International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

A New Technique for Dynamic Load Balancing

Suryakant Patil[1], Pratap Singh[2]

[1]Pune University, Maharashtra, India

[2]Professor, Pune University, Maharashtra, India

Abstract: ***Mission-critical applications must run 24x7, and networks need to be able to scale performance to handle large numbers of client requests without unwanted delays. A "server cluster" is a group of independent servers managed as a single system for achieving availability and scalability. It consists of two or more servers connected by a network, and cluster management software. The software provides services such as failure detection, recovery, load balancing, and manages the servers as a single system. Load balancing is a technique that allows the performance of a server-based program, such as a Web server, to be scaled by distributing its client requests across multiple servers within a cluster of computers. Load balancing is used to enhance scalability, which improves throughput while keeping response times low. In this paper we have proposed a dynamic load balancing system in which servers in distributed system uses multicasting to communicate with each other and uses decentralized approach for load balancing.***

Keywords: Load balancing, Static Algorithms, Dynamic Algorithms, Machine Load, Fastest machine, Metrics for load measurement

1. Introduction

1.1 Load Balancing

The process of distributing or redistributing the load of the system equally among all the nodes of the system is called as load balancing. Load balancing helps the system to use its resources effectively and improve the response time of the job.

1.2 The goals of load balancing are:

- To improve the performance of the system
- To maintain the availability of the system
- To keep system stable

1.3 Types of Load balancing algorithms

1] Load balancing algorithms are divided into threecategories based on who initiated the process of load balancing.

- Client Initiated: If the load balancing algorithm isinitialized by the client.
- Server Initiated: If the load balancing algorithm isinitiated by the server.
- Symmetric: It is the combination of both client initiatedand server initiated

2] Depending on the current state of the system, loadbalancing algorithms can be divided into 2 categories:-

- **Static:** It doesn't take into account the current state of thesystem. past knowledge of the system is needed.
- **Dynamic:** Decisions on load balancing are based oncurrent state of the system. No past knowledge is needed. Soit is better than static approach.

[i] Static Algorithms
Static algorithms distribute the traffic evenly betweenservers. By this approach the traffic on the servers will be
disdained easily and as a result it will make the situation moreimproperly. In this algorithm, based on the past knowledge ofperformance, master server distributes the load among theslave servers. Static algorithms are non-preemptive.

[ii] Dynamic Algorithms
Dynamic algorithms distribute the workload among thenodes at run time. It assigns proper weights on servers and bysearching in entire network a lightest server chosen tobalance the traffic. However, selecting the lightest serverrequires real time communication with the networks, whichwill lead to extra traffic added on system.

1.4 Types of Dynamic Load Balancing Algorithms

1] Centralized dynamic load balancing: -
Centralized dynamic load balancing takes smaller amount ofmessages to reach a decision

2] Distributed dynamic load balancing: -
Each node needs to exchange status information with every other node in the system.

1.5 Policies or Strategies in dynamic load balancing

There are 4 policies:

- Transfer Policy: The policy which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy.
- Selection Policy: It specifies the processors involved in the load exchange.
- Location Policy: It selects a destination node for a transferring the task.
- Information Policy: It collects information about the nodes in the system.

2. Proposed System

Before proceeding further we must understand the following two terms regarding our system

1) **Machine Load:** The overall workload (utilization) of a machine - in our case, this is the sum of the weighted

averages of all performance counters (monitored for load balancing);
2) **Fastest machine:** The machine with the least current load.

We proposed Load balancing software which has three parts.
1) **Machine Server**- Machine server will calculate theload of the machine it is running on and will report it to Master server.
2) **Master Server-** This server will collect the loadinformation from all machine servers.
3) **Library-** Library will ask master server to find out leastloaded server i.e. fastest server.

The Machine Server on each machine joins a specialmulticasts group, and sends messages, containing the machine's load to the group's multicast IP address. Becauseall Master Servers join the same group at startup, they allreceive each machine load, so if you run both Machine andMaster servers on all machines, they will know each other'sload.

All Master Servers store the machine loads in a specialdata structure, which lets them quickly retrieve the leastmachine load at any time. So all machines now know whichthe fastest one is. Each Master Server registers a specialsingleton object with the .NET Remoting runtime, so theLibrary can get an instance of that object, and ask it for theleast loaded machine. The problem is that LBL cannot asksimultaneously all machines about this, so it should chooseone machine (i.e. the machine, it is running on) and will handthat load to the client application that needs the informationto perform whatever load balancing activity is suitable.

