of parameters of quality of which nature and overhead-associated can be considered for differentiation of algorithm (E. Waraich *et al*, 2006). The mainly, processors in parallel and distributed systems in relation to load balancing algorithms are divided into three groups based the workload level.

To advance the performance of a program, performance bottlenecks must be situated, the causes recognized, and the solutions anticipated and implemented. Therefore, performance issues specifically concentrated on focusing on the first two components leaving the rest to the programmers. There are several schemes has been proposed, which is often the case, programmers need to exhaustively try

each to desire the superlative for performance. Significant efforts are required to implement each of the options. In the next component, after presenting the new algorithm for balancing load, it will be studied based on categorization in in (A. Osman, H. Ammar *et al*, 2002). To reach to the higher system throughput and cut down the client's feedback time, the load balancing algorithm adopted both static and dynamic parameters so it achieve the performance.

### 2.4 Static performance parameter

Every web server includes resources (software and hardware) this parameter decide its capability. When processing of the web server cluster, these parameters are everlasting, therefore they are also called static performance parameters. CPU processing ability in the cluster system, it monitors inspects the number and type of CPU, according to previous used value, the server with a controlled CPU processing ability value and the process of sever monitor inspects the size of computer's physical memory and virtual memory, so as to acquire the memory parameter.

### 2.5 Dynamic performance parameter

During the processing of cluster system, each web server's load is changing as time goes on, the system has to estimate the load-balance according to the real-time server load, and these are called dynamic performance parameters.

Processor utilization ratio: it can reflect the busyness degree. The process of sever monitor inspects the CPU utilization ratio term, so as to confirm the CPU's load. Memory utilization ratio is the size of the server memory changes as the system runs. The process of sever monitor inspects the utilization ration of physical memory term, so as to confirm the server memory's load. Network flow: the network data are mainly transferred through TCP mode in cluster system, the process of sever monitor inspects the packages received and sent by server term, so as to get the server's load. In dynamic load balancing algorithm load distribution decisions based on the current work load at each node in any system. Consequently, this algorithm must provide a mechanism for collecting and managing system status information. The part of a dynamic load balancing responsible for collecting information about nodes in the system is referred to as information strategy in the literature. Also, a dynamic load balancing algorithm must include a mechanism to assist each node in deciding which job is eligible for load balancing. The part of a dynamic load balancing algorithm which selects a job for transfer from a local node to a remote node is referred to as transfer strategy. Furthermore, a dynamic load balancing algorithm must provide a mechanism on which a destination node for a transferred job is determined. The part of a dynamic load balancing algorithm which selects a destination node for a transferred task is

referred to as location strategy (Ali M. Alakeel, *et al* 2010). In earlier researchers had examined methods for load balancing in pipelined term-distributed architectures, and propose a suite of techniques for reducing net querying costs. In particular, they explored the load distribution behavior that pipelining displays and show that the imbalances can be addressed by techniques that include predictive index list assignments to nodes, and selective index list replication (Alistair Moffat, *et al* 2006). The algorithm has been proposed for load balancing strategy always converges, and tends to be in a steady state in a negligible processing time. This include, the load status and the locations of the nodes regarding the system's topology are irrelevant to load balancing process (Ahmad Dalal'ah, *et al* 2006).

One of the dynamic load balancing algorithms is Central Queue Algorithm (William Leinberge, *et al* 2000) works on the principle of dynamic distribution. It stores new activities and unfulfilled requests as a cyclic FIFO queue on the main host. Each new activity arriving at the queue manager is inserted into the queue. Then, whenever a request for an activity is received by the queue manager, it removes the first activity from the queue and sends it to the requester. If there are no ready activities in the queue, the request is buffered, until a new activity is available. If a new activity arrives at the queue manager while there are unanswered requests in the queue, the first such request is removed from the queue and the new activity is assigned to it. When a processor load falls under the threshold, the local load manager sends a request for a new activity to the central load manager.

The central load manager answers the request immediately if a ready activity is found in the process-request queue, or queues the request until a new activity arrives. In another case of local queue algorithm (William Leinberge, *et al* 2000) which is also dynamic load balancing algorithm there is dynamic process migration support. The basic idea of the local queue algorithm is static allocation of all new processes with process migration initiated by a host when its load falls under threshold limit, is a user-defined parameter of the algorithm. The parameter defines the minimal number of ready processes the load manager attempts to provide on each processor. Initially, new processes created on the main host are allocated on all under loaded hosts.

The number of parallel activities created by the first parallel construct on the main host is usually sufficient for allocation on all remote hosts. From then on, all the processes created on the main host and all other hosts are allocated locally. When the host gets under loaded, the local load manager attempts to get several processes from remote hosts. It randomly sends requests with the number of local ready processes to remote load managers receives such a request, it compares the local number of ready processes with the received number. If the former is greater than the latter, then some of the running processes are

transferred to the requester and an affirmative confirmation with the number of processes transferred is returned.

### 3. The Proposed load balancing algorithm

At the very start of this algorithm it should be state that a centralized server with defined maximum number of neighbors (servers). The aim of proposed algorithm is that server does not get overloaded and when it gets overloaded with incoming task it algorithm begin distribution of works based on certain parameter. In other words centralized server takes decision of distribution of workload on specific calculated parameters. The algorithm proposed to balance workload load among servers.

