

Implementing Parameterized Dynamic Load Balancing Algorithm Using CPU and Memory

Pradip Wawge¹, Pritish Tijare²

Master of Engineering, Information Technology, Sipna college of Engineering, Amravati, Maharashtra, India¹

Associate Professor, Department of Computer Science and Engineering, Sipna College, Amravati, Maharashtra, India²

Abstract: In today's world, more or less, every activity belongs to internet. The increase of E-Commerce has leads many businesses to carry out the most of their day-to-day business online transaction on data. As a result of the popularity of the web, providers of web sites and storage space providers want to ensure the availability of access to information for their users and the guarantee that requests are processed as quickly as possible. If a server gets more requests than it can handle, this can be combated by using multiple hosts to provide the same service. Dynamic Load balancing is the process by which inbound internet protocol (IP) traffic can be distributed across multiple servers. Load balancing enhances the performance of the servers, leads to their optimal utilization and ensures that no single server is overloaded. By studying pros and cons of different techniques used for load balancing, we are specifically giving priority to Dynamic load balancing method rather than Static load balancing. Further we have discussed different balancing algorithms, techniques and their drawbacks. By taking advantage of Dynamic logic load balancing method, we have deployed one load balancing solution.

Keywords: Load Distribution, Load Balancing, Parameter Load Balancing, Balancing Load on server.

I. INTRODUCTION

Today's need for Dynamic load balancing[1] is speedily increasing because of increasing advance in scientific the necessity of high-speed processing which 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 necessity of the uniform distribution of workload among these servers.

Now a day, in technical life everything is going online. Web applications plays big role on internet to provide 24*7 hr services to customers. When application becomes popular, traffic to that is also in a state of growing. For availability, scalability, performances boosting more and more servers are required for example in online examination. Load balancing is a key issue in these types of large scale situation. Load balancing is to achieve optimal resource utilization, maximize throughput, best services, minimize response time, and avoid overload, disconnectivity. In many scenarios, however, it's not simply a matter of distributing traffic equally across many servers once deploys; other factors also comes into play. Here we have discussed different load balancing strategies, algorithms and methods. By investigating the comparative behavior of load balancing with different parameters, dynamic load balancing[1] is more reliable. So; here we implemented application load balancer by parameterized load balancing algorithm. Actually we perform load balancing by integrating more than two physical servers with Parameterized dynamic load balancing algorithm. The load balancer calculates the utilization of sub servers and select least utilized server and forward the request to sub server. In this system if you are balancing the load across several servers and one of the servers fails, then the service will still be available to your users, as the traffic will be diverted to the other servers in your server farm.

1.1 Dynamic Load Balancing

The use and access of fast processors and also multi-processors has vastly improved and are mostly used in cases that are very time-consuming and time is a valuable source. Thus, in multi-processor systems [2], scheduling on a number of processors is of great interest for such purposes. Parallel computers perform their calculations by executing different computational tasks on a number of processors concurrently. The processors within a parallel computer

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

generally exchange information during the execution of the parallel code. This exchange of information occurs either in the form of explicit messages sent by one processor to another or different parallel processors sharing a specified common memory resource within the parallel computer. The core idea of load balancing is to use multiple hosts to replace only one host, thereby it can enhance the computing power and reliability of the host at a low cost. In this way, the problem about a large number of concurrent accesses can be solved. Load-balancing system is a cluster system composed of multiple hosts. Each individual host, which can provide services independently without any external assistance of other hosts, has equivalent status in the system. The reliability of load-balancing is relatively higher than that of a single machine.

1.2 Load Balancing Concept

Dynamic Load balancing is the process by which inbound internet protocol (IP) traffic can be distributed across multiple servers. Load balancing enhances the performance of the servers, leads to their optimal utilization and ensures that no single server is overloaded. By studying pros and cons of different techniques used for load balancing, we are specifically giving priority to Dynamic load balancing method rather than Static load balancing. Further we have discussed different balancing algorithms, techniques and their drawbacks. By taking advantage of Dynamic logic load balancing method, we have deployed one load balancing solution.

