Cloud Computing (INFS3208)

Lecture 2: VPC Network & Load Balancing

Lecturer: Dr Sen Wang

School of Electrical Engineering and Computer Science

Faculty of Engineering, Architecture and Information Technology

The University of Queensland

Recap

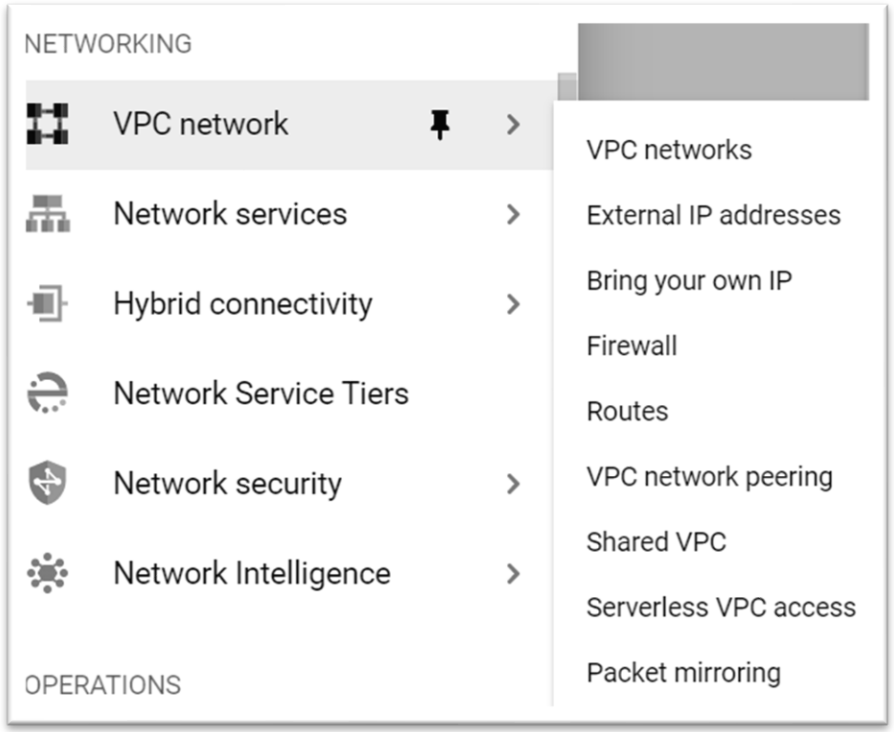
Introduction to Cloud Computing

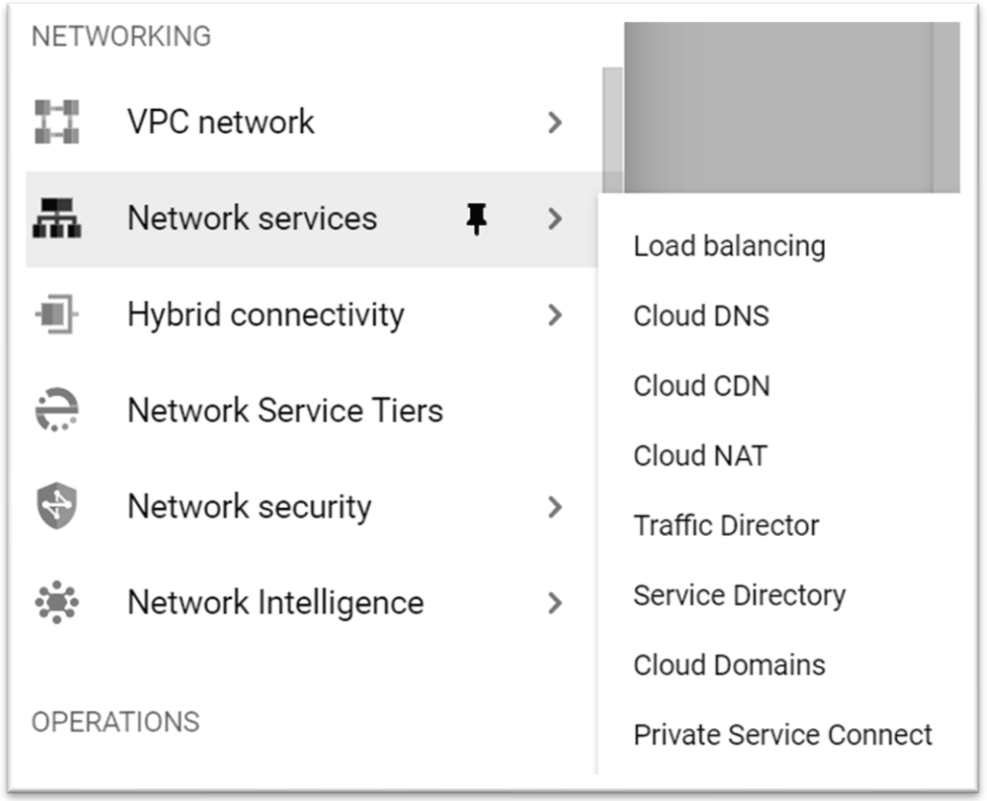
- History & Definitions
- Business Drivers
- Technology Innovations
- Cloud Characteristics
- Cloud Delivery Models & Cloud Deploy Models
- Cloud-enabling technologies: Broadband Networks and Internet Architecture, Virtualisation Technology (VT), Data Centre Technology, Web Technology, and Multitenant Technology
- Goals and Benefits
- Risks and Challenges
- Cloud-based Applications in the World

Outline

- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

Cloud Networking





https://en.wikipedia.org/wiki/Domain_Name_System

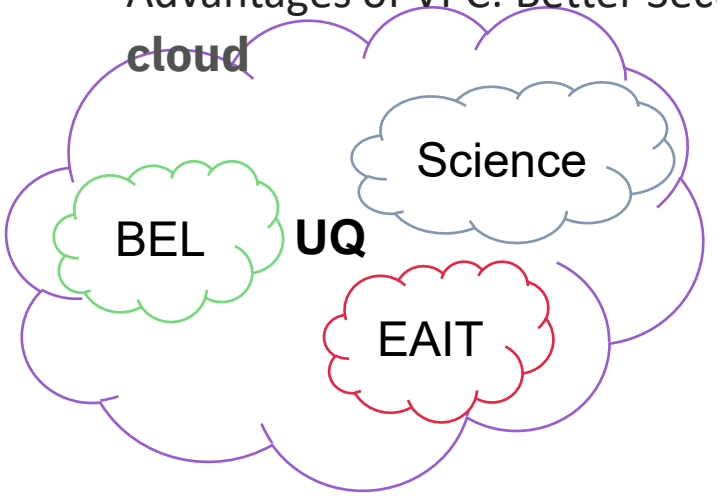
Virtual Private Cloud (VPC)

Cloud Deployment models:

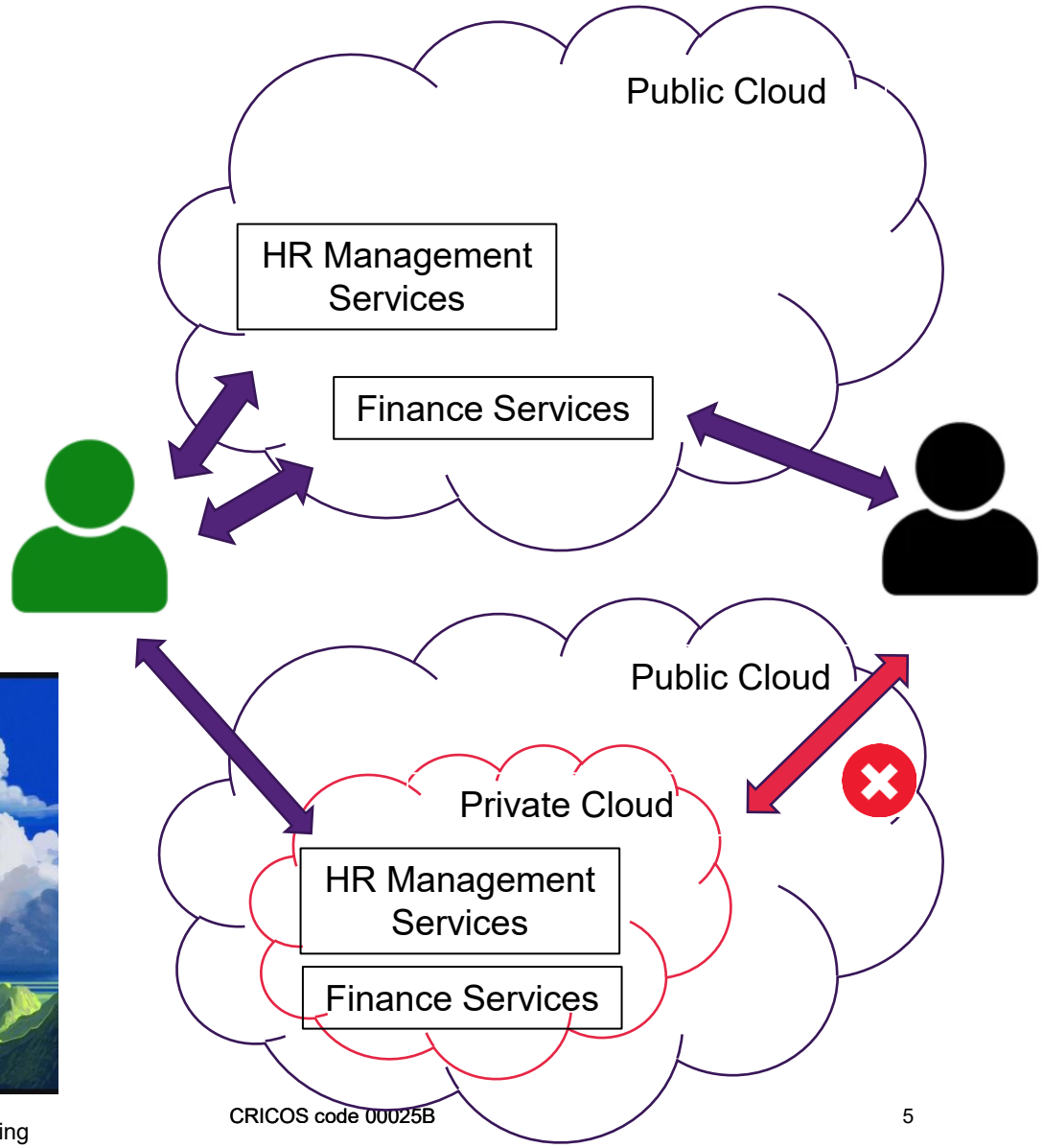
- Public Cloud (e.g. AWS, GCP) vs. Private Cloud (UQCloud)
- Human Resource department vs. Finance department in one company

A Virtual Private Cloud (VPC) is a logically isolated network within a public cloud that allows you to define your own IP address ranges, create subnets, and manage routing and firewall rules.

Advantages of VPC: Better Security + **All benefits of public cloud**



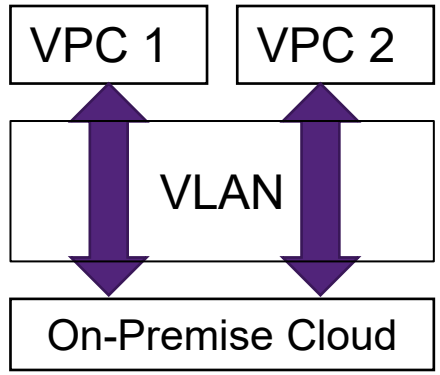
https://ichthyoid.writeas.com/castle-in-the-sky-a-study-in-world-building



Virtual Private Cloud (VPC) Isolation Technologies

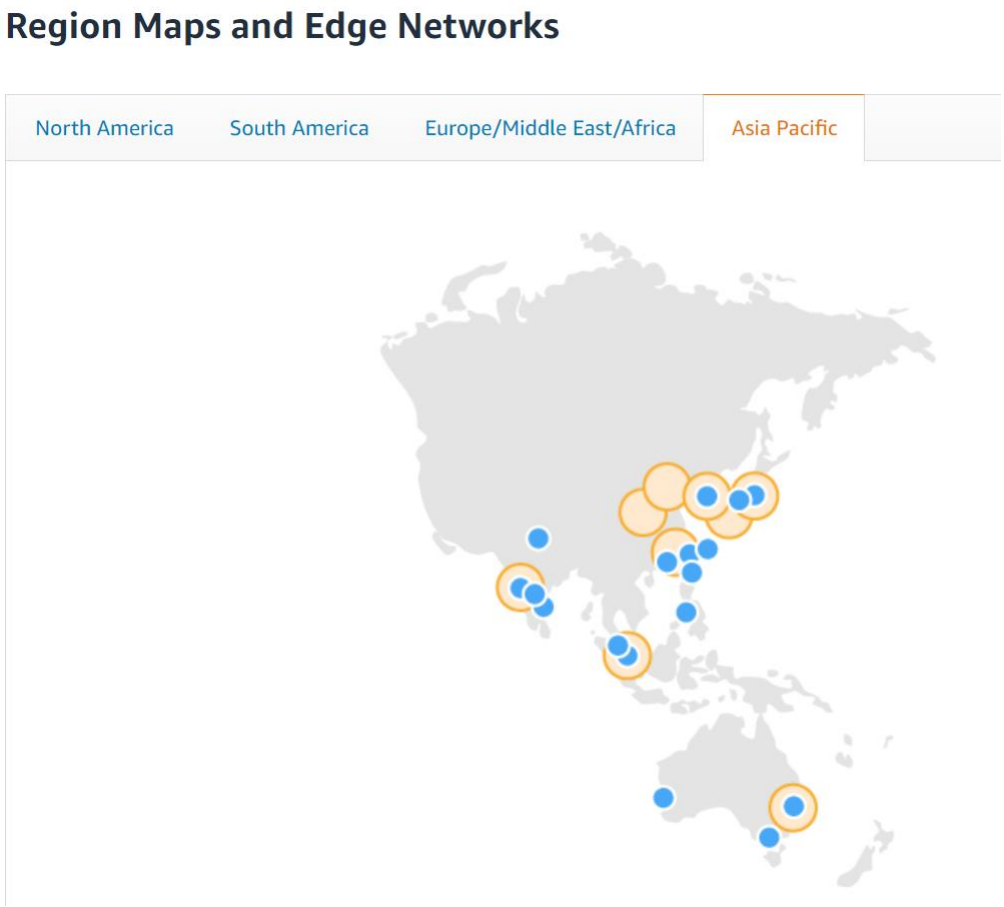
The key technologies for isolating a VPC from the rest of the public cloud are:

- **Subnets:**
  - A subnet (a range of IP addresses) is reserved (not available to everyone) within the network - for private use.
  - In a VPC, cloud providers will allocate private IPs (not accessible via the public Internet, e.g. 10.xxx.xxx.xxx).
  - Example: my-vpc (10.0.0.0/16, 65,534 addresses) has two subnets: hr-subnet (10.0.1.0/24) and fin-subnet (10.0.2.0/24)
- **VPN:**
  - A virtual private network (VPN) uses encryption to create a private network.
  - VPN traffic passes through publicly shared Internet infrastructure – routers, switches, etc.
- **VLAN (Virtual Local Area Network):**
  - A VLAN is a virtual LAN (layer 2) and it's used to partition a network in a data centre.
  - Performance solution for hybrid connection setups (e.g. VPCs + On-Premise Private Cloud)
- **NAT (Network Address Translation):**
  - NAT translates private IP addresses to a public IP address for connections with the public Internet.
  - With NAT, a public-facing website or application could run in a VPC.



Regions and Zones

- Cloud Providers organise IT resources by regions and zones
- Availability Regions
  - the geographic locations of the data centres
    - E.g, China, North America, Southeast Asia, East Asia, Europe, Middle East, etc.
  - collection of isolated zones
  - specific location to run resources
- Availability Zones
  - one or more discrete data centres with redundancy in a Region (independent zones lead to resilience)
  - Multiple zones are interconnected with encryption
- Prices of IT resources in different zones and regions could be very different! (e.g., Asia-Pacific regions often cost more than US regions).



