1/7/24

Cloud computing:

what is cloud computing:

Delivery of computing service including servers, storage, networking, software, analytics and intelligence - over the Internet to offer faster innovation, flexible resources

Services in cloud computing:

Software
Platform
Infrastructure
} → major 3 services

Definition: Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common internet protocol and networking standards.

The word 'cloud' refers to two essential concepts:

1. abstraction — eg. Gdocs → editor, viewer access
2. Virtualization → virtualize the system by pooling & sharing
   - Physically present and not virtually

By NIST definition - cloud computing is a model for enabling ubiquitous, convinient on demand network access to a shared pool of configurable computing resources

eg: (networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal

Management effort, or, service provider interaction.

11/4/24

Characteristics of cloud computing → comparison study
Benefits of cloud computing

1) On Demand Self-services:

No need of communication, just a server request is enough

2) Broad-Network access:

Accessing through various devices in any location

3) Resource ~~Pooling~~ Pooling:

Gathering the needed resource and sharing when needed

4) Rapid elasticity / scaling:

Vertical scaling / scaling up → capacity increasing
Horizontal / scaling down → using a

scaleout → Resources are increased

extending 50MB to 512MB

excess storage over for certain period then giving it back

5) Measured ~~Service~~ service:

Amount will be charged according to the usage

Measurement (no. of TPU's, GPU's, servers)

6) Performance:

Dynamic allocation of workload → loadbalancing
eg: round robin

7) Reduced cost / cost effective:

Scaling up/down facility makes it cost effective

8) Outsource management:

Managing the server

9) Ease of Utilization:

Easy access from anywhere and any device

10) Reliability:

Through, because one server is crashed, they will have a copy in other server
loadbalancing
No attacks, avoid crashing mostly

1) Multi-tendency:

The Resources is shared by many people

↓

The two/more resources is joined by pooling of resources.

Two types

Virtual Multi-tendency - computing, storage eg: google chrome, docs

Organic " " - server, networking, OS, DB's

Types of Clouds / Types of Deployment Model

Types of Cloud

Public cloud - eg: MS Azure

Private cloud - eg: CISCO, VMware, Dell

Community cloud - eg: salesforce, Commvault, govt owned clouds

Hybrid cloud - eg: Azure arc, Azure stack, Google Anthos

private → within organization

public → open to all, pay as per according to usage

hybrid → public+private, eg: Amazon outposts & salesforce

community → Multiple organization, govt owned clouds.

eg: Thiagarajar Engg, Arts, Management, school, Mill

↓

single community

Two Distinct Models of Cloud Computing:

* Service Model → something as a service
* Deployment Model - (Public, Private, Community, Hybrid clouds)

Service Model Types:

SaaS → Software as a service → No License issue → eg: Google docs, sheets,

PaaS → Platform as a service → eg: Android studio, Zoho creator

IaaS → Infrastructure as a service → eg: VMware

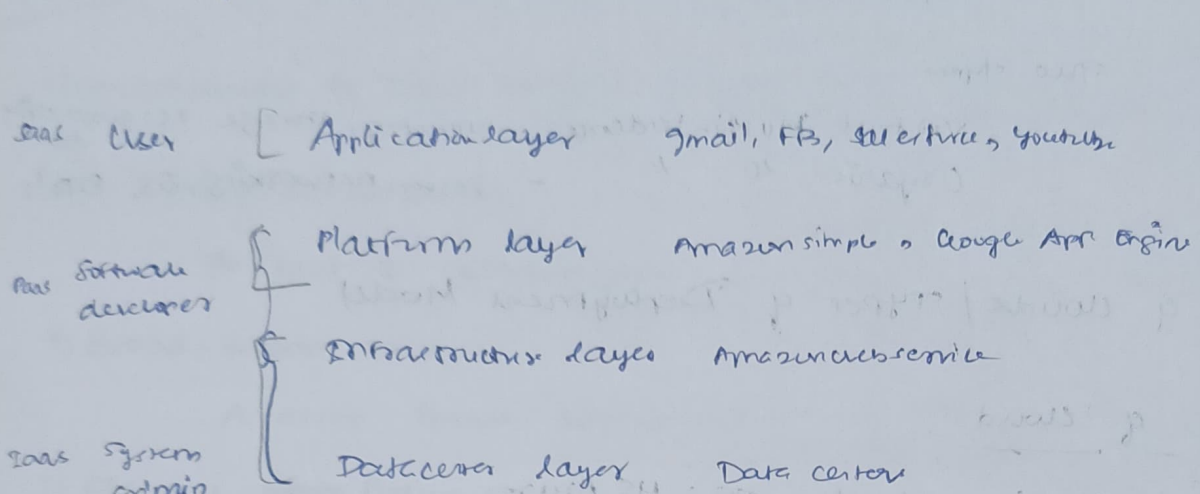
↓ ↳ managed by you/Third party

TPU, GPUs, OS.