1.3 Basic Idea of Parameterized Dynamic Load Balancing

In this paper we are Implemented Parameterized Dynamic load balancing using tree parameter

- CPU Usage
- Physical Memory
- Request Execution

II. LITERATURE REVIEW

In the previous work ,We had studied the comparison is made between various techniques but static load balancing algorithm are more stable and it is also easy to predict their behaviour, but at a same time dynamic distributed algorithm are always considered better than static algorithm. Experimental results of performance modelling show that diffusive load balancing is better than round robin and static load balancing in a dynamic environment, which manifest in frequent clients' object creation requests and in short objects' lifetimes. Ankush P.Deshmukh and Prof. Kumarswamy Pamu [1] have discussed/described different load balancing strategies, algorithms and methods. They investigate that comparative behaviour of load balancing with different parameters; dynamic load balancing is more reliable and after that they conclude that efficient load balancing can clearly provide major performance benefit.

2.1 Load balancing algorithms

Load balancing algorithms can be divided into two major categories: static load balancing algorithm and dynamic load balancing algorithm. In static load balancing algorithm, on the basis of the time needed to complete any given task, tasks are assigned to processors during the compile time and their relation is determined. No decision regarding a shifting of a task from one processor to another during execution time. But in dynamic load balancing algorithms (DLB), load status at any given moment is used to decide on task shifts between processors [2, 3, 4, and 5]. Random, Central and Rendezvous are among the existing load balancing algorithms which can be seen in [4, 6, 7]. In parallel and distributed systems processors are divided into three categories according to their workload level. Heavily loaded processor/Overloaded processor: which have a large number of tasks in waiting? Lightly loaded processor/Under loaded processor: which have a small number of tasks in waiting? Idle processors: which have no tasks to execute [2].

2.2 Definition - What does DNS Load Balancing mean?

We refer to user as anyone who is accessing the information on the World Wide Web, while we done client as a program, typically a Web browser that establishes connections to Internet for satisfying user requests. Clients are connected to the network through gateways; we will refer to the network sub-domain behind these local gateways as domain. The purpose of a Web server is to store information and serve client requests. To request a document from a Web-server host, each client rest needs to resolve the mapping of the host-name contained in the URL to an IP address.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

