Load Balancing

Darko Bozhinoski,
Ph.D. in Computer Science
Email: D.Bozhinoski@innopolis.ru

What is Load Balancing?

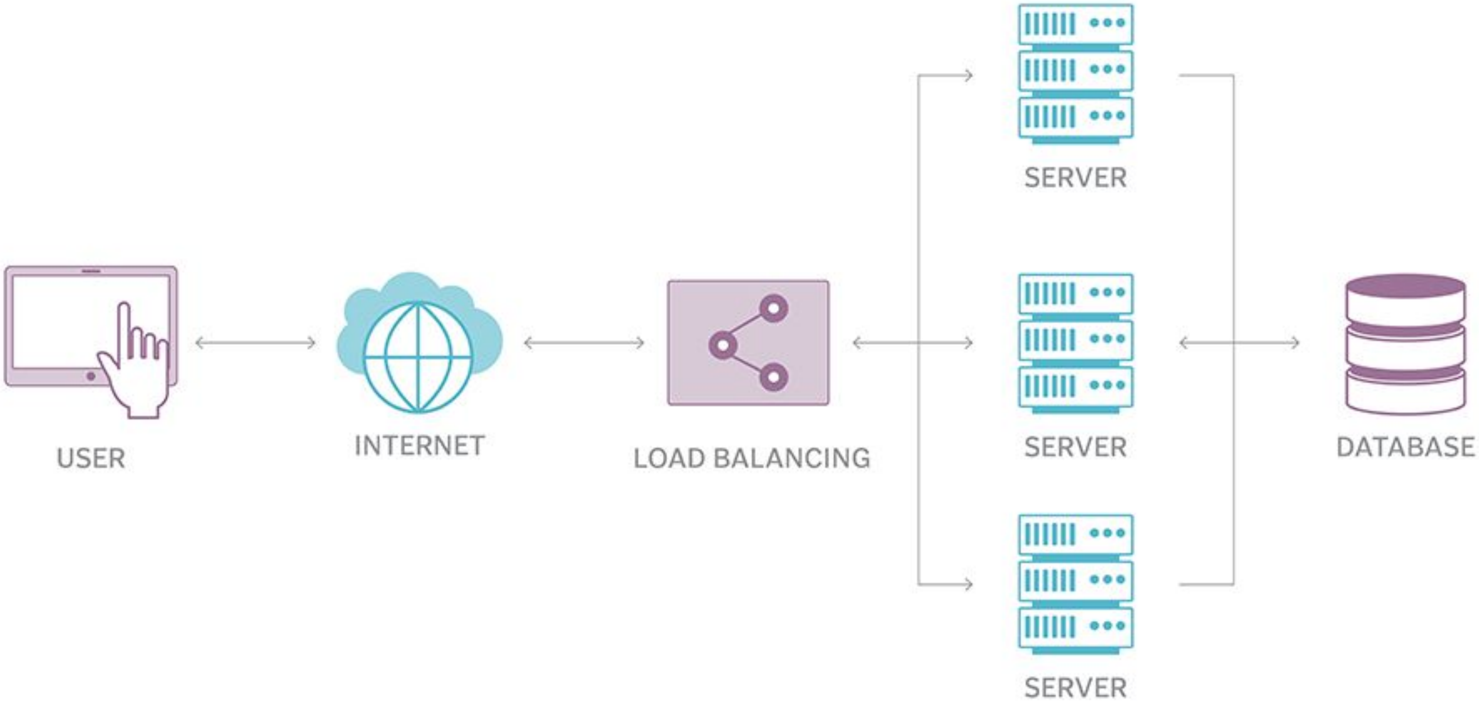
- Most performance problems are related to competition for shared resources (processor, memory, disk, network capacity, etc.)
- Load balancing is the process of distributing workloads across computing resources in a cloud computing environment and carefully balancing the network traffic accessing those resources.
- Load balancing enables organizations to meet workload demands by routing incoming traffic to multiple servers, networks or other resources, while improving performance and protecting against disruptions in services.

Why Load Balance is needed?

Why Load Balance?

- Improve performance
- Improve redundancy
- More cost effective scaling
- More cost effective redundancy

How load balancing works



Intro: Load Balancer

Once a load balancer is introduced, it acts as a network front end and often uses a single IP address to receive all network traffic intended for the target workload. The load balancer evenly distributes the network traffic to each available workload instance or throttles traffic to send specific percentages of traffic to each instance.

