



University of Colorado  
Boulder

# LOAD BALANCING STRATEGIES, ALGORITHMS AND COMPARISON

**CSCI-ECEN 5273: Network Systems**

Alex Costinescu and Suman Hosmane



# Contents

- ❖ Load balancing and advantages
- ❖ Motivation for the project
- ❖ Load Balancing Strategies
- ❖ Technical Aspects and Implementation
- ❖ Demo
- ❖ Q&A + Discussion



# Load-Balancing

- ❖ Load balancing is a computer networking method to distribute workload across multiple processing computers or a cluster, network links, central units, disk drives, or other resources.
- ❖ Ease of administration / maintenance
  - Easily and transparently remove physical servers from rotation to perform maintenance.
- ❖ Resource sharing
  - Run multiple instances of an application/service on a server
  - Load-balance to different port based on data analyzed.



# Motivation for the Project

- ❖ Learn about load balancing
- ❖ Achieve Optimal Resource Utilization
- ❖ Maximize Throughput
- ❖ Minimize Response Times
- ❖ Avoid Overload and Crashing of Infrastructure
- ❖ Performance Comparison with each strategy

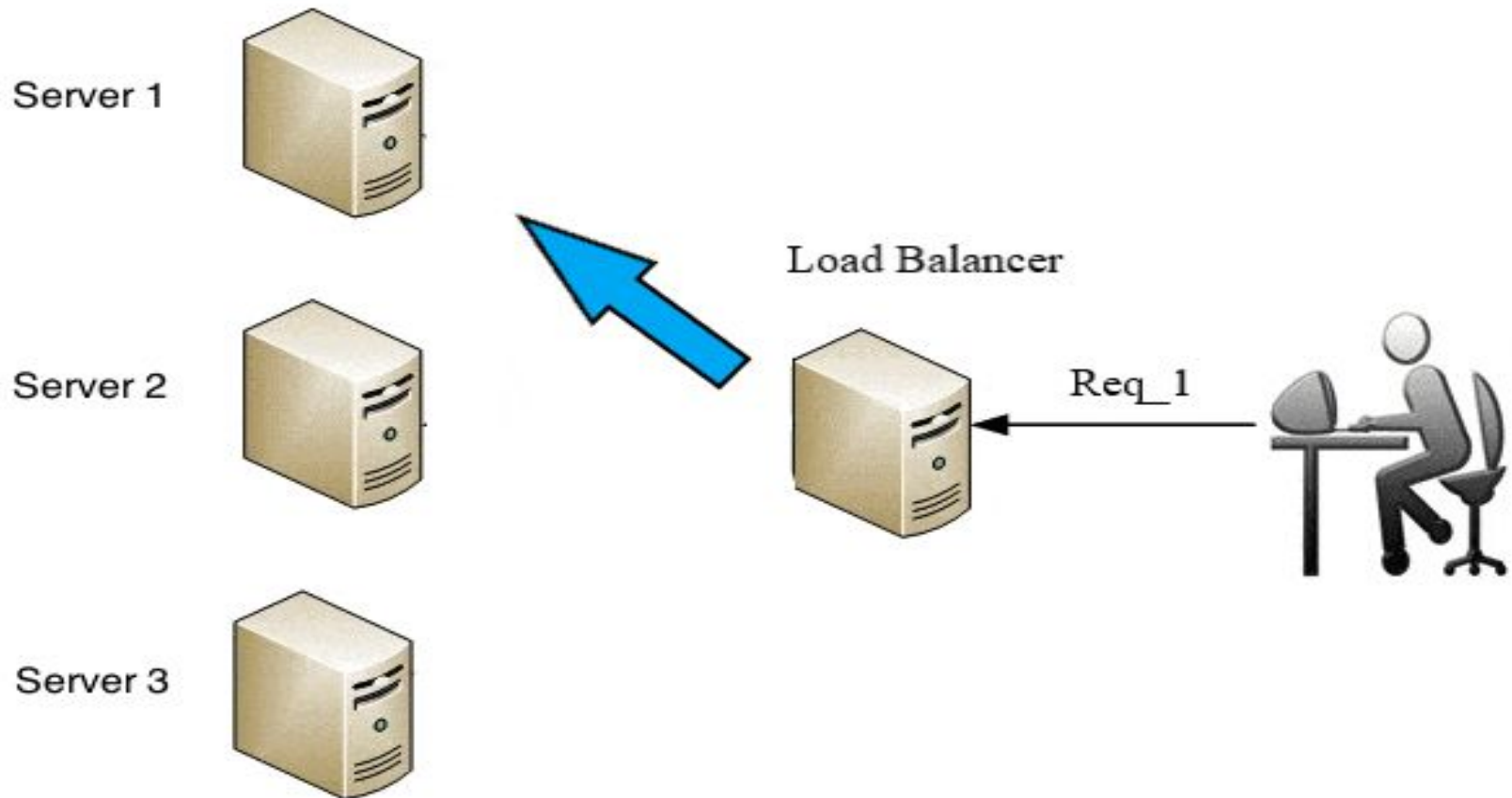


# Load Balancing Strategies

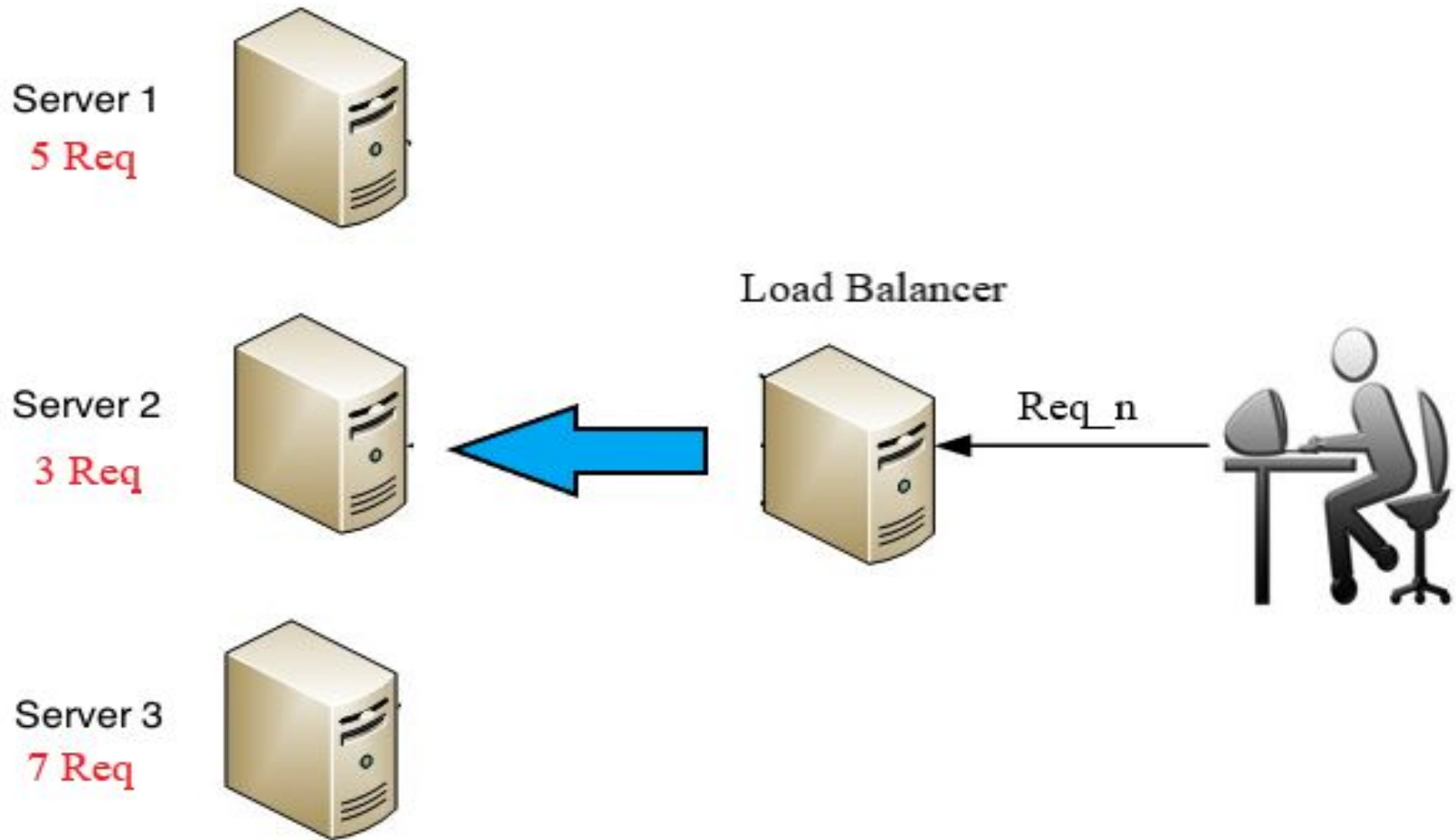
- ❖ Static Load Balancing
- ❖ Dynamic Load Balancing
- ❖ *Round-Robin*
- ❖ *Least connections*
- ❖ *Chained Failover*
- ❖ Weighted Round-Robin
- ❖ Weighted Response Time