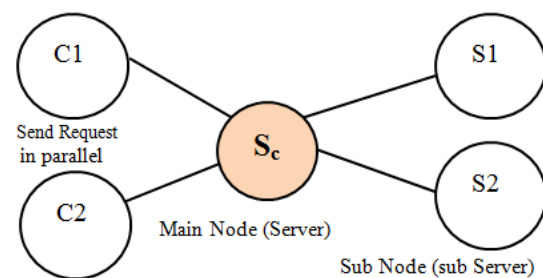


Fig.1 Simulation model of proposed system

There are two clients and two subserver in this system shown in fig1. The algorithm is as follows; as soon as a centralized server receives batch of queries from client it decides whether to distribute queries or execute it by itself. This decision depends upon the parameter overhead which is round trip time of packet between server and subserver. In order to determine the overhead time of a packet from main server to subserver, we are using algorithm which gives average time for roundtrip from centralized server to subserver. Centralized server may receive multiple requests depending that whether the it is overloaded or not it takes decision of distribution of the request to the subservers. Suppose consider  $S_c$  is main server and  $S_{b1}$ ,  $S_{b2}$  are the subserver which are balancing load when centralized server gets overloaded.

The centralized server takes decision of distribution when it gets overloaded by received queries from the client. When server  $S_c$  gets overloaded under such situation our load balancing algorithm works.

The idea load balancing algorithm is perspective from execution time of queries.

When server  $S_c$  receives batch of queries it calculate the average time of all received queries since execution time of basic queries is already calculated. Then it compare with overhead time between  $S_c$  and  $S_b$ .

A new Load balancing algorithm as follows

**Process 1**-Representing main server  $S_c$  and its subserver by  $S$  ( $S_{b1}$ ,  $S_{b2}$  ... $S_{bn}$ ) and Client requests

(task)Tc. This helps to constructs structure with main (Centralized) server and subserver before proceeding.

**Process 2-** Client queries construct with batches which passes multiple requests Tc to Sc.

**Process 3-** Sc decides the distribution of incoming queries on the basis of certain evaluations.

**Process 4-** Sc calculates overhead between Sc and Sb. Let say overhead time is Tovd.

**Process 5-** Sc calculates average of execution time of received request Let say Aq and

**Process 6-** compare Aq with Tovd. If  $Tovd > Aq$  (Average time of received query) Queries executed by main server Sc else No. of queries executed by each server is  $Td = Tq/Sn$

**Process -7** Td is queries get executed by each server. Repeat until all servers get occupied by the workload.

## Conclusions

The proposed algorithm for load balancing in distributed and parallel systems environment works with new parameter consideration. In this paper suggested new load balancing algorithm which includes distribution of tasks on basis of dynamic processing.

The proposed system tries to improve efficient work of main nodes that helps for performance for load distribution between the main nodes and sub nodes. This dynamic load balancing algorithm suggest how sharing takes place when centralized nodes accept the process, it makes every server gets occupied by workload that workload distribute uniformly among sub servers to achieve load balancing in distributed and parallel computer system.

## References

- A. Chhabra, G. Singh, E. Waraich, B. Sidhu, G. Kumar (November 2006), *Qualitative parametric comparison of load balancing algorithms in parallel and distributed computing environment*, Proc. World Academy of Science, Engineering and Technology (PWASET), vol. 16.
- C. Fonlupt, P. Marquet, J. Dekeyser (September 1995), *Analysis of synchronous dynamic load balancing algorithms*, (PARCO'95) Parallel Computing: State-of-the Art Perspective Advances in Parallel Computing, Elsevier Science Publishers.
- C. Fonlupt, P. Marquet, J. Dekeyser (1December 1996), *Data-parallel load balancing strategies*, West Team, High Performance Computing, Laboratoire d'Informatique Fondamentale de Lille, Université de Lille.
- A. Grama, A. Gupta, G. Karypis, V. Kumar (January 2003), *Introduction to parallel computing*, Second Edition, Addison Wesley.
- H. Kuchen, A. Wagener (1990), *Comparison of dynamic load balancing strategies*, Rwth Aachen, Department of Computer Science, Technical Reports, Aachener Informatik-Berichte (AIB).
- Ali M. Alakeel(April 2010), *Load Balancing in Distributed Computer Systems*, IJCSIS, International Journal of Computer Science and Information Security, vol.8.
- R. Lüling, B. Monien, F. Ramme(1992), *A study on dynamic Load balancing algorithms*, Proc. the 3rd IEEE SPDP, pp. 686-689.
- A. Osman, H. Ammar(2002), *Dynamic load balancing strategies for parallel computers*, Scientific Annals Journal of Cuza University International Symposium on Parallel and Distributed Computing (ISPDC), vol. 11, pp. 110-120.
- M. Wu (February 1997), *On runtime parallel scheduling for processor load balancing*, IEEE Transactions on Parallel and Distributed Systems Press Piscataway, vol. 8, pp. 173-186., 453
- Ali M. Alakeel(June 2010), *A Guide to Dynamic Load Balancing in Distributed Computer Systems*, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6.
- Alistair Moffat William Webber Justin Zobel (2006), *Load Balancing for Term Distributed Parallel Retrieval*, ACM SIGIR.
- Ahmad Dalal'ah (April 2006), *A Dynamic Sliding Load Balancing Strategy in Distributed Systems*, The International Arab Journal of Information Technology, Vol. 3, No. 2.
- William Leinberger, George Karypis, Vipin Kumar (2000), *Load Balancing Across Near-Homogeneous Multi-Resource Servers*, 0-7695-0556-2/00.