Load-Balancer

CS 249
Team #2
Rashmeet Khanuja, Anusha Vijay, Steven Yen

A challenge in distributed system

- An application is deployed on multiple nodes that form a cluster.
- Appears as one to the user.
- The nodes in the cluster may or may not have the same computing capacity.

With different configurations and capabilities, how do we ensure minimized response times and reduce idle time of CPU?

Answer: Load Balancer ❗
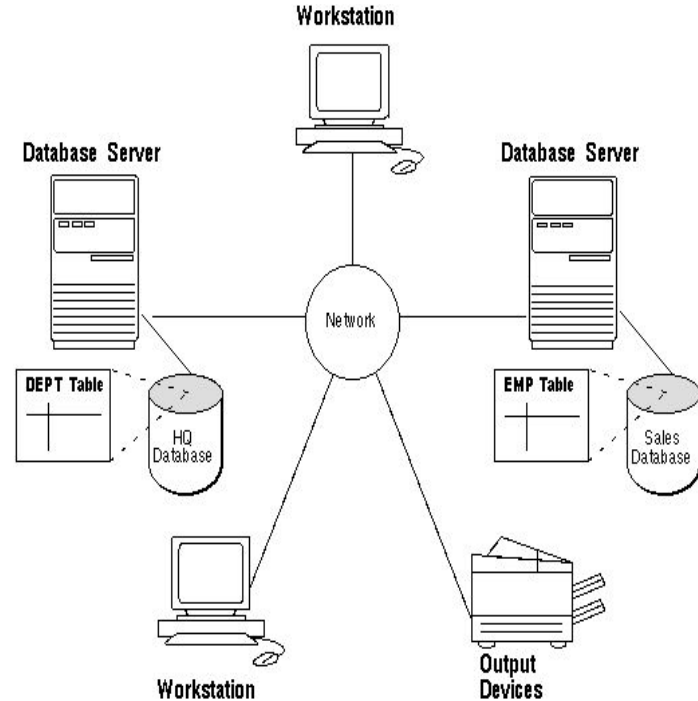
Role of a Load Balancer

- It is a software or hardware component that distributes the workload amongst the nodes.

- Ensures no single server is overloaded.

  - Evenly distributes the load on each server based on their capacity.

- Acts as a single point of entry for the end user

  - Complexities of redirection is abstracted from users.

- Increases reliability, efficiency, scalability, reusability, and availability.
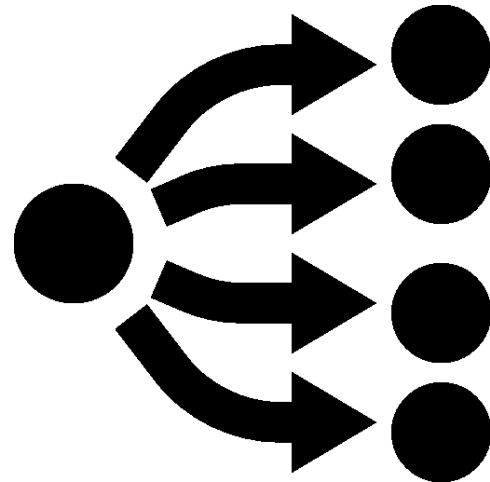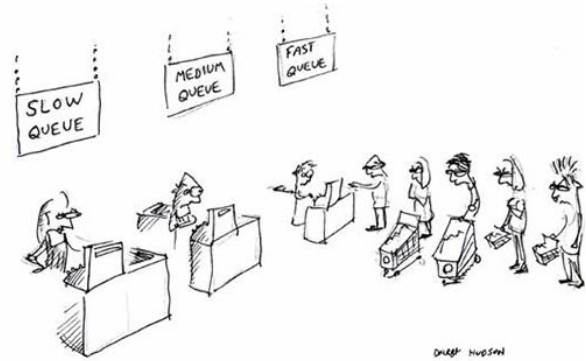
Examples of Load Balancers

