Module 8
Part 2
Architectures for the Cloud -2

Harvinder S Jabbal

SE ZG651/ SS ZG653 Software Architectures

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Architectures for the Cloud - 2

Session Outline

- Base Mechanisms
- Sample Technologies
- Architecting in a Cloud Environment
- Summary

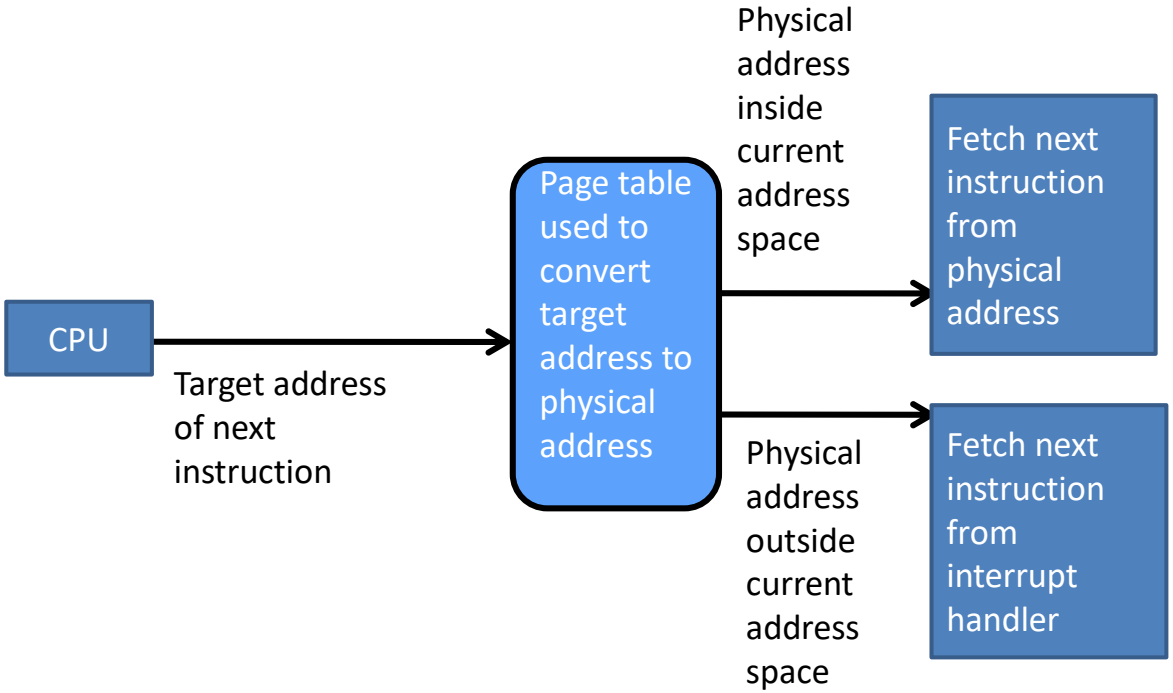
Basic Mechanisms

- Hypervisor
- Virtual Machine
- File system
- Network

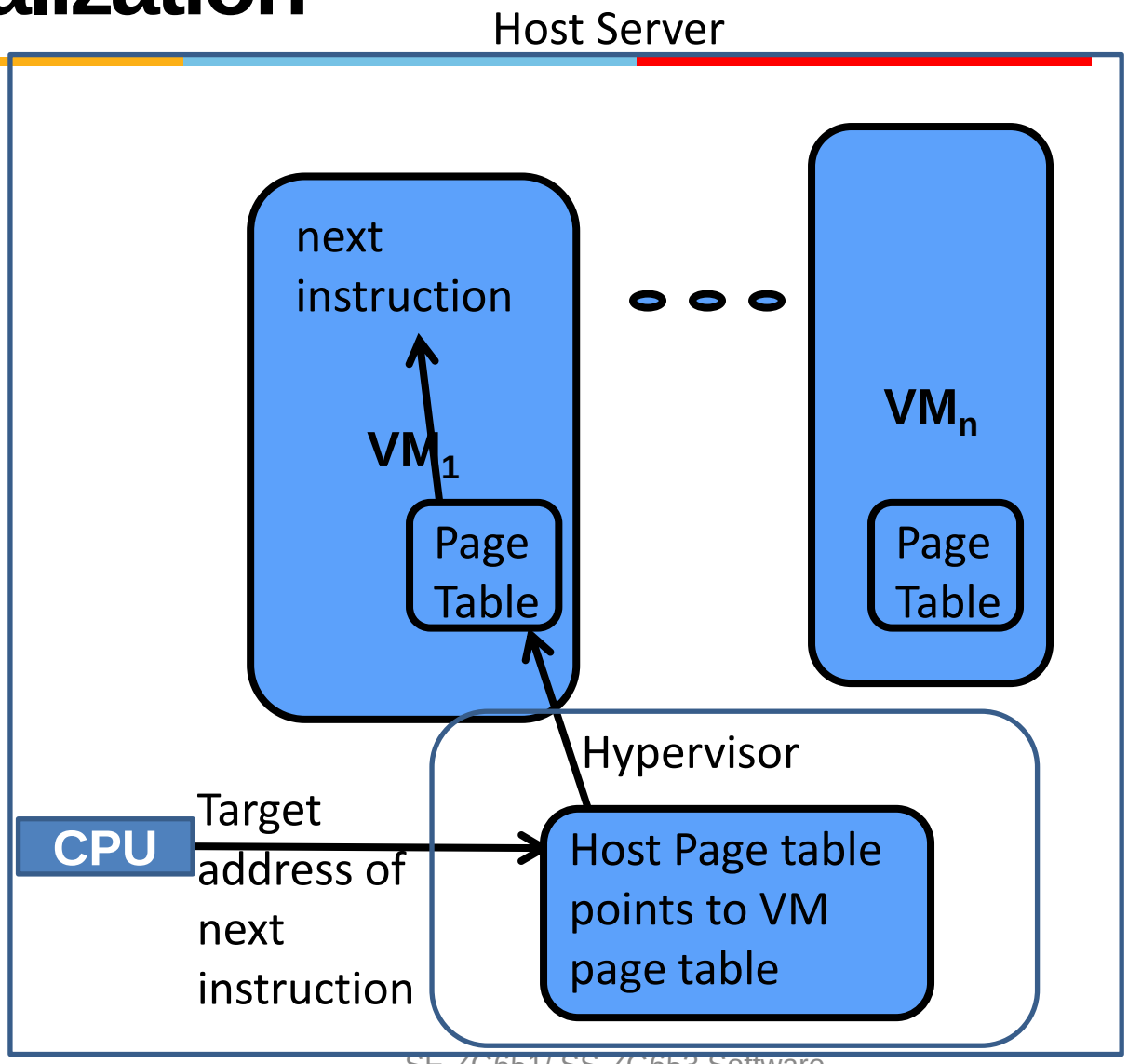
Virtual Memory Page Table

Virtual Memory Page Table

Virtual memory for non-virtualized application



Hypervisor Manages Virtualization