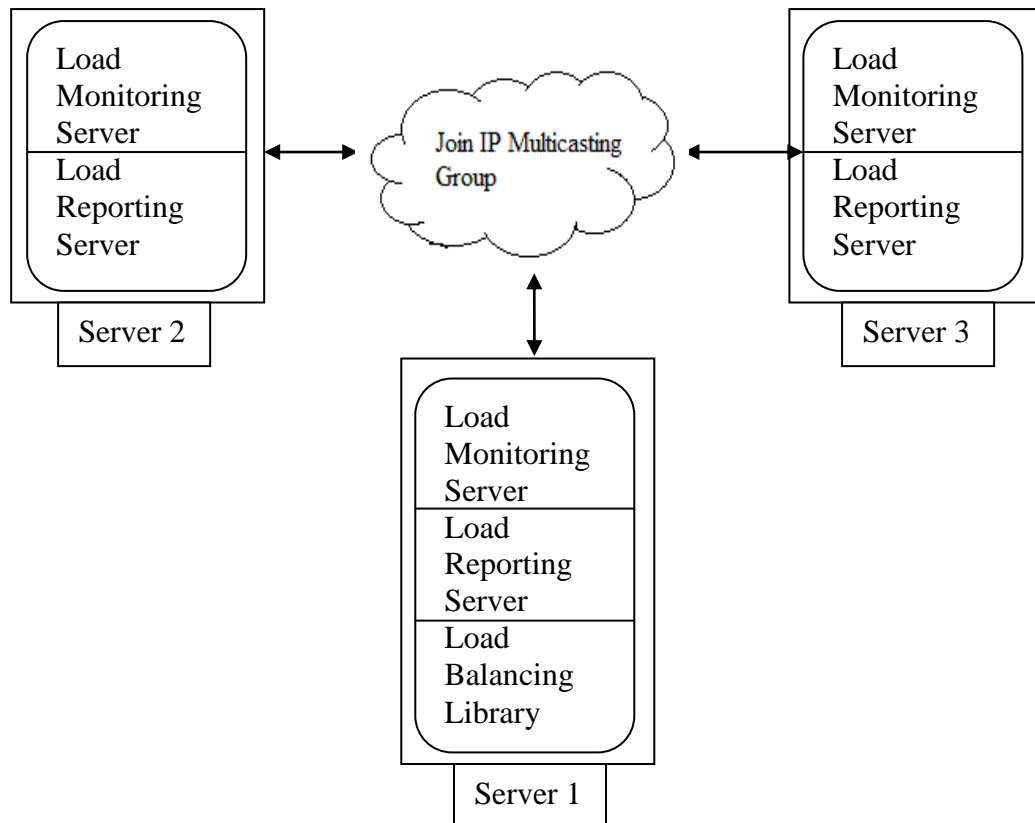


Fig 2.1 Dynamic load balancing Approach [1]

III. ANALYSIS OF PROBLEM

In the previous work we analyze that when load balancing across several servers and any servers fails, then the service may be disturb. Now take a scenario where three servers are clustered and exchanging data with each other. Suddenly server 2 stops informing the machine load to load balancer i.e. server 2 is down. It the least loaded machine among others for say 10 min. In between if request comes from the user then request is forwarded to server 2. But server 2 is down, so here comes a fault. Solution to this problem; after receiving data from server 2, Collector runs timer for 3second and if server 2 doesn't send data again within 3seconds then collector delete the server 2 entry from the list. The remaining processes still carry on with other servers. Application Load balancer shows fastest machine name telling that next request is being processed by that machine. Every time when user hit link page is refreshes and again shows fastest machine name.

3.1 Static Load Balancing

Static load balancing algorithm [4][8] allocate the tasks of a parallel program to workstations based on either the load at the time nodes are allocated to some task, or based on an average load of our workstation cluster. The advantage in this sort of algorithm is the simplicity in terms of both implementation as well as overhead, since there is no need to constantly monitor the workstations for performance statistics. The static algorithm is easily carried into execution and takes less time, which doesn't refer to the status of the servers. But however, static algorithms only work well when there is not much variation in the load on the workstations. Clearly, static load balancing algorithms aren't well suited to an environment, where loads may vary significantly at various times in the day, based on the issues discussed earlier.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

3.2 Random Scheduling

The Random algorithm [4] is self-explanatory. Traffic is directed arbitrarily to any server in your farm. In a random Scheduling, the requests are assigned to any server picked randomly among the group of servers. Pros: Random Scheduling load balancing algorithm is simple to implement. Cons: It can lead to overloading of one server while under-utilization of others.

3.3 Disadvantage

1. If you are balancing load across several servers and one of the servers fails, then the service has been disturb.
2. The main load balancing server will be crash then the other system doesn't work.
3. Calculating load only by a single server may cause heavily loaded system.

IV. IMPLEMENTATION AND RESULTS

4.1 Implementing Parameterized Dynamic Load Balancing Algorithm

Parameterized dynamic load balancing algorithm makes changes to the distribution of work among workstations at run-time; they use current load information when making distribution decisions. Parameterized dynamic load balancing algorithm can provide a significant improvement in performance over static algorithms. However, this will be achieve at the additional cost of collecting and maintaining load information, so it is important to keep these overheads within reasonable limits. The parameterized dynamic load balancing algorithm is self-adaptive algorithm, which is better than static algorithm. Self-adaptive load balancing system [9] mainly includes two processes: monitoring the load status of servers and assigning the request to the servers. The state supervision, which depends on the load information of each server in the cluster monitored and collected periodically by the front-end balancer, raises the effect of load balance by monitoring load variety, however, this will burden the workload of balancer which is the bottleneck of the cluster system. In this work we implemented individual work load calculation technique and also provide the authority to decide who will perform this work.