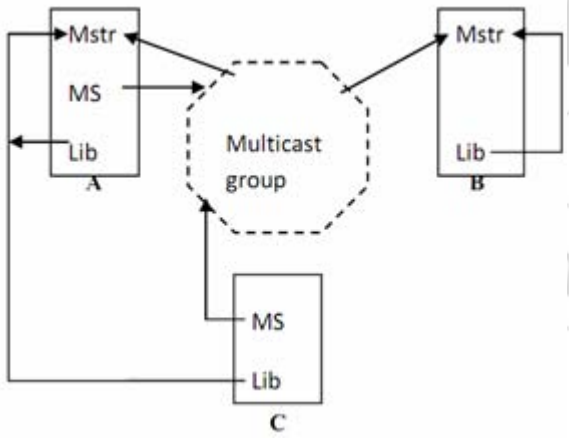

Figure 1: Proposed System architecture

Now look at the "figure". When a machine joins a multicast group, it receives all messages sent to that group, including the messages that the machine has sent. Machine A receives its own load, and the load, reported by C. Machine B receives the load of A and C (it does not report its load, because there's no MS(Machine server) server installed on it). Machine C does not receive anything, because it has not Mstr (Master server) installed. Because the machine C's Lib (library) should connect to an Mstr server, and it has no such server installed, it could connect to machine A or B and query the remote object for the fastest machine. On the "diagram" above, the Lib of A and C communicate with the remote object on machine A, while the Lib of B communicates with the remote object on its machine.

2.1 Metrics for load measurement

1) **Process time**
It is the percentage of elapsed time that the processorspends to execute a non-idle thread. It is calculated by measuring the duration of the idle thread is active in thesample interval, and subtracting that time from interval duration. (Each processor has an idle thread that consumescycles when no other threads are ready to run). This counteris the primary indicator of processor activity, and displays theaverage percentage of busy time observed during the sampleinterval. It is calculated by monitoring the time that theservice is inactive and subtracting that value from 100%.
2) **User Time**
It is the percentage of elapsed time the processor spendsin the user mode. User mode is a restricted processing modedesigned for applications, environment subsystems, andintegral subsystems. The alternative, privileged mode isdesigned for operating system components and allows directaccess to hardware and all memory. The operating systemswitches application threads to privileged mode to access operating system services. This counter displays the averagebusy time as a percentage of the sample time.
3) **ASP Request time**
It is the number of requests currently executing.
4) **Disk Time**
It is the percentage of elapsed time that the selected diskdrive was busy servicing read or writes requests

2.2 The Proposed Load Balancing Algorithm:-

1) Machine Server calculate the load of the machine it isrunning on*MachineLoad*= Sum (WeightedAverage (EachCounter))
2) The Machine Server on each machine joins a specialmulticasts group.
3) Master Server receives the load of all the other machinesand store the machine loads in special Data Structure called as *priority_queue*. *priority_queue*is nothingbut an Array List. Machine with the least *MachineLoad*willbe store at the first position and will have higher priority.
4) Load Balancing Library will ask master server to find out least loaded server. Master Server will retrieve the first element from priority queue which is nothing but the Fastest Machine and forwards this information to Load balancing Library.
5) LBL checks if the Fastest Machine is the different Machine than the Machine it is running on, then LBL will forward the request to the URL of fastest Machine.

2.3 Characteristics of Proposed System.

1) Dynamic Load Balancing.
2) Decentralized Approach.
3) Communication between nodes using Multicasting.

References

[1] Urjashree Patil, RajashreeShedge, " Improved Hybrid Dynamic Load Balancing Algorithm for Distributed Environment", International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013 1 ISSN 2250-3153
[2] AnkitaSinghal and Padam Kumar, "Seizetoken: A Dynamic LoadBalancing Algorithm for Distributed System", International Journal of Advanced Research in Computer Engineering & Technology, Volume1, Issue 3, May 2012.
[3] Tarek Helmy, Fahd S. Al-Otaibi, "Dynamic Load-Balancing Based on a Coordinator and Backup Automatic Election in Distributed Systems", International Journal of Computing & Information Sciences Vol. 9, No. 1, April 2011.
[4] Abbas Karimi, FaranehZarafshan, Adznan b. Jantan, A.R. Ramli, M.Iqbal b.Saripan, "A New Fuzzy Approach for Dynamic Load Balancing Algorithm", (IJCSIS) International Journal of ComputerScience and Information Security, Vol. 6, No. 1, 2009.
[5] Sagar Dhakal, Majeed M. Hayat, Jorge E. Pezoa, Cundong Yang, andDavid A. Bader, "Dynamic Load Balancing in Distributed Systems inthe Presence of Delays: A Regeneration-Theory Approach", IEEETransaction On Parallel And Distributed Systems, Vol. 18, No. 4, April 2007.
[6] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma,"Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology 38 2008.
[7] Ali M. Alakeel, "A Fuzzy Dynamic Load Balancing Algorithm forHomogenous Distributed Systems", World Academy of Science, Engineering and Technology 61 2012.