International Journal of Research in Advent Technology, Vol.2, No.2, February 2014

E-ISSN: 2321-9637

Parameterized Dynamic Load Balancing
Using Concept of Round Robin and threshold Value

Pradip U. Wawge , Prof. Pritish A. Tijare
Student of M.E. (Information Technology), Associate Professor, *CSE dept*
Sipna college of Engineering
Amravati, India
pwawge@gmail.com

Abstract— 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. For availability, scalability, performances boosting more and more servers are required. Load balancing is a main issue in these types of large situation. Load balancing is to achieve optimal resource utilization, maximize throughput, best services, minimize response time, and avoid overload, disconnectivity. In latest scenarios, however, it's not simply a matter of distributing traffic equally across many servers once deploys; other factors also comes into play. Here some brief introduction to different load balancing technique and algorithms. By developing the comparative behaviour of load balancing with dynamic parameters, dynamic load balancing is more effective. So; here we are discussing the implementation of 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 and all servers also having authority to decide least utilization server and then forward the request to that server. In this system if you are load balancing 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.

Index Terms— parameterized load balancing, dynamic parameter, dynamic nature load balance, Dynamic Round Robin.

1. Introduction

In todays 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. A web farm of multiple physical hosts grouped together will share the total load of the client requests, thus reduce the response time and increase the QoS (Quality of service), ultimately resulting in satisfied customers. By using the parameterized load balancing. we proposed best load balancing technique that utilizes 99.99 percent CPU time. That means The server should only be down .1 per cent of the time, which over duration of one year contributes to about 52 minutes of unavailability. Another benefit of having a web farm is redundancy, which helps to achieve both high availability as well as high performance. It is possible to perform upgrades on a host without shutting down the total service.

Today's need for Dynamic load balancing is speedily increasing because of increasing advance in scientific endeavor and 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 processors having a queue of tasks in waiting condition. So there is important of the uniformly distribution of workload among these servers.

2. Literature Review & Related Work

In the previous work I have studied the comparison 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 modeling show that diffusive load balancing is better than round robin and static load balancing in a dynamic environment Ankush P.Deshmukh and Prof. Kumarswamy Pamu [1] discussed/ different load balancing technique, 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 and benefit.

3. Load balancing

The load balancer forwards requests to one of the "slave" servers, which usually replies to the load balancer which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports. If load balancing done across several servers and one of the servers fails, then the service may be disturb.

A. Network Load Balancing

Network load balancing [2] is the distribution of traffic based on network variables, such as Internet Protocol address and ports number.

B. Round Robin Concept

Robin Scheduling Algorithm [3] is that the IP sprayer assigns the requests to a list of the servers on a rotating basis. For the subsequent requests, the IP sprayer follows the circular order to redirect the request. Once a server is assigned a request, the server is moved to the end of the list. This keeps the servers equally assigned. Pros: Better than random allocation because the requests are equally divided among the available servers in an orderly fashion. Cons: Not enough for load balancing based on processing overhead required and if the server specifications are not identical to each other in the server group.

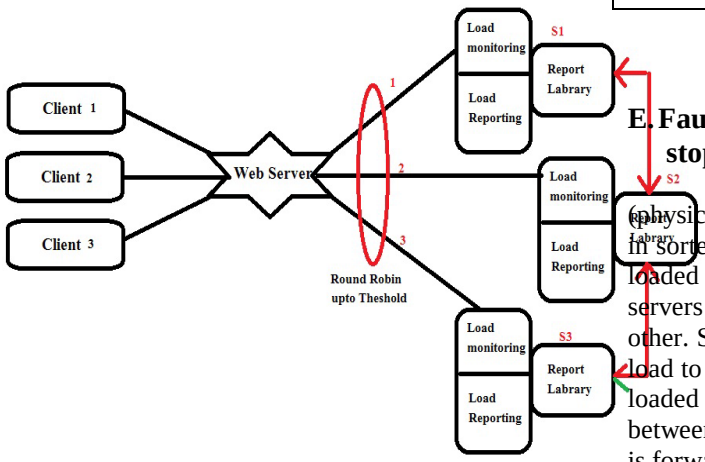


Fig 1 Dynamic load balancing in Round Robin concept

C. Application Load Balancing

Application load balancing is the distribution of requests based on multiple variables, from the network layer to the application layer. It can direct requests based on any single variable as easily as it can a combination of variables. Applications are load balanced based on their peculiar situation on server (operating system or virtualization layer) information. Application of load balancing is used by different companies that conduct huge business transactions using internet. It is normally done to ensure that all their clients are able to conduct business activities with ease and accuracy.

D. Calculations to find machine load

The previous paper result considering four parameters to calculate the load of machine and to find out which machine is least loaded. Every calculation is in milliseconds. If total Average value is less than any other machine's average total then in a sorted list this machine is at first position. When request comes, load balancer will give first priority to this machine to process the request. In a situation where average total of two or more machines is same, then sequence of those machines in current list is same as previous sorted list.

Table-1: Performance of load calculation

Parameter source	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	sum	AVG value
Processor	12	20	9	23	16	80	16
User time	23	28	36	20	15	122	24.4
Disk time	15	4	3	21	7	50	10
Request Exe	3	16	44	11	18	92	18.4
Total							68.8

E. Fault tolerance Solutions "When server stops reporting load":

In this load balancing solution machine (physical server) [3] and its respective load is stored in sorted order so that first machine in list is the least loaded machine. 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

4. Proposed Work And Objective

F. Idea to implement load balancing

While using system sometimes user feels that the machine is getting very slow, then launch the task manager and look at the CPU utilization and uses of resource. If it is low, then the memory was more used, and disk must be trashing. Well, this works if user is around the machine and has one or two machines to monitor. When there are more machines,

one couldn't monitor machines constantly and even if he somehow manages it but, you can't distribute their workload manually. So, you need load monitoring and load distributing features all together to enhance the whole assembly. For 24*7 running application online, performance of total assembly is more depends on how servers are performing. Idea is to monitor server performance by collecting parameter information of processor, Disk utilization, Memory health, User time, seek time of resource access etc.