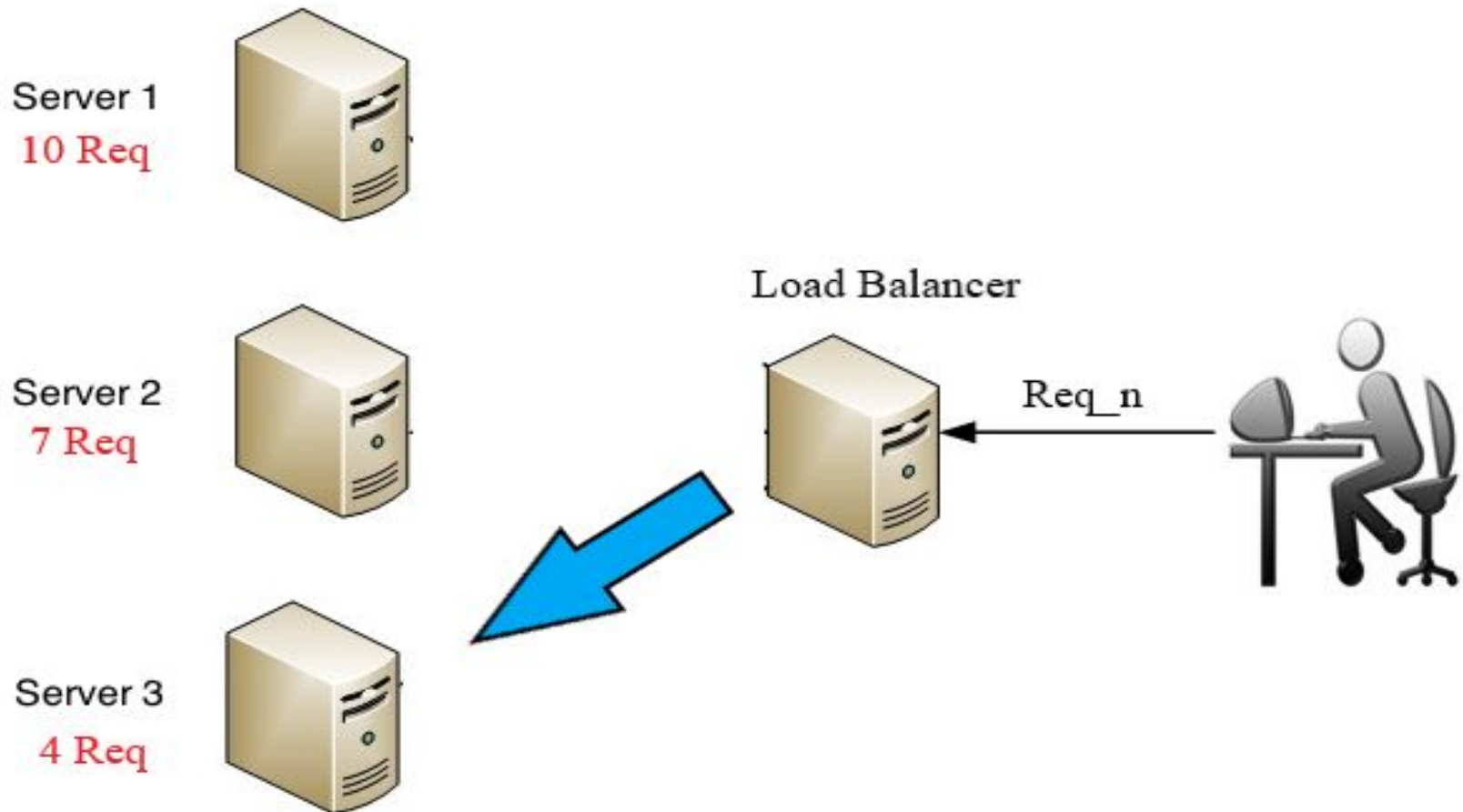
# Round-Robin



# Least Connection

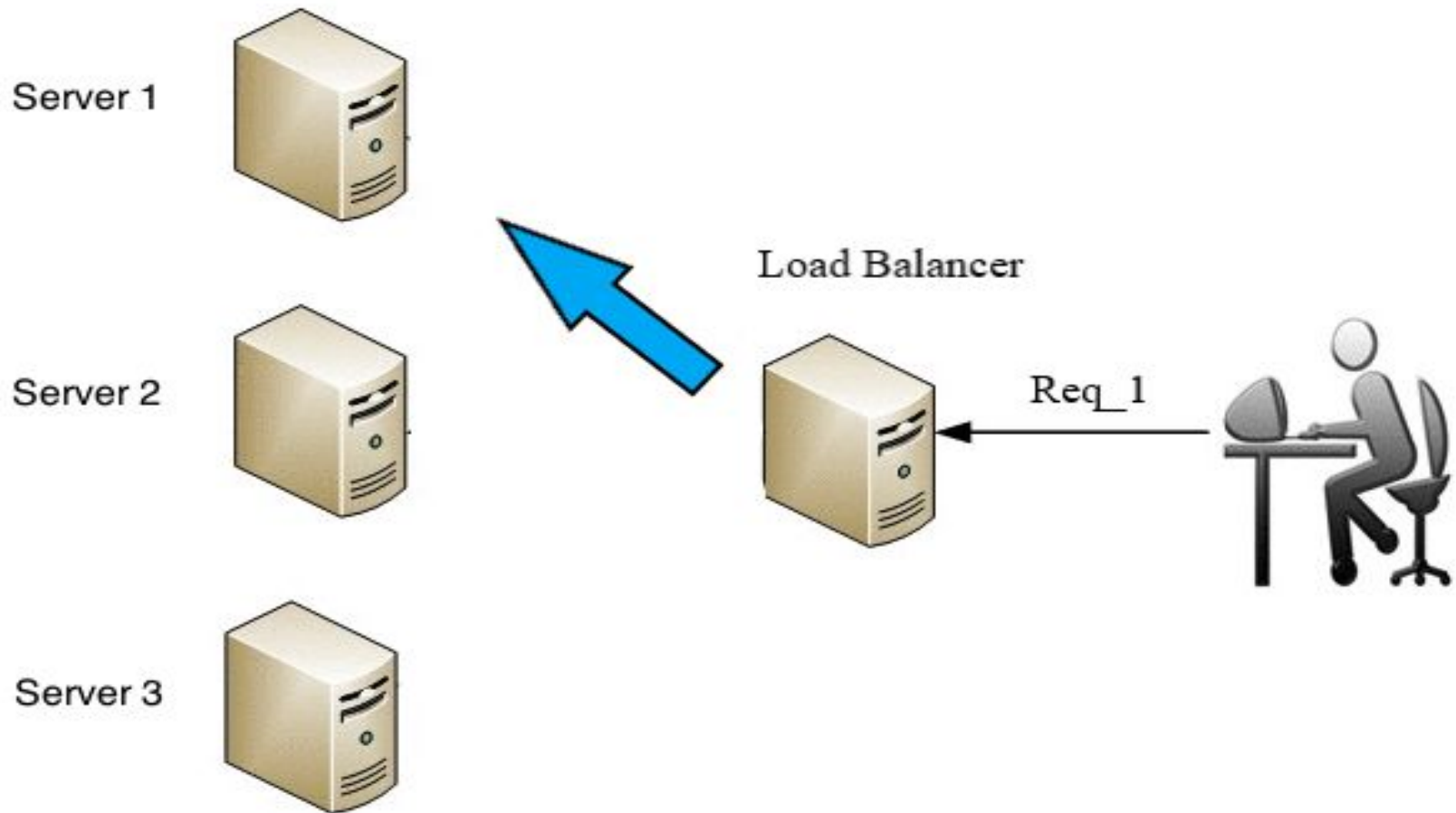


# Least Connection





# Chained Failover



# Implementation Framework

- ❖ Created a framework with Python “http” and “socket” packages to establish network pipes and handling requests.
- ❖ Initially tested the infrastructure on localhost, where one instance was the balancer at port X and spawned servers with distinct ports.
- ❖ Randomised execution scripts for more real-time testing.
- ❖ Final testing on GCP VMs over external network.



# GCP

Google Cloud Platform

My First Project

Search resources and products

Compute Engine

VM instances

Instance groups

Instance templates

Sole-tenant nodes

Machine images

Disks