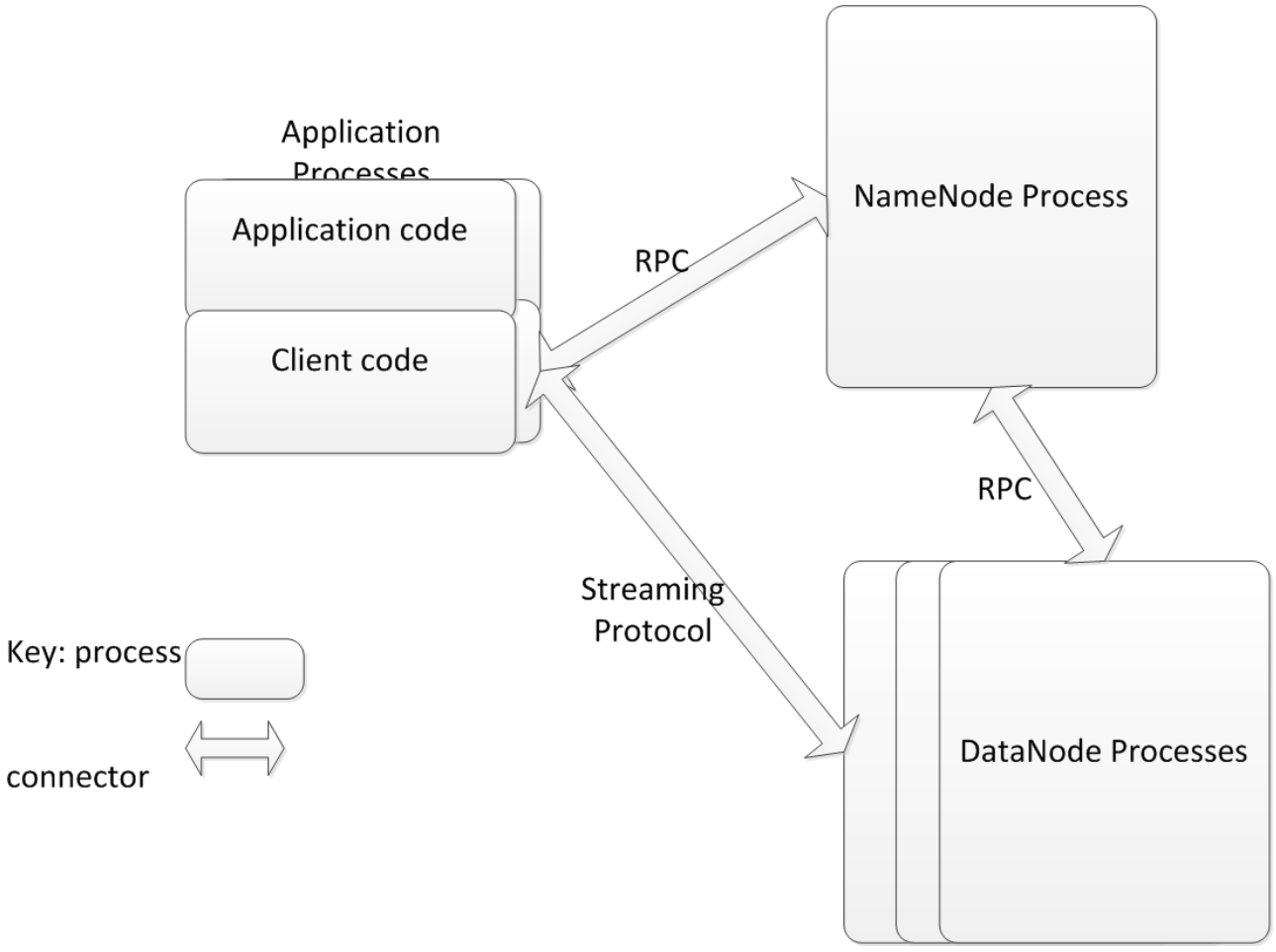
Virtual Machine

- A virtual machine has an address space isolated from any other virtual machine.
- Looks like a bare metal machine from the application perspective.
- Assigned an IP address and has network capability.
- Can be loaded with any operating system or applications that can execute on the processor of the host machine.

File System

- Each virtual machine has access to a file system.
- We will present HDFS (Hadoop Distributed File System) – a widely used open source cloud file system.
- We describe how HDFS uses redundancy to ensure availability.

HDFS Components



HDFS Write – Sunny Day Scenario

- Application writes as to any file system
- Client buffers until it gets 64K block
- Client informs NameNode it wishes to write a new block
- NameNode returns list of three DataNodes to hold block
- Client sends block to first DataNode and informs DataNode of other two replicas.
- First DataNode writes block and sends it to second DataNode. Second DataNode writes block and sends it to last DataNode.
- Each DataNode reports to client when it has completed its write
- Client commits write to NameNode when it has heard from all three DataNodes.