4.2 Weighted Round-Robin Scheduling

The weighted round-robin scheduling[4][10] can assign a weight to each server in the group so that if one server is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the IP sprayer will assign two requests to the powerful server for each request assigned to the weaker one. Pros: Takes care of the capacity of the servers in the group. Cons: Does not consider the advanced load balancing requirements such as processing times for each individual request

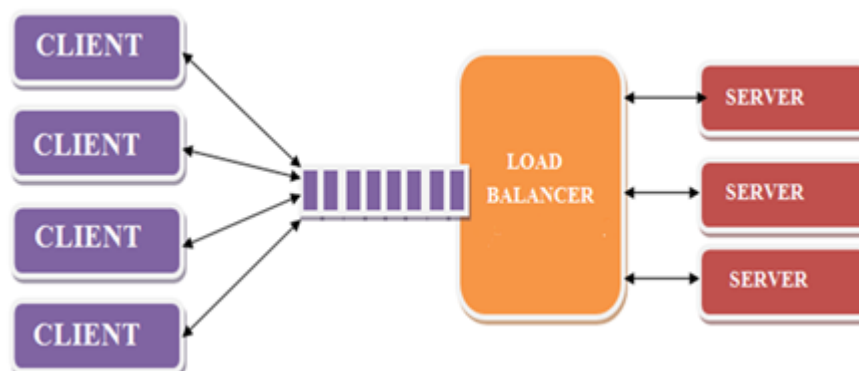


Fig. 4.1 Load Balancing System

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

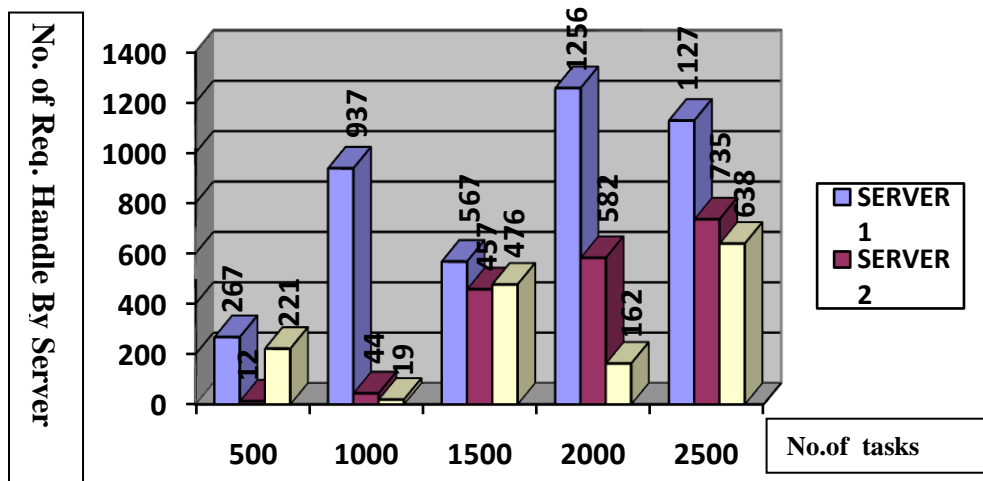
4.3 Result analysis of load balancing algorithm based on Total Load size

In this case the Load balancer balance the load according to load balancing algorithm based on the Addition of CPU Usage size and physical memory size. When the batch of queries received by the Load balancer it first check which server having least utilize on the basis of Total Load once find out that then it shifting the queries among the server. This algorithm tries to balance load of the system by dynamically calculating Total Load at 5ms interval.