Server load balancing architecture

Imagine a distributed environment containing:

– Several clients

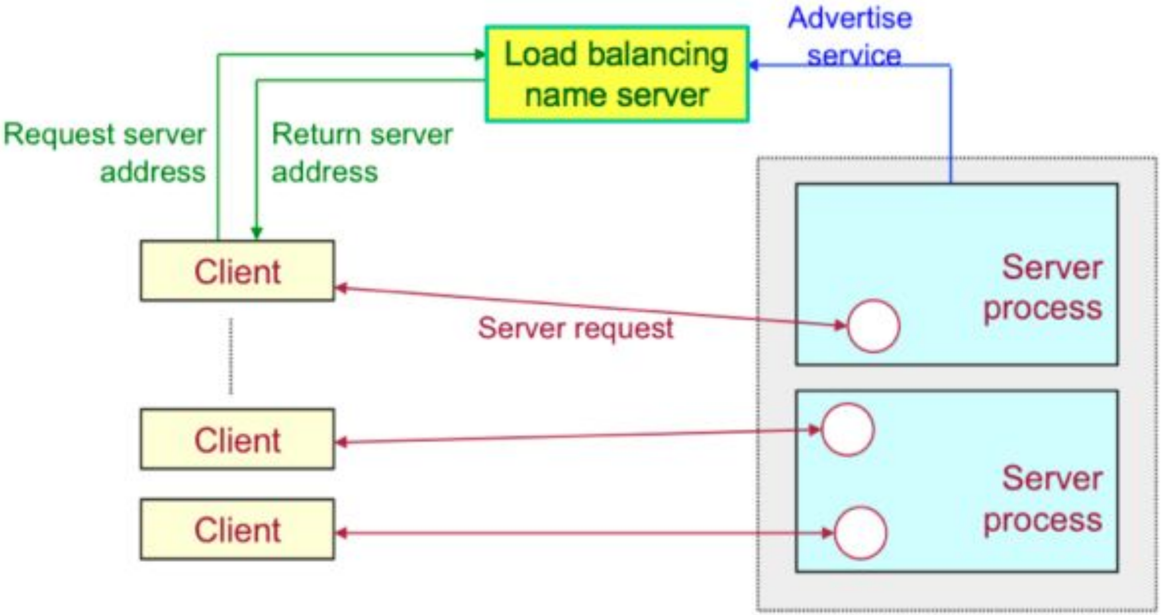
– Multiple replicated servers that can serve client requests

- How are client requests balanced across the servers?

=> This is the essential task of load balancing

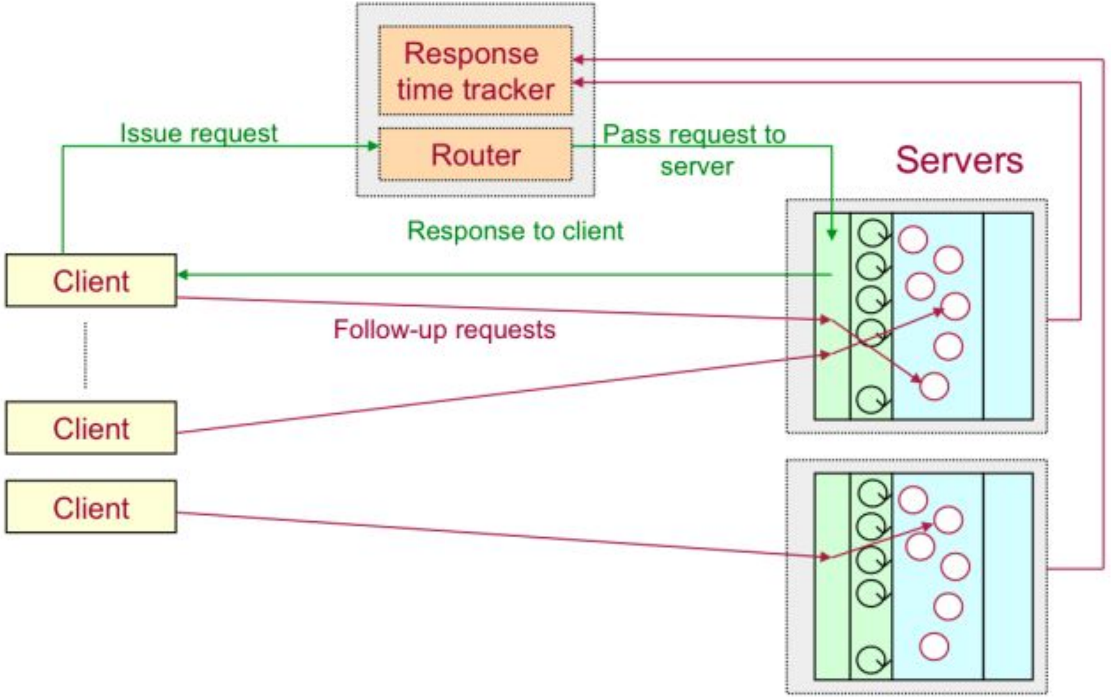
Server load balancing architecture - example 1

