# Concepts  & enabling technologies of cloud computing

# Outline

- Concepts and enabling technologies of cloud computing
    - Virtualization
    - Load balancing
    - Scalability & Elasticity
    - Deployment
    - Replication
    - Monitoring
    - MapReduce
    - Identity and Access Management
    - Service Level Agreements
    - Billing

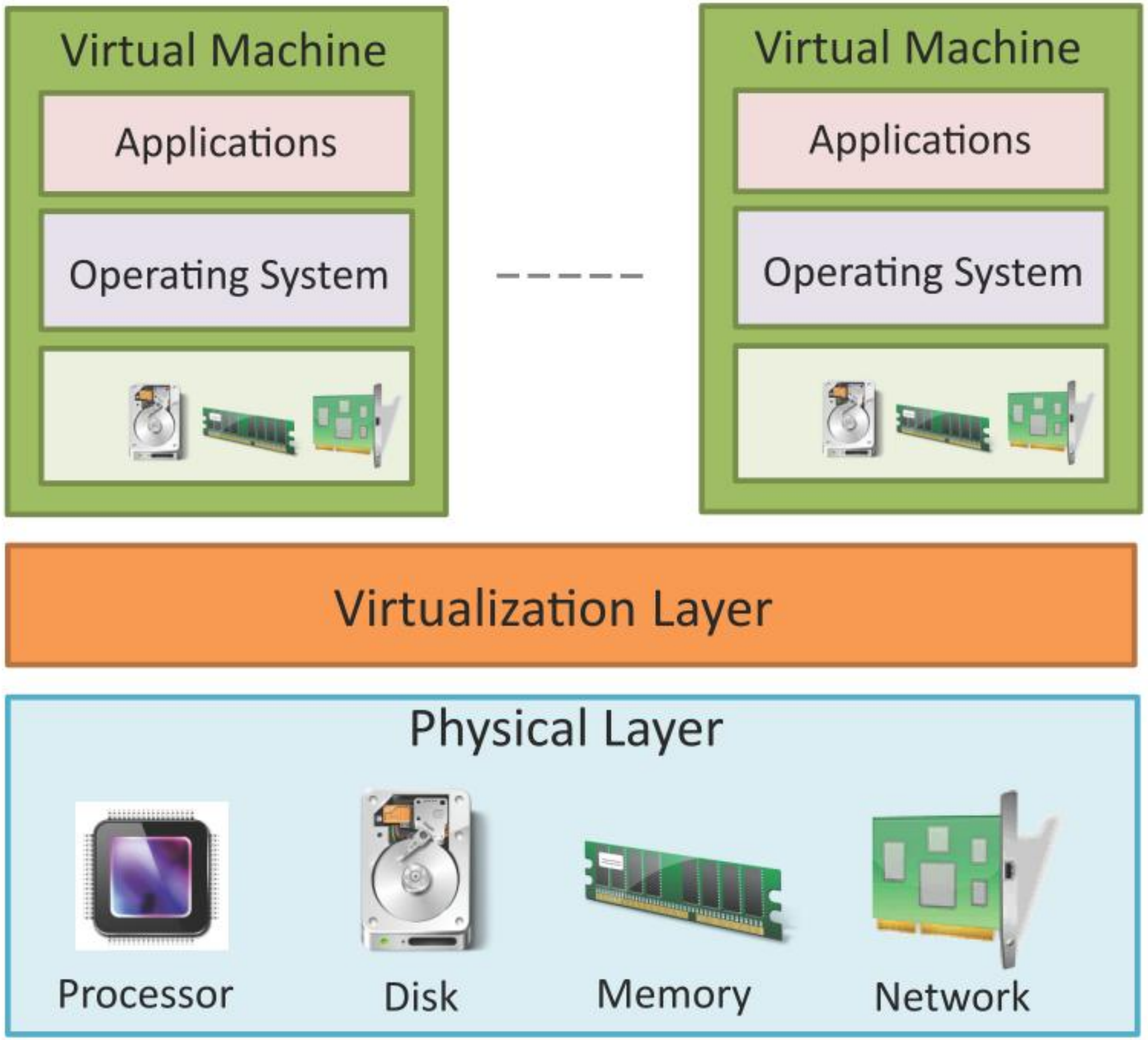
# Virtualization

- Virtualization refers to the partitioning the resources of a physical system (such as computing, storage, network and memory) into multiple virtual resources.

- Key enabling technology of cloud computing that allow pooling of resources.

- In cloud computing, resources are pooled to serve multiple users using multi-tenancy.

