How does Software and Hardware Load Balancer Work? (Loadbalancer Algorithms Explained with Examples)

by LUKE P. ISSAC *on* JANUARY 11, 2016

When you have an enterprise application or website that gets lot of hits, your server might be under heavy load. In that case, you may want to consider distributing the load across multiple servers.

Load balancer will distribute the work-load of your system to multiple individual systems, or group of systems to to reduce the amount of load on an individual system, which in turn increases the reliability, efficiency and availability of your enterprise application or website.

In this article, we'll cover the basics of software and hardware load balancer, and explain the various algorithms used by the load balancers.

The following are the advantages of load balancing your application:

- Reduced the work-load on an individual server.
- Large amount of work done in same time due to concurrency.
- Increased performance of your application because of faster response.
- No single point of failure. In a load balanced environment, if a server crashes the application is still up and served by the other servers in the cluster.
- When appropriate load balancing algorithm is used, it brings optimal and efficient utilization of the resources, as it eliminates the scenario of some server's resources are getting used than others.
- Scalability: We can increase or decrease the number of servers on the fly without bringing down the application
- Load balancing increases the reliability of your enterprise application
- Increased security as the physical servers and IPs are abstract in certain cases.

On a high level, there are two types of load balancers, which implements different types of scheduling algorithms and routing mechanisms.

1. Software load balancer
2. Hardware load balancer

I. Software Load Balancers

Software load balancers generally implements a combination of one or more scheduling algorithms.

The following are the three different basic algorithms used by load balancers. Most modern load balancers use combination of these algorithms to reach high performance and to set a trade off between various parameters.

1. Weighted Scheduling Algorithm

Work is assigned to the server according to the weight assigned to the server. For different types of the server in the group different weights are assigned thus the load gets distributed.

The diagram below depicts a generic scenario where a load balancer is used and how the weighted scheduling will work if there are total of 10 request (R1, R2..... R10) coming to the server farm/cluster.

As you see, the request will be assigned to the server as per its weight. This weight is determined by the administrators wisely by considering the hardware capabilities of each server. Assuming that we have different weights assigned to each server as you can see in the figure below.

The load balancer will compute the percentage of traffic to be sent to a particular server according to the weight assigned to it.

When is this algorithm mostly used?: This is used when there is a considerable difference between the capabilities and specification of the servers present in the farm or

cluster.

This algorithm stands out to be efficient in managing the load without swarming the low capability servers the most and efficiently utilizing the available server resource at any instant of time.