Customer relationship Management (CRM)

12/4/24

3 Layer of cloud computing



Application layer:

Top layer

Automatic - scaling, Fault tolerance, availability and low operational costs

In charge of application co-operations & communication

Responsible for processing IP traffic handling protocol like Telnet and FTP.

Eg: web browser, SNMP protocol,

Platform Layer:

Middle layer, top of infrastructural layer

The operating system and application software

It provides user a platform to develop their software with scalability, dependability and security protection which gives user a space to create their apps, test operational process and keep track of execution outcomes & performance

Objective → to deploy applications directly on virtual machines → to lessen the difficulty of deploying

Eg: Google App engine

Infrastructure Layer

Foundation Layer / Bottom Layer

Layer of Virtualization - physical resources → into collection of virtual resources

serves as a central Hub of the cloud environment

Automated resource provisioning

Refered as virtualization layer

Datacenter Layer:

Managing Physical Resources

Resources to be available & managed in datacenter.

Concepts & Terminologies

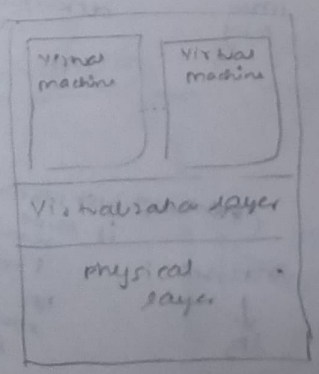
Virtualization — Monitoring

Load Balancing

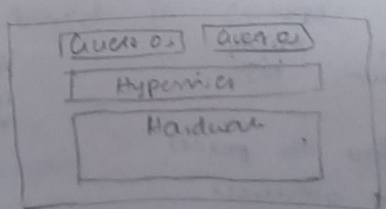
scalability & Elasticity

Deployment

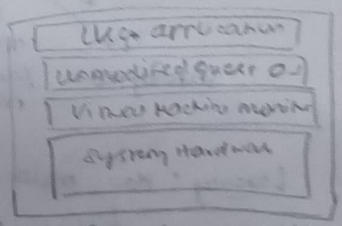
Replication



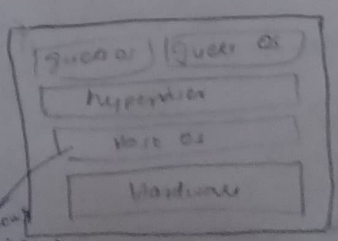
Virtualization Architecture



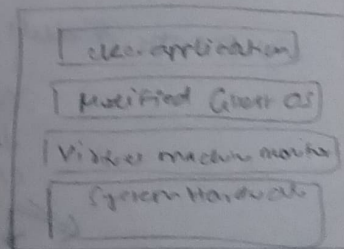
Hypervisor - Type-1



Full virtualization



Hypervisor - Type-2
eg: VirtualBox



Para virtualization

Types of virtualization

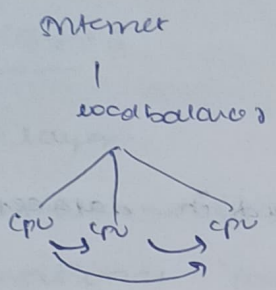
Full virtualization (unmodified OS)
Para virtualization (slightly modified)
Hardware virtualization (resources are virtualized)

15/7/24

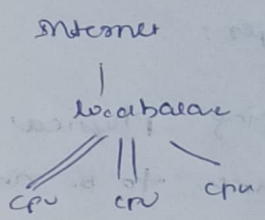
Load Balancing :-

Cloud has 8 techniques for load balancing

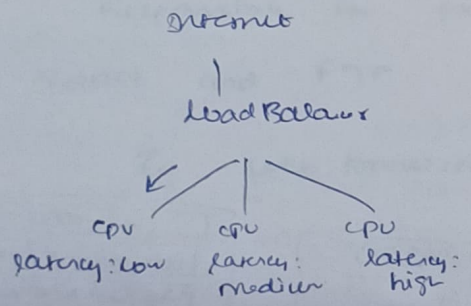
a) Round robin



b) Weighted round robin (combination of priority scheduling & round robin)