Google Cloud: 40 regions and 121 zones
AWS: 32 regions and over 100 availability zones (as of mid-2025).

https://aws.amazon.com/about-aws/global-infrastructure/regions_az/?nc1=h_ls

A Multi-Region Application

An Application that has three main functions:

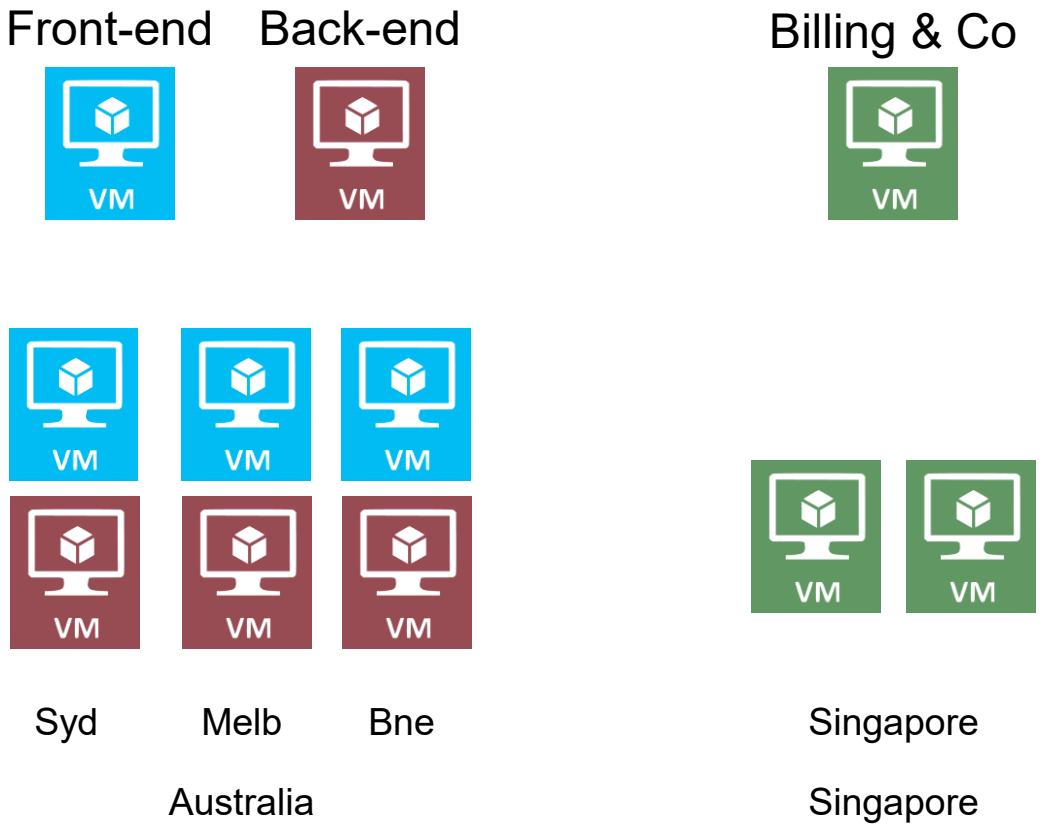
- Front-end
- Back-end
- Other Functions (Billing & Co)

The customers are from Three Major Cities:

- Sydney
- Melbourne
- Brisbane

The HQ is in Singapore

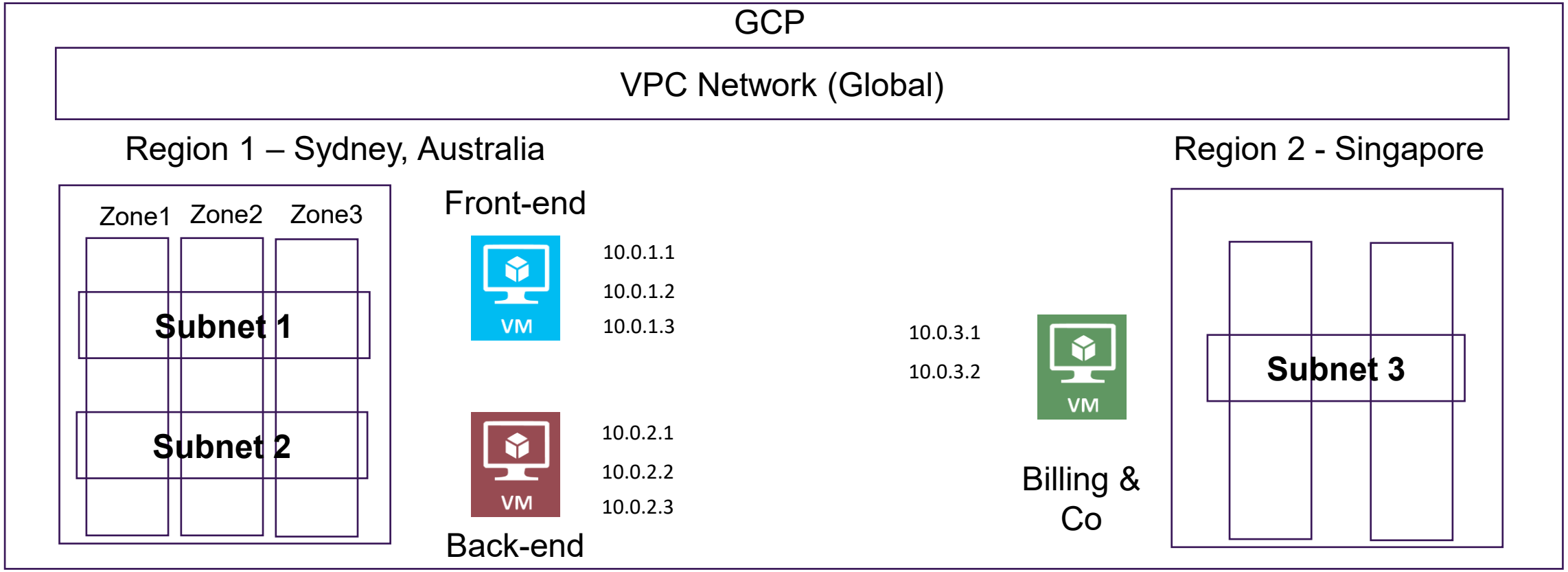
- Running front-end servers in multiple regions improves latency and availability
- Back-end databases may utilise cross-region replication.



VPC and Subnets in GCP and AWS

Subnets and VPCs in GCP and AWS are organised differently:

- VPC in GCP is **global** (automatic routing for traffic), and Subnets are **regional** in **GCP**

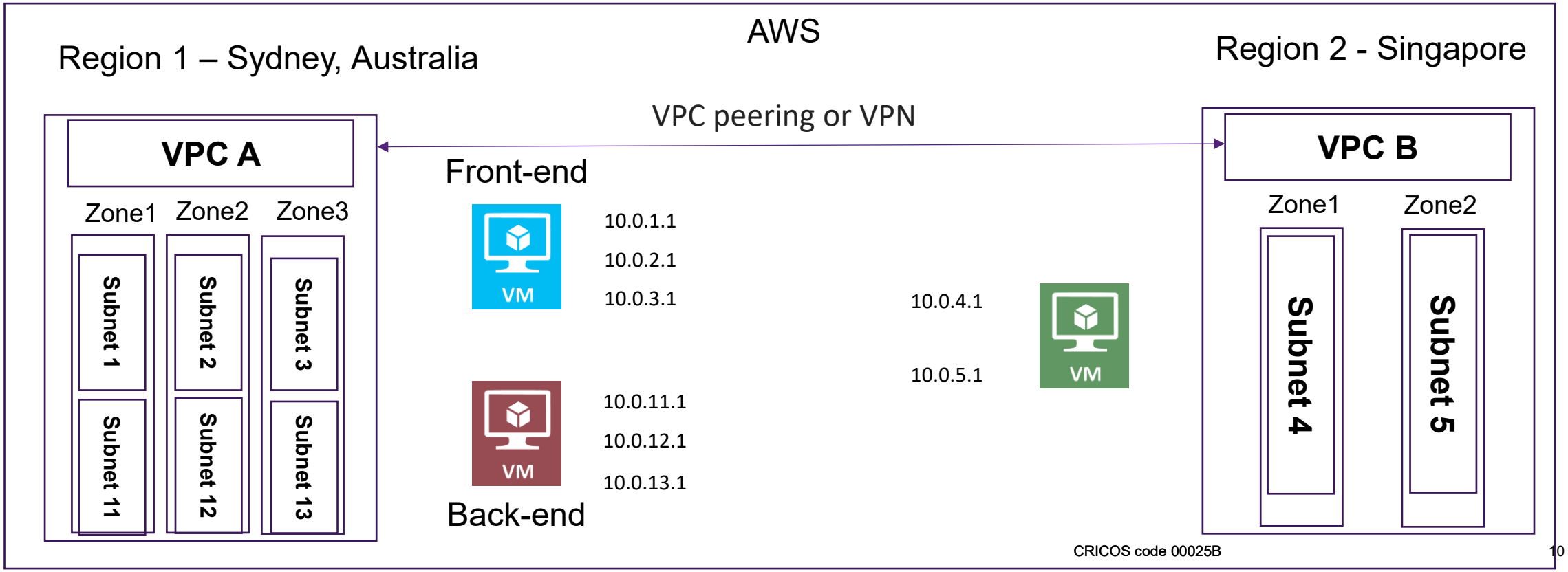


Subnets and VPC in GCP and AWS

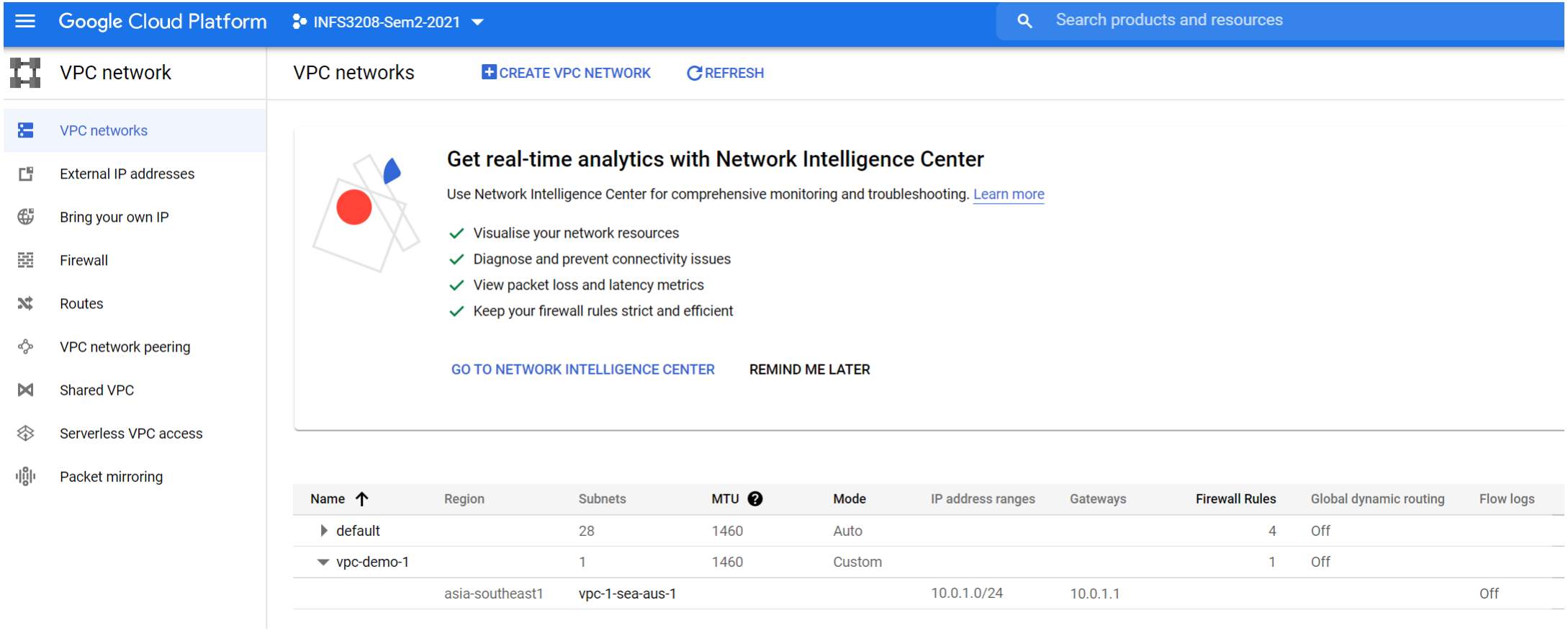
Question:
Can Subnets in AWS overlap each other?

Subnets and VPC in GCP and AWS are differently organized:

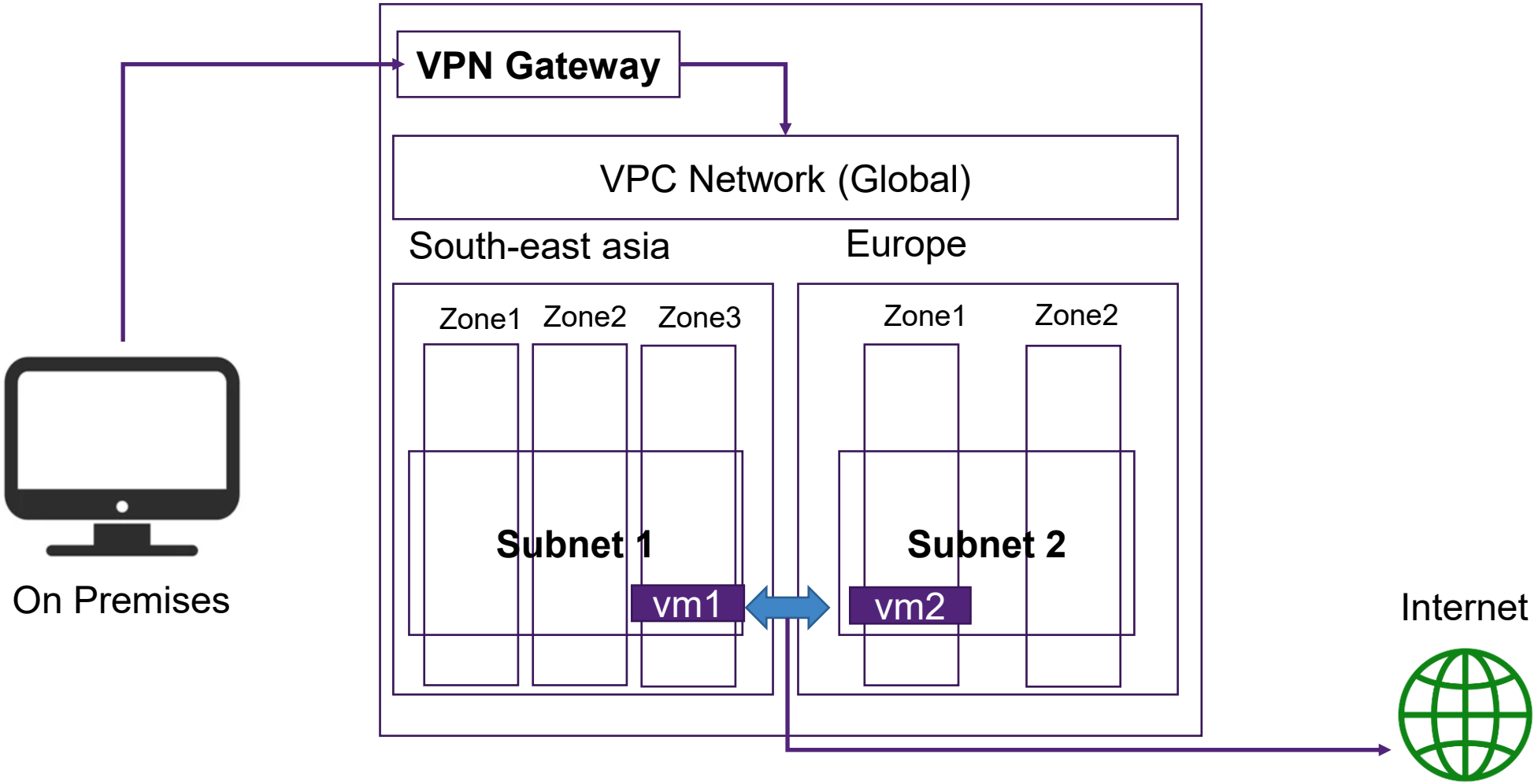
- VPC in AWS is regional (needs VPC peering setup) - each AWS VPC is tied to a single region
- Subnet is confined in zones in AWS (needs routing setup)



Demo – Create VPC Network in GCP



Virtual Private Cloud (VPC)



Routes and Firewall Rules

- Routes define paths for packets leaving instances.
- Routes in Google Cloud are divided into two categories:
  - system-generated and custom.
- Firewall rules aim to protect your VPCs.
- Firewall rules apply to both outgoing (egress) and incoming (ingress) traffic in the network.
- Firewall rules control traffic even if it is entirely within the network.
- In GCP, default VPC network has implied firewall rules;
  - two implied IPv4 firewall rules,
  - two implied IPv6 firewall rules.
  - the implied egress rules allow most egress traffic, and the implied ingress rules deny all ingress traffic.
  - you cannot delete the implied rules, but you can override them with your own rules.
- To monitor which firewall rule allowed or denied a particular connection, see Firewall Rules Logging.

