**Load Balancing**

IT industry is growing each day and so is the need for computing and storage resources. Large quantities of data are generated and exchanged over the network which further necessitates the need of more and more computing resources. Organizations, to better capitalize their investment, are opening their infrastructure to new found virtualization technologies like Cloud computing.

Cloud has helped enterprises leverage the benefits of computing resources which are shared over a virtualized environment. A lot of enterprises are already using cloud-based services in one or the other form. This brings us to the concept of load balancing in cloud.

What is load balancing in Cloud computing?

A website or a web-application can be accessed by a plenty of users at any point of time. It becomes difficult for a web application to manage all these user requests at one time. It may even result in system breakdowns. For a website owner, whose entire work is dependent on his portal, the sinking feeling of website being down or not accessible also brings lost potential customers.

Here, the load balancer plays an important role.

Cloud Load balancing is the process of distributing workloads and computing resources across one or more servers. This kind of distribution ensures maximum throughput in minimum response time. The workload is segregated among two or more servers, hard drives, network interfaces or other computing resources, enabling better resource utilization and system response time. Thus, for a high traffic website, effective use of cloud load balancing can ensure business continuity. The common objectives of using load balancers are:

* To maintain system firmness.
* To improve system performance.
* To protect against system failures.

How does load balancing work?

Here, load refers to not only the website traffic but also includes CPU load, network load and memory capacity of each server. A load balancing technique makes sure that each system in the network has same amount of work at any instant of time. This means neither any of them is excessively over-loaded, nor under-utilized.

The load balancer distributes data depending upon how busy each server or node is. In the absence of a load balancer, the client must wait while his process gets processed, which might be too tiring and demotivating for him.

Various information like jobs waiting in queue, CPU processing rate, job arrival rate etc. are exchanged between the processors during the load balancing process. Failure in the right application of load balancers can lead to serious consequences, data getting lost being one of them.

A [load balancer](https://www.cloudflare.com/load-balancing/) is a software or hardware device that keeps any one server from becoming overloaded. A load balancing algorithm is the logic that a load balancer uses to distribute network traffic between servers (an algorithm is a set of predefined rules). Different companies may use different load balancers and multiple load balancing algorithms like static and dynamic load balancing.

There are two primary approaches to [load balancing](https://www.cloudflare.com/learning/performance/what-is-load-balancing/). *Dynamic load balancing* uses algorithms that take into account the current state of each server and distribute traffic accordingly. *Static load balancing* distributes traffic without making these adjustments. Some static algorithms send an equal amount of traffic to each server in a group, either in a specified order or at random.

**What are the different types of load balancing algorithms?**

**Dynamic load balancing algorithms**

* *Least connection:* Checks which servers have the fewest connections open at the time and sends traffic to those servers. This assumes all connections require roughly equal processing power.Least connection load balancing is a dynamic load balancing algorithm where client requests are distributed to the application server with the least number of active connections at the time the client request is received. In cases where application servers have similar specifications, one server may be overloaded due to longer lived connections; this algorithm takes the active connection load into consideration. This technique is most appropriate for incoming requests that have varying connection times and a set of servers that are relatively similar in terms of processing power and available resources.
* *Weighted least connection:* Gives administrators the ability to assign different weights to each server, assuming that some servers can handle more connections than others.
* *Weighted response time:* Averages the response time of each server, and combines that with the number of connections each server has open to determine where to send traffic. By sending traffic to the servers with the quickest response time, the algorithm ensures faster service for users. The weighted response time load balancing algorithm that uses the application server’s response time to calculate a server weight. The application server that is responding the fastest receives the next request. This algorithm is appropriate for scenarios where the application response time is the paramount concern.
* *Resource-based:* Distributes load based on what resources each server has available at the time. Specialized software (called an "agent") running on each server measures that server's available CPU and memory, and the load balancer queries the agent before distributing traffic to that server. Resource based (or adaptive) load balancing makes decisions based on status indicators retrieved by LoadMaster from the back-end servers. The status indicator is determined by a custom program (an “agent”) running on each server. LoadMaster queries each server regularly for this status information and then sets the dynamic weight of the real server appropriately.

In this fashion, the load balancing method is essentially performing a detailed “health check” on the real server. This method is appropriate in any situation where detailed health check information from each server is required to make load balancing decisions. For example: this method would be useful for any application where the workload is varied and detailed application performance and status is required to assess server health. This method can also be used to provide application-aware health checking for Layer 4 (UDP) services via the load balancing method.