A. Response server name

Test Input Tasks		No. of Task Executed based on CPU Usage + Memory		
		Server 1	Server2	Server3
1	500	267	12	221
2	1000	937	44	19
3	1500	567	457	476
4	2000	1256	582	162
5	2500	1127	735	638

Table 4.1 Task Execution based on CPU Usage and RAM



Graph 4.1 Task Execution based on CPU Usage and RAM

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

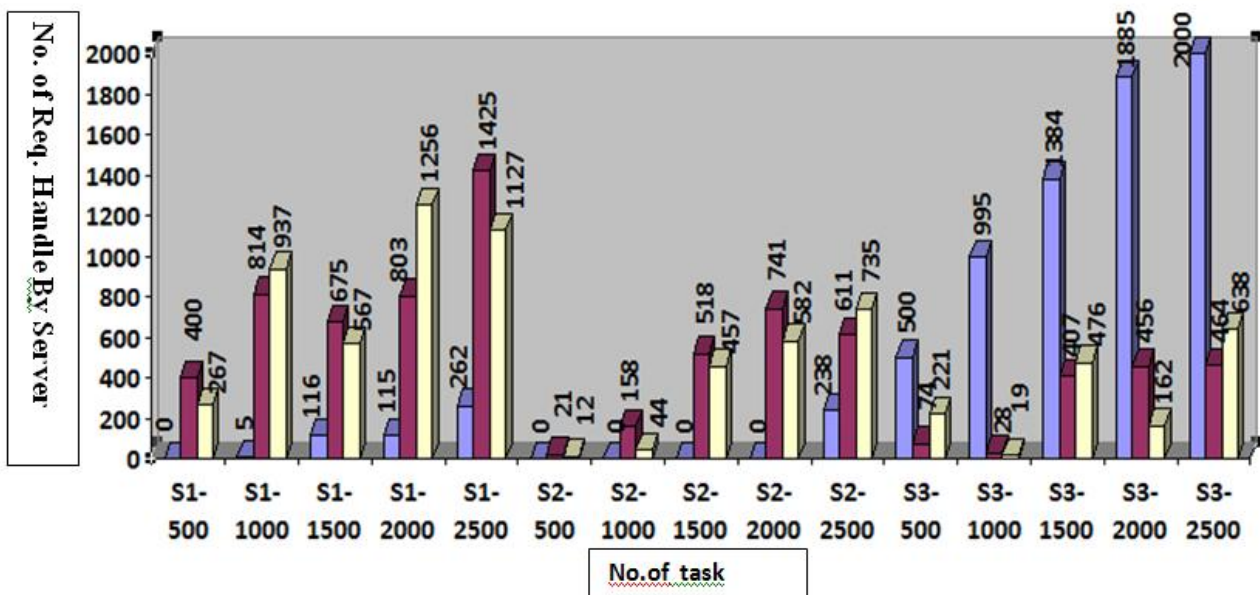
Vol. 3, Issue 5, May 2014

4.4 Experimental result

In this case the Load balancer balance the load according to load balancing algorithm based on the Addition of CPU Usage size and physical memory size. In the following Experiment result shows S1, S2, S3 are the server. When the batch of queries received by the Load balancer it first check which server having least utilize on the basis of Total Load and CPU Usage and memory once find out that then it shifting the queries among the server. In this Experiment result we are using 500 to 25000 bunches of queries and it send to least utilize server. This algorithm tries to balance load of the system by dynamically calculating Total Load at 5ms interval.