Other Networking Products

- Load Balancing
- Cloud DNS
- Cloud CDN
- Cloud NAT
- Traffic Director
- Service Directory
- Cloud Domains
- Private Service Connect
- And more…

Importance of Networking in Cloud Computing

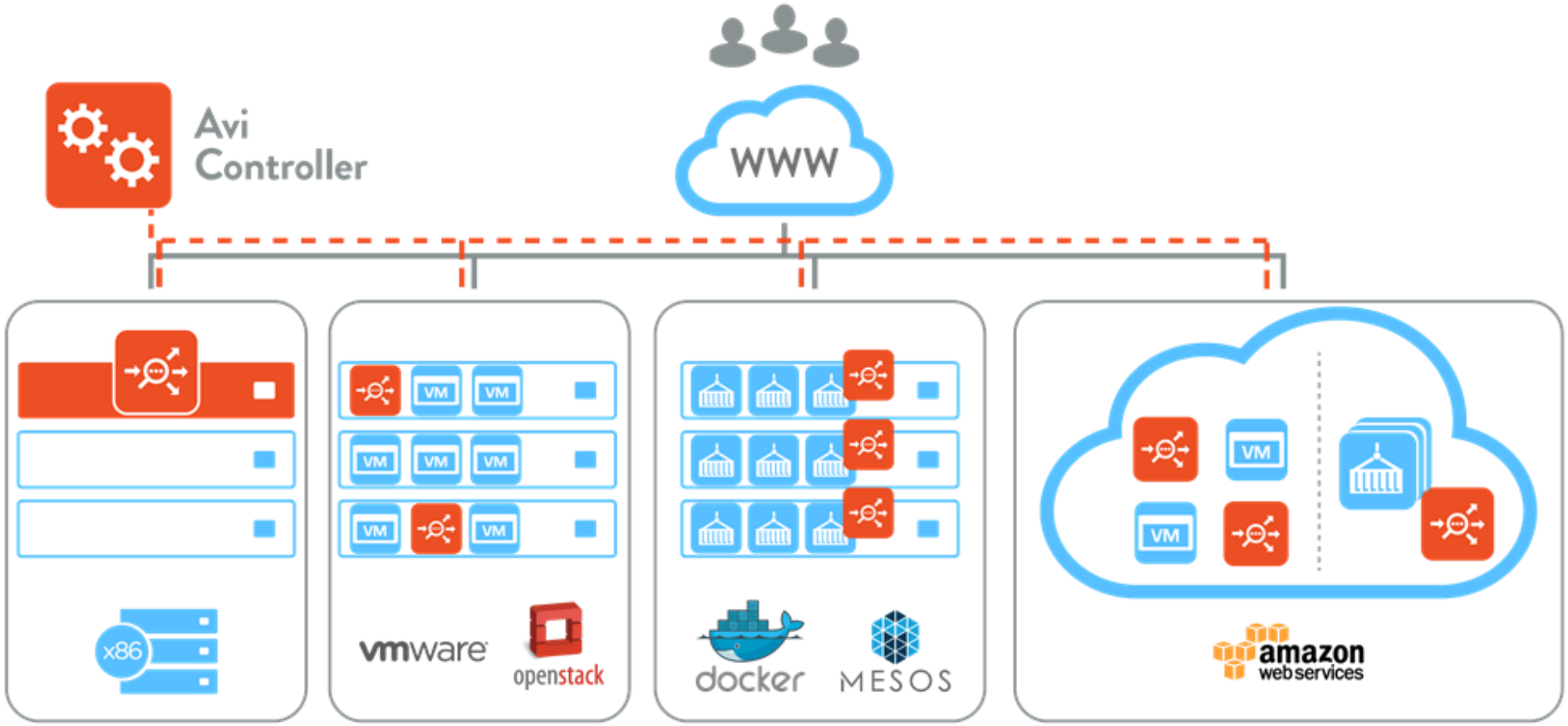
Networking is a crucial aspect of cloud computing for several reasons:

- **Connectivity:** Enables communication between different cloud services, applications, and on-premises infrastructure.
- **Scalability:** Supports dynamic scaling of resources to meet varying demands, ensuring performance and reliability.
- **Security:** Ensures data protection and compliance through network segmentation, encryption, and access controls.
- **Performance:** Optimizes data transfer speeds and reduces latency through efficient routing and load balancing.
- **Cost Management:** Allows for cost-efficient use of network resources by optimizing data flow and reducing unnecessary traffic.

Outline

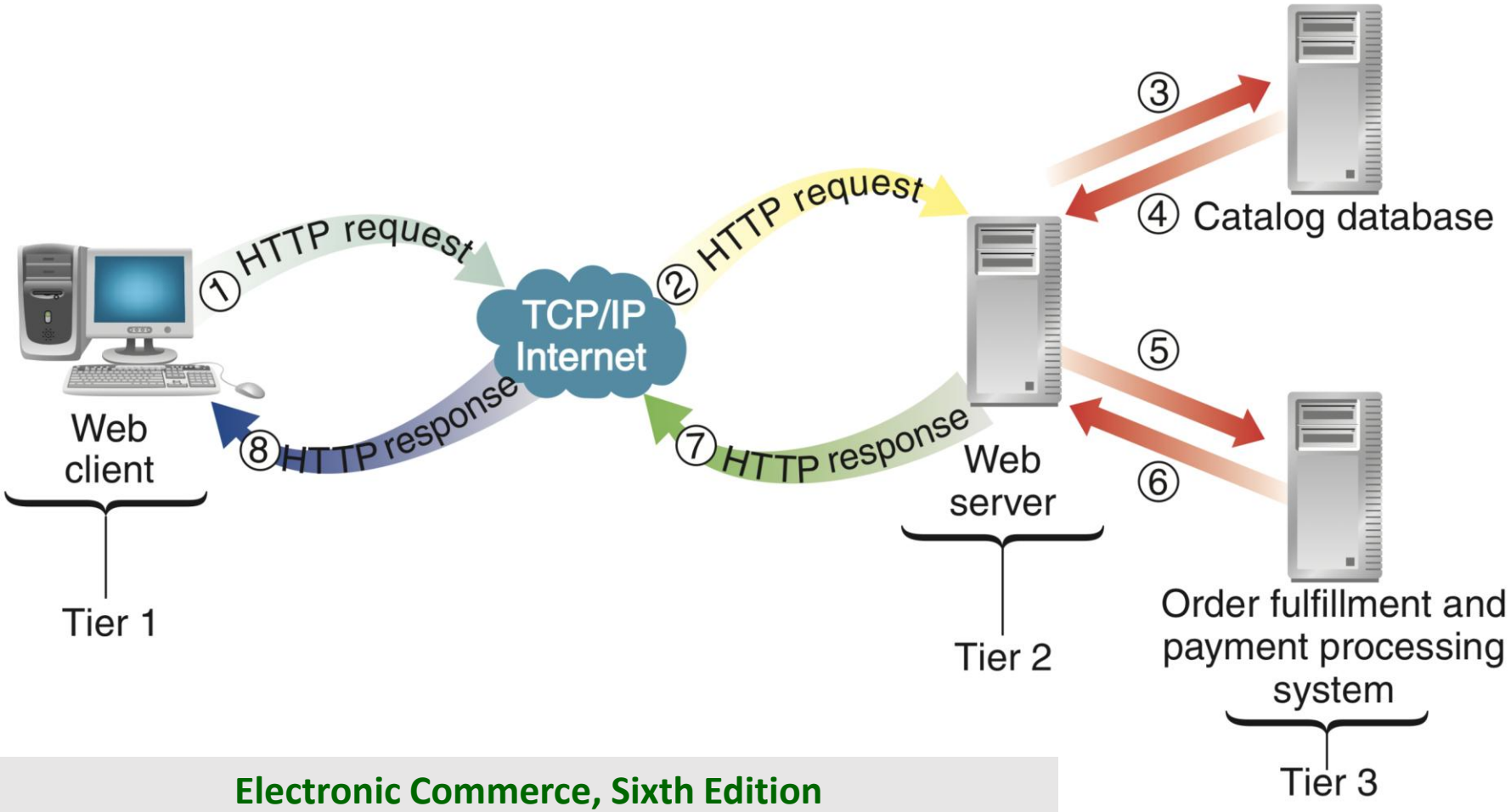
- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

Load Balancing



Picture: https://avinetworks.com/docs/17.2/aws-reference-architecture/

Example: Three-Tier Client/Server Network



Electronic Commerce, Sixth Edition

Load Balancing

What is load balancing ?

- Load balancing improves the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives.
- Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.
- Using multiple components with load balancing instead of a single component may increase reliability and availability through redundancy.
- Load balancing load balancers can operate at different layers of the OSI stack (Layer 4 vs. Layer 7) and can be implemented via **hardware** appliances or **software**.

Why should it be load balanced?

- Improve resource utilization (e.g. 1 full and 2 idle VS 3 busy)
- Improve system performance
- Improve energy efficiency

Unbalanced

Problems of Unbalanced Systems

Server computing capacity problem – The thin clients waged too many applications.

Single-point Data Storage Problem - When a single data resource is (unexpected) demanded by an overwhelming number of clients, (i.e., a single data item is to be requested by many users).

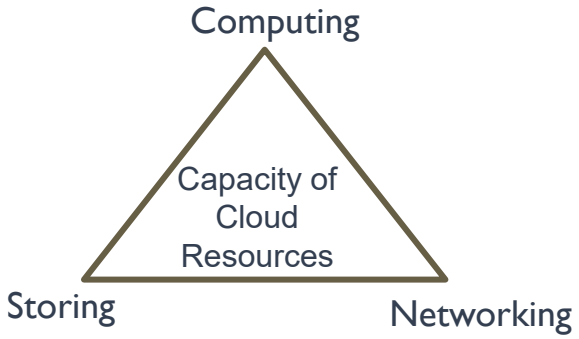
Traffic Problem - When a destination Web Server is to be visited by too many clients.

Storage and Traffic Problem - When a server needs to maintain far too many incoming data streams (upstream) or outgoing data streams (down streams) for file exchanges.

Network Congestion Problem – When the demands of the (web) services is over the server's capacity.

Dynamic Change of the Clients Demands – The clients' demands of services are unpredictable and may change dramatically.

Load balancing is the process of finding overloaded nodes and then transferring the extra load to other nodes.



Outline

- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

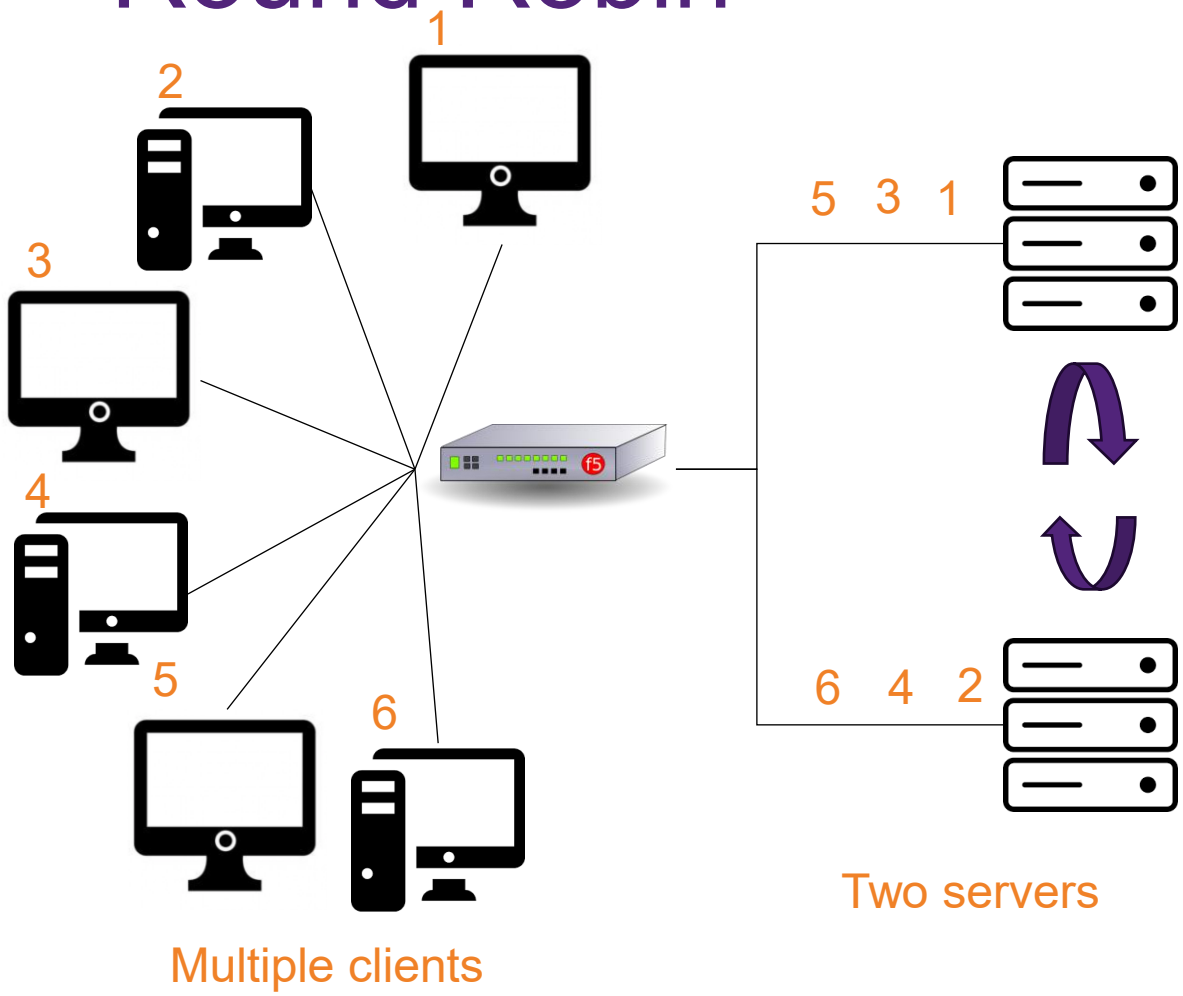
Load Balancing Algorithm – Round Robin

Round-robin load balancing is one of the simplest methods for distributing client requests across a group of servers.

Going down the list of servers in the group, the round-robin load balancer forwards a client request to each server in turn.

