Load balancer:

Load balancing is the process of distributing tasks among more than one computing nodes to improve the system performance/response time /latency and availability.

**Reliability** can be defined as the probability that a system will meet certain performance standard and produce correct outputs up to some given time t. Reliability is enhanced by features that help to avoid, detect, and repair hardware faults. A reliable system does not silently continue and deliver results that include uncorrected corrupted data. Instead, it detects and, if possible, corrects the corruption

**Availability** means the probability that a system is operational at a given time. We can also say the percentage of time that the infrastructure , system, or service is operational under normal circumstances.

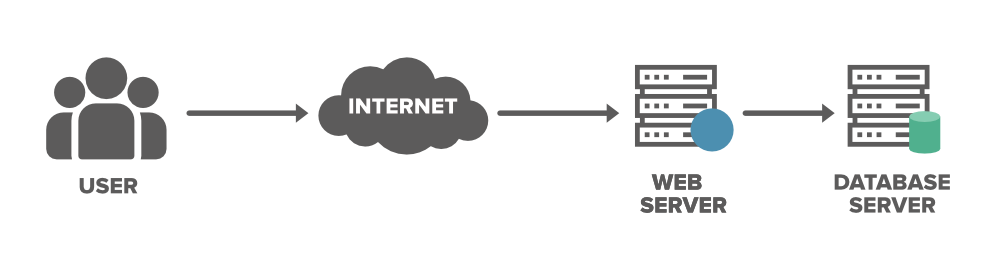
Table

Description automatically generated

A load balancer can be a physical device (Hardware), or a virtualized instance running on a specialized hardware, or software process.

Load balancer plays an important role in distributed systems. It enables the horizontal scaling.

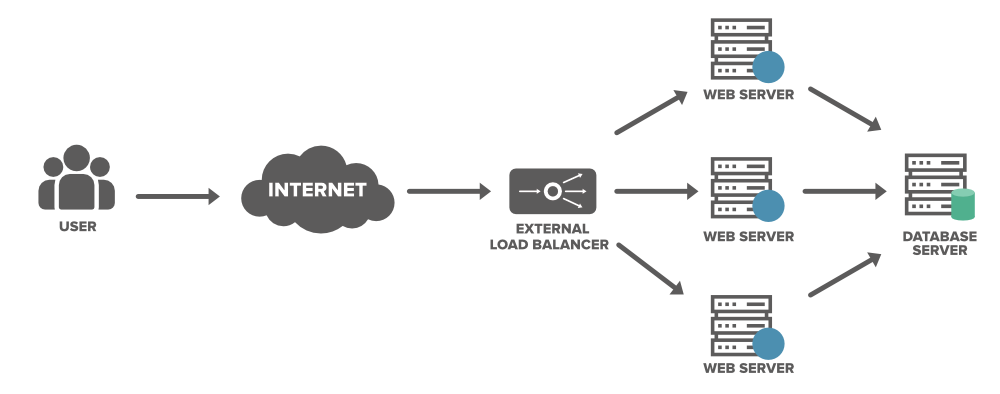
Consider the case of a single serve where the client requests are never load balanced. All the requests are processes by the single server.



As the application gains popularity, the load on the single server may lead to service failure, bad customer experience because of high latency. So, this deployment model will have two issues ***Single point of failure, Overloaded service(Degradation in system performance and over all latency).***

Load on the server can somewhat handled by adding more capacity to the system (Vertical scaling , will have complete service downtime ) but it has a maximum limit.

Best approach to solve both the issues is to add new servers and distribute the load among them based on some load balancing algorithm. This is where, load balancers plays an important role.



## Advantages Of load balancer

* Load balancers minimizes the service response time and maximizes the service throughput.
* Load balancers ensures the high availability and reliability by sending the traffic only to the online servers.
* Load balancers do continuous health checks to monitor the server’s capability of handling the requests.
* Load balancers can also bring up and bring down the services based on the current load on the system. (Dynamic scaling)
* End user only interacts with the load balancer, he doesn’t interact with the actual servers directly, so it guards the application server from cyber-attacks.
* **Continuous** **deployment**: it's possible to roll out software updates without taking the whole service down, by using the load balancer to take out one machine at a time.
* **The load balancers can provide additional security too**. They can reject the suspicious traffic before they could reach the server.

# Positions of a load balancer

A load balancer can be used between

1. Between User/Client application and web servers.
2. Between web servers and application servers.
3. Between application servers and cache servers.
4. Between cache servers and database servers.

# Types of Load Balancers

**Hardware Load balancer**:

* Hardware load balancer, also known as hardware load balancing device, is a proprietary appliance built on customized ASICS, to distribute traffic across a cluster of nodes.
* HW Load balancers generally operate at L4. It uses both L4 and L3 information to distribute the load among the nodes in the cluster.
* To optimize the performance HW load balancers distribute traffic based on customized rules, so that the nodes in the cluster are not overwhelmed.
* Traditionally these HW load balancers are deployed inside a datacenter. To avoid single point of failure these HW load balancers are deployed in pairs (HA).
* Hardware load balancers can’t support dynamic scaling of LB, so it must be over provisioned to avoid performance degradation during the peak hours.

### **Software load balancer**

* A software load balancer can be deployed on a standard x86 server or virtual machine.
* A software load balancer can operate at either L4 or L7 layer.
* It evaluates client requests by examining application-level characteristics (the IP address, the HTTP header, and the contents of the request)
* Software load balancing offers the same functionality of an HLD, but it does not require a dedicated load balancing device. The load-balancing software can run on a regular server, or even a virtual server.
* Load Balancer as a service (LBaaS)
* Software load balancer provides dynamic auto scaling of LB service.