- Client calls on name server to find the location of a suitable server
- Name server can spread client objects across multiple servers
- Client is bound to server until it decides to request new address from name server



Server load balancing architecture - example 2

- Client calls load balancing router when issuing request
- Router passes request to chosen server
- Follow up dialogue directly from client to same server

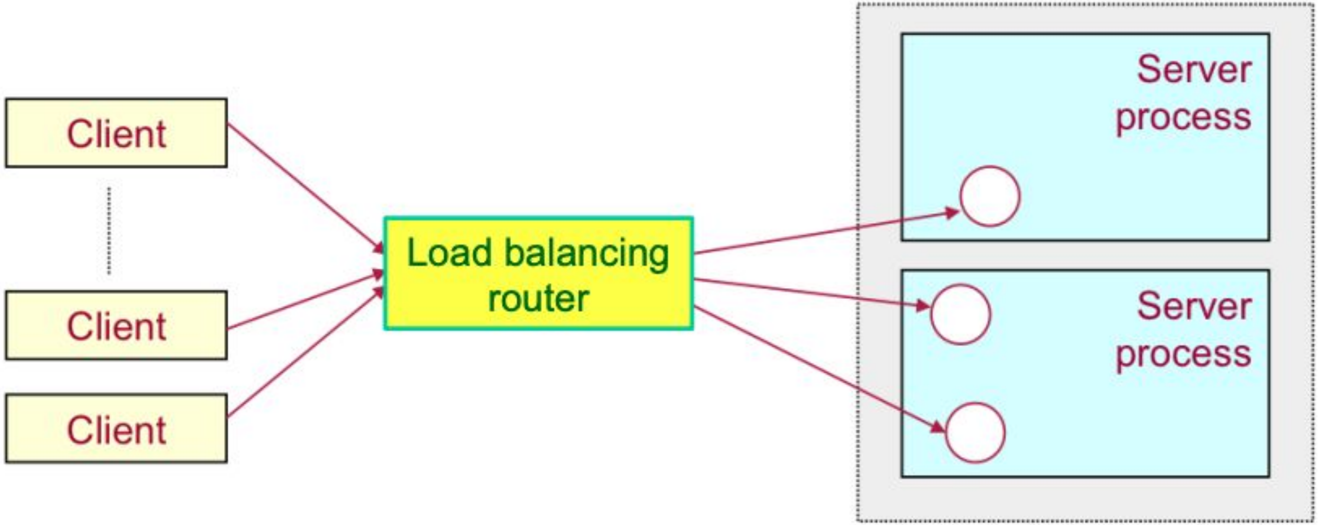


Server load balancing architecture - example 2

- Scales well …
  - – Router only involved when request is issued
  - – Follow-up dialogue directly from client to same server
    - • Good for handling state
  - – Initial server allocation based on response time
    - • Client allocated to least-loaded server
- But …
  - – No support for dynamic rebalancing as server load changes

Server load balancing architecture - example 3

- Some systems involve the router in every browser request
  - Request goes to router who then passes it on to a server process
  - Router can be a bottleneck here
    - Need to consider router scalability and fault tolerance as well!



