HEIDELBERG LAUREATE FORUM

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY Dharwad ज्ञानेन विकासः

acm India Council

Pingala Interactions In Computing

Cloud Virtualization and Storage

Dr. Animesh Chaturvedi

Assistant Professor: IIIT Dharwad

Young Researcher: Pingala Interaction in Computing

Young Researcher: Heidelberg Laureate Forum

Postdoc: King's College London & The Alan Turing Institute

PhD: IIT Indore MTech: IIITDM Jabalpur

KING'S College LONDON

The Alan Turing Institute

Indian Institute of Technology Indore
भारतीय प्रौद्योगिकी संस्थान इंदौर

PDPM
Indian Institute of Information Technology, Design and Manufacturing, Jabalpur

On Demand Computing

- It is one of required feature of Cloud Computing.
- Enterprise-level computing model
- Technology and computing resources are allocated to the organization or its individual users on the basis of demand.
- Example of Computing resources are CPU cycles, bandwidth availability, storage and applications c.
- ODC is needed for: Bandwidth-heavy applications, big data computing, to collect the big data.
- On-demand computing also referred to as utility computing. Thus most of the concept of Cloud Computing follows here such as Pay-as-you-go, HPC, infinite illusion resources.

Distributed Computing

- Field that studies distributed systems.
- Software system in which components located on networked computers communicate and coordinate by using and passing messages.
- Grid, Utility and Cloud computing are the branch of Distributed Computing.
- Fundamentally based on the Remote Procedure Call (RPC), message queues and location transparency.

Virtualization at the infrastructure level

Virtualization

- Creation of a virtual machine that acts like a real computer with an operating system refers to Virtualization.
- Software on virtual machines is independent of underlying hardware resources.
- Machine running a operating system may host another operating system that looks like a computer.

Virtualization

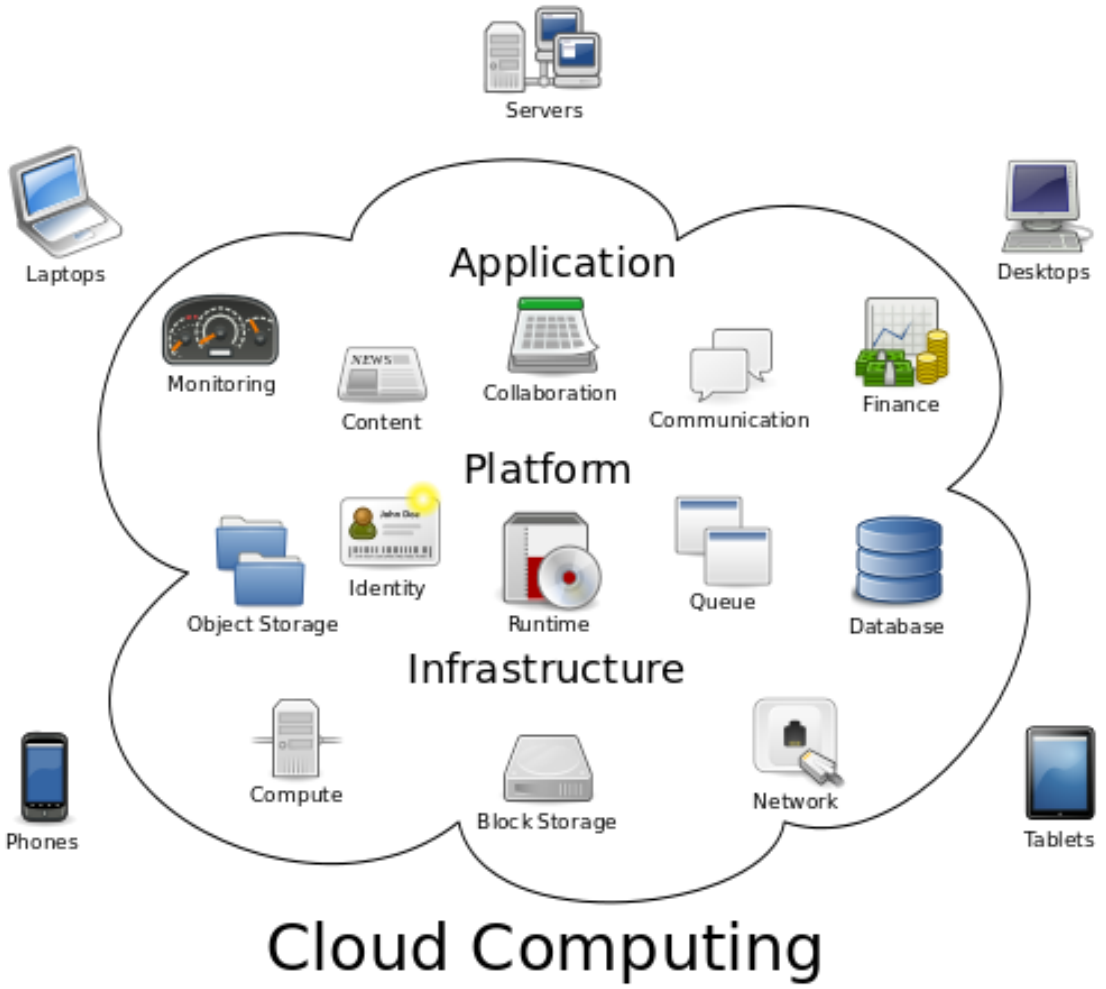
- A **virtual machine** (VM) is a software implementation of a machine (i.e., a computer) that executes programs like a physical machine.
- Software that creates a virtual machine on the host is called a hypervisor or Virtual Machine Manager. A hypervisor provides a uniform abstraction of the underlying physical machine.
- Multiple VMs can execute simultaneously on a single hypervisor.