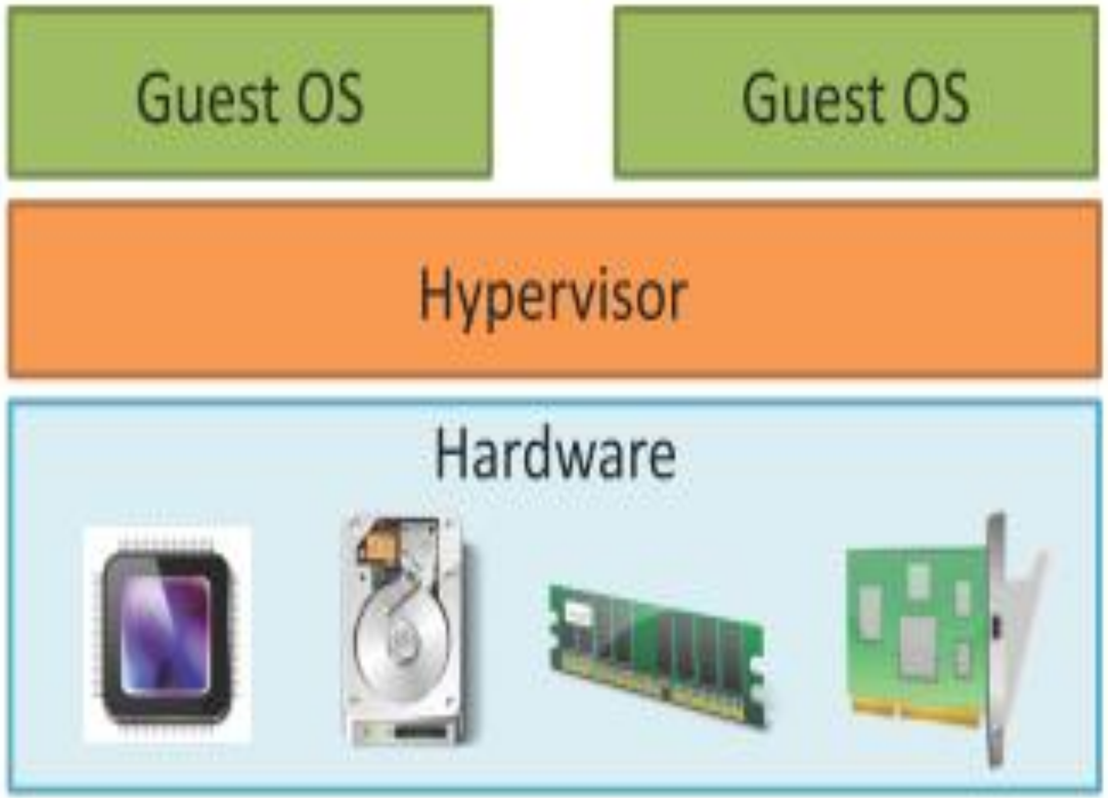
# Virtualization



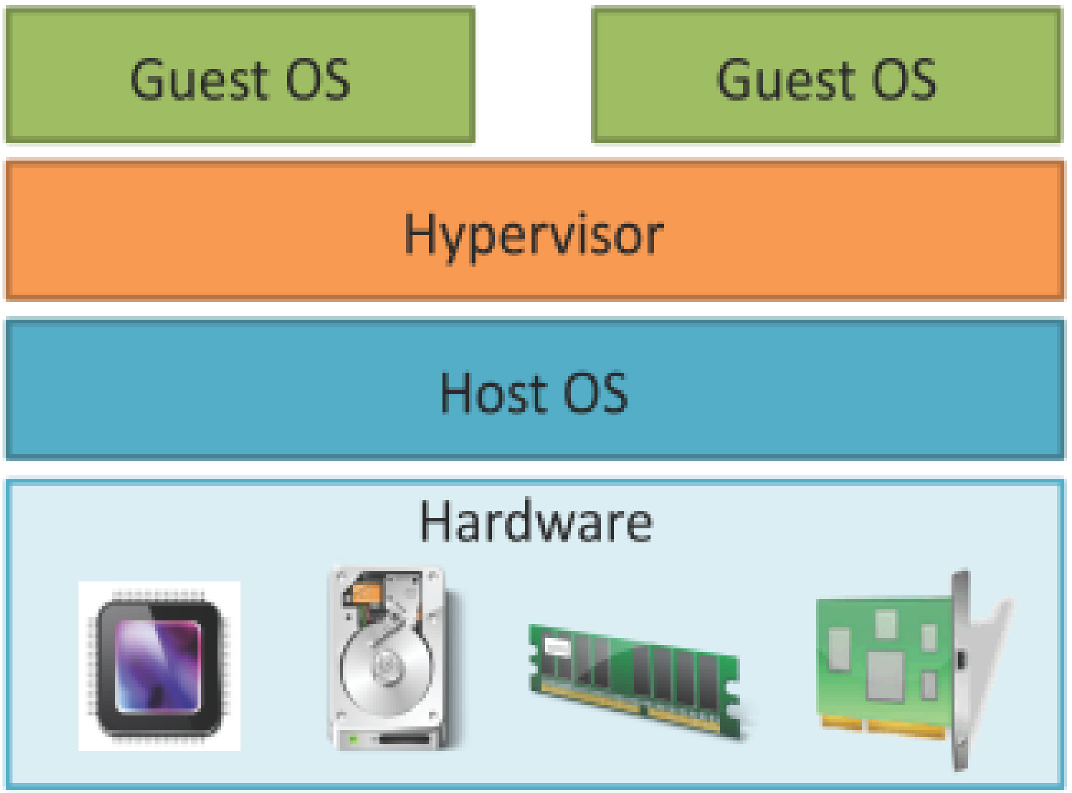
# Hypervisor

- The virtualization layer consists of a hypervisor or a virtual machine monitor (VMM).
- Hypervisor presents a virtual operating platform to a guest operating system (OS).
- Type-1 Hypervisor
  - Type-I or the native hypervisors run directly on the host hardware and control the hardware and monitor the guest operating systems.
- Type-2 Hypervisor
  - Type 2 hypervisors or hosted hypervisors run on top of a conventional (main/host) operating system and monitor the guest operating systems.

# Hypervisor



Type-1/Native Hypervisor



Type-2/Hosted Hypervisor

# Hypervisor

| HYPERVISOR | Type |
|---|---|
| Citrix XenServer | Type-1 / Native |
| Oracle VM Server | Type-1 / Native |
| KVM | Type-1 / Native |
| VMWare ESX/ESXi | Type-1 / Native |
| Microsoft Hyper-V | Type-1 / Native |
| Xen Hypervisor | Type-1 / Native |
| VMWare Workstation | Type-2 / Hosted |
| VirtualBox | Type-2 / Hosted |

# Guest OS

- An Operating System that is installed in a Virtual machine in addition to the host or main OS
- In virtualization, the guest OS can be different from the host OS

# Types of Virtualization

1. Full Virtualization
2. Para-Virtualization
3. Hardware Virtualization

- Full Virtualization
  - The virtualization layer completely decouples the guest OS from the underlying hardware.
  - The guest OS requires no modification and is not aware that it is being virtualized.
  - Full virtualization is enabled by direct execution of user requests and binary translation of OS requests.

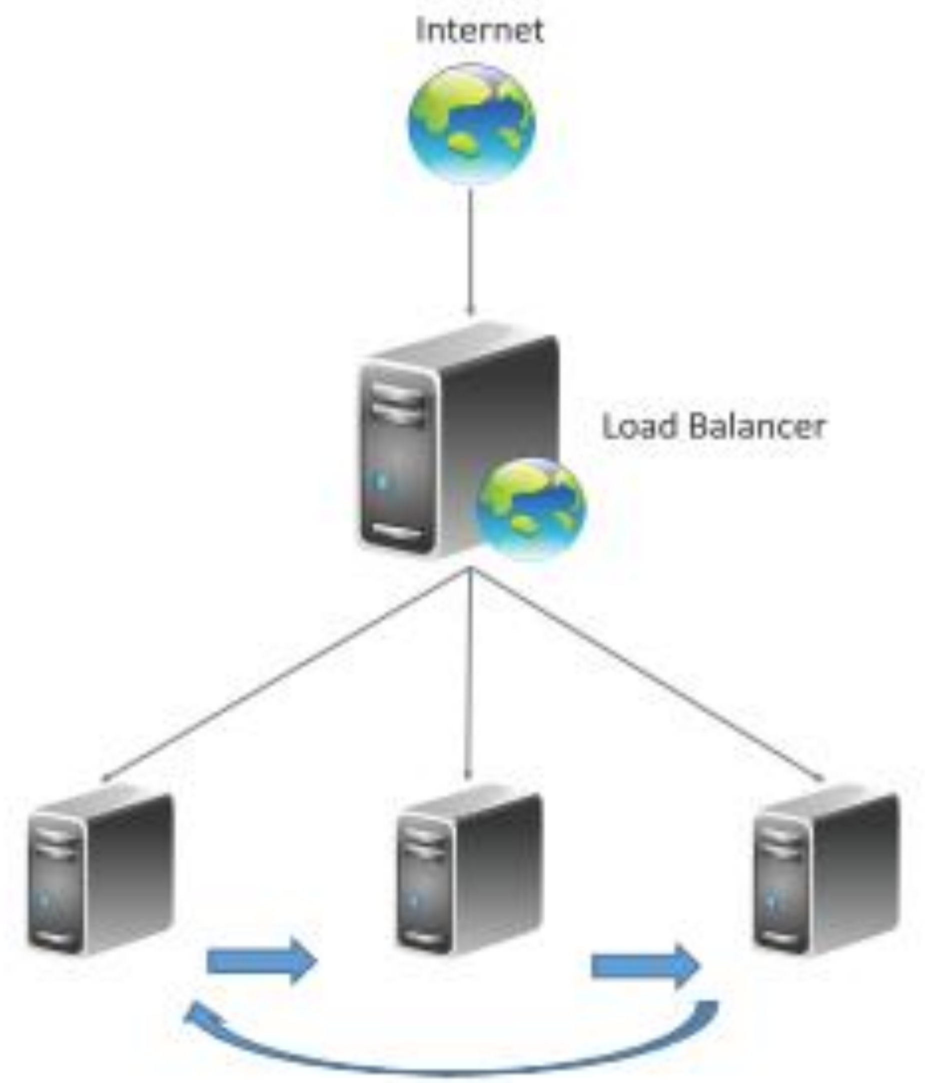
# Types of Virtualization

- Para-Virtualization
  - The Guest OS is modified to enable communication with the hypervisor to improve performance and efficiency.
  - The guest OS kernel is modified to replace non-virtualizable instructions with hyper-calls that communicate directly with the virtualization layer hypervisor.
- Hardware Virtualization
  - Hardware assisted virtualization is enabled by hardware features such as Intel’s Virtualization Technology (VT-x) and AMD’s AMD-V.
  - Privileged and sensitive calls are set to automatically trap to the hypervisor. Thus, there is no need for either binary translation or para-virtualization.

# Load Balancing

- Cloud computing resources can be scaled up on demand to meet the performance requirements of applications.
- 
- Load balancing distributes workloads across multiple servers to meet the application workloads.
- The goals of load balancing techniques include:
  - Achieve maximum utilization of resources
  - Minimizing the response times
  - Maximizing throughput