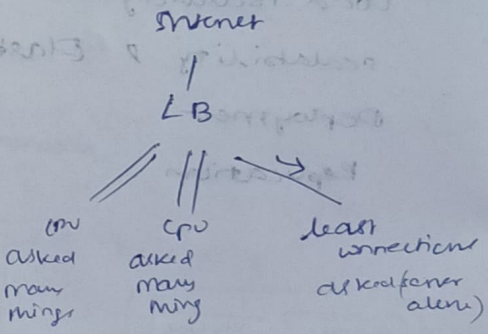


c) ~~Latency~~ Low latency Load balancing

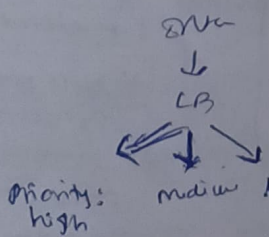


Latency : How much time the process holds ~~this~~ resources

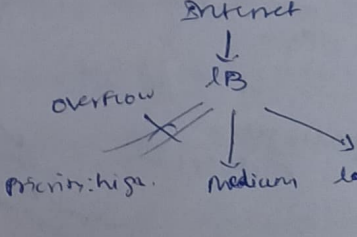
d) Least connection



e) Priority scheduling



f) Overflow (even though high priority if large resources is required not attending him)



g) sticky session
h) session database
i) Browser cookies
j) URL rewriting

→ session and cookies
↳ It handles load balancing & fail over mechanism in these.

Sticky session

takes your session to m server
& session over whole mechanism fails
∴ can't handle failover mechanism

session database

sessions are stored in DB
∴ can handle fail-over mechanism

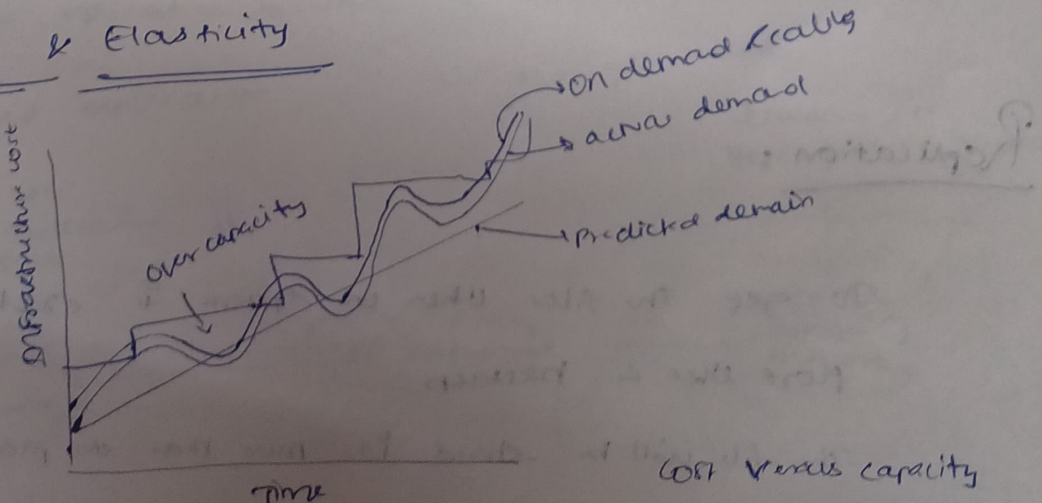
URL rewriting:

URL keeps on changing
when you using a tpu, one again you're running that program on another tpu was assigned. will be assigned because of URL rewriting

Browser cookies:

session is stored in cookies

Scalability & Elasticity



Cost Versus capacity curves

Because of Traditional scale up/scale out techs

It sometimes causes over capacity / under capacity

∴ Using on demand, which adjusts according to demand

Deployment

Deployment design

No. of. tiers

No. of. servers in each tier

compute capacities, memory & storage of servers

server interconnections

load balancing & replication strategies

Performance measurement

Application workload

utilization of servers.
(CPU, memory disk, I/O)