HDFS Write – Failure Cases

- Client fails
  - Application detects and retries
  - Write is not complete until committed by Client
- NameNode fails
  - Backup NameNode takes over
  - Log file maintained to avoid losing information
  - DataNodes maintain true list of which blocks they each have
  - Client detects and retries
- DataNode fails
  - Client (or earlier DataNode in pipeline) detects and asks NameNode for different DataNode.
- Since each block is replicated three times, a failure in a DataNode does not lose any data.

Network

- Every Virtual Machine is assigned an IP address.
- Every message using TCP/IP includes IP address in header.
- Gateway for cloud can adjust IP address for various purposes.

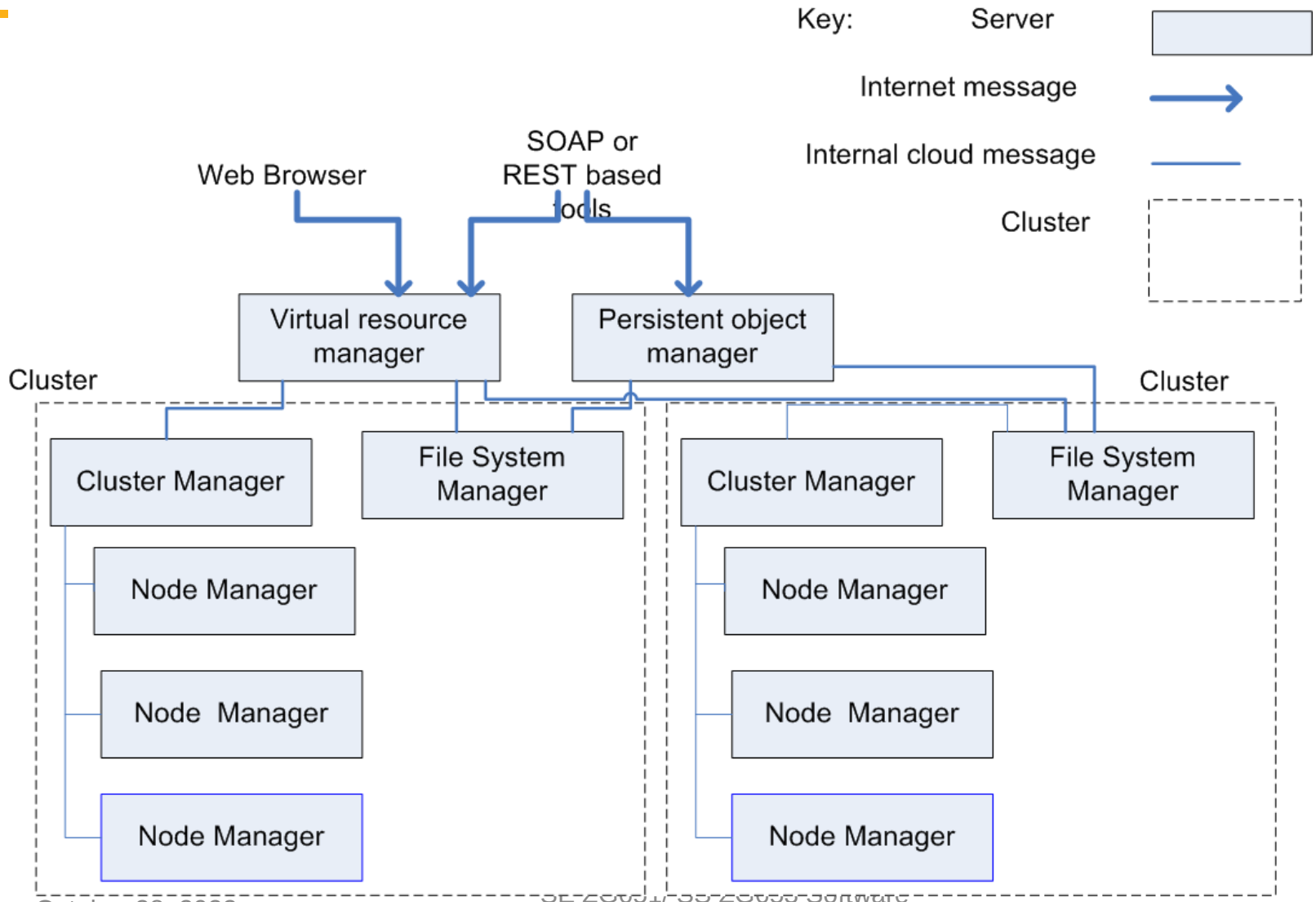
Sample Technologies

- IaaS
- PaaS
- DataBases

IaaS

- An arrangement of servers that manages the base technologies.
  - Servers are arranged in clusters
  - May be thousands of servers in a cluster
  - Some servers are used as the infrastructure of the IaaS
  - Every server has a hypervisor as its base.

IaaS Architecture



IaaS Architecture Components

innovate achieve lead

- Cluster Manager responsible for managing each cluster
- Persistent Object Manager manages persistence
- Virtual Resource Manager manages other resources. It acts as a gateway for messages.
- The File System Manager is similar to HDFS. It manages the network wide file system.

Services Provided by IaaS

- Automatic reallocation of IP addresses in the case of a failure of the underlying virtual machine instance.
- Automatic Scaling. Create or delete new virtual machines depending on load.

PaaS

- Provides an integrated stack for developer.
- E.g. LAMP stack
  - Linux, Apache, MySQL, Python
- The developer writes code in Python and the PaaS manages assignment to underlying layers of the stack.

Databases

- Why relational databases came into question
  - Massive amounts of data are collected from web systems. Much of this data is processed sequentially and so RDBMSs introduce overhead, especially during creation and maintenance.
  - The CAP Theorem shows that it is not possible to simultaneously achieve consistency, availability, and partitioning.
  - The relational model is not the best model for some applications.
- Caused the introduction of new data models
  - Key-value
  - Document centric

Key Value – HBase

- One column designated as a key. The others are all values
- No schema so data can have key + any other values. The values are identified by their variable name.
- Data values are also time stamped