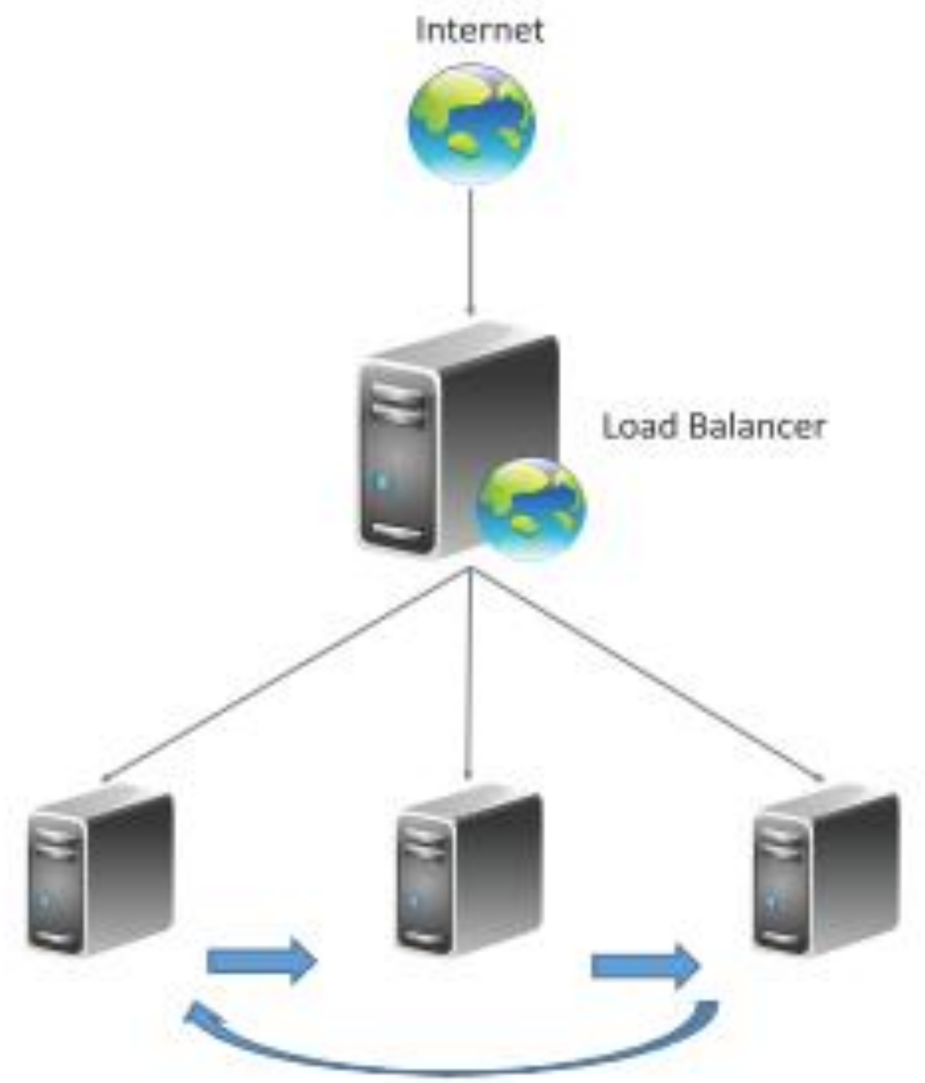
# Load Balancing



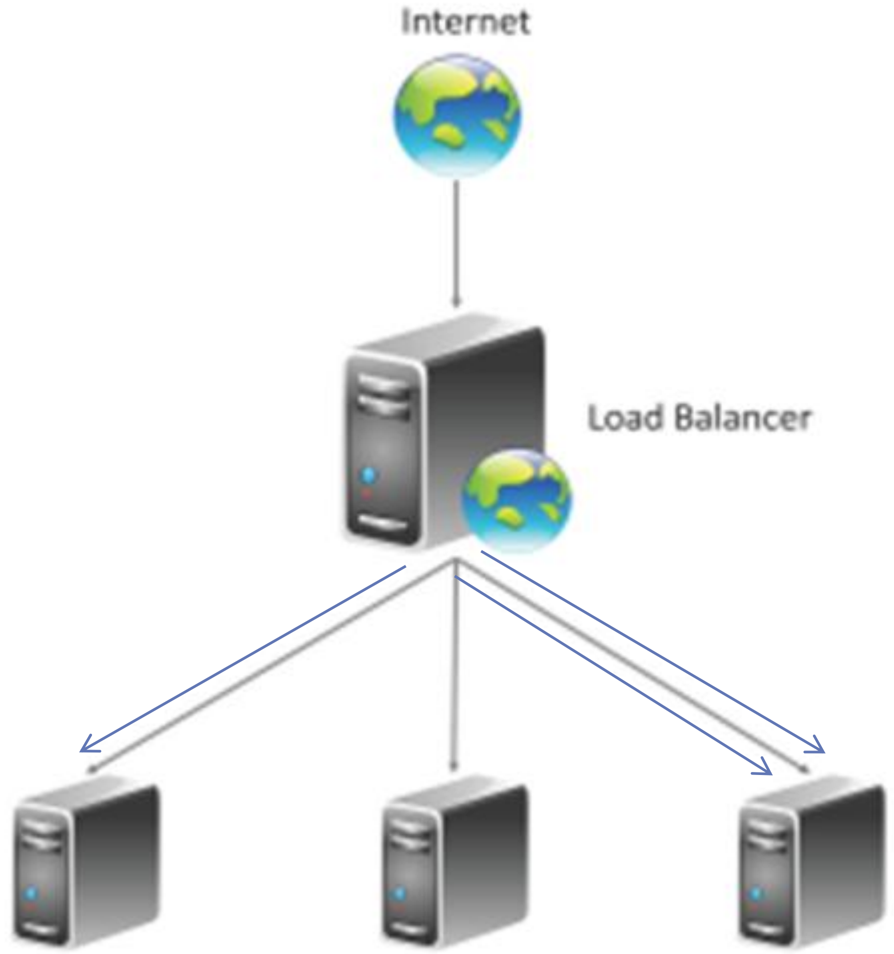
# Load Balancing Algorithms

- Round Robin load balancing
- Weighted Round Robin load balancing
- Low Latency load balancing
- Least Connections load balancing
- Priority load balancing
- Overflow load balancing

