ISSN 1512-1232

LOAD BALANCING ALGORITHMS ROUND-ROBIN (RR), LEAST-CONNECTION, AND LEAST LOADED EFFICIENCY

Dr. Mustafa ElGili Mustafa
Computer Science Department, Community College, Shaqra University, Shaqra, Saudi Arabia,
mustgili@hotmail.comline

Home affiliation
Assistant Professor, Faculty of Computer Science and Information Technology–Neelian University, Khartoum, Sudan

Abstract
This paper aims to discuss Load Balancing Algorithms Round-Robin (RR), Least-Connection, and Least Loaded distribution traffic efficiency, Between 8 HTTP Servers and their CPU utilization over Load Balancer. Opnet software has been used to simulate the network.

Keywords*: Load Balancing Algorithms, Round-Robin, Least-Connection.*

1. Introduction

The Internet is flooded with huge traffic, many applications have millions users, a single server is difficult to bear a large number of clients' access, so many application providers will put several servers as a computing unit to provide support for a specific application, usually people will use distributed computing, load balancing technology to complete the work. A typical load balancing technique is to use a dedicated load balancer to forward the client requests to different servers, this technique requires dedicated hardware support [1].

The Architecture of Web Server Cluster

A cluster-based web system is composed of N sever machines, which are connected through a high-speed network in order to solve user requests. Each server machine acts as a node that has its own disk and operating system and the load balancing is simple and distributed. Although a cluster has large numbers of web servers, it only utilizes one hostname and one virtual IP address to provide a single interface for outside users .So it is necessary to have a mechanism that controls the whole requests of the site and to mask the service distribution among all the servers [2].

Load balancer

One of the most important components is the load-balance controller, which routes the load and dispatches client requests to all the server nodes. Usually, we deploy our load-balancing algorithm into load-balance controller to execute logic, which can help distribute load from heavily-loaded nodes to lightly-loaded nodes in the system.[2]

The Load balancer is the front end to the service as seen by The outside world. The load balancer directs network connections from clients who know a single IP address for services, to a set of servers that actually perform the work [3].

Server load balancing is indispensable in World Wide Web for providing high-quality service. In server load balancing, since the server loads and capacities are not always identical, traffic should be distributed by measuring server performance to improve the service quality [4].

ISSN 1512-1232

load balancing algorithms

Load-balancing algorithm running in web switch is of great significance to boost the cluster performance. When a new request arrives, it chooses the most suitable server and assigns the request to it. Load-balancing algorithms work on the principle that in which situation workload is assigned, during compile time or at runtime [2].

The main purpose of load balancing is to distribute load among a number of nodes to optimize the utilization of the computation capability of every node and reduce the average task response time as well, this is equivalent to maximize the system throughput. The modus operandi is a special computer (also called request distributor) that receives and distributes all task requests to every server in the cluster according to some rules[5]. Below we review some popular load balancing algorithms:

4.1 Round-Robin (RR) Algorithm

The round-robin scheduling algorithm forwards each incoming request to the next server in its list. Thus in a three server cluster (servers A, B and C) request 1 would go to server A, request 2 would go to server B, request 3 would go to server C, and request 4 would go to server A, thus completing the cycling or 'round-robin' of servers. It treats all real servers equally regardless of the number of incoming connections or response time each server is experiencing. Virtual server provides a few advantages over traditional round-robin DNS. Round-robin DNS resolves a single domain to the different IP addresses, the scheduling granularity is host-based, and the caching of DNS queries hinders the basic algorithm. These factors lead to significant dynamic load imbalances among the real servers. The scheduling granularity of virtual server is network connection-based, and it is much superior to round robin DNS due to the fine scheduling granularity [6].

Round-robin Scheduling, in its word meaning, directs the request received from network to the different node in a round-robin manner. It treats all nodes as equals regardless of number of connections. The scheduling granularity is node-based, this will lead to significant dynamic load imbalance among the nodes [5].

4.2 Least-Connection Algorithm

The least-connection scheduling algorithm directs network connections to the server with the least number of established connections. This is one of the dynamic scheduling algorithms; because it needs to count live connections for each server dynamically. For a virtual server that is managing a collection of servers with similar performance, least-connection scheduling is good to smooth distribution when the load of requests vary a lot [6].

Distributes more requests to real servers with fewer active connections. Because it keeps track of live connections to the real servers through the IPVS table, least-connection is a type of dynamic scheduling algorithm, making it a better choice if there is a high degree of variation in the request load. It is best suited for a real server pool where each member node has roughly the same capacity. If a group of servers have different capabilities, weighted least-connection scheduling is a better choice [7]. Least-connection scheduling algorithm assumes that the processing capabilities of all servers are the same and as- signs the newly arrived request to the server with the least connection. However, the system performance is not ideal when the processing capabilities of the servers are different [5].

4.3 Least Loaded Algorithm

In this scheme, the dispatcher assigns the next request to the server that has the lowest workload (workload of a server is defined as the sum of the service time of all requests pending on the server). The baseline algorithm requires knowledge about service time of the client requests. This information is often unknown as the requests arrive, and hence it is very difficult (if not impossible) to use the baseline algorithm in practice. However, one could use the baseline algorithm to establish an upper bound on the performance. Hence, we refer to it as the baseline algorithm and use it as a base for comparing with the performance of the other schemes [8].