Virtualization

- **Virtualization** is the single most effective way to reduce IT expenses while boosting efficiency and agility—not just for large enterprises, but for small and midsize businesses too.
- VMware **virtualization** lets you: Run multiple operating systems and applications on a single computer.
- **Host machine:** Machine which is the hardware on which the virtualization takes place.
- **Guest machine:** is the virtual machine.

Cloud Engines

- Eucalyptus
- Open Stack
- Nimbus
- Open Nebula

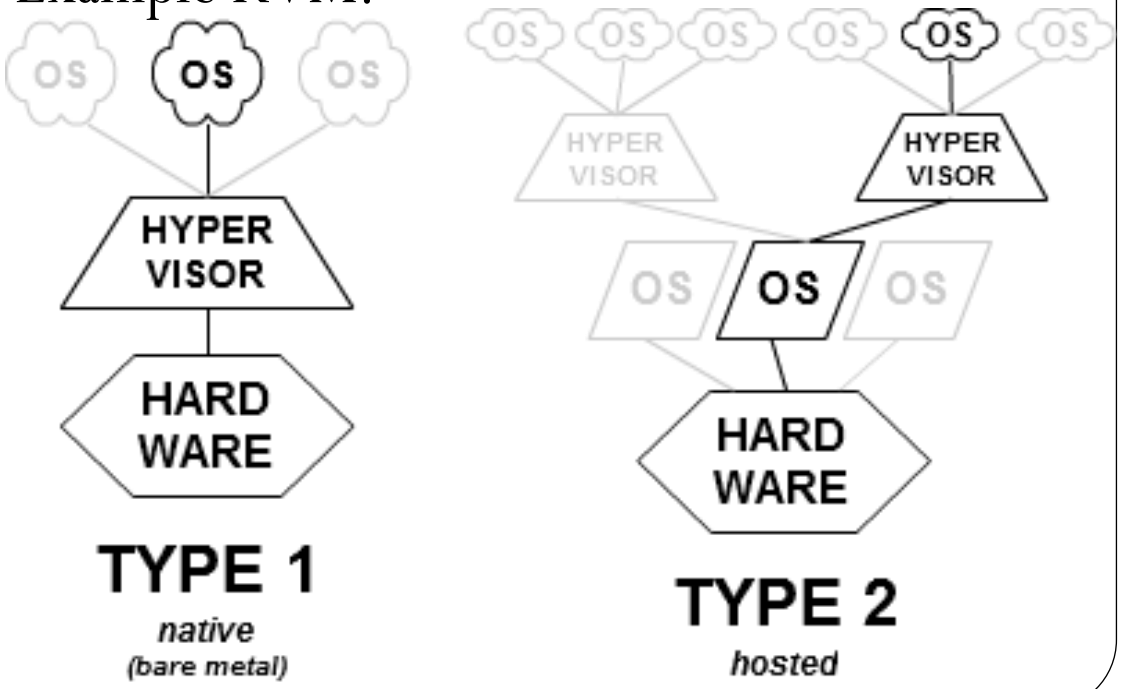


Nested Virtualization

- Running a virtual machine within another.
- Application virtualization can be deployed within a virtual machine.
- Example, Windows 7 is capable of running Windows XP applications inside a built-in virtual machine.
- Nested virtualization can be implemented on a particular computer architecture depends on supported hardware-assisted virtualization capabilities.
- In case a particular architecture does not provide hardware support required for nested virtualization, various software techniques are employed to enable it.

Hypervisor Types

- Type 1
  - Bare Metal : Runs directly on Hardware. On top Hypervisor the OS is loaded. Example Mircrosoft Hyper V, Citrix Xen Server,
- Type2
  - Hosted : The Hypervisor runs on a guest OS. Example KVM.



Types of Virtualizations

Operating System level

- Memory virtualization: aggregating RAM resources from networked systems into a single memory pool