When it reaches the end of the list, the load balancer loops back and goes down the list again (sends the next request to the first listed server, the one after that to the second server, and so on).

works best when servers have equal capacity



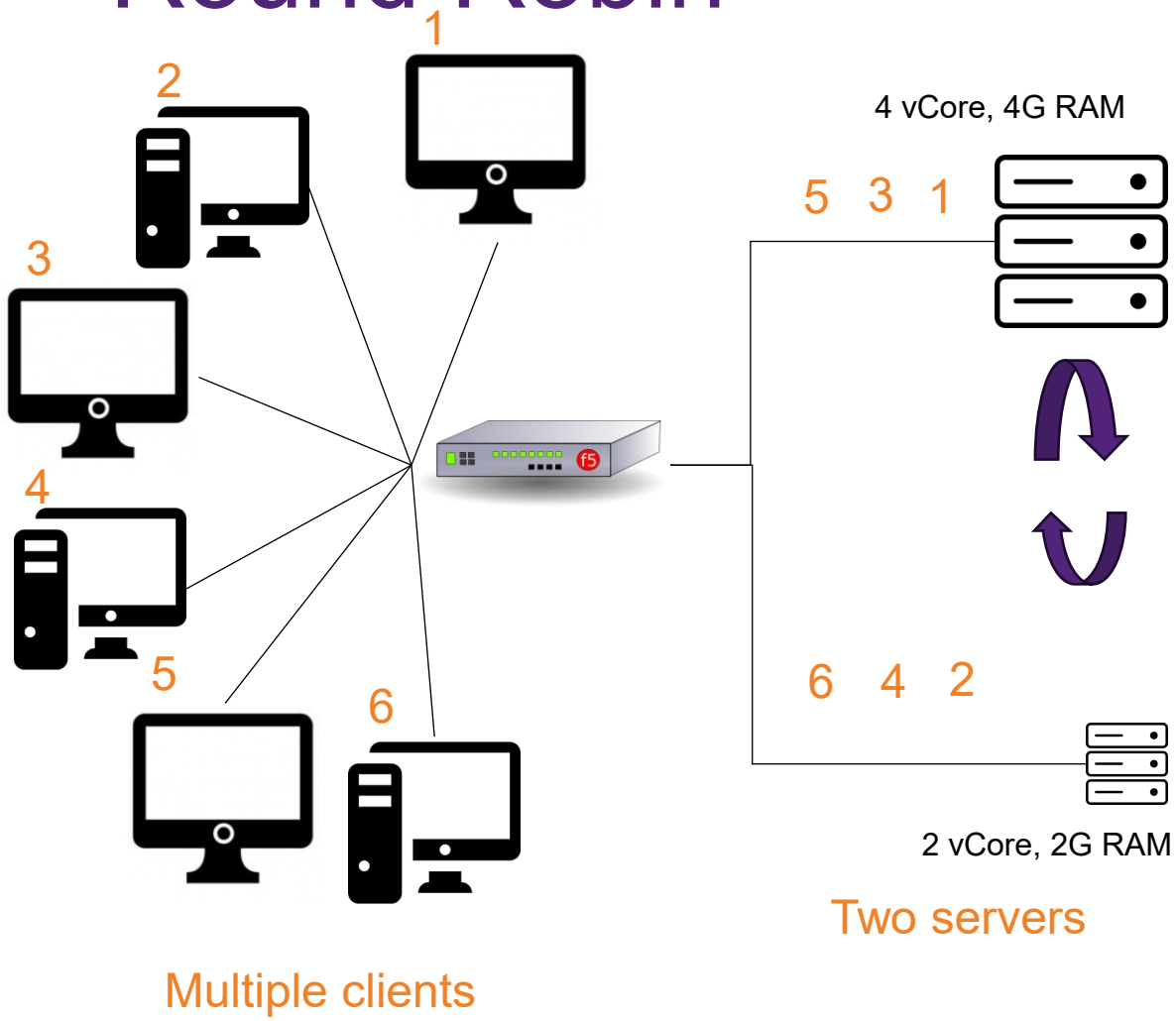
Singh, Navpreet, and Kanwalvir Singh Dhindsa. "Load Balancing in Cloud Computing Environment: A Comparative Study of Service Models and Scheduling Algorithms." *International Journal of Advanced Networking and Applications*8, no. 6 (2017): 3246.

Load Balancing Algorithm – Round Robin

What if the cluster is heterogeneous?

- Different computing capability
- E.g. CPUs (4 cores vs 2 cores)
- E.g. RAMs (4 gig vs 2 gig)

What could be the possible consequence if we still use Round Robin?



Singh, Navpreet, and Kanwalvir Singh Dhindsa. "Load Balancing in Cloud Computing Environment: A Comparative Study of Service Models and Scheduling Algorithms." *International Journal of Advanced Networking and Applications*8, no. 6 (2017): 3246.

Load Balancing Algorithm – Weighted Round Robin

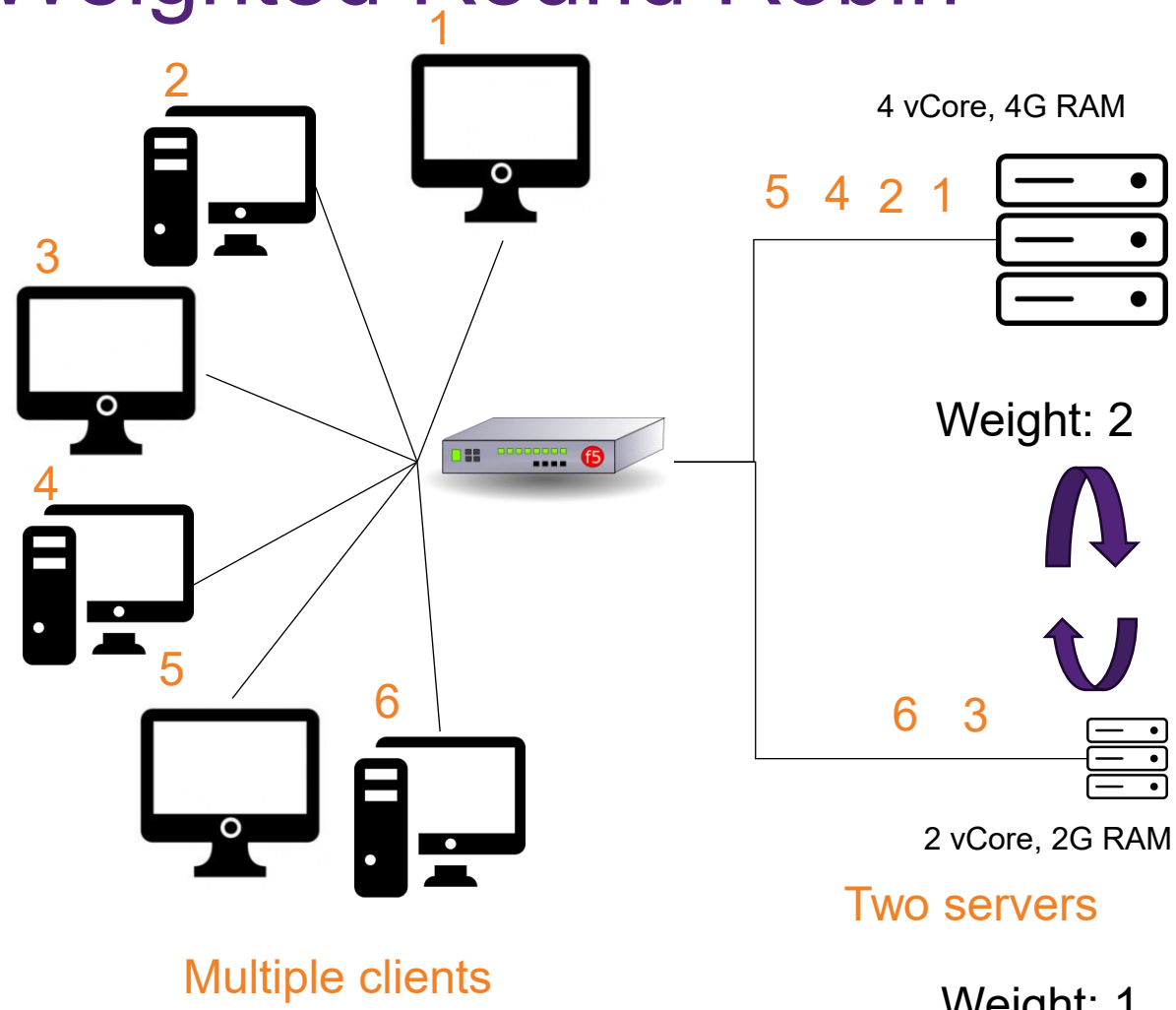
Weighted Round Robin load balancing is similar to the Round Robin (cyclic distribution).

The node with the higher specs will be apportioned a greater number of requests.

Set up the load balancer with assigned "weights" to each node according to hardware specs.

Higher specs, higher weight.

For example, if Server 1's capacity is 2x more than Server 2's, then you can assign Server 1 a weight of 2 and Server 2 a weight of 1.



Singh, Navpreet, and Kanwalvir Singh Dhindsa. "Load Balancing in Cloud Computing Environment: A Comparative Study of Service Models and Scheduling Algorithms." *International Journal of Advanced Networking and Applications*8, no. 6 (2017): 3246.

Load Balancing Algorithm – Least Connections

Identical hardware specs, but different occupied durations by the client. E.g. clients connecting to Server 2 stay connected much longer than those connecting to Server 1.

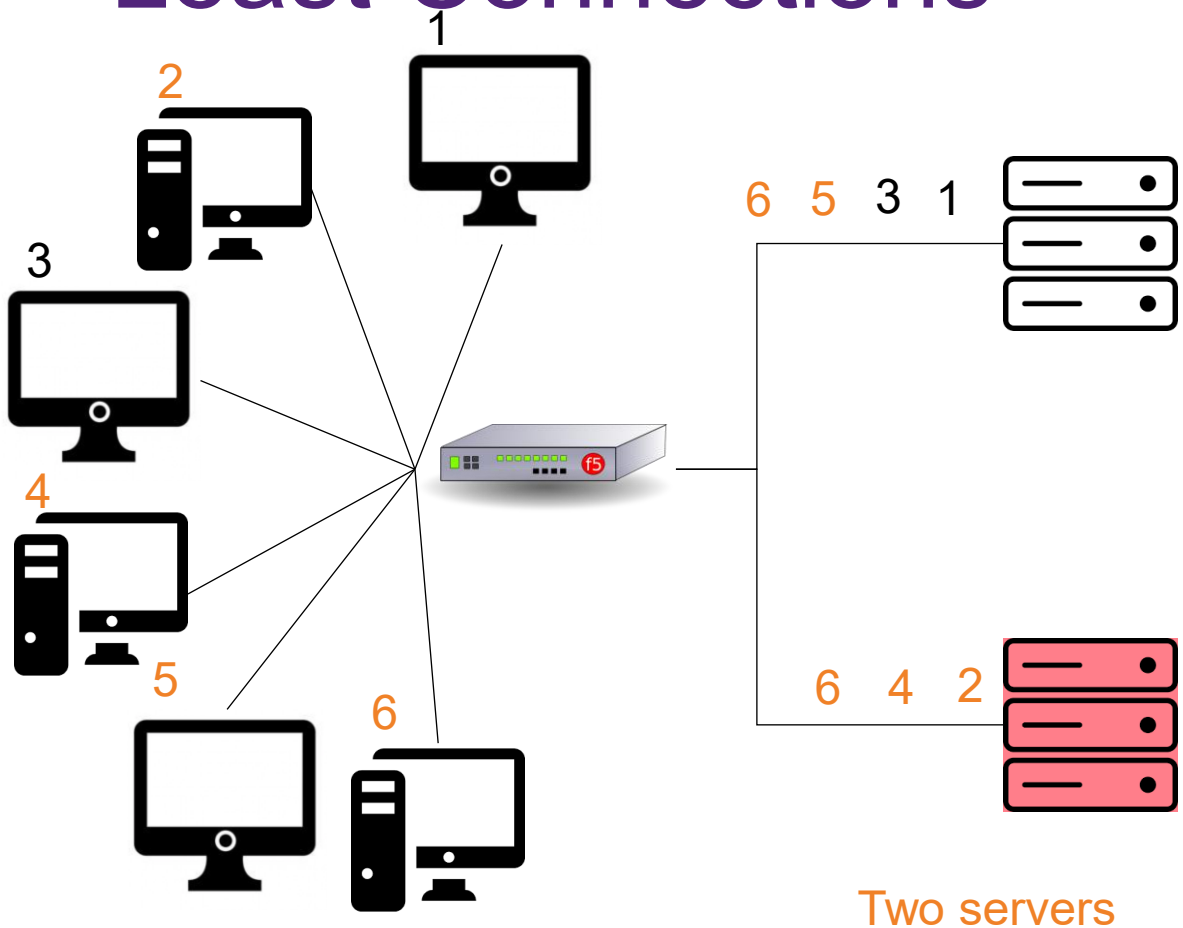
Congestion in Server 2 makes resources run out faster.

Example: clients 1 and 3 have already disconnected, while 2, 4, 5, and 6 are still connected.

The Least Connections algorithm considers the number of current connections each server has when load balancing.

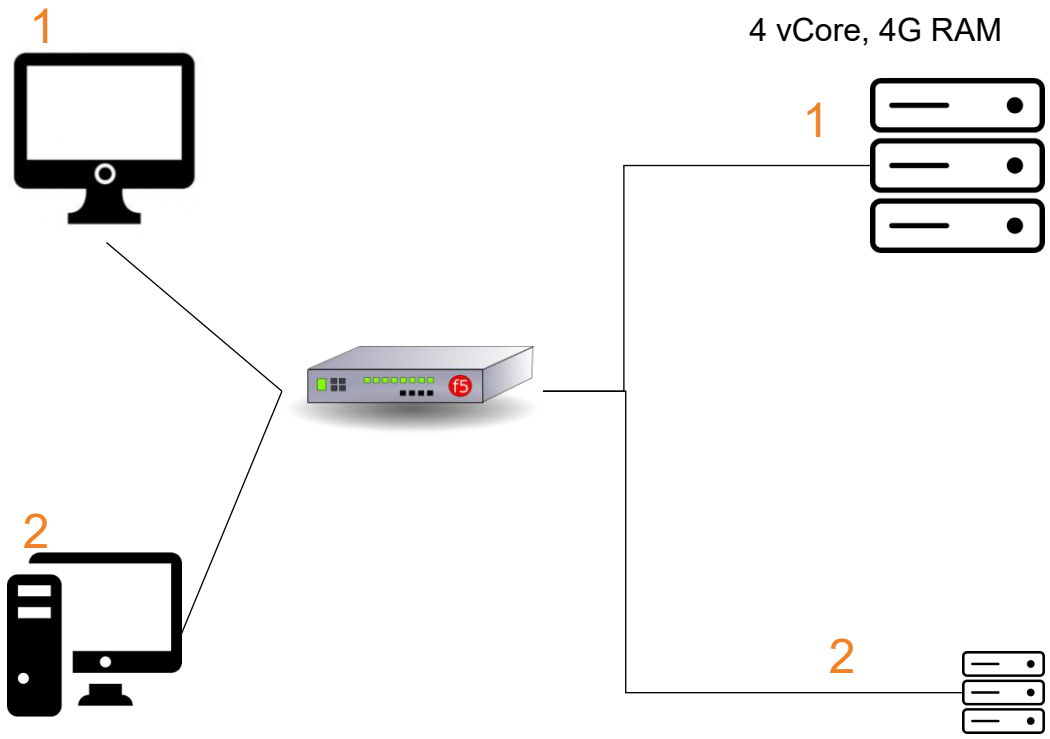
Less connection, higher priority for assignment.

For example, when using the Least Connections algorithm, client 6 will be directed to Server 1 instead of Server 2.