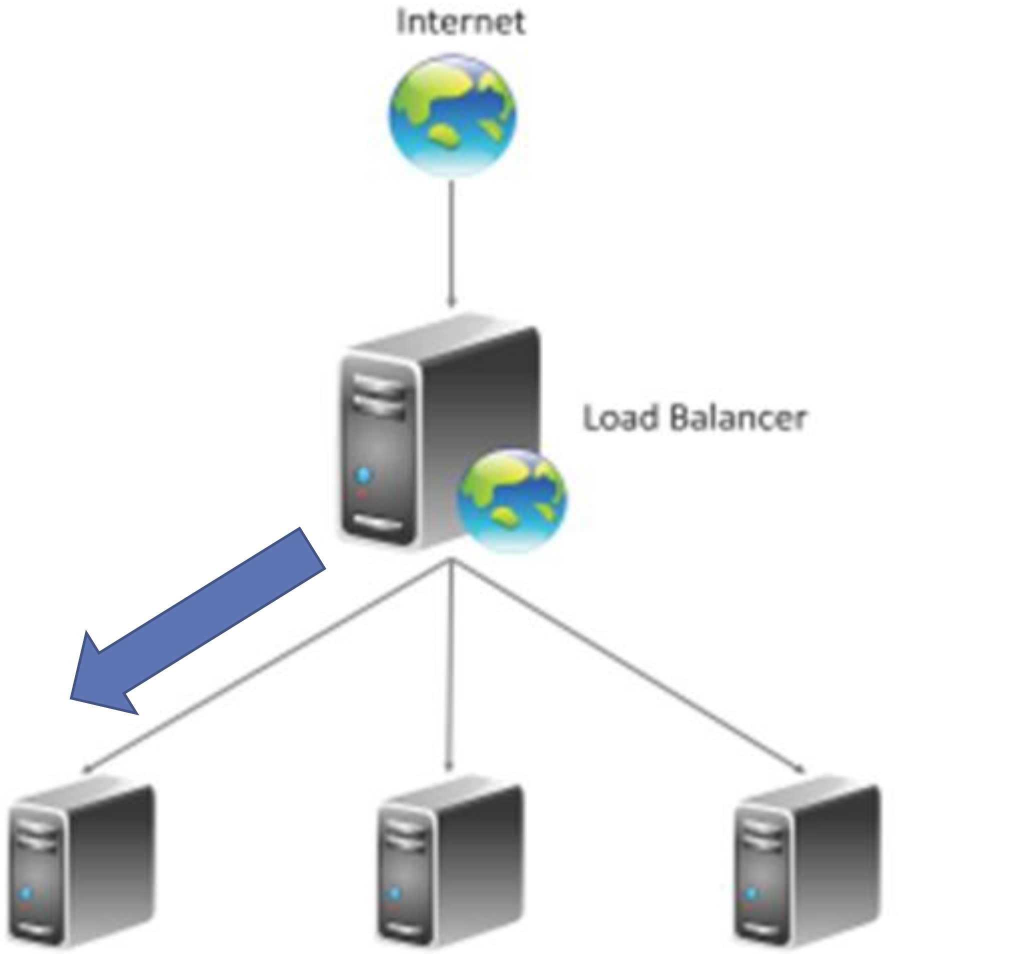
# Round Robin Load Balancing



# Weighted Round Robin load balancing

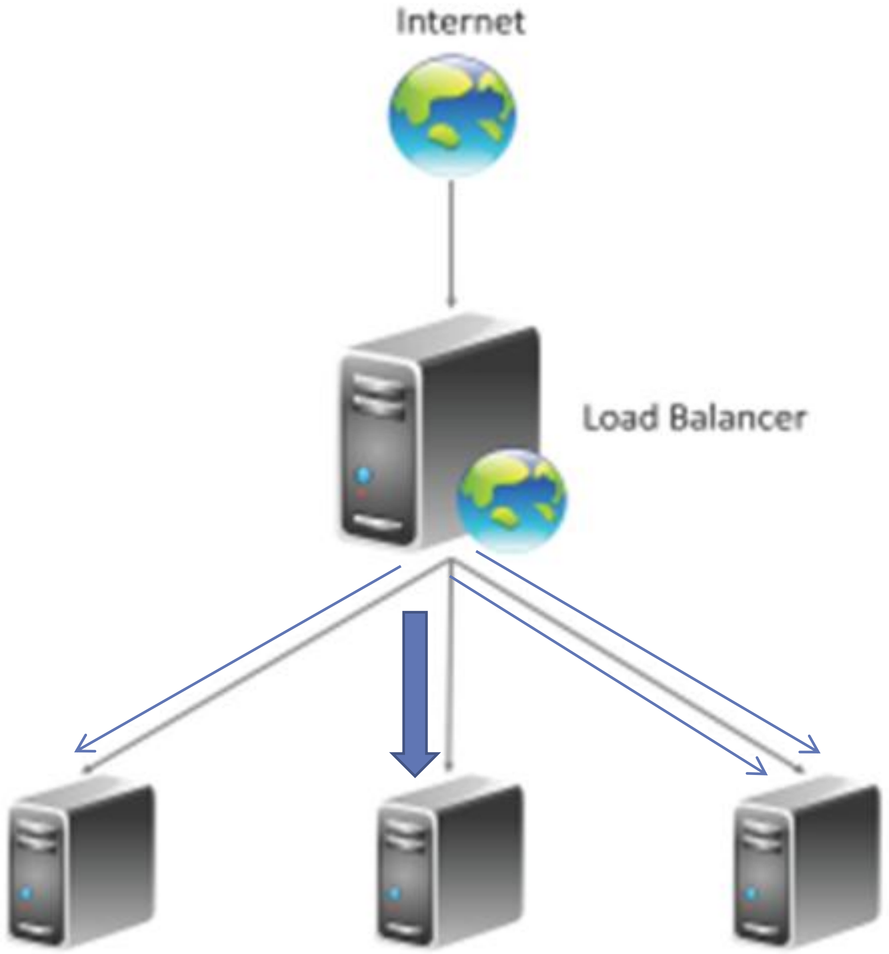


# Low Latency load balancing



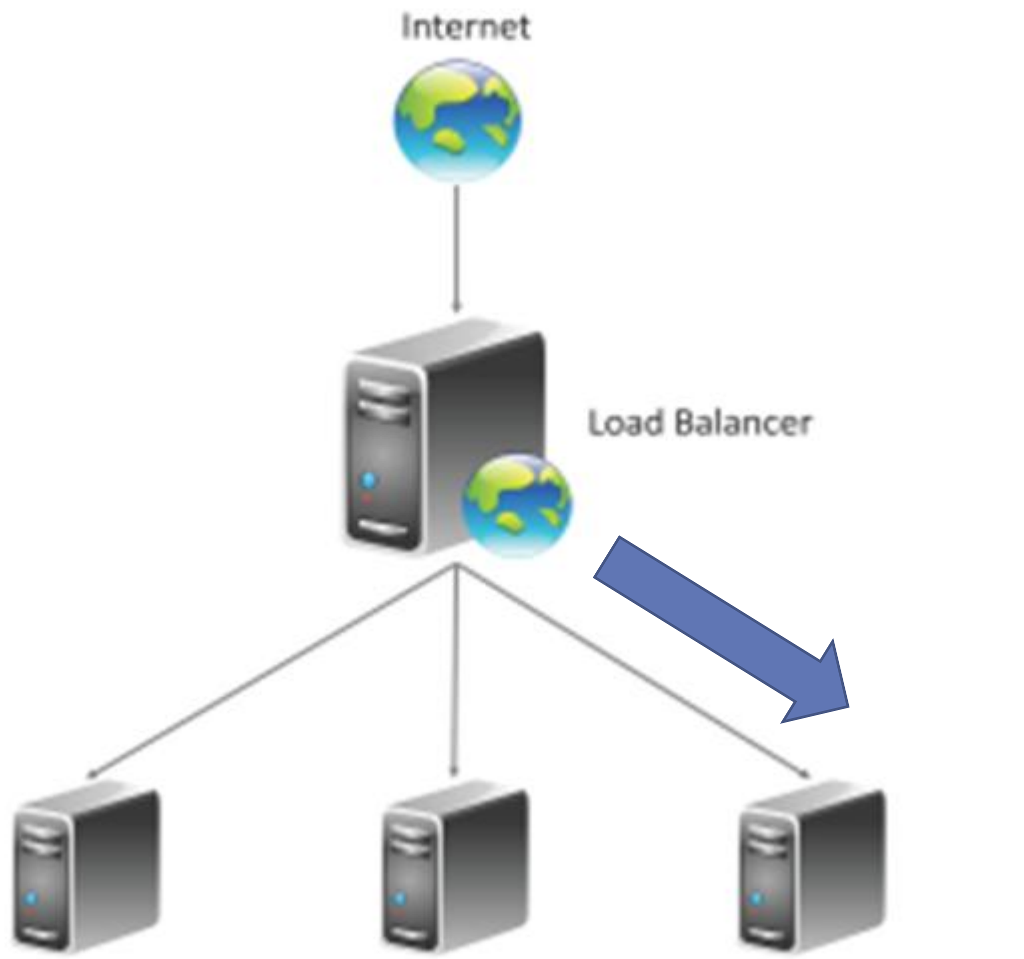
Low Latency Medium Latency High Latency

# Least Connections load balancing



Least Connections

# Priority load balancing

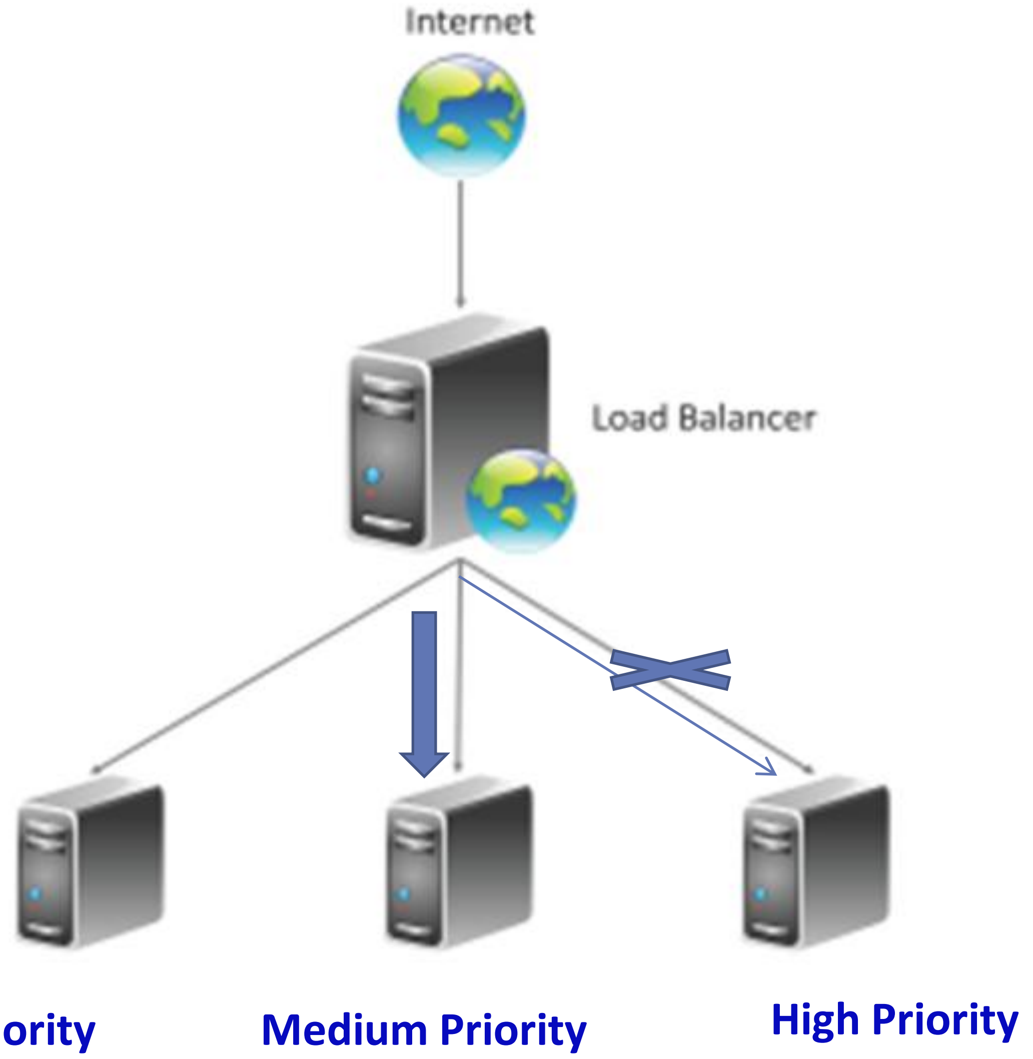


Low  Priority

Medium Priority

High Priority

# Overflow load balancing



# Load Balancing - Persistence Approaches

- Since load balancing can route successive requests from a user session to different servers, maintaining the state or the information of the session is important.

- Persistence Approaches
  - Sticky sessions
  - Session Database
  - Browser cookies
  - URL re-writing

# Load Balancing - Persistence Approaches

- **Sticky Sessions**
  - All Requests belonging to a user session are routed to the same server.
  - These sessions are called sticky sessions
  - Advantage : Session management is simple
  - Disadvantage : If the server fails, all sessions belonging to that serer are lost as there is no automatic failover
- **Session Database**
  - All the session information is stored externally in a separate session database, which is often replicated to avoid a single point of failure
  - Advantage : Automatic failover
  - Disadvantage : Session management has additional overhead of storing session information.

# Load Balancing - Persistence Approaches

- **Browser cookies**
  - Session information is stored on the client side in the form of browser cookies.
  - Advantage : Session management is easy,
    least overhead for the load balancer
- **URL re-writing**
  - A URL re-write engine stores the session information by modifying the URLs on the client side.
  - Advantage : Avoids overhead for the load balancer
  - Disadvantage : Amount of Session information that can be stored is limited.

# Load Balancing - Hardware or Software

- Load Balancing can be implemented in Hardware or Software.
- Software based LBs run on standard OS
- Hardware based LBs implement load balancing algorithms in Application Specific Integrated Circuits (ASICs).

Ex.