- Virtual memory: giving an app the impression that it has contiguous working memory, isolating the underlying physical memory

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

Modern OS Virtualization

- *Hardware-assisted virtualization is a key technological enabler for Cloud Computing*
  - Provides complete isolation on commodity (low-cost) platforms
  - Enables multiplexing of many users onto single server
- Key contribution is minimal performance overhead (few percent) versus non-virtualized
  - However, high I/O applications incur many VM traps (high CPU overhead), limiting scalability and efficiency
- Challenge: true performance isolation for multiple applications
  - Many dimensions! (more in research discussion)

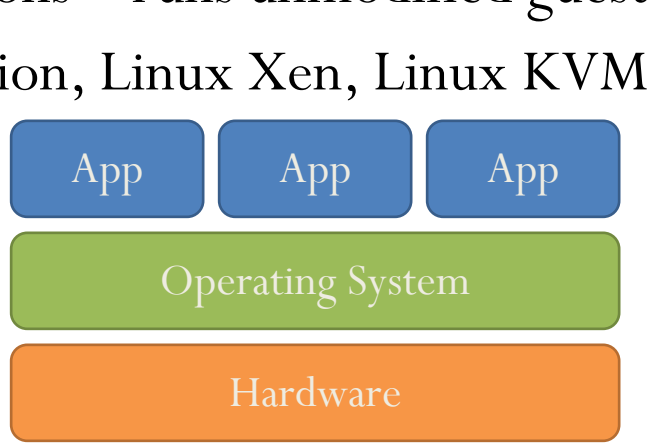
Types of Virtualizations

- Virtual computer hardware platforms, storage devices, and computer network resources.
- *Hardware virtualization* or *platform virtualization* refers to the creation of a Virtual Machine (VM).
  - Full virtualization – complete simulation of the actual hardware to allow software environments, guest operating system and its apps.
  - Paravirtualization – the guest apps are executed in their own isolated domains.
- Application virtualization and Workspace virtualization: isolating individual apps from the underlying OS and other apps
- Service virtualization: emulating the behavior of API, Cloud and SOA