**Static load balancing algorithms**

* *Round robin:* [Round robin load balancing](https://www.cloudflare.com/learning/dns/glossary/round-robin-dns/) distributes traffic to a list of servers in rotation using the [Domain Name System (DNS)](https://www.cloudflare.com/learning/dns/what-is-dns/). An authoritative nameserver will have a list of different [A records](https://www.cloudflare.com/learning/dns/dns-records/dns-a-record/) for a domain and provides a different one in response to each DNS query.

[Round-robin](https://kemptechnologies.com/load-balancer/round-robin-load-balancing) load balancing is the simplest and most commonly-used load balancing algorithm. Client requests are distributed to application servers in simple rotation. For example, if you have three application servers: the first client request is sent to the first application server in the list, the second client request to the second application server, the third client request to the third application server, the fourth to the first application server, and so on.

Round robin load balancing is most appropriate for predictable client request streams that are being spread across a server farm whose members have relatively equal processing capabilities and available resources (such as network bandwidth and storage).

* *Weighted round robin:* Allows an administrator to assign different weights to each server. Servers deemed able to handle more traffic will receive slightly more. Weighting can be configured within [DNS records](https://www.cloudflare.com/learning/dns/dns-records/).

Weighted round robin is similar to the round-robin load balancing algorithm, adding the ability to spread the incoming client requests across the server farm according to the relative capacity of each server. It is most appropriate for spreading incoming client requests across a set of servers that have varying capabilities or available resources. The administrator assigns a weight to each application server based on criteria of their choosing that indicates the relative traffic-handling capability of each server in the farm.

So, for example: if application server #1 is twice as powerful as application server #2 (and application server #3), application server #1 is provisioned with a higher weight and application server #2 and #3 get the same, lower, weight. If there are five (5) sequential client requests, the first two (2) go to application server #1, the third (3) goes to application server #2, the fourth (4) to application server #3. The fifth (5) request would then go to application server #1, and so on.

#### **Fixed Weighting load balancing method**

Fixed weighting is a load balancing algorithm where the administrator assigns a weight to each application server based on criteria of their choosing to represent the relative traffic-handling capability of each server in the server farm. The application server with the highest weight will receive all of the traffic. If the application server with the highest weight fails, all traffic will be directed to the next highest weight application server. This method is appropriate for workloads where a single server is capable of handling all expected incoming requests, with one or more “hot spare” servers available to pick up the load when the currently active server fail.

* *IP hash load balancing algorithm:* Combines incoming traffic's source and destination [IP addresses](https://www.cloudflare.com/learning/dns/glossary/what-is-my-ip-address/) and uses a mathematical function to convert it into a hash. Based on the hash, the connection is assigned to a specific server.

The source IP hash load balancing algorithm uses the source and destination IP addresses of the client request to generate a unique hash key which is used to allocate the client to a particular server. As the key can be regenerated if the session is broken, the client request is directed to the same server it was using previously. This method is most appropriate when it’s vital that a client always return to the same server for each successive connection.

#### **URL Hash load balancing method**

The URL hash load balancing algorithm is similar to source IP hashing, except that the hash created is based on the URL in the client request. This ensures that client requests to a particular URL are always sent to the same back-end server.

## What are the advantages of Cloud Load Balancing?

### 1. High Performing applications

Cloud load balancing techniques, unlike their traditional on-premise counterparts, are less expensive and simple to implement. Enterprises can make their client applications work faster and deliver better performances, that too at potentially lower costs.

### 2. Increased scalability

Cloud balancing takes help of cloud’s scalability and agility to maintain website traffic. By using efficient load balancers, you can easily match up the increased user traffic and distribute it among various servers or network devices. It is especially important for ecommerce websites, who deals with thousands of website visitors every second. During sale or other promotional offers they need such effective load balancers to distribute workloads.

### 3. Ability to handle sudden traffic spikes

A normally running University site can completely go down during any result declaration. This is because too many requests can arrive at the same time. If they are using cloud load balancers, they do not need to worry about such traffic surges. No matter how large the request is, it can be wisely distributed among different servers for generating maximum results in less response time.

### 4. Business continuity with complete flexibility

The basic objective of using a load balancer is to save or protect a website from sudden outages. When the workload is distributed among various servers or network units, even if one node fails the burden can be shifted to another active node.