Deployment refinement

Horizontal Scaling

Vertical Scaling

Alternative server interconnections

Alternative load balancing & application strategies

Cloud application deployment life cycle

In CC:- we are getting whatever we need, that is called as deployment in cloud computing

Replication:

To get the files when one system is crashed

More like a backup

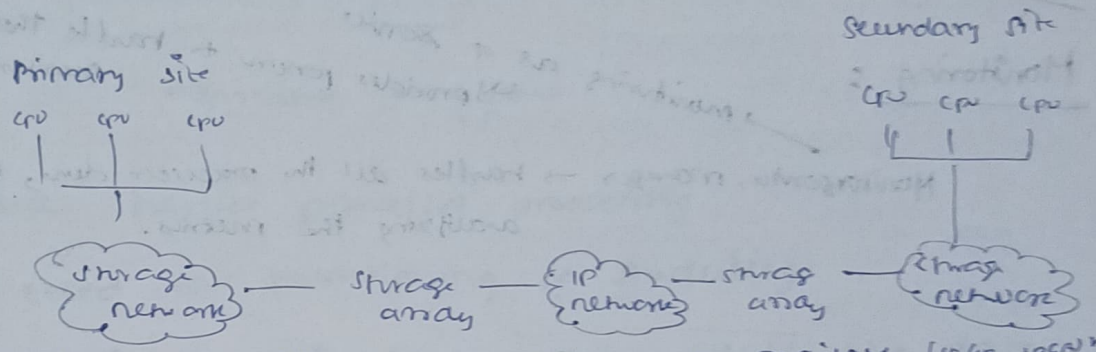
The file will be stored in more than one place.

Two types:

Storing all the files

storing the only have the files in case of very large size files.