  - Hbase does not support transactions. Time stamps are used to detect collisions after the fact.

Document Centric – MongoDB

Stores objects rather than data

Access data through containing object

Objects can also contain links to other objects

No concept of primary or secondary index. A field is indexed or it is not.

What is Omitted From These DBs

Transactions. No locking is performed. The application must detect interference with other users.

Schemas. No predefined schemas. The application must use correct name.

Consistency. The CAP theorem says something must give. Usually consistency is replaced by “eventual consistency”

Normalization and Joins. Performing a join requires that the join field is indexed. Because there is not a guaranteed index field, joins cannot be performed. This means normalization of tables is not supported.

Architecting in a Cloud Environment

innovate achieve lead

- Quality attributes that are different in a cloud
  - Security
  - Performance
  - Availability

Security

- Multi-tenancy introduces additional concerns over non-cloud environments.
  - Inadvertent information sharing. Possible that information may be shared because of shared use of resources. E.g. information on a disk may remain if the disk is reallocated.
  - A virtual machine “escape”. One user can break the hypervisor. So far, purely academic.
  - Side channel attacks. One user can detect information through monitoring cache, for example. Again, so far, purely academic.
  - Denial of Service attacks. One users can consume resources and deny them to other users.
- Organizations need to consider risks when deciding what applications to host in the cloud.

Performance

- Auto-scaling provides additional performance when load grows.
  - Response time for new resources may not be adequate
  - Architects need to be aware of resource requirements for applications
    - Build that knowledge into the applications
    - May applications self aware so that they can be proactive with respect to resource needs.

Availability

- Failure is a common occurrence in the cloud
  - With 1000s of servers, failure is to be expected
- Cloud providers ensure that the cloud itself will remain available with some notable exceptions.
- Application developers must assume instances will fail and build in detection and correction mechanisms in case of failure.

Summary

- The cloud provides a new platform for applications with some different characteristics.
- Architect needs to know how a cloud cluster works and pay special attention to
  - Security
  - Performance
  - Availability