ISSN 1512-1232

5. Simulation Methodology

Network is simulated using OPNET® Modeler. OPNET® is extensive and powerful simulation software tool with wide variety of capabilities. It enables the possibility to simulate entire heterogeneous networks with various protocols [9]. The simulated network designed with 8 http server provide http service to 112 clients and there is load balancer device between the servers and clients as shown in the Fig 1.

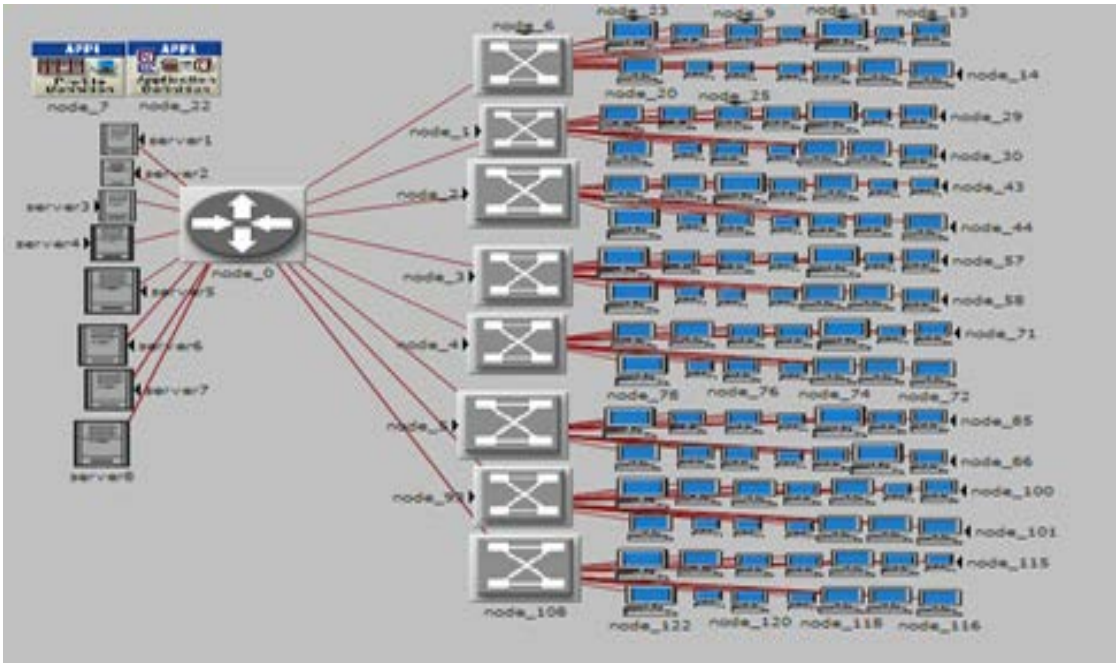

Figure 1. Network diagram

6. Scenarios

The paper proposed three scenarios, first scenario the load balance device implements Round-Robin as scheduling algorithm, the second scenario uses Least-Connection as scheduling algorithm and the last one uses Least Loaded as scheduling algorithm.

7. Results

As shown in Fig 2 Round Robin algorithm and Number of Connection algorithm have same CPU utilization, instead of Server Load has more CPU utilization comparing with Round Robin algorithm and Number of algorithm.

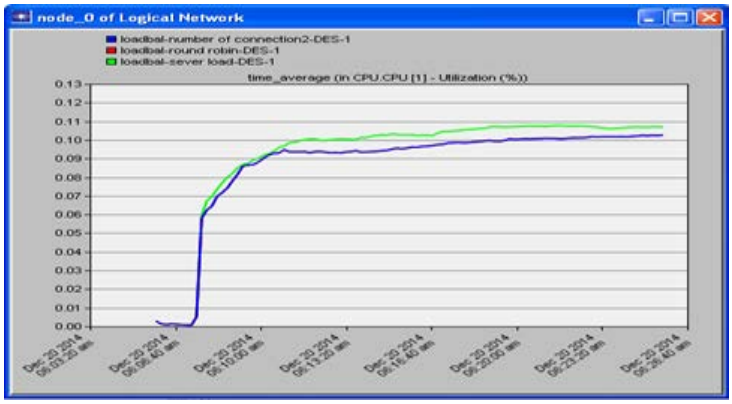

Figure 2. Load balancer CPU utilization

ISSN 1512-1232

Number of Connection algorithm distribute the traffic fairly between HTTP server, small variance in the load between the 8 HTTP Servers, as shown in Fig 3.