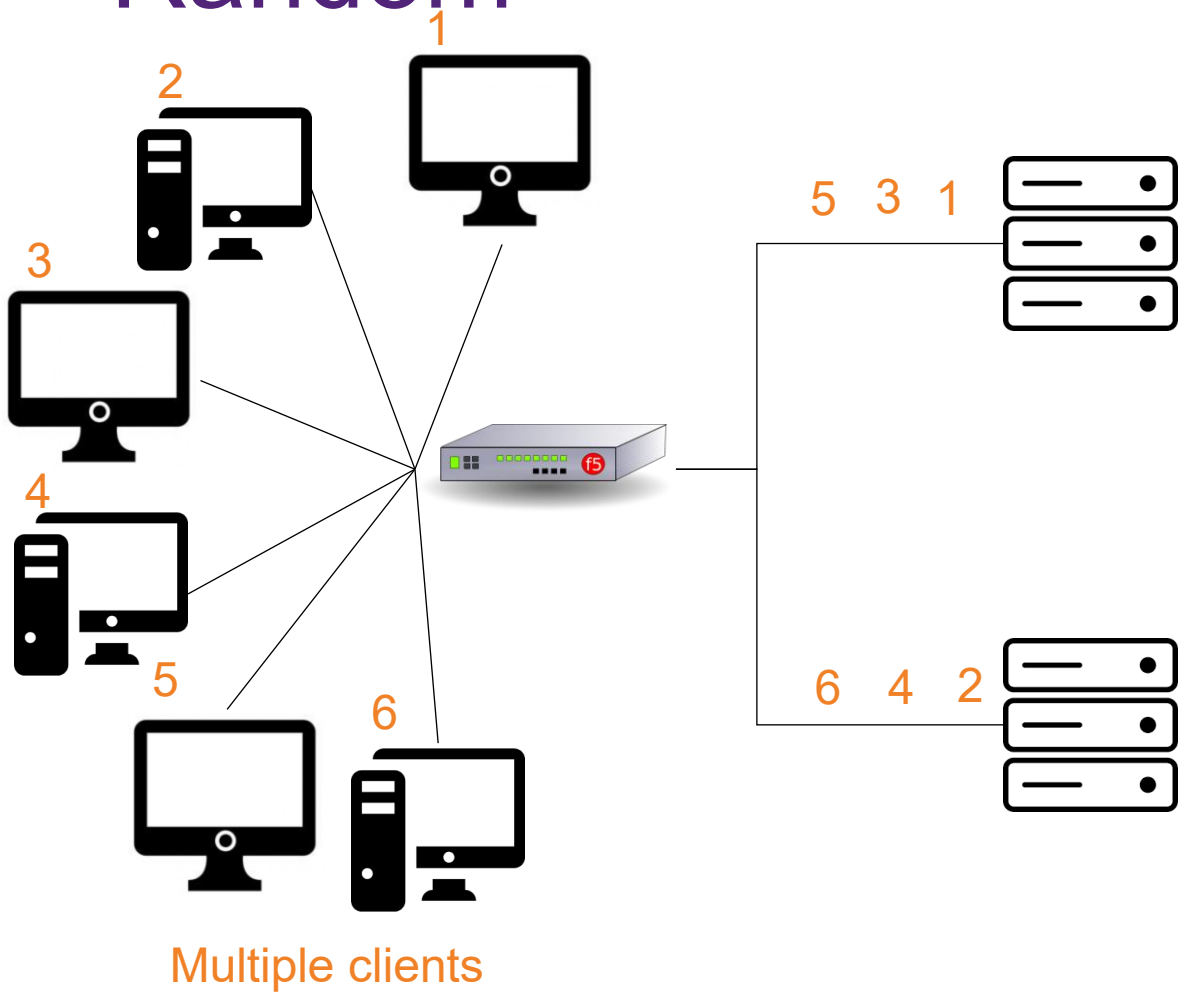
Load Balancing Algorithm – Weighted Least Connections

- The Weighted Least Connections algorithm applies a "weight" component based on the computing capacities of each server.
- Similar with Weighted Round Robin, setup a weight for each server.
- When directing an access request, a load balancer now considers two things:
  - the weights of each server
  - the number of clients currently connected to each server.



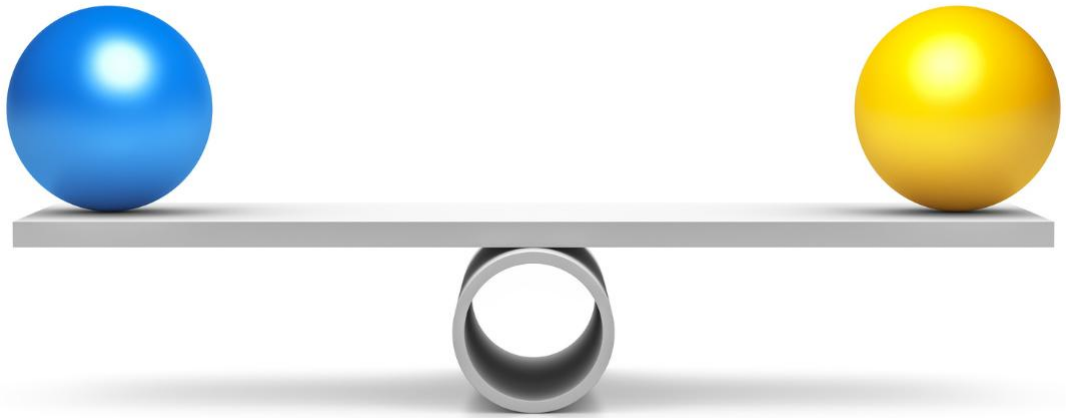
Load Balancing Algorithm – Random

- As its name implies, this algorithm matches clients and servers by random, i.e. using an underlying random number generator.
- In cases wherein the load balancer receives a large number of requests, a Random algorithm will be able to distribute the requests evenly to the nodes.
- Like Round Robin, the Random algorithm is suitable for clusters consisting of nodes with similar configurations (CPU, RAM, etc).
- RR vs Rand
  - Both are simple (Rand is slightly simpler – no tracking)
  - RR is predictable and fair while Rand is unpredictable and potentially unfair



Other Cloud Load Balancing Algorithms

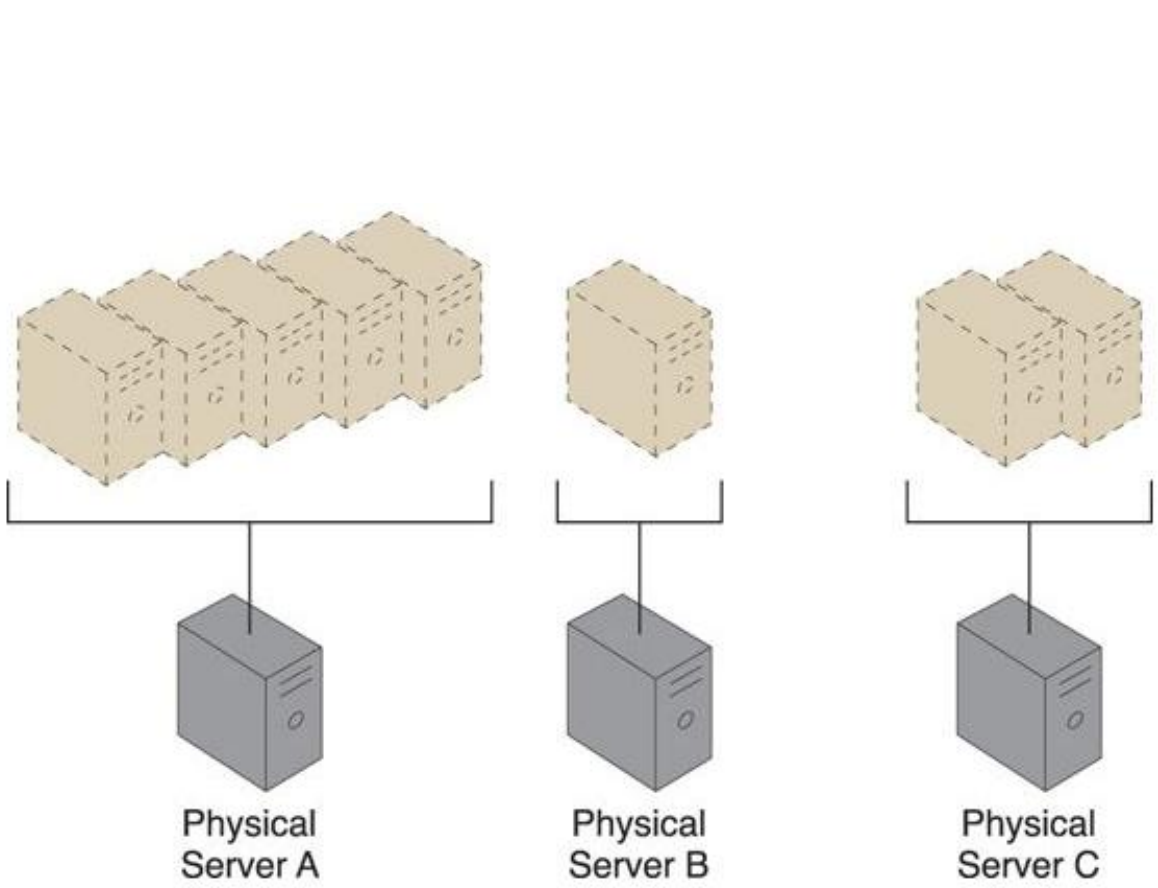
- Agent-based adaptive load balancing
- Chained failover load balancing
- Weighted response time load balancing
- Source IP hashing load balancing
- Layer 4-7 load balancing
- Etc.

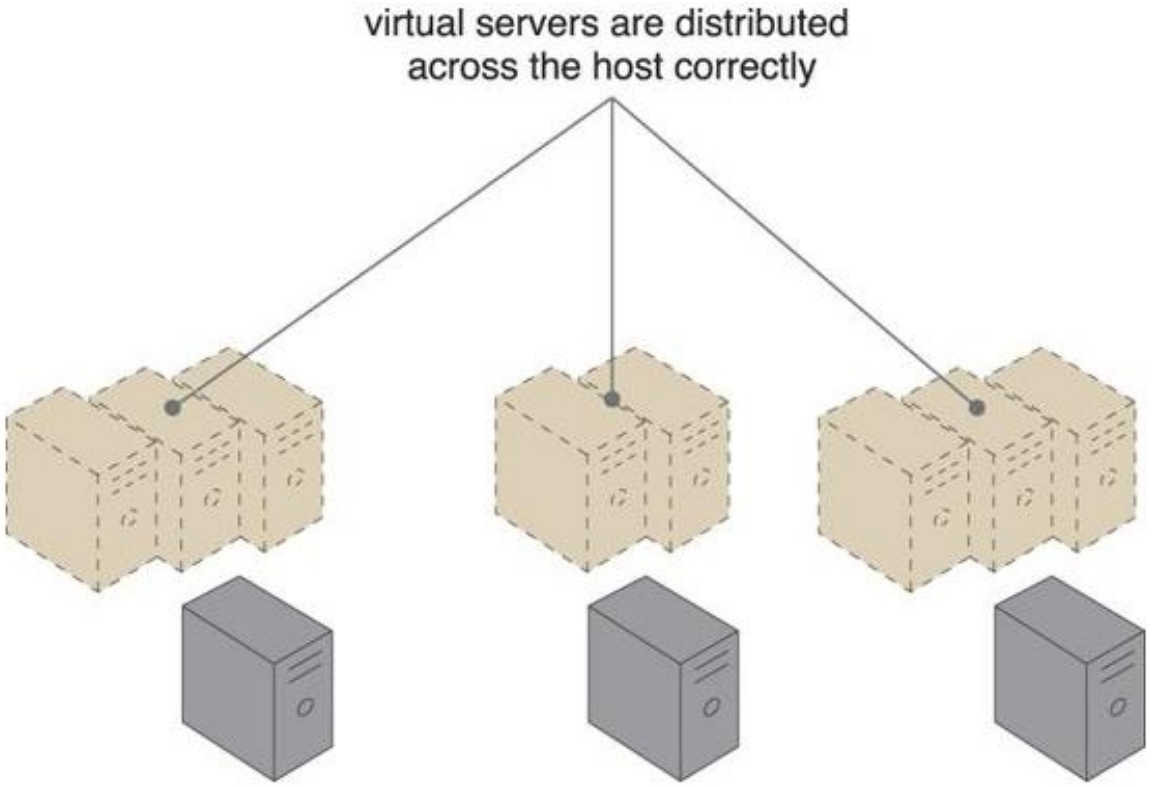


Outline

- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

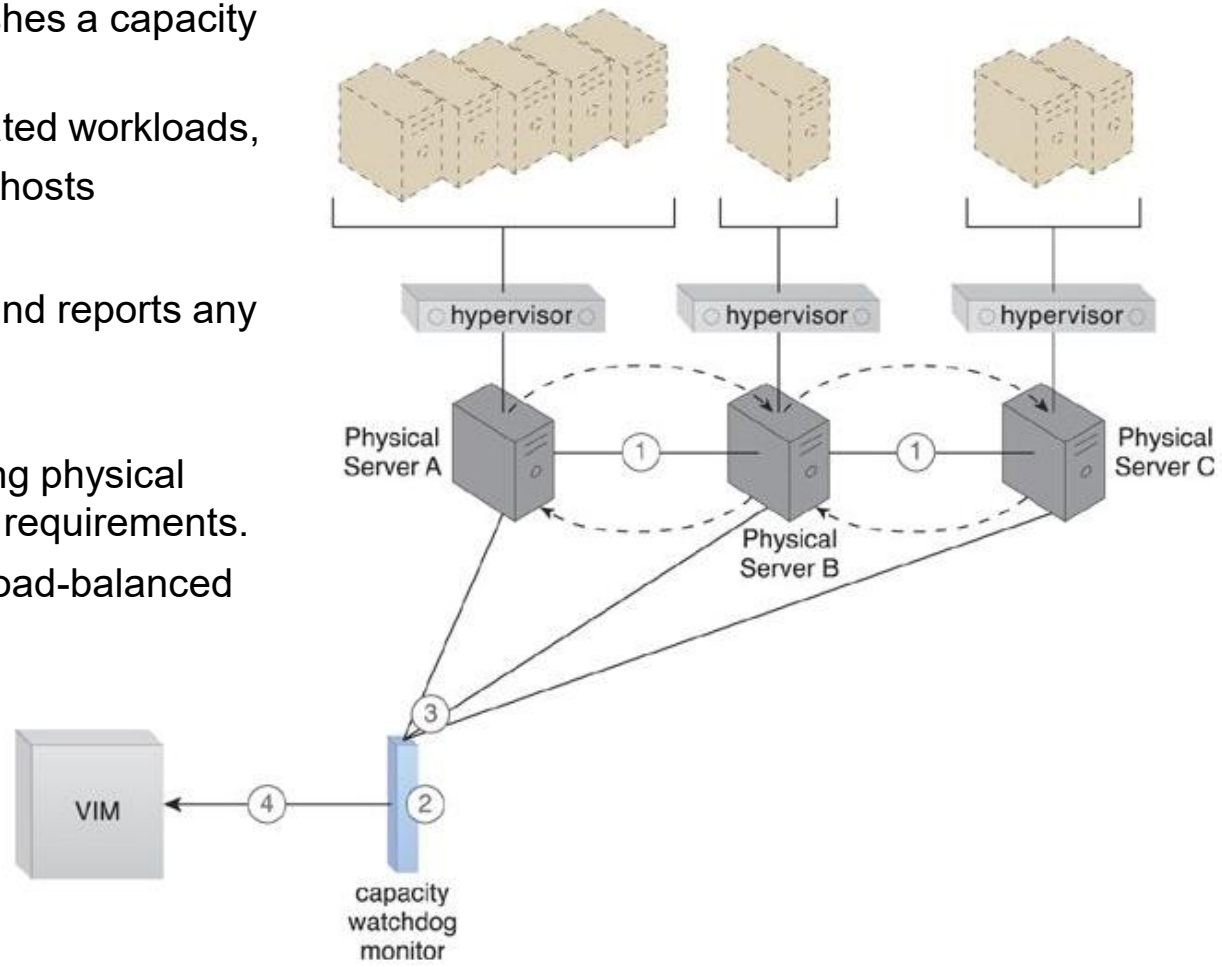
Load Balanced Virtual Server Instances Architecture