a) Strategy 1 → array based replication

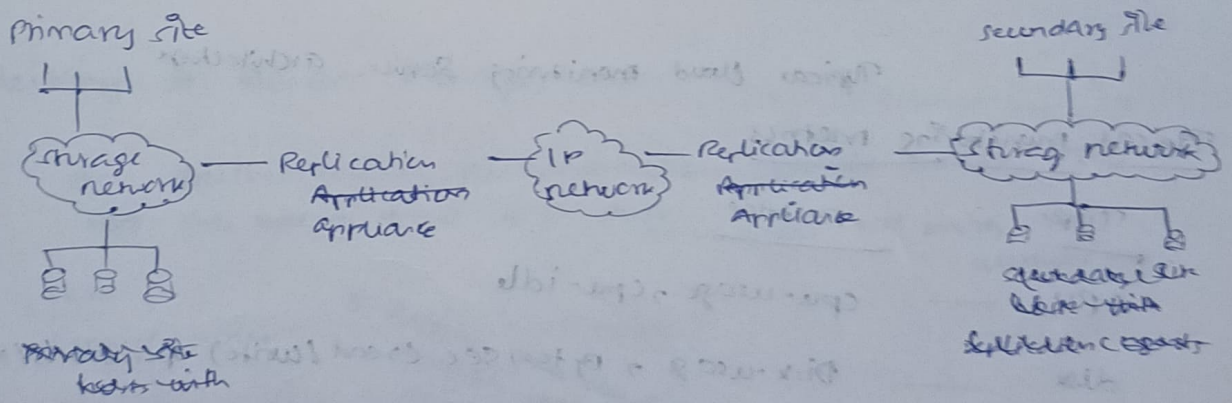


*sometimes same storage array may not be available [10 GB, 10GB ✓ (10GB, 5GB) ✗

b) Network based replication

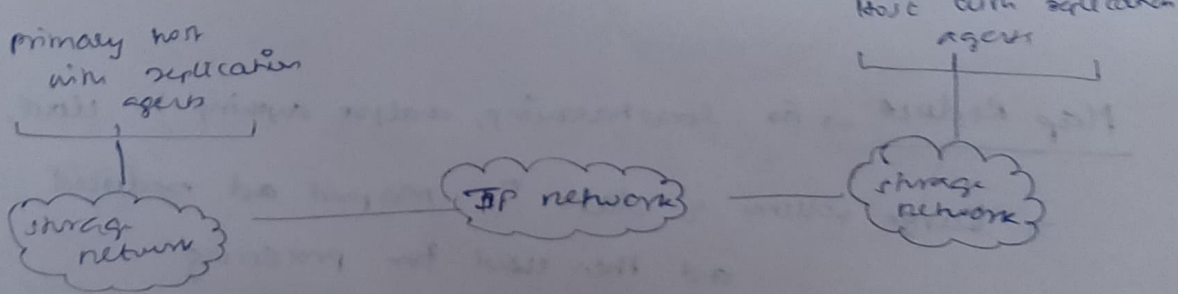
Data — segment - packets - frames - o/p

Replication appliance → converts as packets



* The replication appliance is bit costlier, so moving to next

c) Host based replication



Host → no of computers in TCP/IP

two techniques in Host-based replication:

Block based → entire file, entire data is replicated

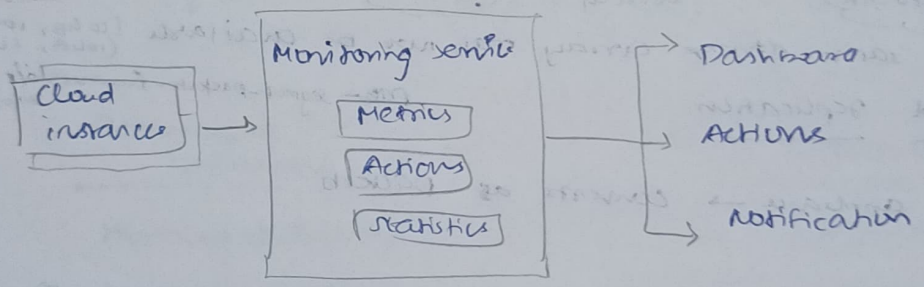
file based → can choose files of your own to be replicated

18/4/24

Monitoring :

→ monitoring as a service provides person to handle their monitoring

Monitorservice, manager → handles all the processes done, by analyzing the metrics.



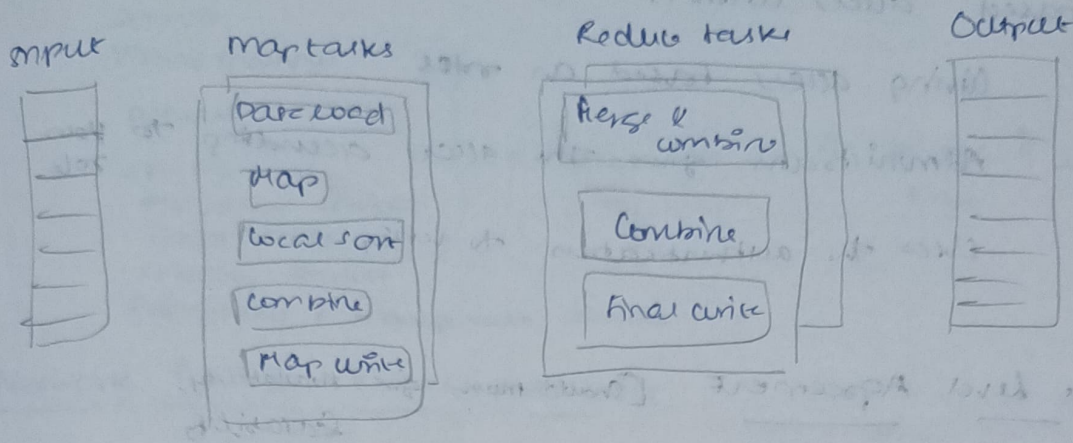
Typical cloud monitoring service architecture

Typical 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) octets/sec (incoming/outgoing)

Map Reduce → for load balancing, analyze anything in cloud, word count

→ Large volume of data is maped and reduced and then used for processing



Mapreduce workflow

* Mapreduce handles parallel processing
* Mostly used for partitioning & scheduling
* Load balancing, taking many resources, distributing data to anything
* will be in key-value pair eg: {'The' : 5}

Identity & Access Management

All authorization, authentication comes under this
Rob based access, O-Auth contains 3rd party for authorizing

User
Resource owner

3. Returns from cloud identity Provider asking resource owner to authenticate

Cloud Identity Provider

Third party

Rob based access control:

Giving access based on roles

Administrator gives the access according to your role

Uses the authentication to verify

Service level Agreement [should mostly give Availability, Scalability flexibility]