**Software load balancers**

Nginx

HAProxy

Pound

Varish

**Hardware load balancers**

Cisco Systems Catalyst 6500

Coyote Point Equalizer

F5 Networks BIG-IP-LTM

Barracuda Load Balancer

# Scalability & Elasticity

- Multi-tier applications such as e-Commerce, social networking, business-to-business, etc. can experience rapid changes in their traffic.
- Capacity planning involves determining the right sizing of each tier of the deployment of an application in terms of the number of resources and the capacity of each resource.
- Capacity planning may be for computing, storage, memory or network resources.

# Scaling Approaches

- **Vertical Scaling/Scaling up:**
    - Involves upgrading the hardware resources (adding additional computing, memory, storage or network resources).
- **Horizontal Scaling/Scaling out**
    - Involves addition of more resources of the same type.

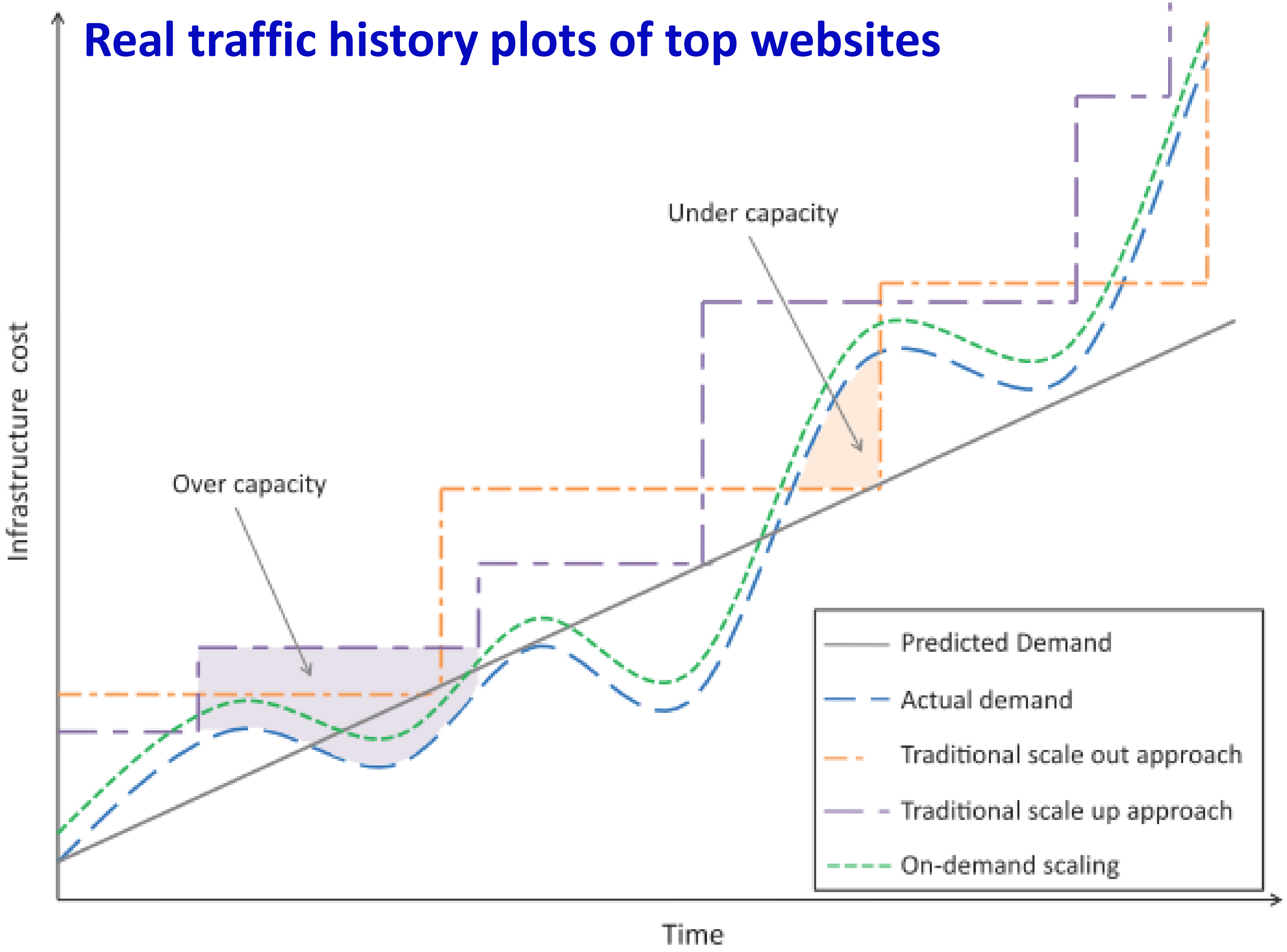
# Scale-up or Scale-out ?

- **Vertical Scaling (Scale-up):** Generally refers to adding more processors and RAM, buying a more expensive and robust server.
  - **Pros**
    - Less power consumption than running multiple servers
    - Cooling costs are less than scaling horizontally
    - Generally less challenging to implement
    - Less licensing costs
    - (sometimes) uses less network hardware than scaling horizontally (this is a whole different topic that we can discuss later)
  - **Cons**
    - PRICE, PRICE, PRICE
    - Greater risk of hardware failure causing bigger outages
    - generally severe vendor lock-in and limited upgradeability in the future

# Scale-up or Scale-out?

- **Horizontal Scaling (Scale-out):** Generally refers to adding more servers with less processors and RAM. This is usually cheaper overall and can literally scale infinitely (although we know that there are usually limits imposed by software or other attributes of an environment's infrastructure)
  - **Pros**
    - Much cheaper than scaling vertically
    - Easier to run fault-tolerance
    - Easy to upgrade
  - **Cons**
    - More licensing fees
    - Bigger footprint in the Data Center
    - Higher utility cost (Electricity and cooling)
    - Possible need for more networking equipment (switches/routers)

# Scalability & Elasticity



**Overprovisioning**

Higher Capital expenditure

Higher operating expenditure

**Underprovisioning**

Traffic overloads

Slow response time

Low throughputs

# Deployment

- Cloud application deployment design is an iterative process that involves:
    - Deployment Design
    - Performance Evaluation
    - Deployment Refinement

- Deployment Design
    - The variables in this step include the number of servers in each tier, computing, memory and storage capacities of severs, server interconnection, load balancing and replication strategies.

# Deployment

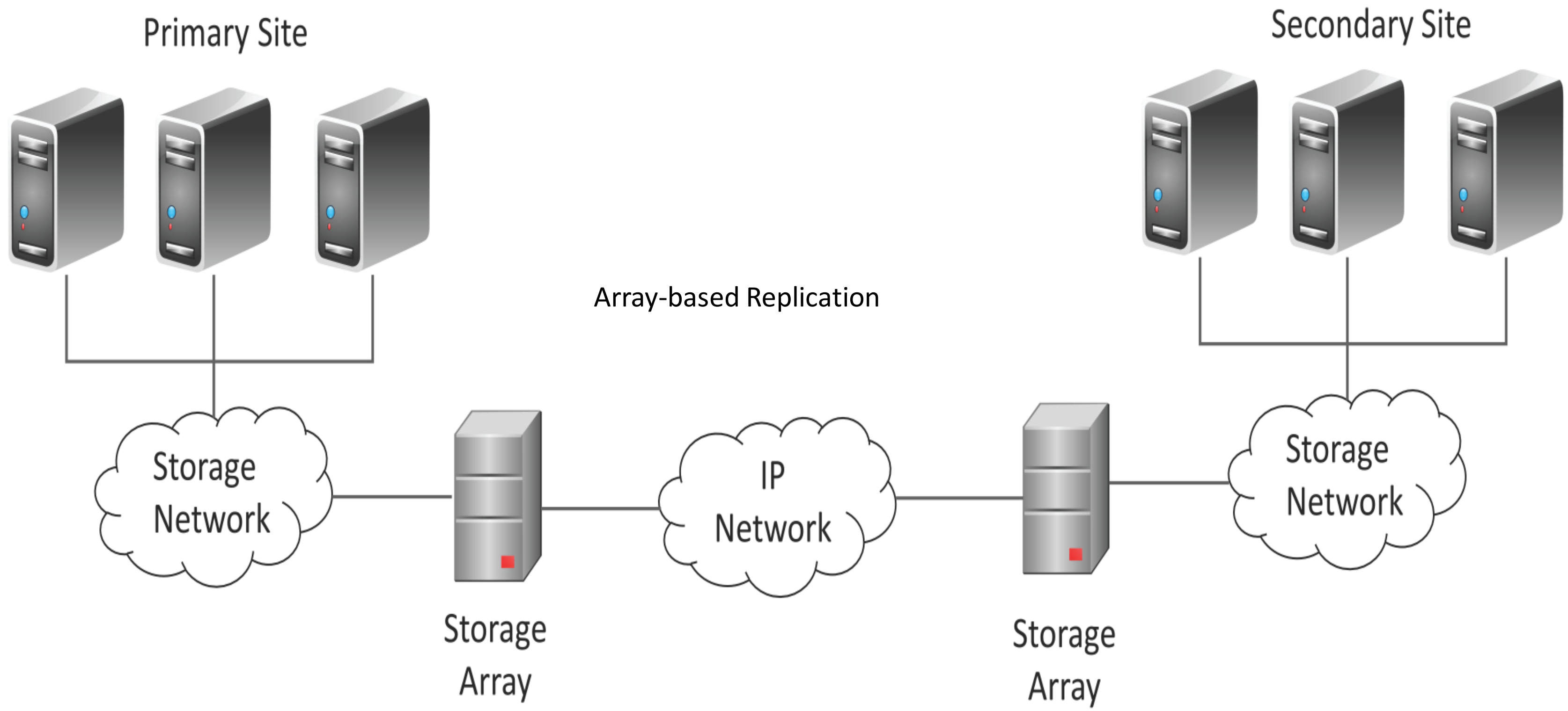
- Performance Evaluation
  - To verify whether the application meets the performance requirements with the deployment
  - Involves monitoring the workload on the application and measuring various workload parameters such as response time and throughput.
  - Utilization of servers (CPU, memory, disk, I/O, etc.) in each tier is also monitored.
- Deployment Refinement
  - Various alternatives can exist in this step such as vertical scaling (or scaling up), horizontal scaling (or scaling out), alternative server interconnections, alternative load balancing and replication strategies, for instance.