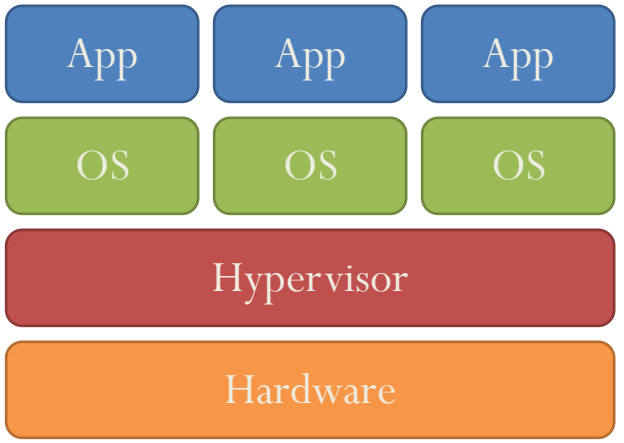
https://en.wikipedia.org/wiki/Virtualization

Types of Virtualizations

- Full virtualization
  - Sensitive instructions (discovered statically or dynamically at run-time) are replaced by binary translation or trapped by hardware into VMM for SW emulation
  - Any OS software can run in the VM
  - Examples: IBM's CP/CMS, Oracle (Sun) VirtualBox, VMware Workstation
- Hardware-assisted virtualization (IBM S/370, Intel VT, or AMD-V)
  - CPU traps sensitive instructions – runs unmodified guest OS
  - Examples: VMware Workstation, Linux Xen, Linux KVM, Microsoft Hyper-V



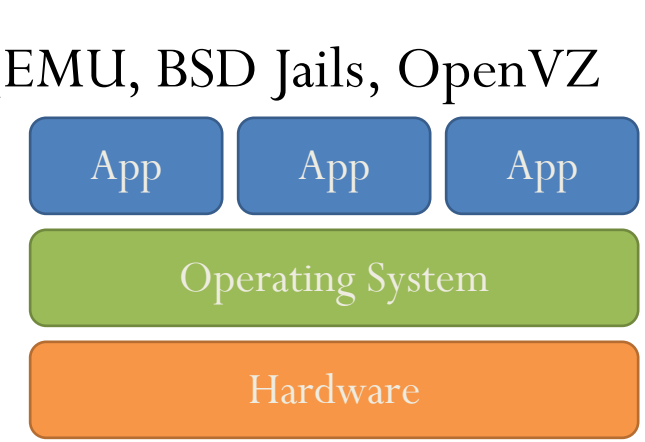
Traditional Stack



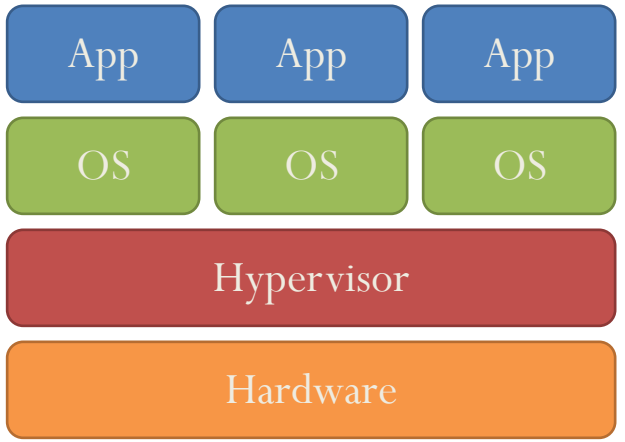
Virtualized Stack

Types of Virtualizations

- Para-virtualization
  - Presents SW interface to virtual machines similar to but not identical to that of the underlying HW, requiring guest operating systems to be adapted
  - Examples: early versions of Xen
- Operating System virtualization
  - OS kernel allows for multiple isolated user-space instances, instead of just one
  - Instances look and feel like a real server
  - Examples: Solaris Zones, QEMU, BSD Jails, OpenVZ



Traditional Stack



Virtualized Stack

Types of Virtualizations

- Network virtualization, creation of a virtualized network addressing space within or across network subnets
- Virtual private network (VPN), a network protocol that replaces the actual wire or other physical media in a network with an abstract layer, allowing a network to be created over the Internet.
- Data virtualization: the presentation of data as an abstract layer, independent of underlying database systems, structures and storage.
- Database virtualization: the decoupling of the database layer, which lies between the storage and application layers .