All policies

List of criteria for cloud SLA's

Criteria	Details
Availability	Percentage of time the service is guaranteed to be available.
Performance	Response time, throughput
Disaster Recovery	Mean time to recover
Problem resolution	Process to identify problems, support options, resolution expectations
Security & privacy of data	Mechanism for security of data in storage and transmission

Billing:

Three types of Billing

* Elastic pricing → pay according to what you need
* Fixed → similar to house rent, whether you used/not you should pay the amount
* Spot pricing - If demand is increased, an auction will be happened, so those who can pay the price, can use that

fixing price according to demand

Resources

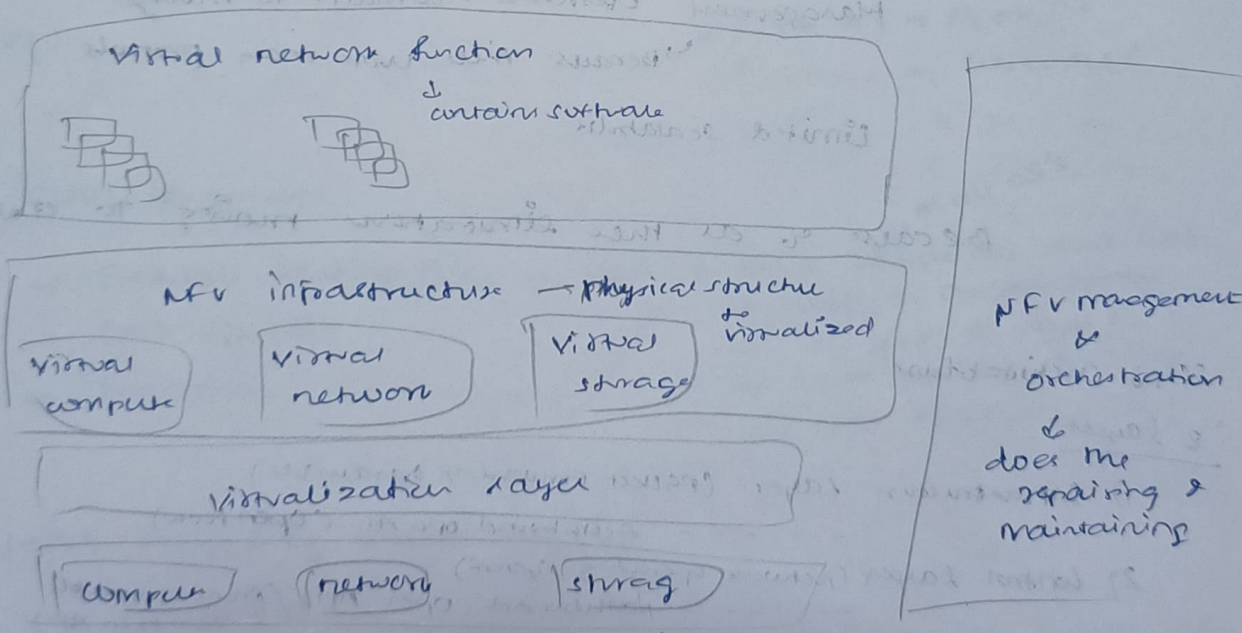
Server
Virtual machines
network
storage
Application services
Data services
security services
Deployment & management services.

→ For these we can fix prices in 5 ways

Network fuction Virtualization :

NFV architecture :-

Handles physical resources, software resources, traffics, network cloud accessings — for eg; DNS, fire wall — these can all be even brought from cloud.



NFV use cases

1) Virtualization of Security functions
2) " " Home enterprise & networks
3) " " content delivery networks (if cache is done in cloud then it is called)
4) " " Mobile, core network
5) " " Mobile BAS stations

Software defined Networking (SDN)

Handles the networking architectures

Control plane, data plane
↓ ↓ carries the traffic
carries the signal

Conventional Network architecture:
distinguishes the control plane & data plane
and prepares a centralized networking ⇒ This avoids traffic

3 limitations
- Complex network devices are needed
- More workload, ∴ more failures
- Management overhead → difficult to manage
  ↳ Because there were many protocols
- Limited scalability

Because of all these limitations moving to SDN

SDN Architecture

3 layers:
1) Infrastructure layer (Packet Forwarding Hardware)
   - - - - - Southbound open API (OpenFlow)
2) Control layer (Network operating System)
   - - - - Northbound open API (Programmable open API)
3) Application layer
   (Firewall, DNS)

OpenFlow seperates the data plane / control plane, & prepares a centralized networking

programmable API ⇒ handles the quality of service,
(delay, Throughput)

Openflow switch, Openflow table

end of unit-1

Hypervisor creates virtual machine in a physical resource. Ex: in help of virtual box.