Load Balanced Virtual Server Instances Architecture

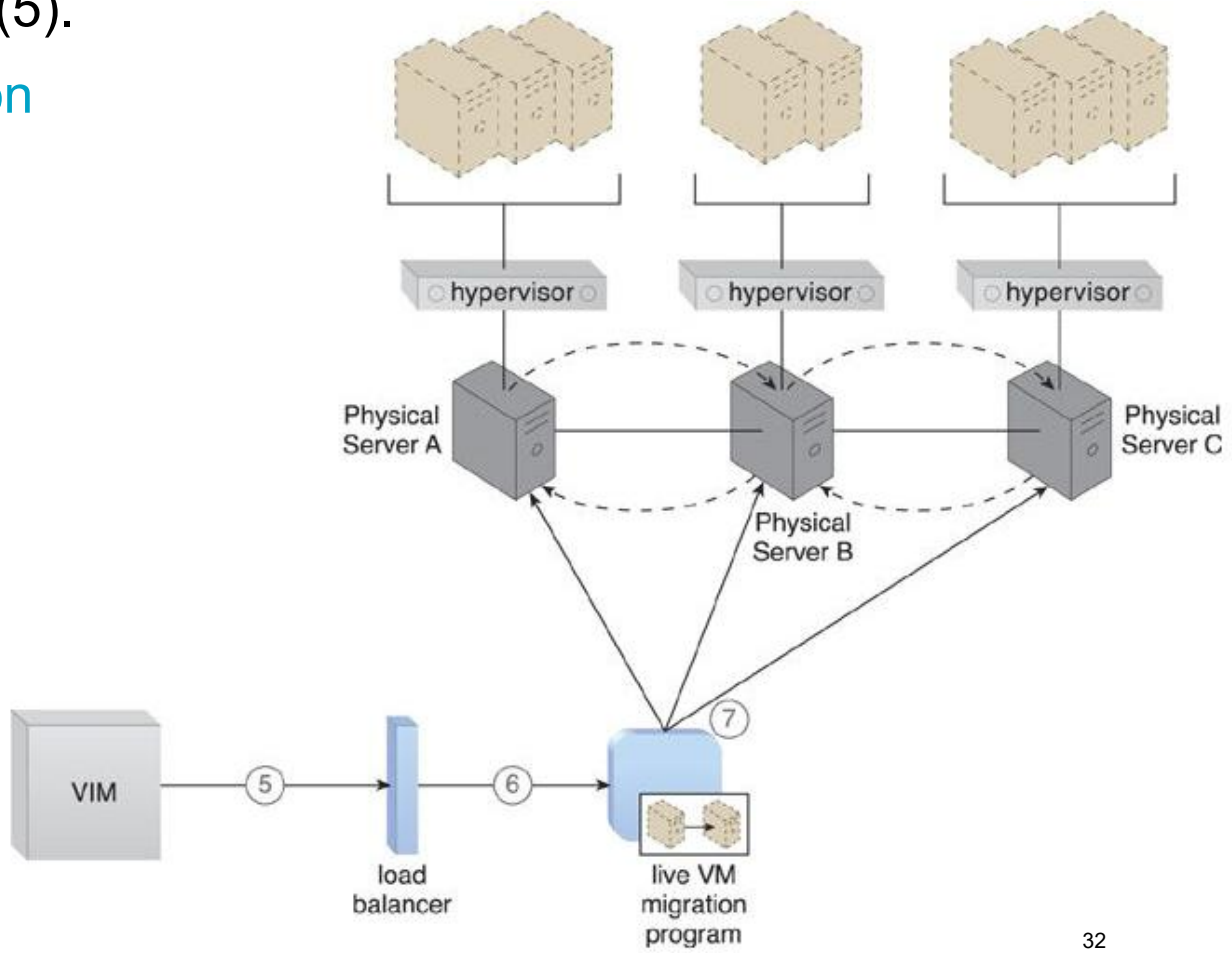
- The *load balanced virtual server instances architecture* establishes a capacity watchdog system
  - dynamically calculates virtual server instances and associated workloads,
  - distributes the processing across available physical server hosts
- The capacity watchdog system has
  - a usage monitor: tracks physical and virtual server usage and reports any significant fluctuations to the capacity planner
  - live VM migration program
  - a capacity planner: is responsible for dynamically calculating physical server computing capacities against virtual server capacity requirements.
- The hypervisor cluster architecture provides the foundation of load-balanced virtual server architecture.



- Policies and thresholds are defined for the capacity watchdog monitor (2), which compares physical server capacities with virtual server processing (3).
- The capacity watchdog monitor reports an over-utilization to the VIM (4).

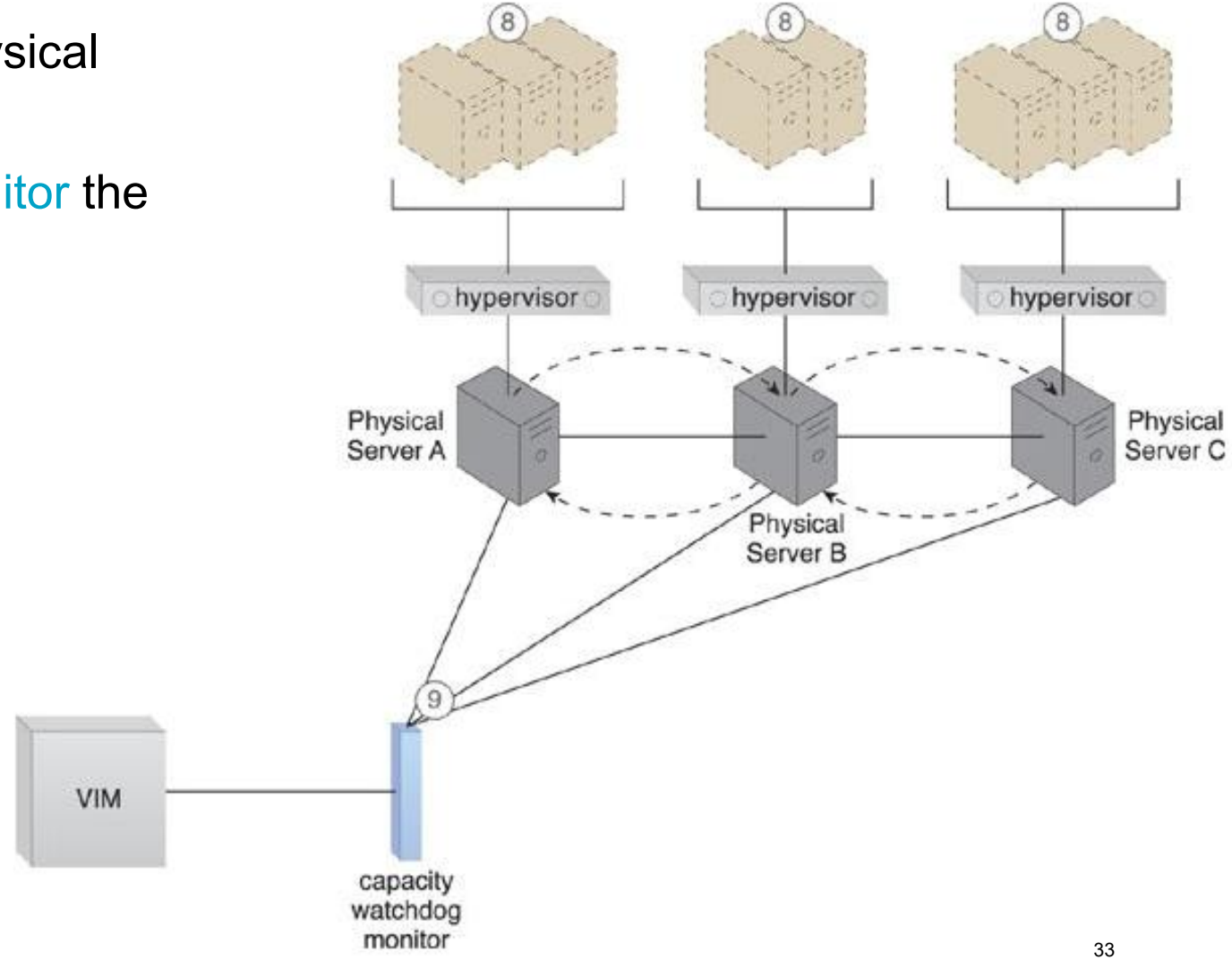
Load Balanced Virtual Server Instances Architecture

- The VIM signals the load balancer to redistribute the workload based on pre-defined thresholds (5).
- The load balancer initiates the live VM migration program to move the virtual servers (6).
- Live VM migration moves the selected virtual servers from one physical host to another (7).



Load Balanced Virtual Server Instances Architecture

- The workload is balanced across the physical servers in the cluster (8).
- The capacity watchdog continues to monitor the workload and resource consumption (9).

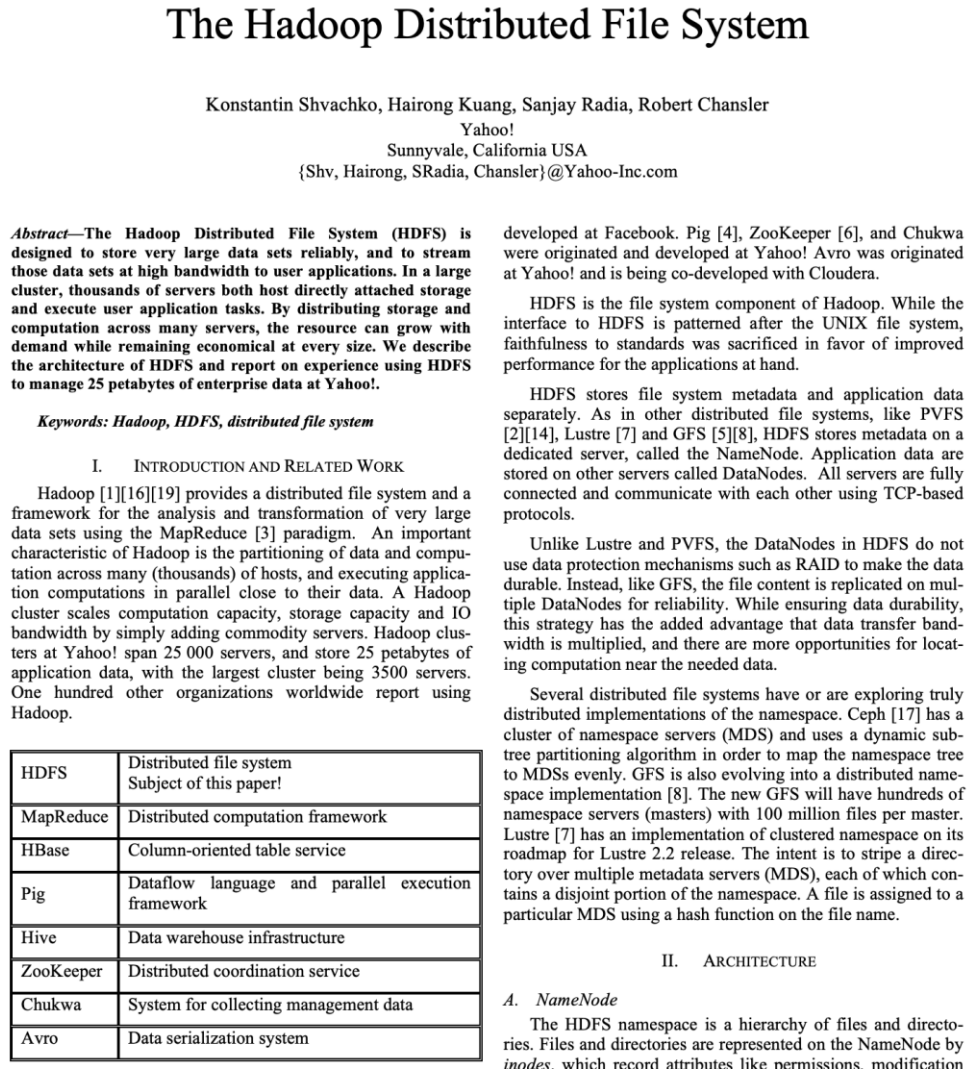


Outline

- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

Hadoop Distributed File System

- Apache Hadoop was proposed in 2010 as a collection of open-source software utilities to deal with big data problem.
- The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model.
  - Hadoop splits files into large blocks and distributes them across nodes in a cluster.
  - It then transfers packaged code into nodes
  - It takes advantage of data locality.
  - faster and more efficiently than a conventional supercomputer architecture



The Hadoop Distributed File System

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler
Yahoo!
Sunnyvale, California USA
{Shv, Hairong, SRadia, Chansler}@Yahoo-Inc.com

***Abstract*—The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks. By distributing storage and computation across many servers, the resource can grow with demand while remaining economical at every size. We describe the architecture of HDFS and report on experience using HDFS to manage 25 petabytes of enterprise data at Yahoo!.**

Keywords: Hadoop, HDFS, distributed file system

I. Introduction and Related Work

Hadoop [1][16][19] provides a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce [3] paradigm. An important characteristic of Hadoop is the partitioning of data and computation across many (thousands) of hosts, and executing application computations in parallel close to their data. A Hadoop cluster scales computation capacity, storage capacity and IO bandwidth by simply adding commodity servers. Hadoop clusters at Yahoo! span 25 000 servers, and store 25 petabytes of application data, with the largest cluster being 3500 servers. One hundred other organizations worldwide report using Hadoop.

HDFS	Distributed file system Subject of this paper!
MapReduce	Distributed computation framework
HBase	Column-oriented table service
Pig	Dataflow language and parallel execution framework
Hive	Data warehouse infrastructure
ZooKeeper	Distributed coordination service
Chukwa	System for collecting management data
Avro	Data serialization system

developed at Facebook. Pig [4], ZooKeeper [6], and Chukwa were originated and developed at Yahoo! Avro was originated at Yahoo! and is being co-developed with Cloudera.

HDFS is the file system component of Hadoop. While the interface to HDFS is patterned after the UNIX file system, faithfulness to standards was sacrificed in favor of improved performance for the applications at hand.

HDFS stores file system metadata and application data separately. As in other distributed file systems, like PVFS [2][14], Lustre [7] and GFS [5][8], HDFS stores metadata on a dedicated server, called the NameNode. Application data are stored on other servers called DataNodes. All servers are fully connected and communicate with each other using TCP-based protocols.

Unlike Lustre and PVFS, the DataNodes in HDFS do not use data protection mechanisms such as RAID to make the data durable. Instead, like GFS, the file content is replicated on multiple DataNodes for reliability. While ensuring data durability, this strategy has the added advantage that data transfer bandwidth is multiplied, and there are more opportunities for locating computation near the needed data.

Several distributed file systems have or are exploring truly distributed implementations of the namespace. Ceph [17] has a cluster of namespace servers (MDS) and uses a dynamic subtree partitioning algorithm in order to map the namespace tree to MDSs evenly. GFS is also evolving into a distributed namespace implementation [8]. The new GFS will have hundreds of namespace servers (masters) with 100 million files per master. Lustre [7] has an implementation of clustered namespace on its roadmap for Lustre 2.2 release. The intent is to stripe a directory over multiple metadata servers (MDS), each of which contains a disjoint portion of the namespace. A file is assigned to a particular MDS using a hash function on the file name.