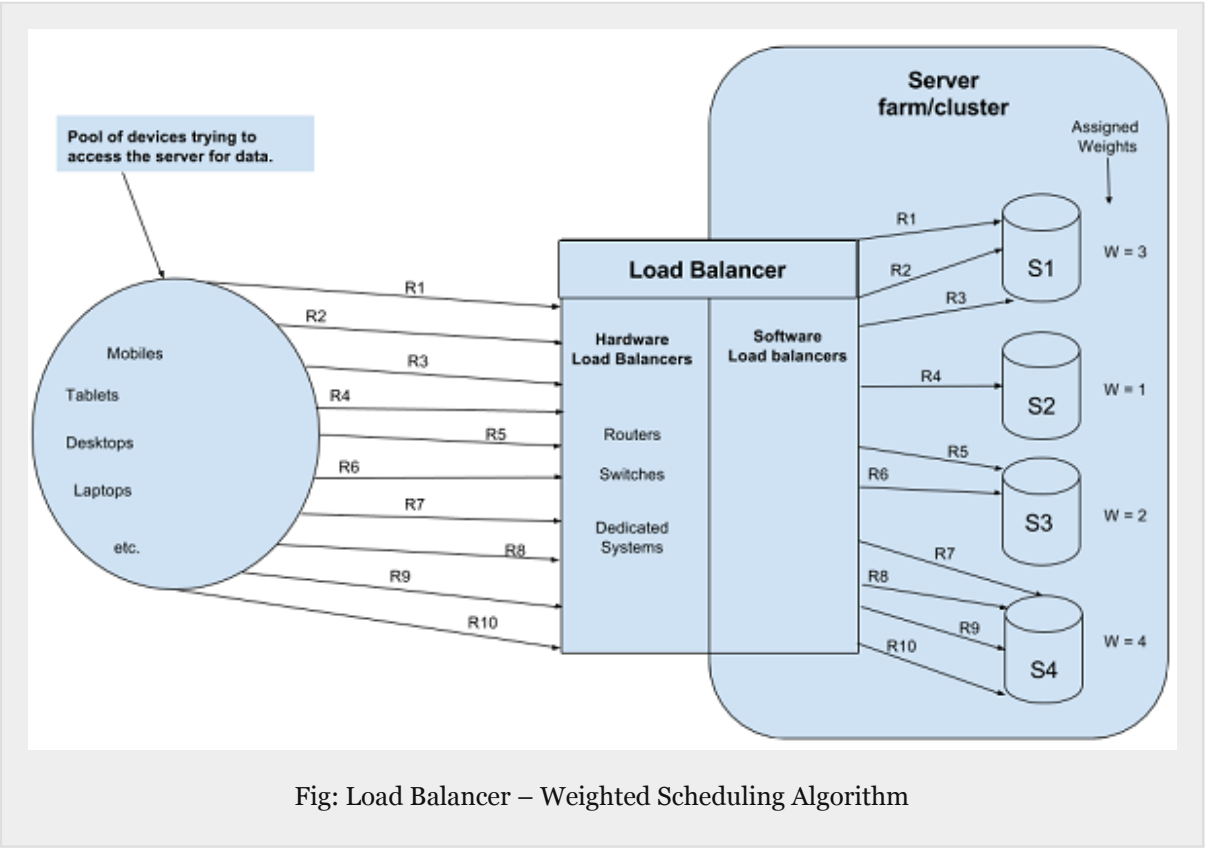


Fig: Load Balancer – Weighted Scheduling Algorithm

The following points can be noted in the above diagram:

- A load balancer can be of two types: hardware load balancer and software load balancer.
- Software load balancer are often installed on the servers and consumes the processor and memory of the servers. So, in the diagram above software load balancer is over lapping the server farm.
- Hard ware load balancers are specialized hardware deployed in-between server and the client. It can be a switching/routing hardware or even a dedicated system running a load balancing software with specialized capabilities.

2. Round Robin Scheduling

Requests are served by the server sequentially one after another. After sending the request to the last server, it starts from the first server again.

The diagram below depicts this approach. Sequentially each request gets assigned to each server one by one and the round goes on. The change in the request assigned can be easily understood by looking into the diagram below.

This algorithm is used when servers are of equal specification and there not much persistent connections.