NetScaler
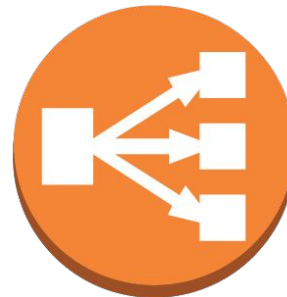
Resonate

NGINX Plus

Zen Load Balancer

Amazon ELB

BalanceNG

LVS

F-5 Big-IP Local Traffic Manager

Cisco Load Balancer

Barracuda Load Balancer ADC

Web Server without load balancer

No Load Balancing
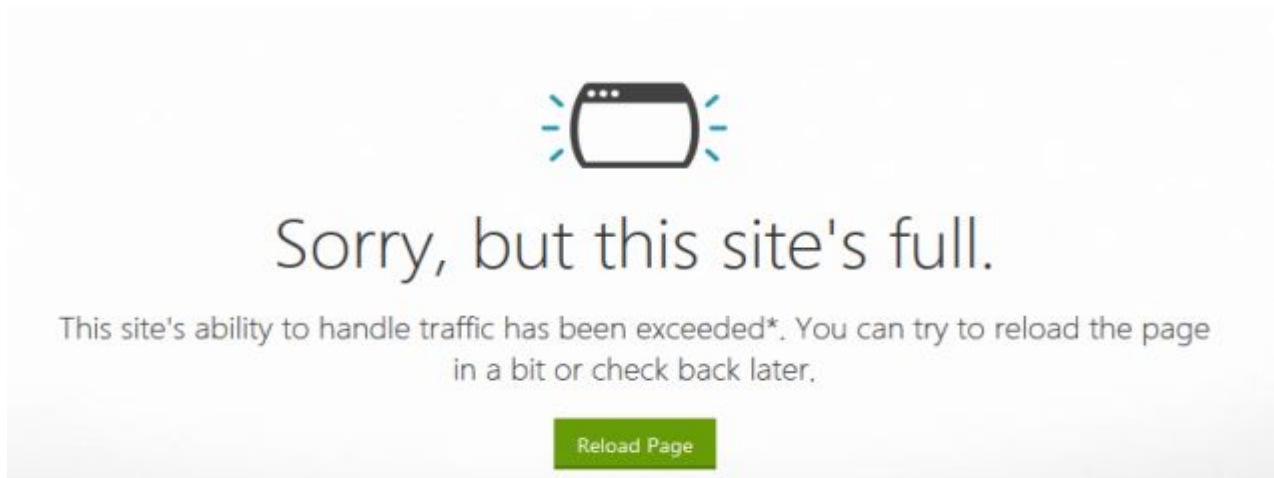
User → Internet → Web Server → Database Server

http://yourdomain.com/

- In this monolithic architecture, as the number of users trying to access the server simultaneously increases , chances of failing to service user requests with optimal response times increases.
- Users may experience slow load times or something worse.

To Avoid This..

Sorry, but this site's full.

This site's ability to handle traffic has been exceeded*. You can try to reload the page in a bit or check back later.

Reload Page

- A website with significant traffic will benefit from a distributed architecture with a load balancer
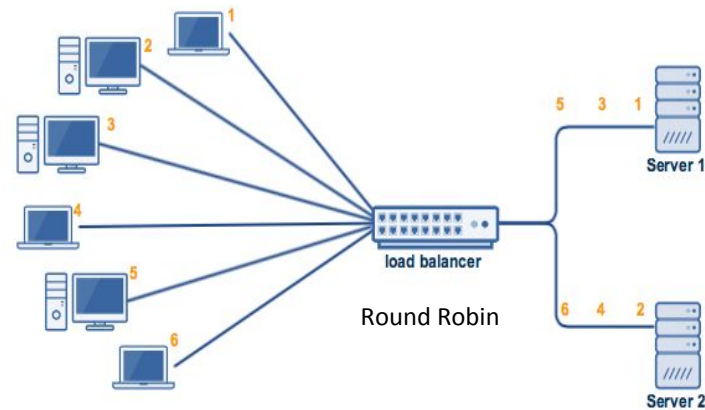
Distribute load between servers

To manage the incoming traffic:

- Load balancer listens on a port for incoming traffic from clients (e.g., a user's browser).
- Redirects any incoming client request (e.g., a HTTP GET request) to one of the many servers in the cluster.
- Server responds to the load balancer which is redirected back to the client.
- This is typically a software load balancer
- Benefits: hides the topology of the network/cluster from the client, providing some security. Faster response time for each client.

User

Internet

Load Balancer

Top Layer

Replicated Servers

Bottom Layer

App 01   App 02
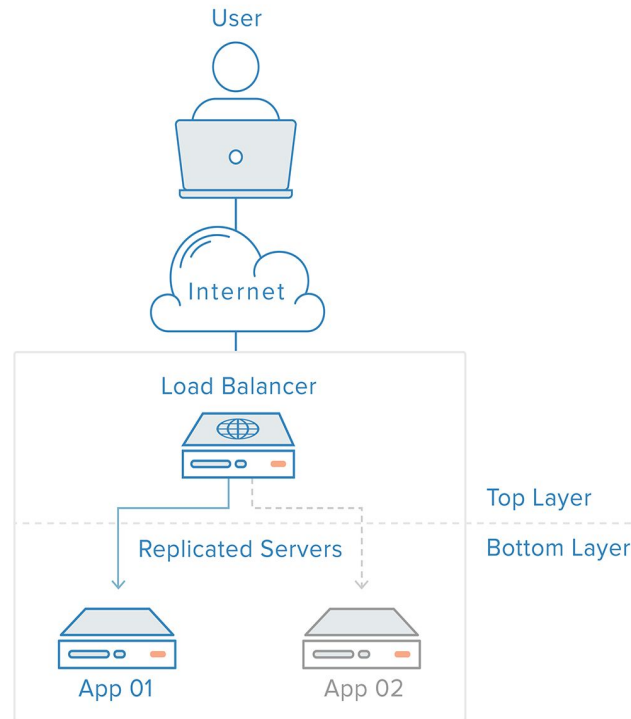
Software Load-Balancer

- Installed on all serving nodes
- Implement a combination of scheduling algorithms such as **Round-Robin** and **Least Connection First**
- **Weighted round-robin and weighted least connection.**
  - Useful when the servers/nodes have different capabilities (e.g. processing power/RAM).
  - Weight associated with the server depends on the hardware capabilities

Round Robin

Drawbacks

- Could be sensitive to OS versions
- Virtual appliance deployments could experience hypervisor dependencies
- Server's RAM and CPU

Hardware Load-Balancer

- Specialized routers or switches deployed between servers and clients

- Rely on firmware to supply the internal code base - the program

- Routing is either randomized (e.g., round-robin), or based on factors such as

    - available server connections,

    - server processing power,

    - resource utilization

- Also knows as *layer 4-7 router* since they are implemented on Transport layer and Application layer of OSI model

Drawbacks

- Cost becomes considerable when global server load balancing or cross data center load balancing needs to be done
    - At least one appliance in each of the data centers
- A central box to manage load distribution between those appliances