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

Types of Virtualizations

- Storage virtualization, the process of completely abstracting logical storage from physical storage.
  - Storage hypervisor, the software that manages storage virtualization and combines physical storage resources into one or more flexible pools of logical storage.
- Distributed file system, any file system that allows access to files from multiple hosts sharing via a computer network
- Virtual file system, an abstraction layer on top of a more concrete file system, allowing client applications to access different types of concrete file systems in a uniform way.
- Virtual disk drive, a computer program that emulates a disk drive such as a hard disk drive or optical disk drive

Eucalyptus and Amazon Machine Image

Eucalyptus

- Eucalyptus deploys instances (i.e., virtual machines) on a hypervisor.
- Eucalyptus can use either Xen or KVM hypervisors.
- To interact with them, Eucalyptus employs libvirt virtualization API.
- Creating nested hypervisors require running one or more hypervisors inside another hypervisor.

Amazon Machine Image (AMI)

- Special type of virtual appliance that is used to instantiate (create) a virtual machine within the Amazon Elastic Compute Cloud ("EC2").
- It serves as the basic unit of deployment for services delivered using EC2
- **Public**: an AMI image that can be used by anyone.
- **Paid**: a for-pay AMI image that is registered with Amazon DevPay and can be used by anyone who subscribes for it. DevPay allows developers to mark-up Amazon's usage fees and optionally add monthly subscription fees.
- **Shared**: a private AMI that can only be used by Amazon EC2 users who are allowed access to it by the developer.

Amazon Machine Image (AMI)

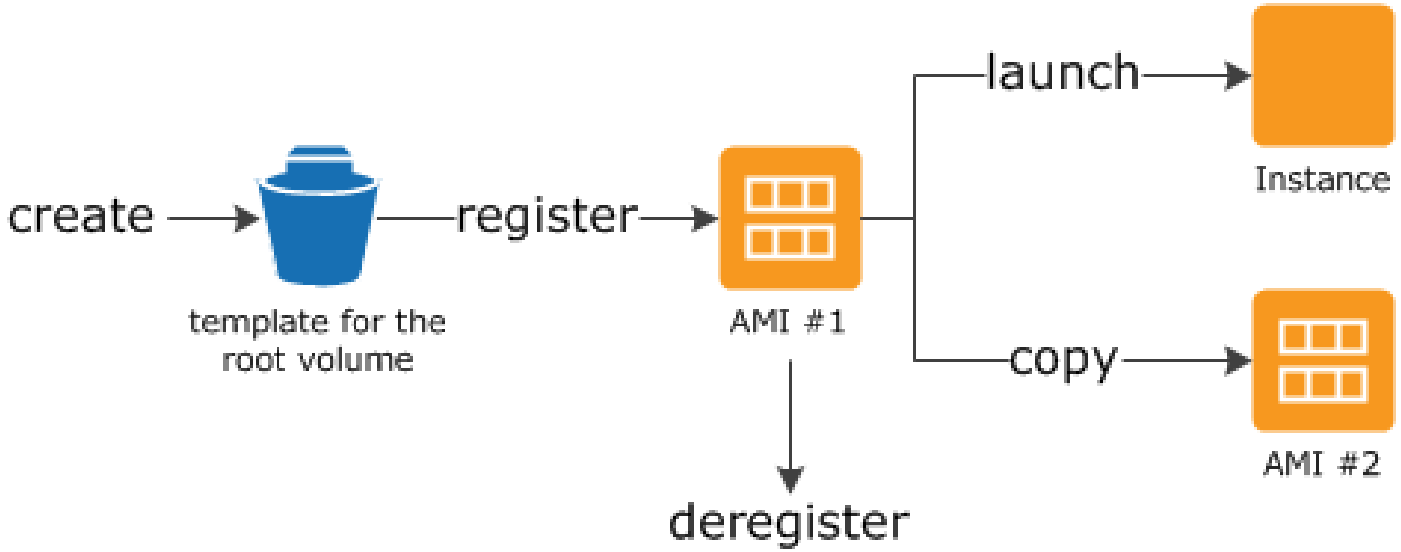
- **Public**: an AMI image that can be used by anyone.
- **Paid**: a for-pay AMI image that is registered with Amazon DevPay and can be used by anyone who subscribes for it. DevPay allows developers to mark-up Amazon's usage fees and optionally add monthly subscription fees.
- **Shared**: a private AMI that can only be used by Amazon EC2 users who are allowed access to it by the developer.
- AMI Demo