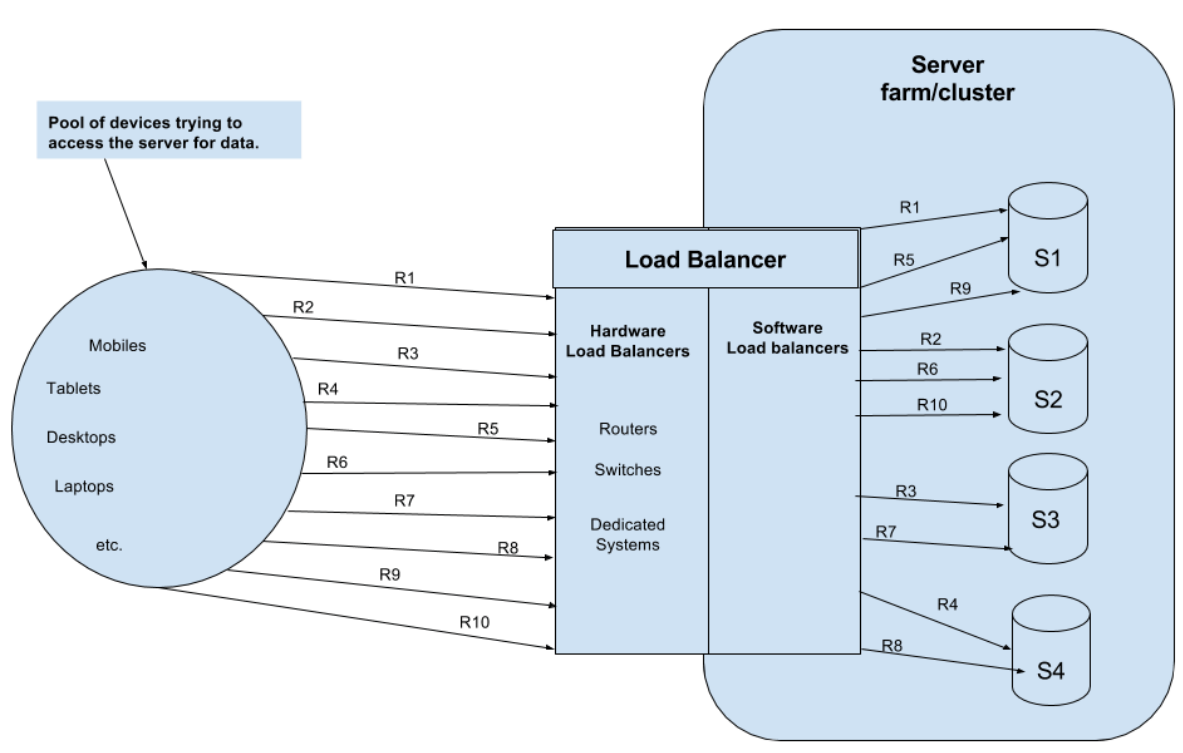


Fig: Load Balancer – Round Robin Algorithm

3. Least Connection First Scheduling

Requests are served first to the server which is currently handling least number of persistent connections.

In the diagram above, if we are using the least connection first scheduling algorithm, the request R5 can be assigned randomly to any of the server as when R5 will be coming every server will be having same number of connections.

Lets say when request R5 comes, request R3 got completed and server S3 is free now. Then, in that case, the request R5 will be assigned to server S3 instead of any other server as server S3 is having least no of connection at that instant of time.

When is this algorithm used?: When we have large number of persistent connections in the traffic unevenly distributed between the servers. It is often coupled with **Sticky Session** or **Session aware** load balancing. In this, all the request related to a session is sent to the same server to maintain the session state and syncronization.

This approach is used when we have session aware write operations in sync with client and the server so that it avoids any inconsistency.

Now, load balancing softwares can have the smart implementation of the combination of the above three basic scheduling algorithm. Such implementations are **Weighted round robin** scheduling and **Weighted least connection** scheduling.

Many hybrid scheduling algorithm for load balancing has evolved using some variations or combinations of the above algorithms.

Software Load Balancer Examples

The following are few examples of software load balancers:

1. HAProxy – A TCP load balancer.
2. NGINX – A http load balancer with SSL termination support. (install Nginx on Linux)
3. mod_athena – Apache based http load balancer.
4. Varnish – A reverse proxy based load balancer.
5. Balance – Open source TCP load balancer.
6. LVS – Linux virtual server offering layer 4 load balancing

II. Hardware Load Balancers

Load balancing hardwares are often referred as specialized routers or switches which are deployed in between the servers and the client. It can also be a dedicated system in between the the client and the server to balance the load.

The hardware load balancers are implemented on Layer4 (Transport layer) and Layer7 (Application layer) of OSI model so prominent among these hardwares are L4-L7 routers.

1. Layer4 Hardware Load Balancing

These kind of load balancers work on transport layer of OSI model and make use of TCP, UDP and SCTP transport layer protocol details to make decision on which server the data is to be sent.

