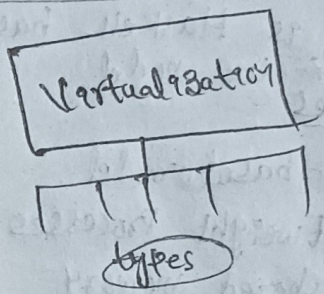
UNIT-3

- Virtualization
- Programming models of cc
  - Map Reduce
  - Cloud Haskell
- s/w development in cloud



① Hardware Vn:

→ multiple VMs run on one physical machine

→ uses hypervisor

ex: VMware, KVM

② OS Vn:

→ Many containers run on one OS

ex: Docker.

③ Seaver Vn:

→ One physical seaver is divided into many virtual servers.

ex: VMware

④ Application Vn:

→ Runs applications without installing.

⑤ Memory Vn:

→ combines physical memory & allocates it to VMs

② **Virtualization** → a technology

→ V^n is the technique of creating virtual resources (VMs, virtual storage, virtual n/ws) - instead of using physical h/w.

Benefits

- cost reduction
- Easy backup & recovery
- Scalability
- Isolation
- Quick deployment
- Platform independence

Drawbacks

- Higher initial setup cost
- single point of failure (If host fails → all VMs fail)
- Security risks
- Complex to Manage

→ Vn has completely transformed the IT industry
- by changing the way organization use h/w, s/w, n/w & data centers.

1. Shift from physical servers to Virtual servers:
Before Vn: one appn ran on one server
After Vn: Many VMs run on a single server
2. Huge cost Reduction:
- Vn reduced: h/w cost, electricity usage
3. Enabled CC
4. Faster Deployment
Earlier: Setting up a server took days or weeks
Now: creating a VM takes seconds or min
4. Disaster Recovery
5. Increased Flexibility & Agility
6. Better Security
7. DevOps & Automation

⇒ Vn is the process of creating a virtual version of h/w, OS, storage
- instead of using physical components.

→ It allows multiple VMs
- Scalability
- Flexibility

CC Mid-II

1). What is the role of hypervisor in virtualization? Briefly explain the different types of hypervisors with a neat diagram.

Hypervisor

→ A hypervisor is a s/w layer that creates and manages Virtual Machines (VMs).

→ It allows multiple OS to run on a single physical machine.

→ Hypervisor is a virtualization s/w
- It creates, runs & manage VM

→ A hypervisor is a Virtual Machine Monitor

→ It is a specialized s/w layer
- It separates the physical h/w from virtual environments.

→ It acts as a middle layer
- allocates h/w resources (CPU, memory, storage) to each VM.

→ VMs are widely used instead of physical machines in the IT industry today.

→ VMs support green IT solutions.

Role of Hypervisor

1. VM Creation & Mgmt – Creates, runs, pauses & delete VMs
2. Resource Allocation – distributes CPU, memory, storage to each VM
3. Isolation – ensures one VM does not affect another VM
4. H/w Virtualization – makes h/w appear as virtual h/w to VMs

5. Live Migration - Moves running VMs from one host to another without downtime.

6. Performance monitoring

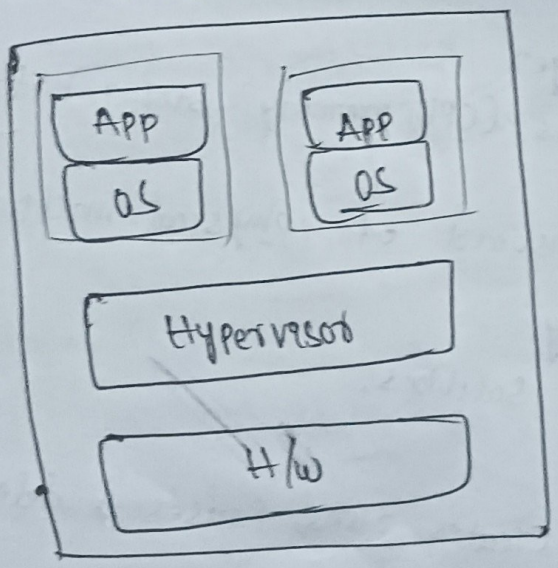
7. High Availability

Types

Type-1 (Bare Metal)

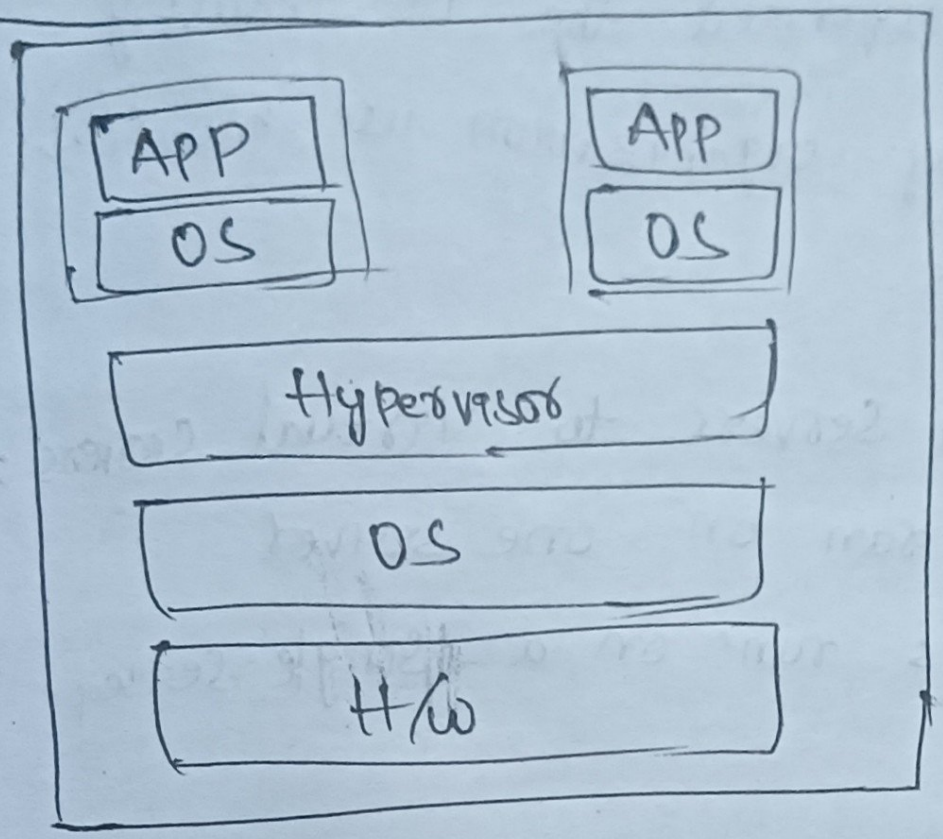
→ Runs directly on h/w
→ no host OS
→ ex: Xen
→ High availability
→ Used in data centers & cloud environments
→ Supports clustering.