Amazon Machine Image (AMI)

- An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.
- An AMI includes the following:
- A template for the root volume for the instance (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

AMI lifecycle

- You can search for an AMI that meets the criteria for your instance. You can search for AMIs provided by AWS or AMIs provided by the community
- Launch new instances (Owner grants you launch permissions)
- Connecting, and using your instance at Amazon EC2 Instances.



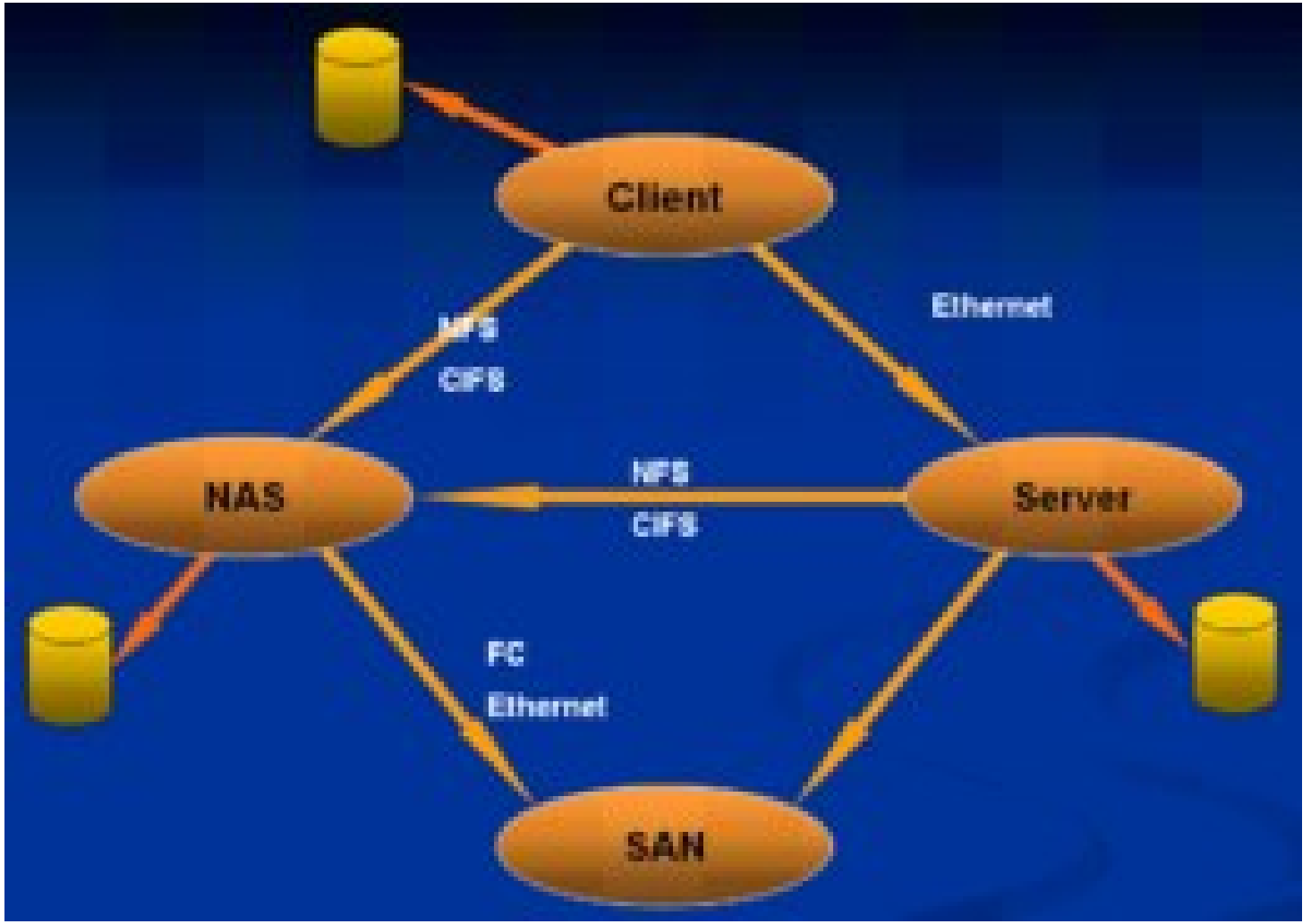
Storage Area Network (SAN)