Types of load balancers

Software-based load balancer:

Software-based load balancers run on less expensive hardware (standard intel x86, for example). As a result, modern businesses, particularly SaaS cloud service providers, rely on software load balancers. Simply install the software in cloud computing environments such as Amazon’s AWS to eliminate the need for a physical appliance.

Hardware-based load balancer:

Hardware-based load balancers are dedicated boxes that contain Application Specific Integrated Circuits (ASICs) that have been optimized for specific use. ASICs enable high-speed network traffic promotion. Since hardware-based load balancing is significantly quicker than software, these are frequently used for transport-level load balancing.

OSI Model

- Layer 1: Physical (cable, electrical)
- Layer 2: Datalink (example: Ethernet)
- Layer 3: Network (example: IP)
- Layer 4: Transport (example: TCP)
- Layer 5: Session
- Layer 6: Presentation
- Layer 7: Application (example: HTTP)

Types of load balancing in cloud computing

- ❖ **Layer 4:** Some cloud load-balancing services operate at Layer 4 to direct data from transport layer protocols *TCP, User Datagram Protocol (UDP)). Load balancing at this lower level of the network stack provides the best performance -- millions of network requests per second with low latencies -- and is a great option for erratic or unpredictable network traffic patterns.
  Layer 4 load-balancing services include AWS Network Load Balancer, Google Cloud Platform (GCP) TCP/UDP Load Balancing and Microsoft Azure Load Balancer.

- ❖ **Layer 7: HTTP(s):** load balancing at this level is the most traditional type of load balancing. This means that load balancing occurs at the operational Layer. It is the most adaptable load-balancing method because it allows you to make delivery decisions based on information obtained from HTTP addresses. Each of the major cloud providers has its own feature or service for this: AWS Application Load Balancer, Azure Application Gateway and Google Cloud HTTP(S) Load Balancing.

Scheduling algorithms – server selection

Several approaches, including:

– Round robin: Apply each successive request to each server in turn

– Highest response time: Based on monitoring server performance

– Lowest load: Based on monitoring server resource utilisation

– Match request size to server performance

– Match request priority to server performance/availability

– Combination of the above

Choice of a load balancer

❖ The choice of a cloud load balancer should extend beyond traffic types alone. Cloud providers also differentiate load-balancing services based on scope and framework.

❖ For example, GCP suggests global load-balancing services when workloads are distributed across multiple regions, while regional load-balancing services are a good fit when all workloads are in the same region. Similarly, GCP suggests external load balancers when traffic comes into the workloads from the internet and internal load balancers when traffic is intended for use within GCP.

Load balancer comparison

AWS

AWS: Amazon’s Elastic Load Balancing (ELB) can be used to distribute traffic across multiple EC2 instances. The service is elastic (i.e. changeable) and fully managed which means that it can automatically scale to meet demand.

There are three types of load balancers available in AWS.

- ❖ **Classic Load Balancer (CLB)** operates on both the request and connection levels for Layer 4 (TCP/IP) and Layer 7 (HTTP) routing. It is best for EC2 Classic instances.
- ❖ **Application Load Balancer (ALB)** works at the request level only. It is designed to support the workloads of modern applications such as containerized applications, HTTP/2 traffic, and web sockets.
- ❖ **Network Load Balancer (NLB)** operates at the fourth layer of the (OSI) Open Systems Interconnection model. It is capable to handle millions of requests per second.

AZURE

Azure: Three types of load balancers in Azure: Azure Load Balancer, Internal Load Balancer (ILB), and Traffic Manager. The various load balancers ensure that the traffic is sent to healthy nodes.

- **Internal Load Balancer (ILB)** has an internal-facing Virtual IP. Meaning, users can apply an internal load balancing for virtual machines (VM) that are connected only to an internal Azure cloud service or a virtual network.
- **Traffic Manager** is an internet-facing solution that balances the traffic loads at various endpoints using a policy engine as well as a set of DNS queries. It can route traffic to any region's service and even to non-Azure endpoints.
- **Azure Load Balancer**: is a high-performance, ultra low-latency Layer 4 load-balancing service (inbound and outbound) for all UDP and TCP protocols. It is built to handle millions of requests per second while ensuring your solution is highly available.