Layer 4 load balancers are mostly the **network address translators** (NATs) which shares load to the different servers getting translated to by these loadbalancer. These routers hide multiple servers behind them and translate every response data packets coming from the server to be coming from same ipaddress.

Similarly, when there is a request they reverse translate the request using the mapping table and distributes it among the multiple servers.

DNS load balancing: In DNS based load balancing method the Domain Name Servers are configured to return different ipaddress for different systems. This approach creates a load balancing effect whenever there is a dns lookup.

The diagram below depicts the highlevel overview of Layer 4 and Layer 7 load balancer working and techniques on OSI layer.

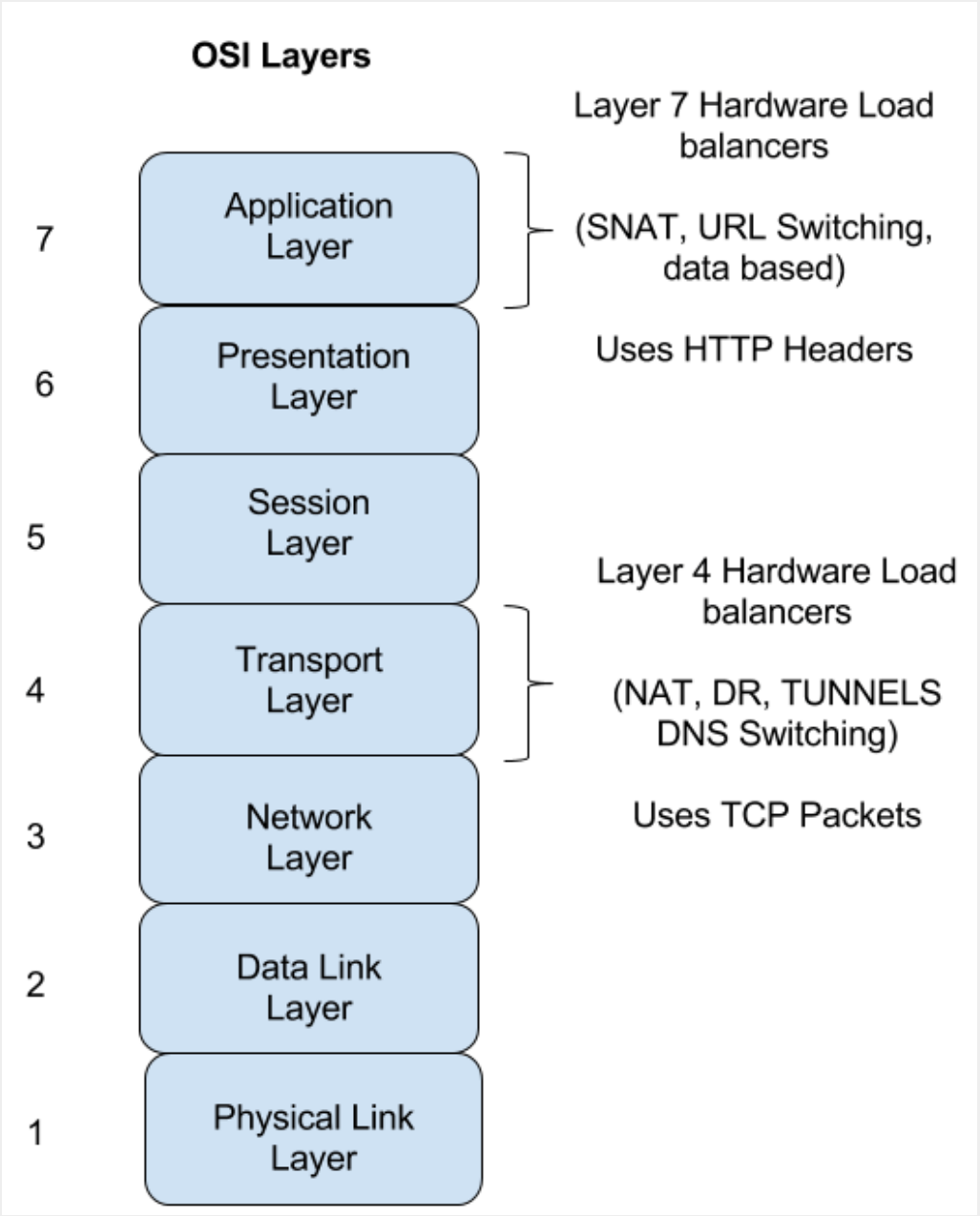


Fig: Load Balancer – Layer 4 and Layer 7

Direct routing: This is a yet another configuration of hardware load balancing where the routers are aware of the server mac addresses and server may be ARP(Address resolution Protocol) disabled.

In direct routing, it is direct in the sense that all the income traffic is routed by the load balancer however all the outgoing traffic direct reaches the client which makes it super fast load balancing configuration.

Tunnel or a **IP tunneling** often looks like Direct routing where response is directly sent to client however the traffic between the router and the server can be routed.

In this, client sends the request to the virtual IP of load balancer which further encapsulates the IP packets, keeps a hast table and distributes it to the different servers as per the configured load balancing technique.

When the server is getting back the response, it decapsulates it and send back to the client directly according to the hash table which it has stored. This record is eventually removed from hash table when the connection is closed or there is a timeout.

2. Layer7 Hardware Load Balancing

This type of load balancers makes the decision according to the actual content of the message (URLs, cookies, scripts) since HTTP exists on the layer7.

These layer7 hardware actually form a **ADN (Application delivery network)** and they pass-on request to the servers as per the type of the content.

For example, the request for image will go to an image server, request for PHP scripts may to another server, HTML, js and css like static content may go to another one and request to any media content may go to yet another server.

So, here a load balancing effect is achieved by distributing load according to the type to content requested.

For this, it is also very helpful to understand the fundamentals of TCP/IP Protocol with the different layers.

The diagram below depicts a Layer7 load balancer.