# Replication

- Replication is used to create and maintain multiple copies of the data in the cloud.
- Effective data replication strategies for disaster recovery in Cloud enables rapid implementation of replication solutions for disaster recovery for organizations.
- With cloud-based data replication, organizations can plan for disaster recovery without making any capital expenditures on purchasing, configuring and managing secondary site locations.

# Replication

- Types: Array-based Replication
  - Network-based Replication
  - Host-based Replication



# Array-based Replication

- Array-based replication is an approach to data backup in which compatible storage arrays use built-in software to automatically copy data from one storage array to another.
- Array-based replication software runs on one or more storage controllers resident in disk storage systems, synchronously or asynchronously replicating data between similar storage array models at the logical unit number or volume block level.
- The term can refer to the creation of local copies of data within the same array as the source data, as well as the creation of remote copies in an array situated off site.

# Network-based Replication

- Network-based Replication usually replicate at the file system-level, which has the benefit of file system metadata awareness, which can be leveraged during the replication process and enables replication based on criteria such as file size and file type.
- It is slower and usually less efficient than block-based replication.
- The performance impact increases with the number of files and folders in a replication set that need to be parsed, and the larger the tree, the longer it takes to parse.
- It usually comes into play in heterogeneous storage environments. It'll work with anyone's array and supports any host platform.

# Host-based Replication

- Host-based replication is the practice of using servers to copy data from one site to another
- Host-based replication is conducted by software that resides on application servers and forwards data changes to another device.
- The process is usually file-based and asynchronous: The software traps write input/output (I/O) and then forward changes to replication targets.
- To enable efficient and secure data copying, host-based replication software products include capacities such as deduplication, compression, encryption, and throttling. Host-based replication can also provide server and application failover capability to aid in disaster recovery.

# Monitoring

- Monitoring services allow cloud users to collect and analyze the data on various monitoring metrics.

- A monitoring service collects data on various system and application metrics from the cloud computing instances.

- Monitoring of cloud resources is important because it allows the users to keep track of the health of applications and services deployed in the cloud.

# Monitoring

## Examples of Monitoring Metrics

| Type | Metrics |
|---|---|
| CPU | CPU-Usage, CPU-Idle |
| Disk | Disk-Usage, Bytes/sec (read/write), Operations/sec |
| Memory | Memory-Used, Memory-Free, Page-Cache |
| Interface | Packets/sec (incoming/outgoing),<br>Octets/sec(incoming/outgoing) |

# Software Defined Networking

- Software-Defined Networking (SDN) is a networking architecture that separates the control plane from the data plane and centralizes the network controller.
- Conventional network architecture
  - The control plane and data plane are coupled. Control plane is the part of the network that carries the signaling and routing message traffic while the data plane is the part of the network that carries the payload data traffic.
- SDN Architecture
  - The control and data planes are decoupled and the network controller is centralized.

# Why We need SDN?

1. Virtualization: Use network resource without worrying about where it is physically located, how much it is, how it is organized, etc.
2. Orchestration: Should be able to control and manage thousands of devices with one command.
3. Programmable: Should be able to change behavior on the fly.
4. Dynamic Scaling: Should be able to change size, quantity
5. Automation: To lower OpEx and minimize manual involvement
    - Troubleshooting
    - Reduce downtime
    - Policy enforcement
    - Provisioning/Re-provisioning/Segmentation of resources
    - Add new workloads, sites, devices, and resources

# Why We need SDN?

6. Visibility: Monitor resources, connectivity
7. Performance: Optimize network device utilization
    - Traffic engineering/Bandwidth management
    - Capacity optimization
    - Load balancing
    - High utilization
    - Fast failure handling
8. Multi-tenancy: Tenants need complete control over their addresses, topology, and routing, security

# Why We need SDN?

9. Service Integration: Load balancers, firewalls, Intrusion Detection Systems (IDS), provisioned on demand and placed appropriately on the traffic path

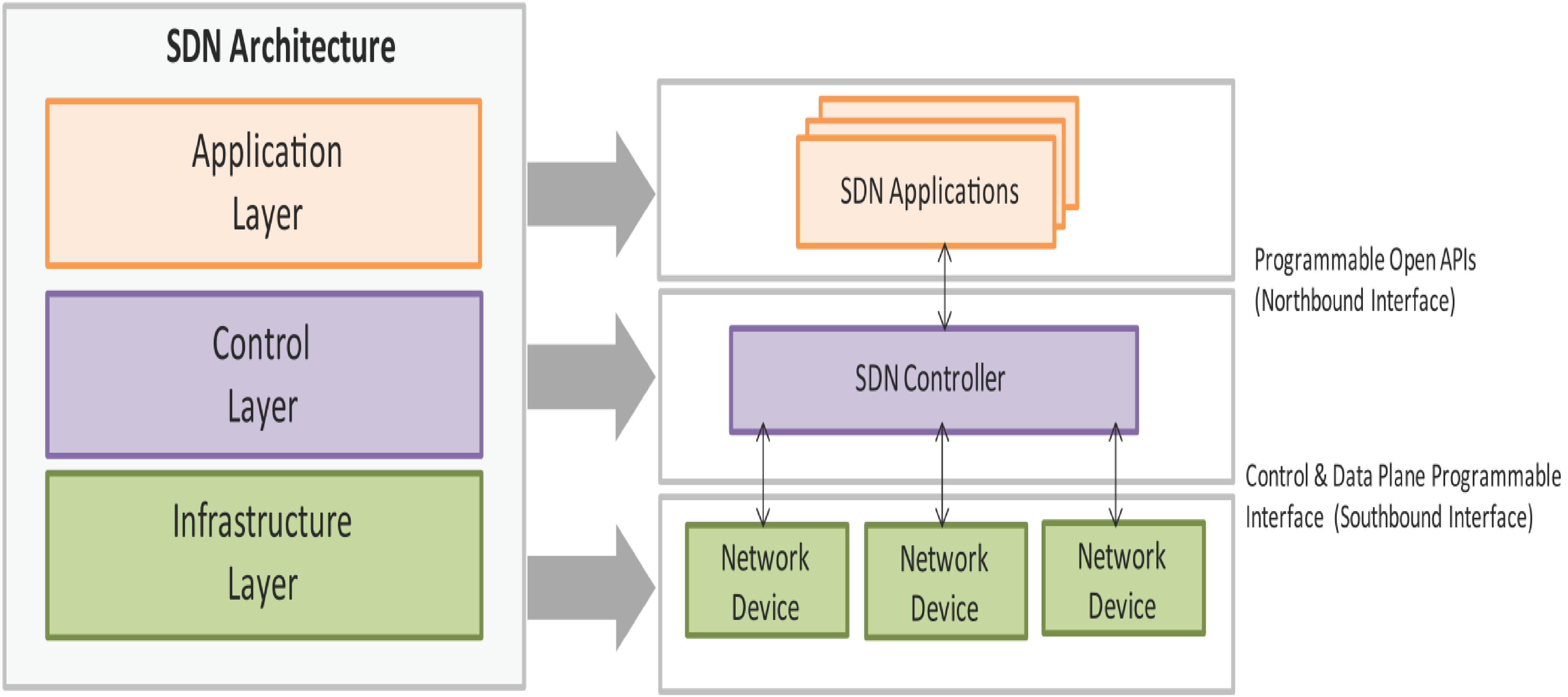
10. Openness: Full choice of "How" mechanisms
    - Modular plug-ins
    - Abstraction:
        - Hide the details.
        - Define tasks by APIs

# SDN Definition

- SDN is a framework to allow network administrators to automatically and dynamically manage and control a large number of network devices, services, topology, traffic paths, and packet handling (quality of service) policies using high-level languages and APIs.

- Management includes provisioning, operating, monitoring, optimizing, and managing faults, configuration, accounting, performance, and security in a multi-tenant environment.

# Software Defined Networking



# SDN - Key Elements

- Centralized Network Controller
  - With decoupled the control and data planes and centralized network controller, the network administrators can rapidly configure the network.
- Programmable Open APIs
  - SDN architecture supports programmable open APIs for interface between the SDN application and control layers (Northbound interface). These open APIs that allow implementing various network services such as routing, quality of service (QoS), access control, etc.