Challenges of virtual box:-

* security - If hacked all VM's will be hacked
* Limited scalability
* Resource allocation /
* Resource compatability / hardware compatability
* No challenge in cost (it is cost effective)
* Performance overhead (slow performance because not running in native machines)

Note:
* Using 1 virtual box - we can create (2) virtual machines

* eg: " " " - " " " } VM - 1 ⇒ App OS
VM - 2 ⇒ Ubuntu

Benefits / Features:-

→ Software testing (when you want to test a software on all OS's but don't want it leak to another system you can use a virtual box and create multiple OS's and test them there)
→ Cost effective
→ Open source

26/7/24 Types of virtual Machine

1) System VM
2) Process VM → virtual Mach. created for a specific process, after the end of process - it gets demolished

Java VM ⇒ to make the code platform independent

example of process VM

to make the code a platform Independent

Contains a code to system understandable code

What happens in JVM as Interpretation happens (process)

System VM examples: Oracle Virtual Box, Xen, VMware,

↳ creating a VM with your own system resources

~~Virtual memory~~

How physically I only have 4 processor & 100 GB. Here how does Virtual Box provides 12 processor & 2TB??

That 4 processor is ÷ into 12 so sometimes it gets hang

Also for 2TB (your physical resources will have a virtual memory - There a file zipped file will be stored
↓
Through that this 2TB can be asked)

Calc.

5 × 2 = 10 A
5 × 5 = 25 B
15 × 1 = 15 C

virtual box
6 steps
container

Installation steps? Oracle virtual box

Notes

System VM

→ full virtualization machine (also called)

→ Acts in a real machine

→ eg:

→ Hardware resource

Process VM

* Application VM / Runtime VM / temporary VM (also called)

* One process at a time

* Platform independent

eg: JVM,

↳ can achieve high level of abstraction (access specifiers)

Binary Translation & Interpretation

* There are many Algorithm for this
  ↓
  written in PERL language — a coding language

* what does compiler do & what does interpreter do?
  ↓ whole
  convert code to machine understandable language

  ↓ line by line
  converts line by line code to machine understandable lang

  Why line by line:
  debugging
  so when there is error in a specified line it stops after resolving it moves next

  That's why we use to spyder, jupyter

Why should we bring Interpretation concept ⇒ boost in cloud computing?

* Scalability & Elasticity
  ↳ scale up, scale down
  ‼ while doing this the code content should get adapted so that Interpretation concept is incorporated

* when the workload ⇒ dynamic → platform / infrastructure / should get adapted to it so main thing

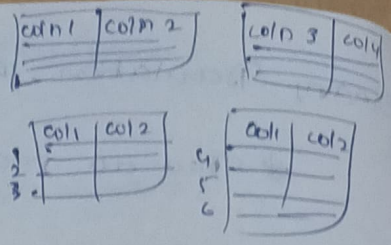
Algorithm:
Recursive Descent Parsing
Byte code Interpretation
Thread code Interpretation
} popular Interpretation Algorithm

Byte code Interpretation → changes into Byte code during Interpretation

Thread " " → change into Bytecode with address during Interpretation

attribute divided → vertical scaling

rows divided → Horizontal scaling →



Binary Translation:

* JIT → Just in time [when there is a repetitive code it only compile it and stores it so that instead of running each time it takes from the stored part]

same concept is used in Binary Translation the same packages as what we stored as Bytecode

* Uses of Binary Translation in cloud computing

TO shift from one architecture to another architecture

* AWS, GCP → they use Binary Translation

using it for to support cross platform support

google cloud platform

using it for cost efficiency

* code accessability, compatibility, makes the execution higher → uses of Binary Translation

3 Algorithms:

1) Static Binary Translation (SBT) → same to target
2) Dynamic " " (DBT) → first it converts to bytecode only during execution it gets optimized
3) Trace-Based dynamic binary Translation (Trace-based DBT)
   ↳ same as above but happens at the same time both translation & optimization