### **L4 load balancer**

* Layer 4 load balancers are also referred to as Network Load Balancer (NLB).
* An L4 load balancer acts on information found in the network and transport layers of the request. This includes the protocol (TCP, UDP, etc.), source and destination IP addresses, and source and destination port numbers.
* Layer-4 load balancer mostly use the IP address, and sometimes port number to distribute the traffic among a group node.
* Load balancer acts as reverse proxy. It uses virtual IP addresses for communicating with end users.
* When the Layer 4 load balancer receives a request and makes the load balancing decision, it also performs Network Address Translation (NAT) on the request packet, changing the recorded destination IP address from its own to that of the content server it has chosen on the internal network. It will also change the source IP address of the packet to one of its addresses.
* Similarly, while forwarding the server response back to the client, it will do NAT to change the source IP from server address to its own address and destination address from its own to actual client address.
* So, this required L4 load balancers to maintain state.

Pros

* L4 load balancer is very simpler compared to L7 load balancer. It is efficient and secure as it doesn’t require to investigate the L7 data.
* There is one connection or pipe between the user and the application server.

Cons

* As it doesn’t investigate data it can’t use load balancing based on data.
* It can’t be used with microservices.
* L4 load balancer suffers from ***sticky segments.* Let’s understand what sticky segment is.**  We have a TCP connection between the sender and receiver, based of the MSS negotiated between the TCP end points L7 packet will be sent in multiple TCP segments. For all these segments IP addresses remain same. So, once load balancer selects the server end of the connection, it MUST send all the segments to the same server.
* With L4 load balancer we can’t use cache at load balancer level to improve the latency. For a given application even tough same content is send to multiple users from the application server we can’t cache them.

**East-west traffic**, in a networking context, is the transfer of data packets from server to server within a data Center. Traffic among the nodes within a datacenter.

North-South traffic refers to the traffic that flows into or out of the datacenter.

### **L7 load balancer**

* Layer 7 load balancer also know as the **Application Load Balancer (ALB) or HTTP (S) load balancer.**
* It will be able to see the data carried in the request, both plain and encrypted ones. Then use this data to take the routing decision.
* For example, they can route requests for video content to a pool of servers optimized for video, requests for static content to a different set of servers, etc.
* They can also route requests based on the user, so that the same user always lands on the same server (for session stickiness) or the same pool of servers (for performance reasons).
* For HTTPS traffic there will be one TLS connection between the client and the load balancer. Load balancer will have its own valid TLS certificate.
* The connection between the Load balancer and the actual servers may or may not be encrypted.
* With L7 load balancer there will be a two TCP connections or tunnels between the client and the server.

Pros

* You can do smart load balancing
* We can do caching at load balancer level
* It can be used with microservices

Cons

* As it looks inside the data it impacts the overall performance and latency. With modern high-end hardware, it might be very negligible.
* TLS termination leads to storing of multiple certificates for different services served via the load balancer.
* It has two TCP connection we need to link them together

# Load Balancing Algorithms

We need a load balancing algorithm to decide which request should be redirected to which backend server.

Round Robin

Weighted Round Robin

Least Connection Method

Least Response Time Method

Least Bandwidth Method

IP hashing

# Features

1. The software-defined, scale-out architecture for autoscaling of elastic load balancers.

We need to think why LB needs scale out can’t we just have 2 node cluster in Active-Active or active backup?

1. With L7 load balancer we can see into the data and use it in our decision process. For example, consider Instagram service. Here we can have different server for pictures and comments. If the rest API end point is /picture, then the load balancer will send the request to picture service. So, if we have multiple picture servers then load balancer can decide which one to send.
2. Sticky Segments
3. Load Balancer Need to do health check of the servers.
4. How to configure the server details to the load balancer. Initially we can start with configuration file. Later when we go for containerization then it has to be from the orchestrator.
5. Whit L7 load balancer how many backend connections we must maintain with the actual servers. We need to link the client connection to a serving backend connection. Multiple client connection will be linked to the same backed connection so how we do that?
6. Stateful vs Stateless Load balancer.

<https://avinetworks.com/glossary/hardware-load-balancer/>

<https://igotanoffer.com/blogs/tech/load-balancing-system-design-interview>

<https://www.geeksforgeeks.org/load-balancer-system-design-interview-question/>

[Software Load Balancer (SLB) for SDN in Azure Stack HCI and Windows Server - Azure Stack HCI | Microsoft Docs](https://docs.microsoft.com/en-us/azure-stack/hci/concepts/software-load-balancer)

<https://oshyshkov.com/2018/07/20/load-balancing/> (Details of Cloud and SW Load balancer)

<https://www.resonatenetworks.com/2020/04/29/layer-4-vs-layer-7-load-balancing/>

[Software Load Balancer (SLB) for SDN in Azure Stack HCI and Windows Server - Azure Stack HCI | Microsoft Docs](https://docs.microsoft.com/en-us/azure-stack/hci/concepts/software-load-balancer)

How to decide if we need a software load balancer or a HW load balancer.

One factor is scalability. If an elastic scalability is desired it is always better to go with the s/w load balancer.

# Notes

Virtual IP address (Think like loopback address, No Physical interface is required )

Virtual Router Redundancy Protocol

Zookeeper

<https://www.youtube.com/watch?v=Zgy1miPsTNs>

In Kubernetes, an Ingress is an object that allows access to your Kubernetes services from outside the Kubernetes cluster.