Test	Input Tasks	Response Server name based on Total Load								
		Memory			CPU Usage			Total load		
		S1	S2	S3	S1	S2	S3	S1	S2	S3
1	500	0	0	500	400	21	74	267	12	221
2	1000	5	0	995	814	158	28	937	44	19
3	1500	116	0	1384	675	518	407	567	457	476
4	2000	115	0	1885	803	741	456	1256	582	162
5	2500	262	238	2000	1425	611	464	1127	735	638

Table 4.2 Tasks Execution based on CPU Usage, RAM and Total load



Graph 4.2 Tasks Execution based on CPU Usage, RAM and Total Load

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

The above experimental graphical result shows that utilization of servers. When we are using different parameters to distribute the queries and check each and every 5ms which server having less utilization. Whenever we are sending 500 queries on server s1 then we observe that the comparison on the basis of memory utilization of s1 is 0. Second time we are comparing on the basis of CPU usage then utilization is 400 queries and last we are comparing on the basis of combination of CPU and Memory the utilization is 267 queries. We are increasing the queries up to 2500 and observe that the server machine handles more requests whenever the comparison done between Total load or CPU Usage.

V. CONCLUSION

The Parameterized Dynamic Load Balancing Algorithm has valuable features for balancing a load on any server system such as cloud computing, which makes it well-suited for time critical and efficient applications.

This Paper attempted to conclude algorithm for Dynamic Load Balancing. This algorithm shows how load distribution takes place over server. So every server gets occupied by workload when any server overloaded as well as shows that workload distribute uniformly among server to achieve load balancing.

REFERENCES

- [1] Ankush P.Deshmukh and Prof. Kumarswamy Pamu "Applying Load Balancing: A Dynamic Approach" (IJARCSSE), vol. 2, issue 6, June 2012.
- [2] R. Buyya and D. Abramson and J. Giddy and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing", int Journal of Concurrency and Computation: Practice and Experience, Volume 14, Issue.13-15, pp. 1507-1542, Wiley Press, December 2002.
- [3] Amit Chhabra, Gurvinder Singh, Sandeep Singh Waraich, Bhavneet Sidhu, and Gaurav Kumar "Qualitative Parametric Comparison of Load Balancing Algorithms in Parallel and Distributed Computing Environment" Proc. World Academy of Science, Engineering and Technology (PWASET) Vol 16, pp. 39-42, November 16, 2006,
- [4] Der-Chiang Li, Fengming M. Chang: "An In-Out Combined Dynamic Weighted Round-Robin Method for Network Load Balancing" Published by Advance Access publication, The Computer Journal Vol. 50 No. 5, 2007.
- [5] Satoru Ohta and Ryuichi Andou : "WWW Server Load Balancing Technique Based on Passive Performance Measurement" Published by IEEE 978-1-4244-3388-9/09.
- [6] Shahzad Malik: "Dynamic Load Balancing in a Network of Workstations" from Research Report. Valeria cardellini, University of Rome Tor Vergata, Michele Colajanni, University of Modena.
- [7] Philip S. Yu, IBM T.J. Watson Research Center: "Dynamic load balancing on web-server systems" Published by IEEE Internet Computing 1089-7801/99.
- [8] Li Wenzheng, Shi Hongyan: "Novel Algorithm for Load Balancing in Cluster Systems" Published by IEEE Proceedings of the 2010-978-1-4244-6763-1/10.
- [9] William Leinberger, George Karypis, Vipin Kumar, "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556-2/00, 2000 IEEE.
- [10] D. Andresen, T. Yang, V. Holmedahl, O.H. Ibarra, "SWEB: Toward a scalable World Wide Web-server On multicomputers", Proc. of 10th IEEE Int'l. Symp. on Parallel Processing, Honolulu, pp. 850-856, April 1996.

AUTHOR BIOGRAPHIES:



Mr. Pradip U. Wawge is a student of Sipna College of Engineering and Technology, Information Technology [M.E] Amravati, Sant Gadge Baba Amravati University. He has presented 2 papers in International Conference. His area of interest is network security and network load balancing.



Prof. Pritish A. Tijare is working as an Associate Prof. in Computer Science and Engineering department of Sipna College of Engineering and Technology, Amravati. He is having 9 years of experience in teaching field. He has published various state, national and international conference and journal. He has guided BE, ME students.