User can monitor system performance by launching Task Manager and by looking at the CPU utilization in the Performance tab start monitoring the CPU utilization. Now notice the counter values and values stay almost constant. Now close Task Manager, run media player or any other application, wait about 5 seconds and start it again. A big peak in the CPU utilization should be noticed. In several seconds, may be the peak vanishes. Here if performance counters values are reported instantly one could think that our machine was extremely busy (almost 100%) at that moment. That's why rather than reporting instance values, several samples of the counter's values are collected and will report their average.

G. Parameterized Dynamic Load Balancing Algorithm:

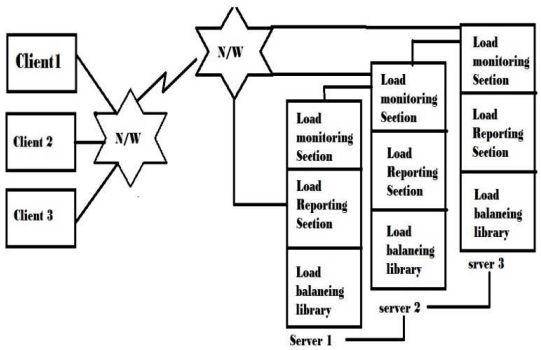
Parameterized dynamic load balancing algorithm makes changes to the distribution of work among workstations at run-time; they use current or recent load information when making distribution decisions. Parameterized dynamic load balancing algorithm can provide a significant improvement in performance 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 [4][5]. The parameterized dynamic load balancing algorithm is self-adaptive algorithm, which is better than static algorithm. Self-adaptive load balancing system 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 proposed individual work load calculation technique and also provide the authority to decide who will perform this work.

Algorithm:

1. Load reporting section first comes into action, start collecting the machine load information on which it is running.
2. Send collected data to load monitoring section which is located on same machine.
3. Load Monitoring section stores the collected data in data structure.
4. Calculate Sum and average of data for particular performance parameter.
5. Systems send data to other system which are connected in cluster.
6. Receive data from other machine and stay updated at particular time duration.
7. (At load balancer) when request comes, Load Balancing Library finds least loaded machine and return the address of that machine.

H. Basic idea behind parameterized dynamic load balancing algorithm:

User can monitor system performance by launching Task Manager and by looking at the CPU utilization in the Performance tab start monitoring the CPU utilization. Now notice the counter values and values stay almost constant. Now close Task Manager, run media player or any other application, wait about 5 seconds and start it again. A big peak in the CPU utilization should be noticed. In several seconds, may be the peak vanishes. Here if performance counters values are reported instantly one could think that our machine was extremely busy (almost 100%) at that moment. That's why rather than reporting instance values, several samples of the counter's values are collected and will report their average. The CPU utilization is not enough for a realistic calculation of the machine's workload; more than one counter should be monitor at a time such as disk utilization, memory usage, I/O etc. Now the machine load will be calculated as the sum of the weighted averages of all monitored performance counters. A counter is set to 0. All parameter values are collected in certain interval till counter value becomes 5. Then sum and average of parameter values are calculated. Depends on that calculated values, less loaded server or high performance system is being selected for load balancing and serving the coming request at best.



Basically two servers are needed because you can't do load balancing on single server. Here we have used three servers for deployment of load balancer. On server 1,server2,server3: Load Monitoring section

for monitoring load of its own system, Load Reporting section which reports load collected by Monitoring Section to Load Balancing Library, Load Balancing Library which accepts data sent by Load Reporting section and does calculations to find least loaded or high performance machine.

parameterized factor:

1. CPU Utilization
2. Disk time Utilization
3. User Time Utilization
4. Execution time
5. Physical memory
6. Kernel memory
7. Network I/O

Five or six Parameter are used for Calculating System load and decide how to distribute a client Request.

CONCLUSION

This paper attempted to proposed algorithm for Implementing Parameterized Dynamic Load Balancing Algorithm. This Concept presents a new load balancing algorithm which includes distribution of tasks on basis of certain parameter. This algorithm proposed the solution of how to distribute load. 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.

ACKNOWLEDGEMENT

It is a matter of great pleasure by getting the opportunity of highlighting fraction knowledge, I acquired during my technical education through this paper. This would not have been possible without the guidance and help of many people. This is the only page where I have opportunity of expressing my emotions and gratitude from the care of my heart to them. This paper would not have been successful without enlightened ideas, timely suggestion and keen interest of my respected Guide **Prof. Pritish A. Tijare** without his best guidance this would have been an impossible task to complete.

REFERENCES

[1] Ankush P.Deshmukh and Prof. Kumarswamy Pamu “Applying Load Balancing: A Dynamic Approach” (IJARCSSE), vol. 2, issue 6, June 2012.

[2] 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.

[3] Satoru Ohta and Ryuichi Andou : “WWW Server Load Balancing Technique Based on Passive Performance Measurement” Published by IEEE 978-1-4244-3388-9/09.

[4] Satoru Ohta and Ryuichi Andou : “WWW Server Load Balancing Technique Based on Passive Performance Measurement” Published by IEEE 978-1-4244-3388-9/09.

[5] 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.