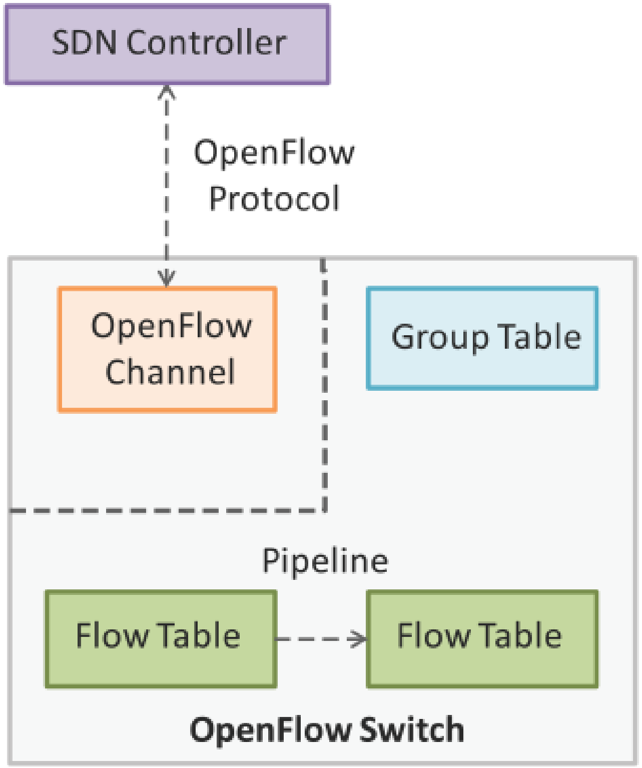
# SDN - Key Elements

- Standard Communication Interface (OpenFlow)
  - SDN architecture uses a standard communication interface between the control and infrastructure layers (Southbound interface).
  - OpenFlow, which is defined by the Open Networking Foundation (ONF) is the broadly accepted SDN protocol for the Southbound interface.

# OpenFlow

- OpenFlow is the broadly accepted SDN protocol for the Southbound interface.
- With OpenFlow, the forwarding plane of the network devices can be directly accessed and manipulated.
- OpenFlow uses the concept of flows to identify network traffic based on pre-defined match rules.
- Flows can be programmed statically or dynamically by the SDN control software.
- OpenFlow protocol is implemented on both sides of the interface between the controller and the network devices.

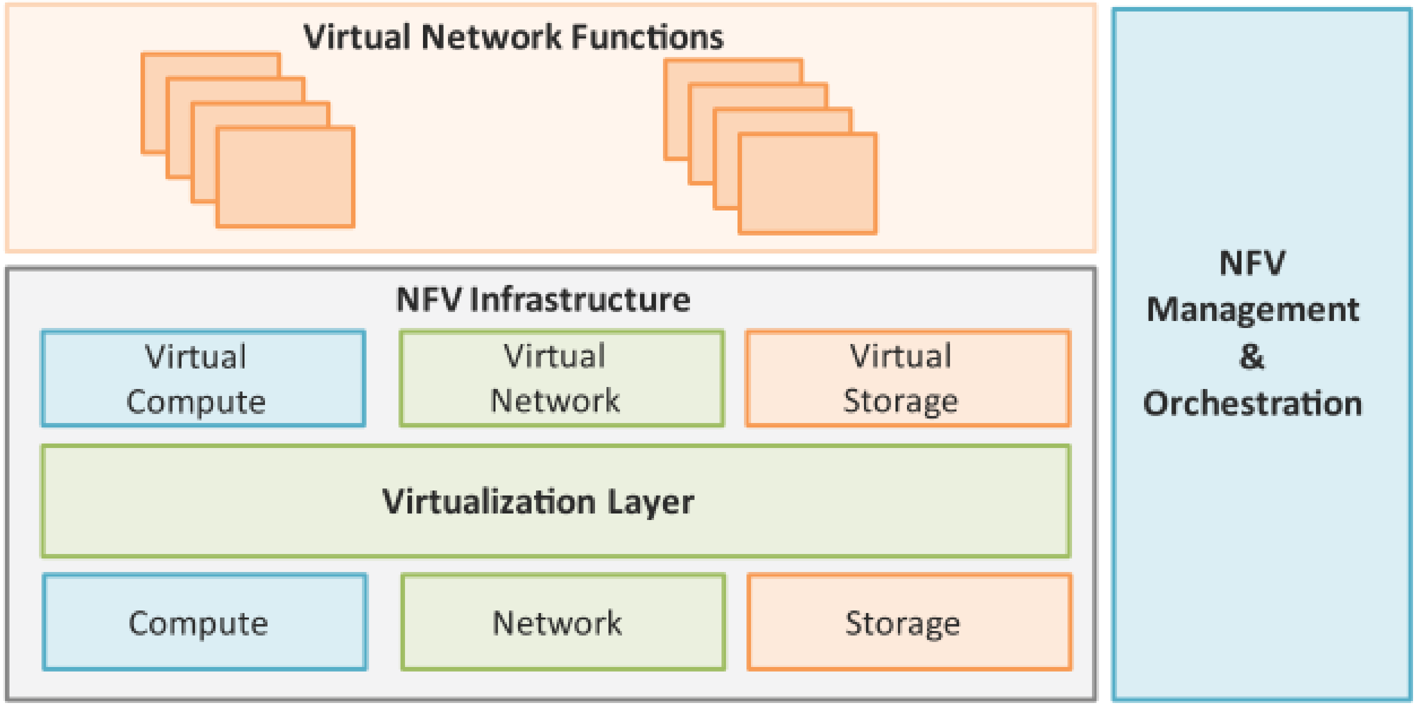
# OpenFlow



OpenFlow switch comprising of one or more flow tables and a group table, which perform packet lookups and forwarding from an OpenFlow channel to an external controller.

# Network Function Virtualization

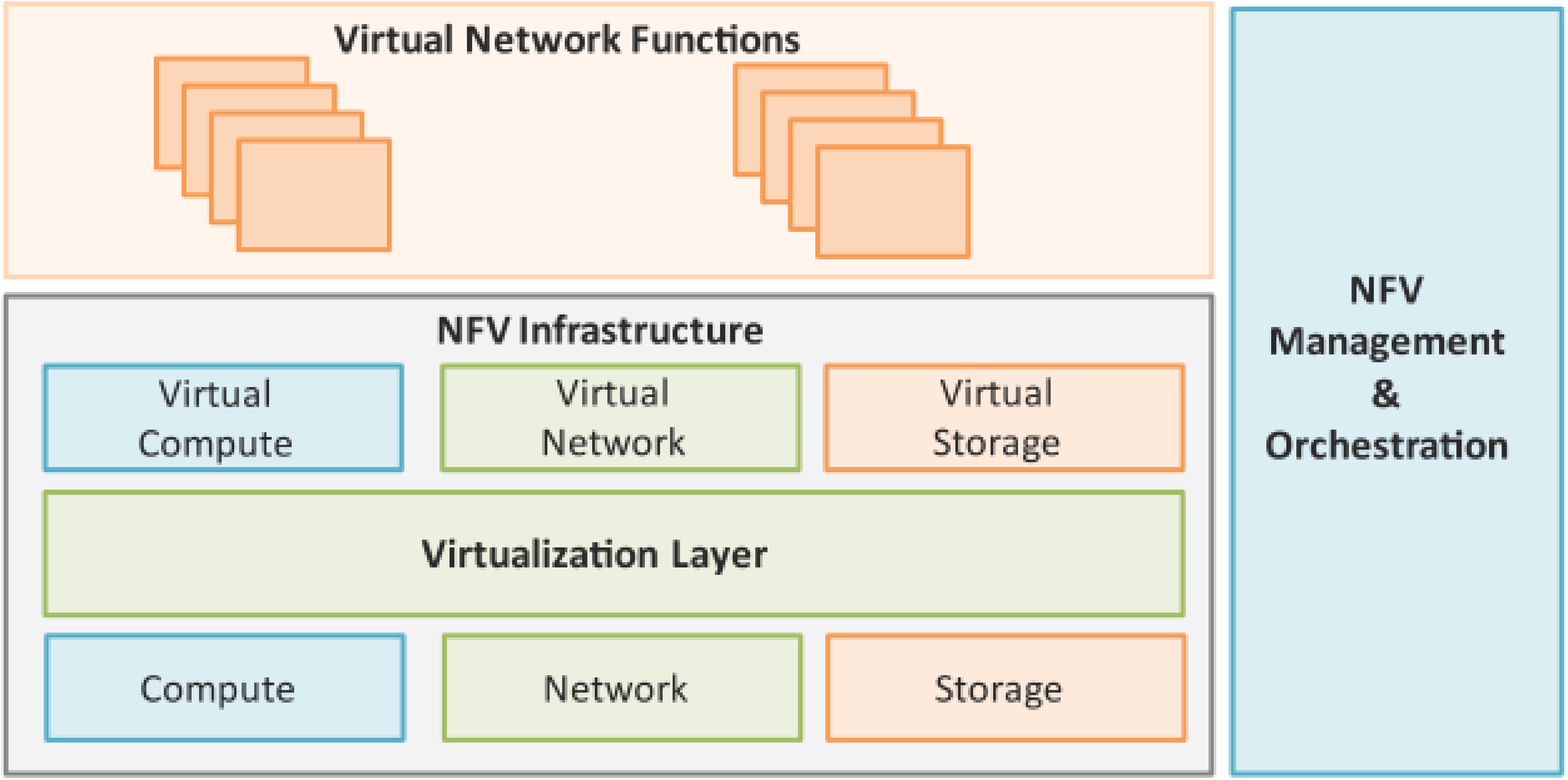
- Network Function Virtualization (NFV) is a technology that leverages virtualization to consolidate the heterogeneous network devices onto industry standard high volume servers, switches and storage.