II. Architecture

A. NameNode

The HDFS namespace is a hierarchy of files and directories. Files and directories are represented on the NameNode by *inodes*, which record attributes like permissions, modification

Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. "The hadoop distributed file system." In *MSST*, vol. 10, pp. 1-10. 2010

Design Motivations (similar with GFS)

- Many inexpensive commodity hardware and failures are very common
- Many big files: millions of files, ranging from MBs to GBs
- Two types of reads
  - Large streaming reads
  - Small random reads
- Once written, files are seldom modified
  - Random writes are supported but do not have to be efficient
- High sustained bandwidth is more important than low latency

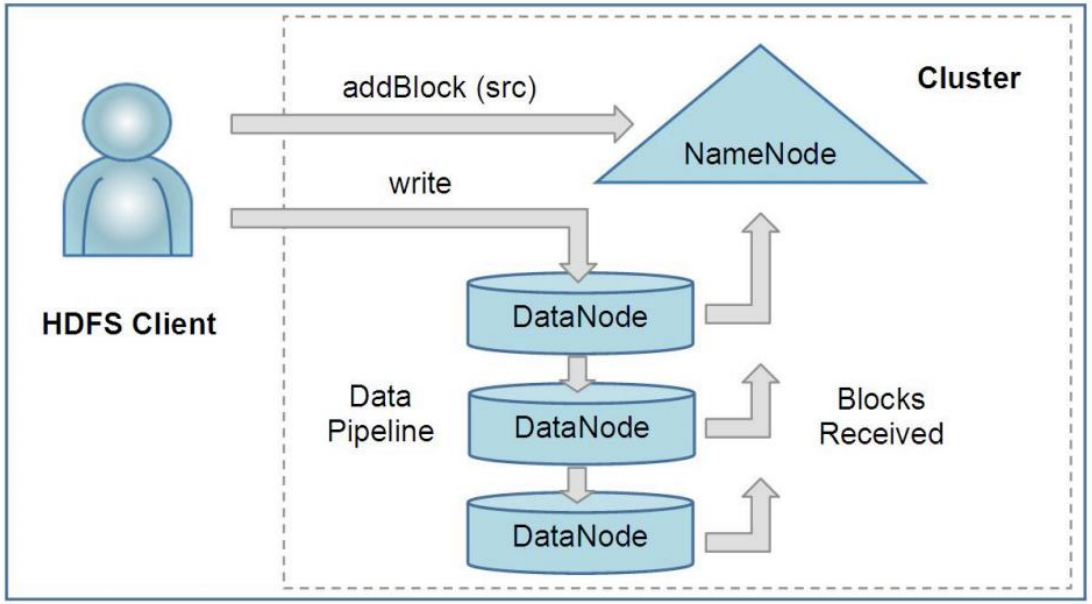
HDFS – Architecture Overview

NameNode

- Maintains meta-data in RAM
- maintains the namespace tree and the mapping of file blocks to DataNodes

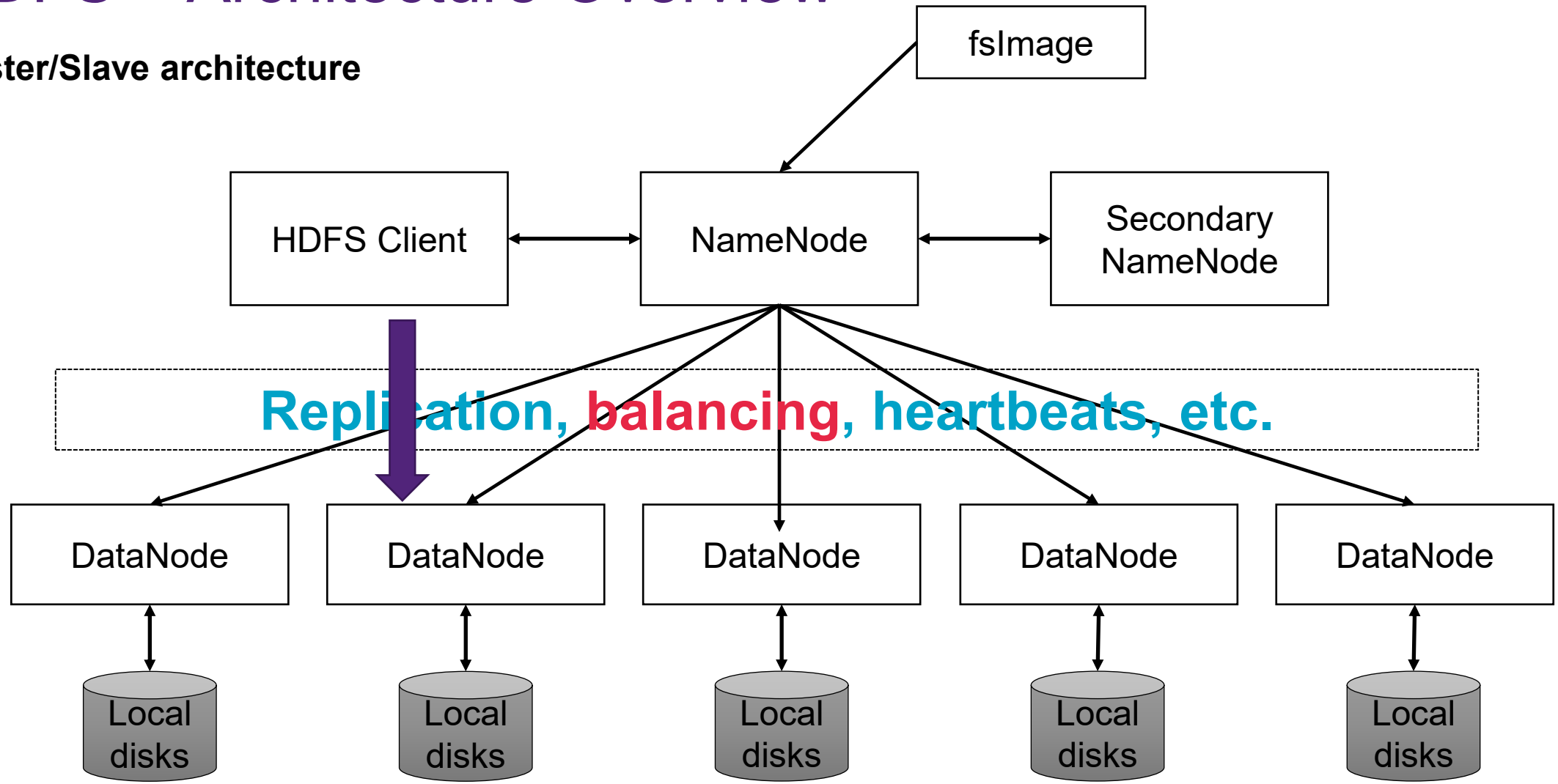
DataNode

- Store data and replicas
- Send heartbeats to NameNode
- receives maintenance commands from the NameNode indirectly (in replies to heartbeats).
  - replicate blocks to other nodes;
  - remove local block replicas;
  - re-register or to shut down the node;
  - send an immediate block report.



HDFS – Architecture Overview

Master/Slave architecture



Load Balancing in NginX

The following load balancing mechanisms (or methods) are supported in nginx:

round-robin — requests to the application servers are distributed in a round-robin fashion,

least-connected — next request is assigned to the server with the least number of active connections,

ip-hash — a hash-function is used to determine what server should be selected for the next request (based on the client's IP address).

```
http {
    upstream myapp1 {
        server srv1.example.com;
        server srv2.example.com;
        server srv3.example.com;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://myapp1;
        }
    }
}
```

Round-robin in Nginx configuration file

```
upstream myapp1 {
    server srv1.example.com weight=3;
    server srv2.example.com;
    server srv3.example.com;
}
```

Weighted round-robin in Nginx configuration file

```
upstream myapp1 {
    least_conn;
    server srv1.example.com;
    server srv2.example.com;
    server srv3.example.com;
}
```

Least connection in Nginx configuration file

https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/

Load Balancing in NginX cont'd

More LB algorithms are available in NginX Plus (commercial version):

Least Time (NGINX Plus only) – For each request, NGINX Plus selects the server with the lowest average latency and the lowest number of active connections**.**

```
upstream backend {
    least_time header;
    server backend1.example.com;
    server backend2.example.com;
}
```

Random (NGINX Plus only) – Each request will be passed to a randomly selected server.

```
upstream backend {
    random two least_time=last_byte;
    server backend1.example.com;
    server backend2.example.com;
    server backend3.example.com;
    server backend4.example.com;
}
```

https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/

Outline

- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

Network Model

OSI (Open Systems Interconnect) Model:

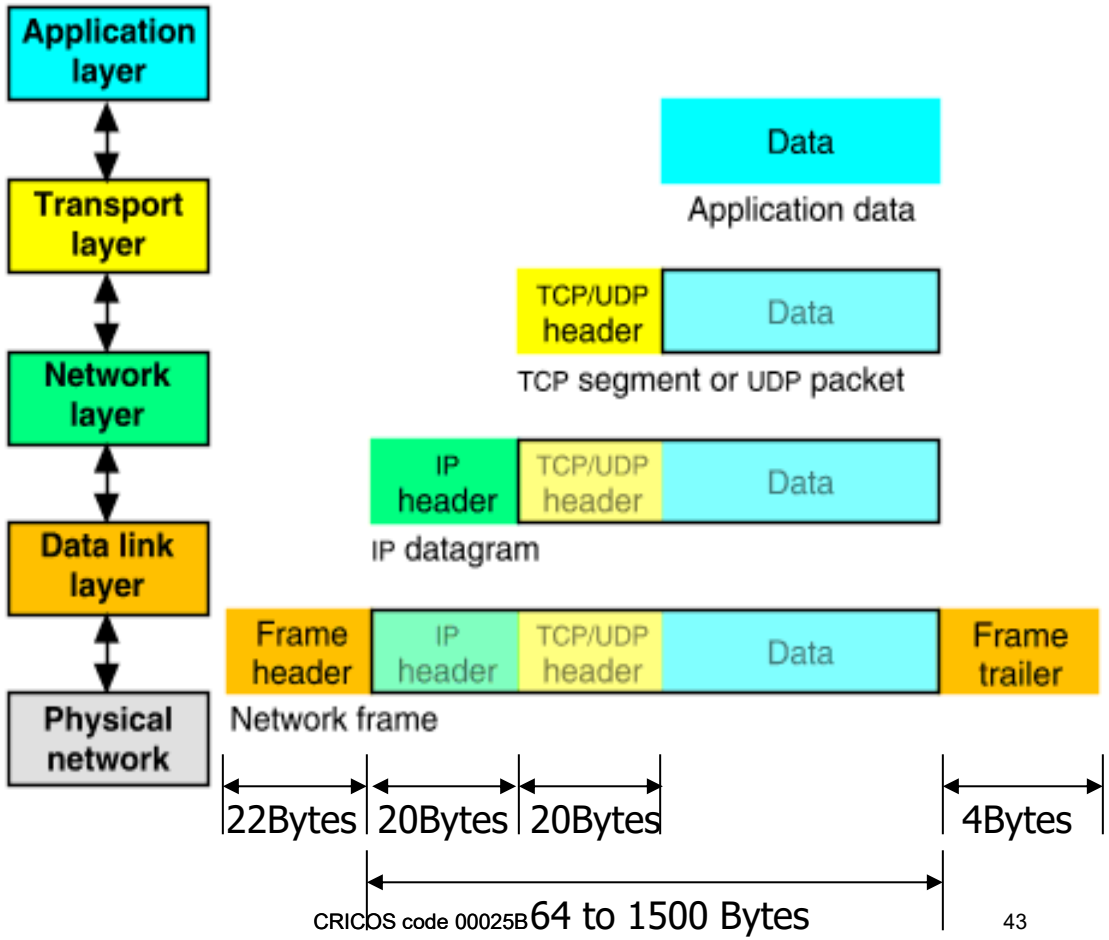
- is a conceptual model that characterizes and standardizes the communication functions of a telecommunication without regard to its underlying internal structure and technology.
- Its goal is to interpret diverse communication systems with standard communication protocols.
- The model partitions a communication system into 7 abstraction layers.

OSI model				
Layer			Protocol data unit (PDU)	Function[6]
Host layers	7	Application	Data (HTTP, SMTP, POP3, FTP, etc.)	High-level APIs, including resource sharing, remote file access
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram (TCP/UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1	Physical	Symbol	Transmission and reception of raw bit streams over a physical medium

Packet Encapsulation

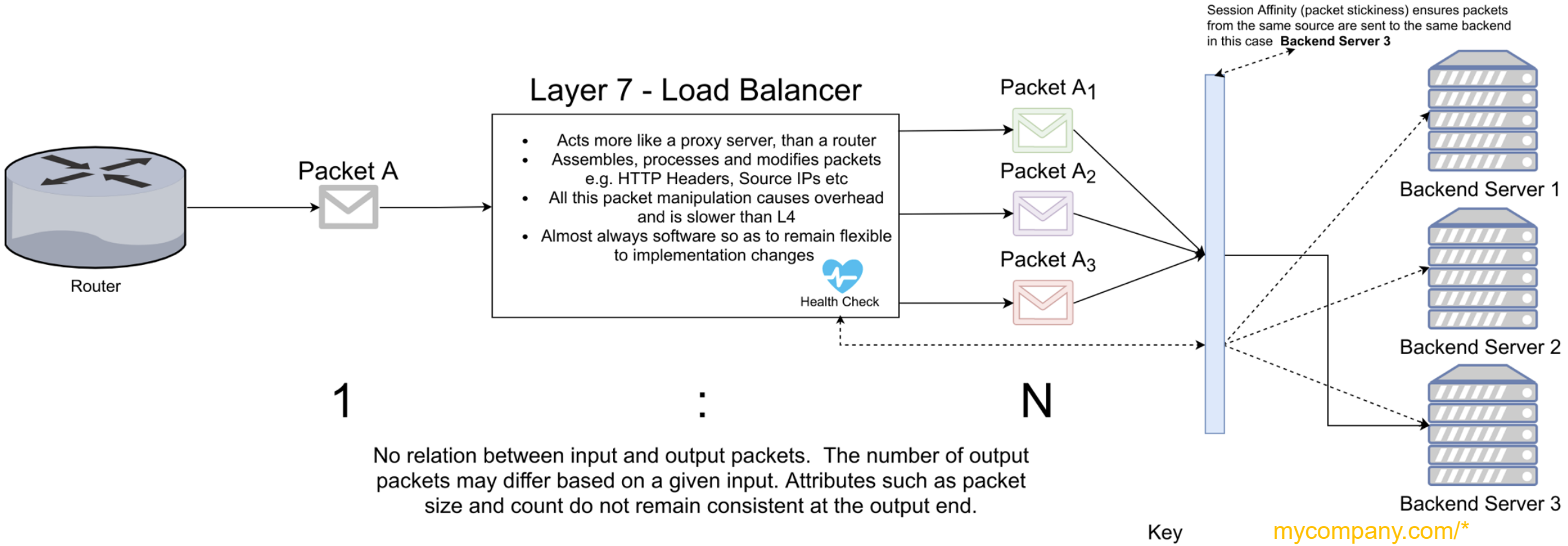
TCP/IP Suite:

- The data needs to be encapsulated before physically transfer to another location
- Each layer in TCP/IP adds the data by prepending headers
- For a Layer-n Load Balancer, higher n means more encapsulated in the packet.



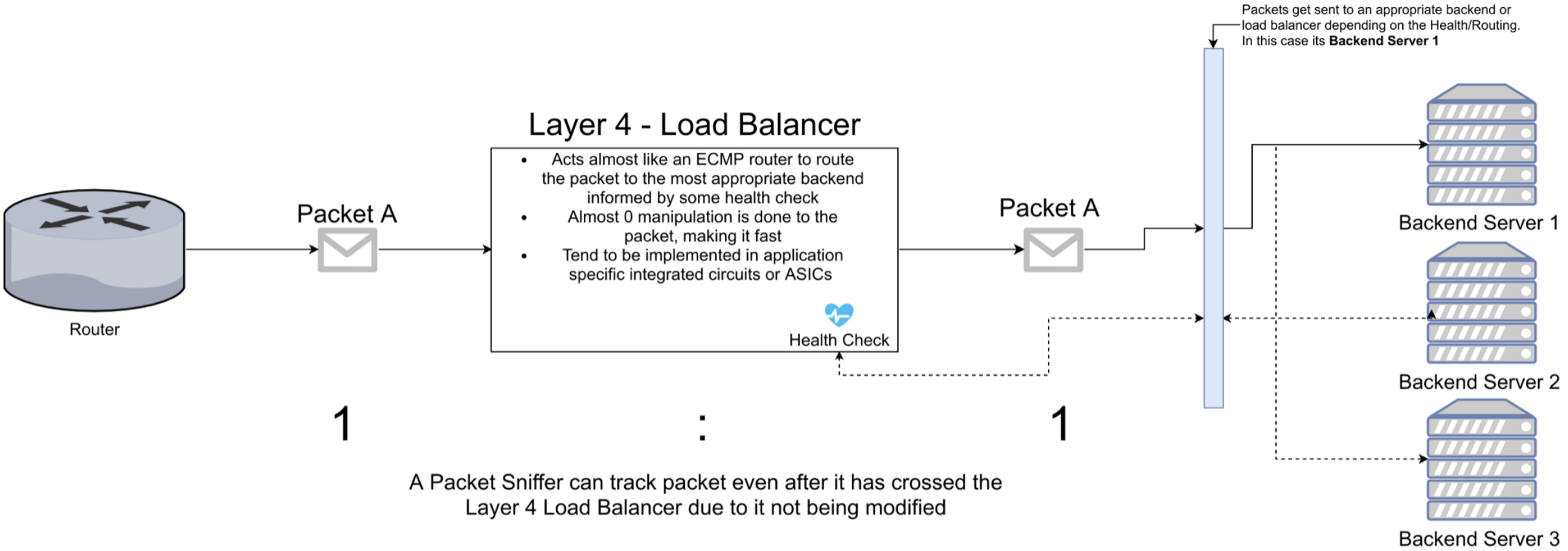
[Explanation: OSI and TCP IP Models](#) **(19'19")**

HTTPs (Layer 7) Load Balancing



TCP vs HTTP(S) Load Balancing. https://medium.com/martinomburajr/distributed-computing-tcp-vs-http-s-load-balancing-7b3e9efc6167

TCP (Layer 4) Load Balancing



TCP vs HTTP(S) Load Balancing. https://medium.com/martinomburajr/distributed-computing-tcp-vs-http-s-load-balancing-7b3e9efc6167

Differences between Layer 4 and Layer 7 Load Balancing

	Layer 4 LB (TCP)	Layer 7 LB (HTTPs)
Layer	Transport Layer	Application Layer
Packet Manipulation	No	Yes
SSL Traffic	No	Yes
Logging & Monitoring	Not suitable	Yes
Implementation	Dedicated hardware	Typically software
Throughput Speed	Fast	Relatively lower

TCP vs HTTP(S) Load Balancing. https://medium.com/martinomburajr/distributed-computing-tcp-vs-http-s-load-balancing-7b3e9efc6167

Outline

- Networking and Virtual Private Cloud
- Load Balancing
  - What & Why Load Balancing
  - Algorithms
  - LB in Cloud Architecture
  - LB in Distributed Systems
  - LB in Network Communications
  - LB in Cloud Product

Load Balancing in GCP – Overview

Worldwide autoscaling and load balancing

- Scale your applications on Compute Engine from small to big.
- Distribute your load-balanced compute resources in single or multiple regions
  - close to your users to meet your high availability requirements.
- Put your resources behind a single IP (IP Anycast technology).
  - a single anycast IP front-ends all your backend instances in regions around the world.
  - It provides cross-region load balancing
    - E.g. automatic multi-region failover, which gently moves traffic in fractions if backends become unhealthy.
  - In contrast to DNS-based global load balancing solutions, Cloud Load Balancing reacts instantaneously to changes in users, traffic, network, backend health, and other related conditions.