Use cases
- AWS EC2
- Azure VMs



- Runs on h/w
- High Performance
- Used in data centers
- More secure

Type-2 (Hosted)



→ Runs on existing OS → Runs on OS

→ Easier to use

→ less efficient than Type-1

→ lower performance

→ used in personal laptops

→ less secure

3). @. Mapreduce.

→ MapReduce is a parallel processing framework.

- Introduced by Google
- used on Hadoop
- used to process huge datasets
- It is a distributed data processing model.

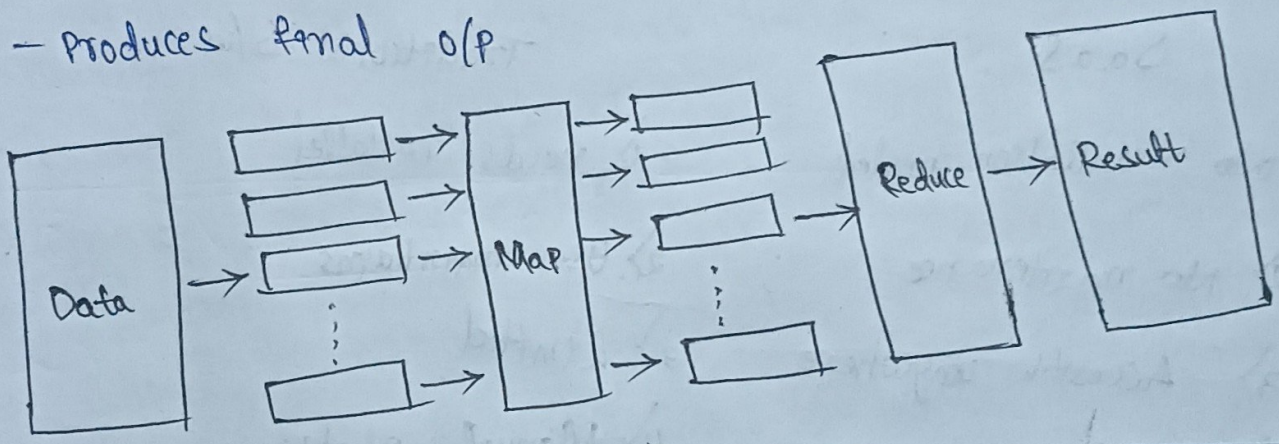
Two main phases

1. Map Phase Map(key1, val1) → List(key2, val2)

- Input is split into small chunks
- Mapper processes each chunk, produces key-value pairs.

2. Reduce phase Reduce(key2, List(val2)) → List(val3)

- Reducer collects all values of same key
- Produces final o/p



MapReduce	CGL-MapReduce
1). Batch oriented	1). Streaming oriented
2). used on Hadoop	2). used on cloud environments
3). High latency	3). low latency
4). Not suitable for real-time	4). Suitable for real-time
5). Limited Scalable	5). Highly Scalable
6). Disk-based execution	6). Memory-based execution

ex: Map: ("apple", 1), ("banana", 1), ("apple", 1)

Reduce: ("apple", 2), ("banana", 1)

Mapreduce

Adv

- Easy
- Scalable
- Fault detection

Dis

- Expensive
- Inefficient
- High latency

→ Developed by Google
↳ used in Hadoop

Programming Models for CC?

→ It define how applications are developed, deployed, executed on cloud.

Cloud Haskell

→ It is a domain-independent.
→ It is distributed programming framework.
- that brings Erlang-style concurrency, message passing, & fault tolerance to
→ It is Haskell-based programming framework.
→ Actor model.

Features

→ Actor-based model
→ Lightweight Processes
→ No shared memory
→ Fault detection
→ Fault monitoring

→ Haskell-based.
→ domain-independent.

~~Architecture~~
- Nodes
- ~~Transport layer~~

Adv
- Scalabale
- Safety
- reliability

* S/w Development on cloud

→ It uses cloud Infrastructure, cloud IDEs, CI/CD Pipelines.

→ Inorder to build, test, deploy & delever applications uses cloud.

Stages

- Requirements stage
- Design stage
- Development stage
- Testing stage
- Deployment stage
- Monitoring stage

⇒ S/w development on cloud means:
- building, testing, deploying & maintaining applications using cloud tools.

⇒ Instead of installing tools on your computer, developers use:
- cloud servers
- cloud storage
- cloud IDEs
- cloud infrastructure.

→ Everything done using cloud