Google Cloud

Google Cloud: The Google Cloud Load Balancer (GCLB) provides server-side load balancing to distribute incoming traffic to multiple virtual machine instances. It allows users to direct applications across any region and scale compute with very little configuration. It can load 0-1 million requests per second with no pre-warming. Three deployment types of load balancing services in Google: Global, Network and Internal.

- ❖ **Global Load Balancing** supports HTTP(S) traffic for modern web-based applications. Traffic is distributed to the region that is closest to the calling user, provided the region has available capacity.
- ❖ **Network Load Balancing** directs traffic across virtual machine (VM) instances in the same region in a VPC network. Any TCP and UDP traffic can be load balanced on the basis of source, destination port, and protocol so that the traffic from the same connection reaches the same server.
- ❖ **Internal Load Balancing** is a regional load balancer that distributes the internal traffic across a set of back-end instances without requiring a public IP address.

Alibaba Cloud

Alibaba Cloud: Alibaba’s Cloud Server Load Balancer (SLB) redirects the incoming traffic among various instances to balance and improve the service capabilities of applications. It can process up to millions of requests all at the same time and quickly meet the requirements during large demands. This avoids service outages.

It checks the service availability of ECS instances by performing health checks. In the case of unhealthy instances, it automatically removes them to avoid a single point of failure. You can also reduce the frequency of health check by increasing the interval time or changing a layer-7 health check to layer-4 health check based on the service condition.

Alibaba Cloud

Alibaba Cloud (2): SLB also provides URL-based routing that allows users to redirect the incoming traffic to the backend server, based on URLs. This lets you configure the SLB across different zones of the region, so even if communication to one zone is interrupted, SLB automatically directs the traffic to zone 2 that is working normally.

Cross Region Disaster Tolerance: you can configure SLB instances in different regions, and add ECS instances in various zones of the regions along with DNS service. The DNS service can resolve domain names, add IP addresses of different regions to different address pools of the Server Load Balancer and perform health checks. So, if a region becomes unavailable, this will automatically stop the domain name resolution for that unavailable region.

Exercise

In this lab you should install a web server on a “master” instance and configure a load balancer to direct traffic to multiple replicas of this. These replicas will be created manually rather than with auto-scaling.

Part A. “Master” instance

1. Launch a standard Amazon/Alibaba/Google Cloud/Yandex Linux instance:

- Choose the smallest instance (e.g. nano, micro in step 1)
- Tag the instance with a suitable name
- Ensure it allows inbound SSH and HTTP.

2. Connect to your instance with SSH/PuTTY.

3. Install the apache web server (httpd), enable at startup, and start the service.

You should now be able to browse to the public IP address or the public DNS name of your instance and see the apache default page.

Part A. “Master” instance

4. Create a new default page: sudo nano /var/www/html/index.html Anycontent is fine here. Refresh the page in your browser to check that it worked.

5. Create an image of the instance. Give the image a name and description. It takes a couple of minutes for the image to become available.

Part B. Load Balancing

6. Launch three instances based on the image. Place the instances in availability zones by your choice.

7. Configure the instance details based on your preferences, but ensure that the security group that is chosen allows inbound HTTP and SSH traffic.

8. Log into each of the instances separately and make a small edit to the home page to help identify it.

9. Go to Load Balancers and create your Load Balancer. Create listeners. Register all running instances with the load balancer and specify the different weights. Choose any schedule algorithm.

10. Now you should test the load balancer with a browser.

11. Examine the apache log files on each of your instances and try to trace a particular refresh to find out which instance handled that request.You can find apache logsin /var/log/httpd/. A useful command: `sudo tail -f /var/log/httpd/access_log`

12. Try to change the schedule algorithm and the weights if the corresponding servers. Repeat the test again. What has changed?

Part C. Load Testing

Bonus Task:

- Create an application or a simple script to create a large number of user requests on the load balancer.
- Try to note down how many of those requests end up on in which of the instances of your server.
- You can use one of the following load testing tools: https://testguild.com/load-testing-tools/

Expected Deliverables

1. Write report for all work related to this lab.

2. Include description of:

    a. workflow explanation (include all commands you used to make it work)

    b. Report all the configuration you created, what you choose, why you did it and what was the result.

    c. Screenshots:

    - of your web server directly without load balancer (has to show incoming traffic from multiple locations and through the load balancer)
    - of visiting the load balancer page