# Network Function Virtualization

- Relationship to SDN
  - NFV is complementary to SDN as NFV can provide the infrastructure on which SDN can run.
  - NFV and SDN are mutually beneficial to each other but not dependent.
  - Network functions can be virtualized without SDN, similarly, SDN can run without NFV.
- NFV comprises of network functions implemented in software that run on virtualized resources in the cloud.
- NFV enables a separation the network functions which are implemented in software from the underlying hardware.

# NFV Architecture



Virtual Network Functions

NFV Infrastructure

Virtual Compute

Virtual Network

Virtual Storage

Virtualization Layer

Compute

Network

Storage

NFV Management & Orchestration

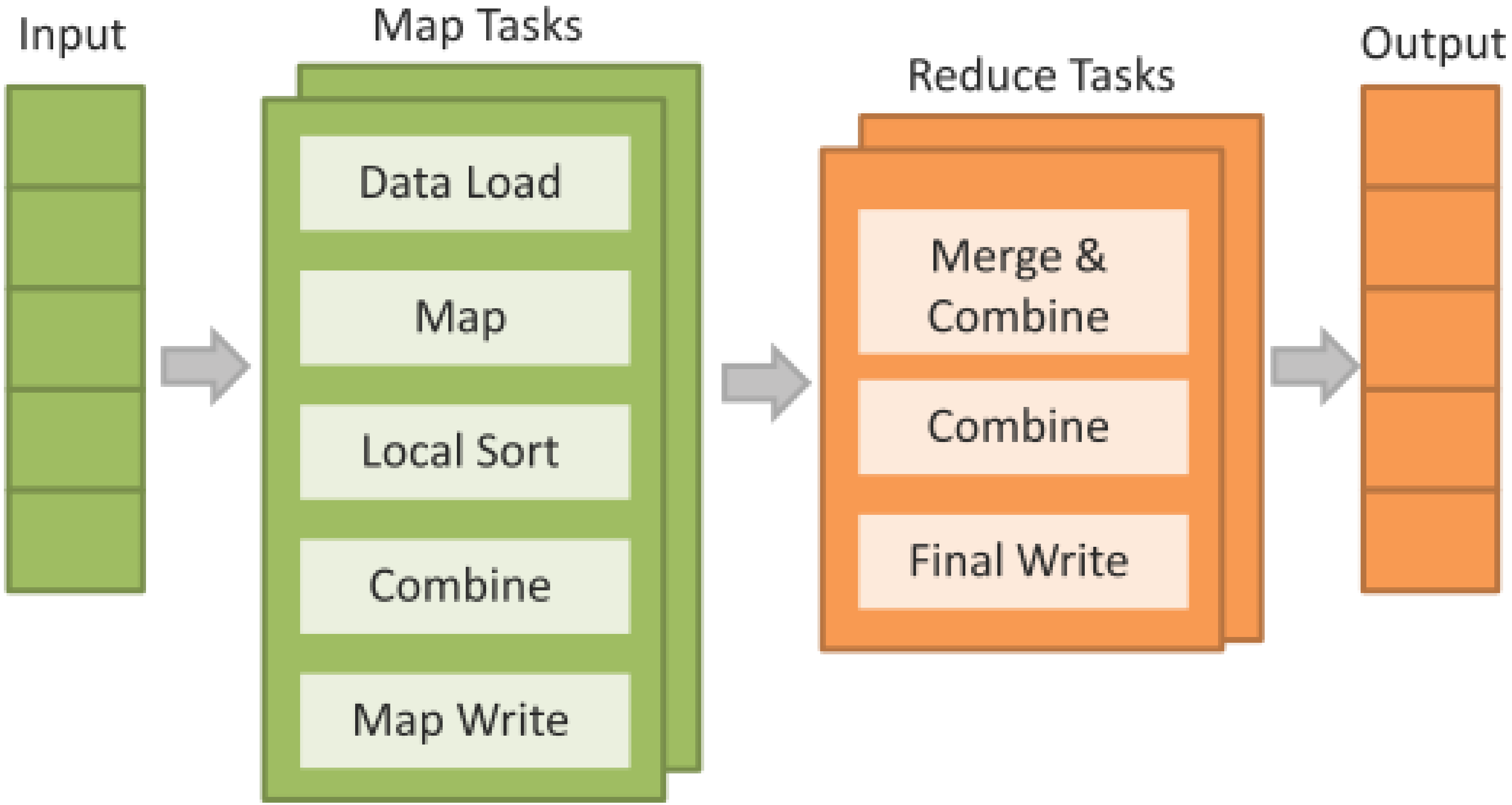
# NFV Architecture

- Key elements of the NFV architecture are
  - Virtualized Network Function (VNF): VNF is a software implementation of a network function which is capable of running over the NFV Infrastructure (NFVI).
  - NFV Infrastructure (NFVI): NFVI includes compute, network and storage resources that are virtualized.
  - NFV Management and Orchestration: NFV Management and Orchestration focuses on all virtualization-specific management tasks and covers the orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualization, and the lifecycle management of VNFs.

# MapReduce

- MapReduce is a parallel data processing model for processing and analysis of massive scale data.
- MapReduce phases:
    - Map Phase: In the Map phase, data is read from a distributed file system, partitioned among a set of computing nodes in the cluster, and sent to the nodes as a set of key-value pairs.
    - The Map tasks process the input records independently of each other and produce intermediate results as key-value pairs.
    - The intermediate results are stored on the local disk of the node running the Map task.
    - Reduce Phase: When all the Map tasks are completed, the Reduce phase begins in which the intermediate data with the same key is aggregated.

# MapReduce


Input
Map Tasks
Data Load
Map
Local Sort
Combine
Map Write
Reduce Tasks
Merge & Combine
Combine
Final Write
Output

# Identity and Access Management

- Identity and Access Management (IDAM) for cloud describes the authentication and authorization of users to provide secure access to cloud resources.
- Organizations with multiple users can use IDAM services provided by the cloud service provider for management of user identifiers and user permissions.
- IDAM services allow organizations to centrally manage users, access permissions, security credentials and access keys.

# Identity and Access Management

- Organizations can enable role-based access control to cloud resources and applications using the IDAM services.
- IDAM services allow creation of user groups where all the users in a group have the same access permissions.
- Identity and Access Management is enabled by a number of technologies such as OpenAuth, Role-based Access Control (RBAC), Digital Identities, Security Tokens, Identity Providers, etc.

# Service Level Agreements - SLA

- A SLA is a contract between a service provider and its internal or external customers that documents what services the provider will furnish.
- SLAs are output-based, its purpose is specifically to define what the customer will receive and defines the level of service expected from the service provider.
- SLAs measure the service provider’s performance and quality in a number of ways.

# Service Level Agreements - SLA

- Some metrics that SLAs may specify include:
  - Availability and uptime -- the percentage of the time services will be available
  - The number of concurrent users that can be served
  - Specific performance benchmarks to which actual performance will be periodically compared

# Service Level Agreements - SLA

- Application response time
- The schedule for notification in advance of network changes that may affect users
- Help desk response time for various classes of problems
- Usage statistics that will be provided.

# Service Level Agreements - SLA

- In addition to establishing performance metrics, an SLA may include a plan for addressing downtime and documentation for how the service provider will compensate customers in the event of a contract breach.
- Specify a range of enforceable consequences, such as penalties, for non-compliance with SLA performance measures.
- SLAs, once established, should be periodically reviewed and updated to reflect changes in technology and the impact of any new regulatory directives.

# What does typical SLA document consists of:

- An introduction to the SLA, what does this agreement propose
- A Service description, what service this SLA supports and details of the service
- Mutual responsibilities, who’s responsible for what part of the service
- Scope of SLA
- Applicable service hours, from what time till what time is the service available according to the agreement

# What does typical SLA document consists of:

- Service availability, how much is the service available during the service window and outside of service window
- Reliability
- Customer support arrangements
- Contact points and escalation; a communication matrix
- Service performance
- Security
- Costs and charging method used

# Billing

- Cloud service providers offer a number of billing models described as follows:
- Elastic Pricing
  - In elastic pricing or pay-as-you-use pricing model, the customers are charged based on the usage of cloud resources.
- Fixed Pricing
  - In fixed pricing models, customers are charged a fixed amount per month for the cloud resources.
- Spot Pricing
  - Spot pricing models offer variable pricing for cloud resources which is driven by market demand.

# Further Reading

- Network Functions Virtualization, http://www.etsi.org/technologies-clusters/technologies/nfv
- OpenFlow Switch Specification, https://www.opennetworking.org
- J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, OSDI 2004
- VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assist, 2007