**IP Anycast, is a networking technique that allows for multiple machines to share the same IP address.*

Cloud Load Balancing

Software-defined load balancing

- a fully distributed, software-defined, managed service for all traffic.
- It is not an instance- or device-based solution, won't be locked into physical load balancing infrastructure.
- Multiple traffic supported: HTTP(S), TCP/SSL, and UDP.

Over one million queries per second

- the same frontend-serving infrastructure that powers Google.
- supports 1 million+ queries per second with consistent high performance and low latency.
- Traffic enters Cloud Load Balancing through 80+ distinct global load balancing locations, maximizing the distance traveled on Google's fast private network backbone.

Seamless autoscaling

- Cloud Load Balancing can scale seamlessly and automatically when users and traffic grow

Cloud Load Balancing

Internal load balancing

- without load balancer exposure to the internet.
- GCP internal load balancing is architected using Andromeda.
- VPN supported for clients.

Support for cutting-edge protocols

- includes support for the latest application delivery protocols.
  - E.g. It supports HTTP/2 with gRPC when connecting to backends.

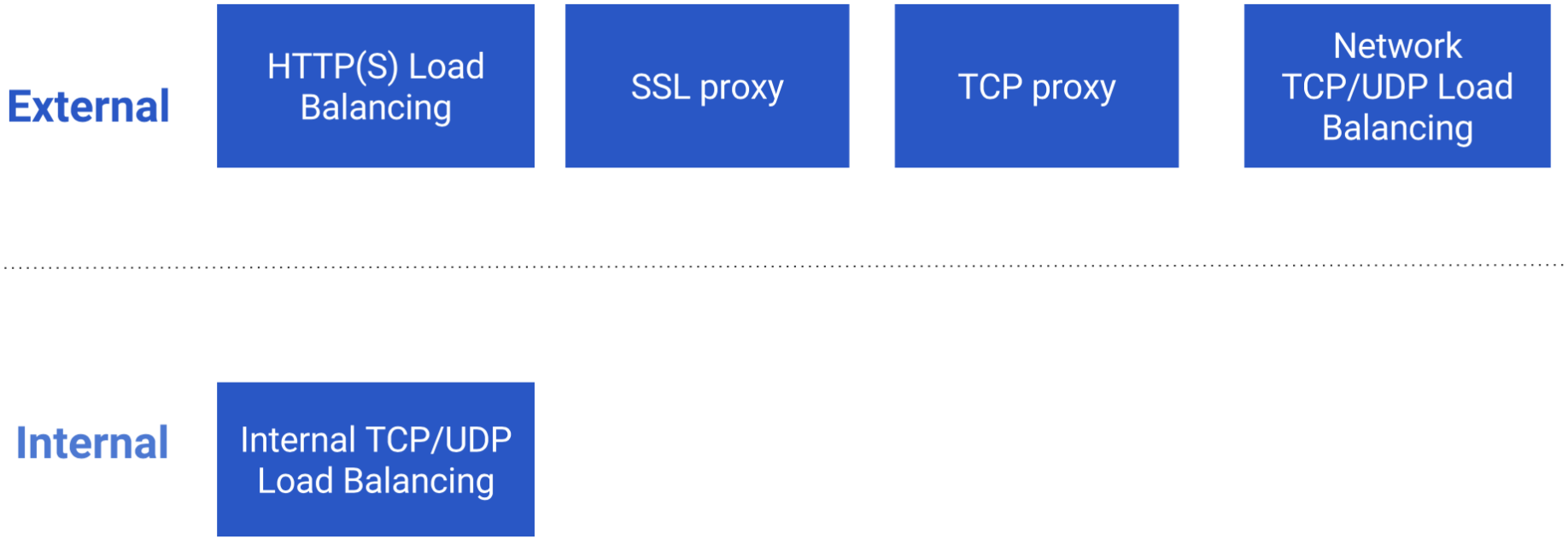
What is HTTP/2?

It's the second version of HTTP to have faster, simpler, and more robust applications.

External and Internal Load Balancing

GCP's load balancers can be divided into external and internal load balancers.

- External load balancers distribute traffic coming from the *internet* to your *GCP network*.
- Internal load balancers distribute traffic within your GCP network.

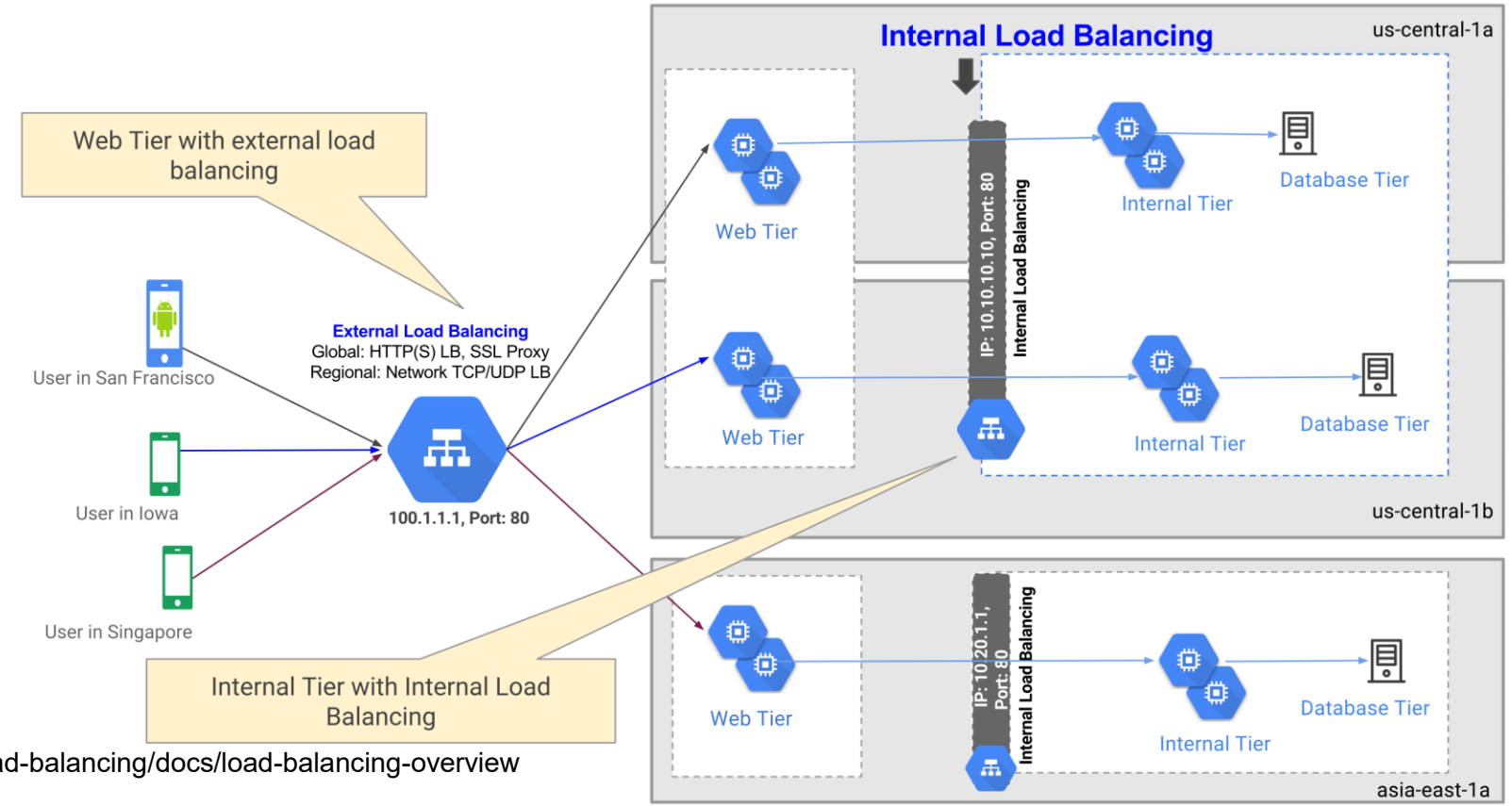


https://cloud.google.com/load-balancing/docs/load-balancing-overview

Hybrid Load Balancing Example on GCP

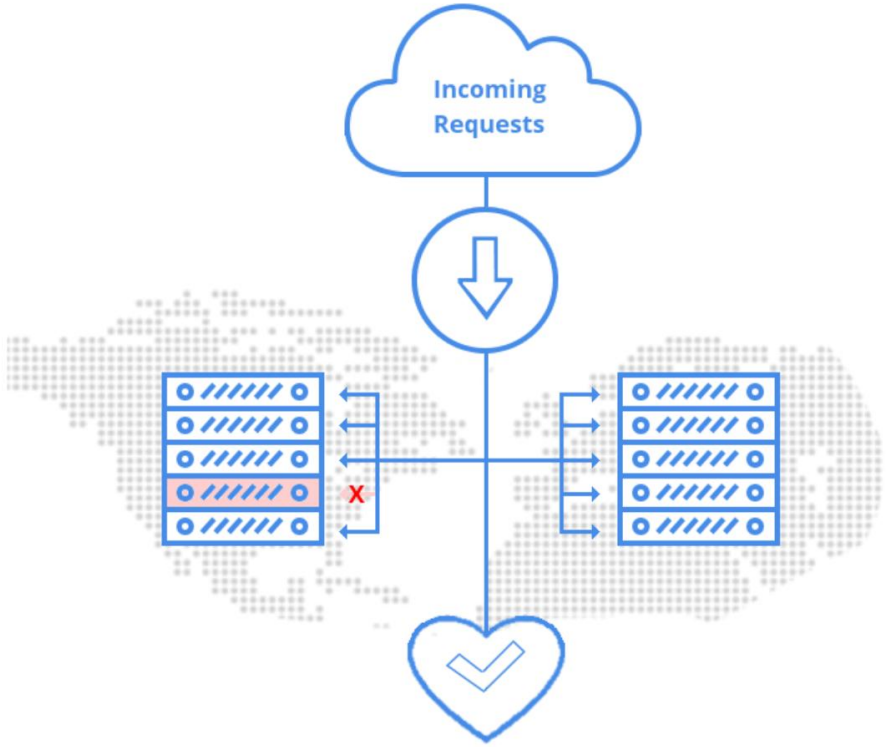
Traffic from users in San Francisco, Iowa, and Singapore is directed to an external load balancer, which distributes that traffic to different regions in a GCP network.

An internal load balancer then distributes traffic between the us-central-1a and us-central-1b zones.

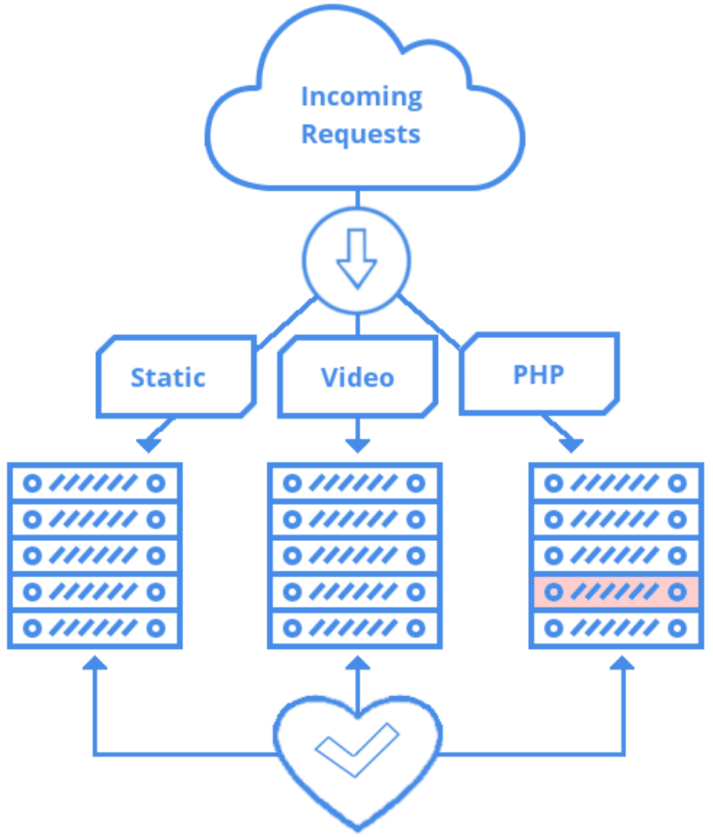


https://cloud.google.com/load-balancing/docs/load-balancing-overview

HTTP(S) Load Balancing Setup



Cross-region load balancing



Content-based load balancing

https://cloud.google.com/load-balancing/docs/load-balancing-overview

No Free Lunch

To have such Load Balancers, you need to manually setup

The Load Balancing services are not free

Load balancing and forwarding rules

The following applies to all types of load balancing and forwarding rules (protocol forwarding).

Iowa (us-central1)

Item	Price per Unit (USD)	Pricing Unit
First 5 forwarding rules	$0.025	Per Hour
Per additional forwarding rule	$0.010	Per Hour
Ingress data processed by load balancer	$0.008	Per GB

1-month vanilla load balancing services with 5 forwarding rules

Rule cost: 24*30*0.025 = $18

Ingress cost: 1024GB *30 *0.008 =$245.76

Total cost = $18 + 245.76 = $263.76

https://cloud.google.com/load-balancing/docs/load-balancing-overview

References

1.“Cloud computing: concepts, technology & architecture”. Erl, Thomas, Ricardo Puttini, and Zaigham Mahmood. Pearson Education, 2013.

2.https://www.jscape.com/blog/load-balancing-algorithms

3.https://en.wikipedia.org/wiki/Load_balancing_(computing)

4.What Is Layer 4 Load Balancing? https://www.nginx.com/resources/glossary/layer-4-load-balancing/

5.What Is Layer 7 Load Balancing? https://www.nginx.com/resources/glossary/layer-7-load-balancing/

6.Gaochao Xu, Junjie Pang and Xiaodong Fu. (2013) A Load Balancing Model based on Cloud Partitioning for the Public Cloud, Tsinhhua Science and Technology, ISSN l1007-0214l, l04/12l, pp34-39, Vol. 18, No. 1.

7.Cardellini, V., Colajanni, M., & Yu, P. S. (1999) Dynamic load balancing on web-server systems, IEEE Internet Computing, 3(3), 28-39. DOI: 10.1109/4236.769420.

8.Rajwinder Kaur and Pawan Luthra (2014) Load Balancing in Cloud Computing, Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC, pp374-381.

9.http://www.javatpoint.com/cloud-computing-tutorial

10.https://blog.newrelic.com/2016/04/07/importance-local-load-balancing/

11.https://www.cloudflare.com/en-au/learning/cloud/what-is-a-virtual-private-cloud

Tutorial and Practical

Tutorial:

- Answer the questions below to enhance the understanding of Lecture 1.
- Learn Linux Basics and Command Lines.

Practical:

- [Optional]. Create a VPC network on GCP and set up a Firewall Rule to allow SSH access to the VPC.
- Run basic Linux command lines.