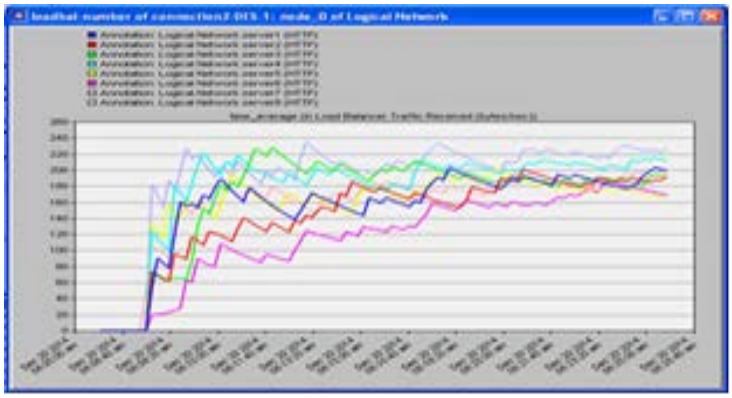

Figure 3. Number of Connection HTTP Servers load

Round Robin algorithm distribute the traffic fairly between HTTP server, but Server 1 has more load than other 7 servers, as shown in Fig 4.

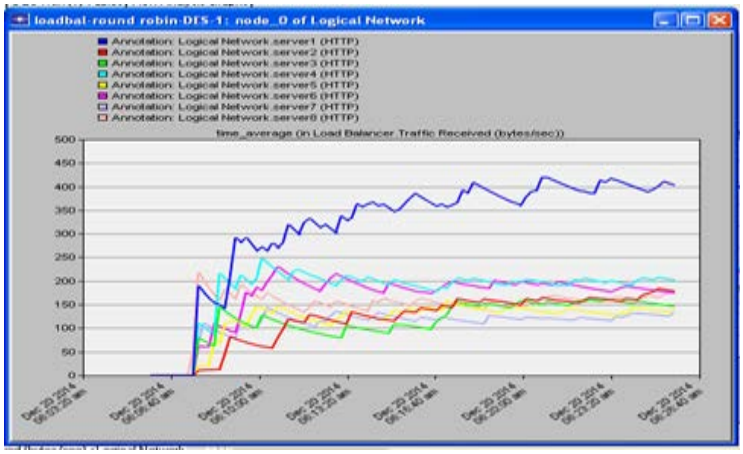

Figure 4. Round Robin HTTP Servers load

As shown in Fig 5 Server Load algorithm there is big variance in the load between the 8 HTTP Servers .

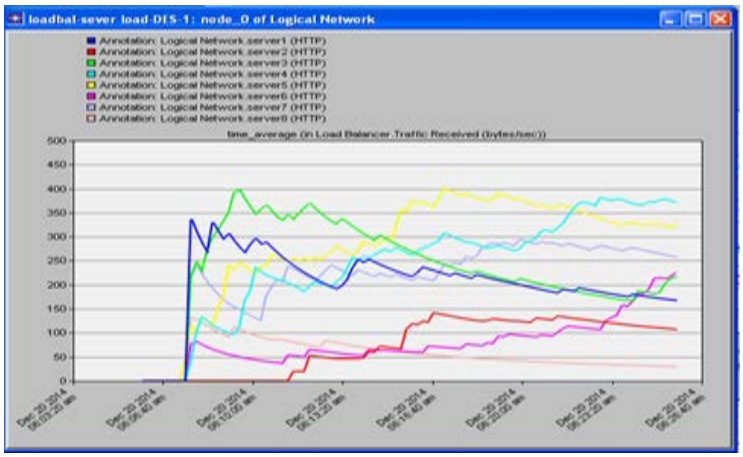

Figure 5. Server Load HTTP Servers load

ISSN 1512-1232

8. Conclusion

Server Load algorithm has more CPU utilization comparing with Round Robin algorithm and Number of Connection algorithm and when Server Load algorithm has been used there will be big variance in the load between the 8 HTTP. Number of Connection algorithm distribute traffic in a fair way rather than others algorithms.

References

1. Zhihao Shang and others, Design and implementation of server cluster dynamic load balancing based on OpenFlow.
2. XU Zongyu, A Modified Round-robin Load-balancing Algorithm for Cluster-based Web Servers, Proceedings of the 33rd Chinese Control Conference July 28-30, 2014, Nanjing, China,.
3. Wensong Zhang, Linux Virtual Server For Scalable Network Services, National Laboratory for Parallel & Distributed Processing
4. Satoru Ohta and other, WWW Server Load Balancing Technique Based on Passive Performance Measurement, 2009 IEEE
5. YU SHENGSHENG and others,Least-Connection Algorithm based on variable weight for multimedia transmission.
6. Comparison of Load Balancing Algorithms for Clustered Web Servers, Proceedings of the 5th International Conference on IT & Multimedia at UNITEN (ICIMU 2011) Malaysia, November 2011
7. Scheduling Algorithms http://www.centos.org/docs/5/html/Virtual_Server_Administration /s2-lvs-sched-VSA.html
8. YM Teo and R Ayani, Comparison of Load Balancing Strategies on Cluster-based Web Servers, Transactions of the Society for Modeling and Simulation (accepted for publication), 2001.
9. S.G. Thorenoor (2010) Communication Service Provider's Choice between OSPF and IS-IS Dynamic Routing Protocols and Implementation Criteria Using OPNET Simulator. Second International Conference on Computer and Network Technology, Bangkok, 23-25 April 2010, 38-42.

Article received: 2017-01-12