Snapshots

Images

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

START

STOP

RESET

DELETE

Instance "netsys-instance-lb" is underutilised. You can save an estimated \$35 per month by switching to the machine type: custom (1 vCPU, 3 GB memory). [Learn more](#)

Filter VM instances

Columns

<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/> netsys-instance-1	us-west1-b			10.138.0.2 (nic0)	35.247.73.142 ↗	SSH ▾ ⋮
<input type="checkbox"/> netsys-instance-2	us-west3-a			10.180.0.2 (nic0)	34.106.248.143 ↗	SSH ▾ ⋮
<input type="checkbox"/> netsys-instance-3	us-west4-a			10.182.0.2 (nic0)	34.125.74.70 ↗	SSH ▾ ⋮
<input type="checkbox"/> netsys-instance-lb	us-west1-b	Save \$35/mo		10.138.0.3 (nic0)	35.197.5.17 ↗	SSH ▾ ⋮



# Technical Implementation

- ❖ Multi-threaded using Python threading library
  - Main thread just listens for connections
  - Hands off to child threads for the rest
- ❖ Child threads handle:
  - Decision making (Round robin, chained failover, least connection)
  - Packaging and delivering client request + server response
- ❖ Balancer:
  - Tracks total number of requests handled
  - Tracks open requests for each server
  - Can be configured to listen at any port
  - Provides a verbose mode for debugging



# DEMO





# Challenges

- ❖ Didn't have enough time to build out the UI for dynamic server management
- ❖ During testing, we had someone pinging the server which caused inconsistent results
- ❖ Inconsistent broken pipe error on GCP



# Questions?



THANK  
YOU