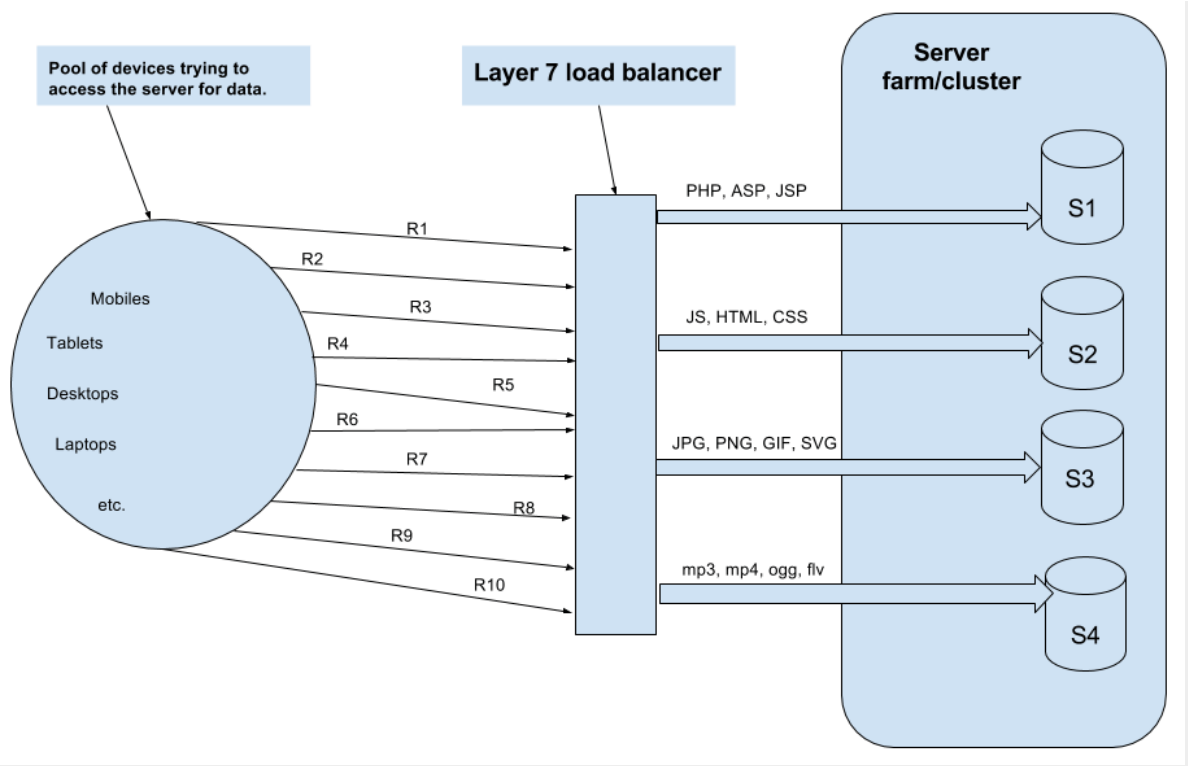


Fig: Load Balancer – Layer 7

Layer 7 load balancing uses the following three techniques:

1. URL parsing: From this they come to know about different type of contents.
2. Cookie sniffing: This helps them for a session aware routing.
3. HTTP reading: This method looks for http header information.

Hardware Load Balancer Examples

1. F5 BIG-IP load balancer (Setup HTTPS load balance on F5)
2. CISCO system catalyst
3. Barracuda load balancer
4. Coytepoint load balancer

Add your comment

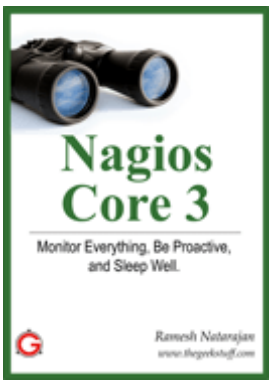
If you enjoyed this article, you might also like..

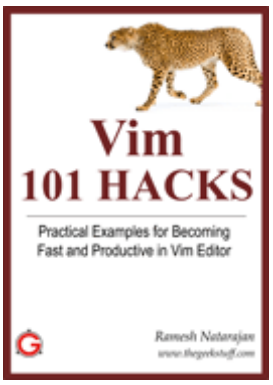
1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples

3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook **Free**

- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons





Tagged as: Hardware Load Balancer Example, Load Balancer Algorithm Examples, Load Balancer Fundamentals, Load Balancer Introduction Tutorial, Software Load Balancer Example, What is Load Balancer

Comments on this entry are closed.

Oleg January 12, 2016, 2:08 pm

Thank you for a great article!

LINK

Ramana January 12, 2016, 7:51 pm

Thank you for an informative post.

LINK

Vzboy January 12, 2016, 7:51 pm

Great writeup.

LINK

vicace January 21, 2016, 5:41 am

without load balancers.. we would end with skewness in load on resources.. thank you for posting

LINK

Slava February 1, 2016, 3:51 am

Thanks a lot for all that you are doing here, very good website!

LINK

Kevin February 1, 2016, 4:53 pm

Since most of the hardware load balancers also come in a virtual version, there is no functionality difference between hardware and software loadbalancers. The major difference used to be SSL acceleration and capacity, but those have become less important as better processors and distributed environments have changed the architectures.

As an example, Brocade sells the Stingray (formerly Zeus) software load balancer that installs on your favorite unix, but is as capable as any of the other commercial products.

LINK

steve November 24, 2016, 6:31 am

very good website. The information is very informative…..thanks a lot team

LINK

Next post: How to Install and Setup LXC Linux Containers on CentOS / RHEL / Ubuntu
Previous post: Happy New Year 2016 – From Geek and the Dolls