https://sites.google.com/site/animeshchaturvedi07/academic-teaching/cloudcomputing

SAN

- A dedicated network that provides access to consolidated, block level data storage.
- SANs are primarily used to enhance storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system.
- A SAN typically has its own network of storage devices that are generally not accessible through the local area network (LAN) by other devices.
- The cost and complexity of SANs dropped in the early 2000s to levels allowing wider adoption across both enterprise and small to medium sized business environments.

SAN


Client
Ethernet
NFS
CIFS
NAS
NFS
CIFS
Server
FC
Ethernet
SAN

SAN Benefits

- Sharing storage usually simplifies storage since cables and storage devices do not have to be physically moved to shift storage from one server to another.
- Ability to allow servers to boot from the SAN itself. This allows for a quick and easy replacement of faulty servers since the SAN.
- While SAN technology is still new, many view it as being the future of the enterprise datacenter.
- SAN extension over IP networks.
- SANs also tend to enable more effective disaster recovery processes.
- A SAN could span a distant location containing a secondary storage array. This enables storage replication either implemented by disk array controllers, by server software, or by specialized SAN devices.

Small Computer System Interface (SCSI)

- Small Computer System Interface is a set of standards for physically connecting and transferring data between computers and peripheral devices.
- The SCSI standards define commands, protocols and electrical and optical interfaces.
- SCSI is most commonly used for hard disks and tape drives
- The traditional physical SCSI layer could only support a few meters of distance - not nearly enough to ensure business continuance in a disaster.
- SAN is the economic consolidation of disk arrays has accelerated the advancement of several features including I/O caching, snapshotting, and volume cloning.

SAN QoS

- Key factors that affect Storage Area Network QoS (Quality of Service) are:
  - Bandwidth – The rate of data throughput available on the system.
  - Latency – The time delay for a read/write operation to execute.
  - Queue depth – The number of outstanding operations waiting to execute to the underlying disks (Traditional or SSD).

No SQL and CAPS

https://sites.google.com/site/animeshchaturvedi07/academic-teaching/cloudcomputing

No SQL

- “Not Only SQL” i.e. database systems that works on other than the tabular relations used in relational databases.
- No SQL systems may also support SQL-like query languages.
- No SQL database systems has data structures different from relational databases (key-value, graph, document).
- High usage in big data and real-time web applications.
- Huddle in adoption of NoSQL stores are low-level query languages, lack of standardized interfaces, and huge investments in existing SQL.
- Some NoSQL systems do not provides all four ACID properties together (atomicity, consistency, isolation and durability).
- Mostly NoSQL databases systems follows the CAP theorem.

CAPS Theorem

- Also known as Brewer's theorem

"It is impossible for a distributed computer system to simultaneously provide following three together with guarantees in single instance"

- **Consistency:** all nodes can view the same data at the same time
- **Availability:** a guarantee that every request will receives response for success or failure
- **Partition:** the system continues to operate irrespective of the loss or failure of a node.

Categories of NoSQL databases

- **Column Oriented:** Accumulo, Cassandra, Druid, HBase, Vertica
- **Document-Oriented:** Clusterpoint, Apache, CouchDB, Couchbase, MarkLogic, MongoDB, OrientDB
- **Key-value:** Dynamo, FoundationDB, MemcacheDB, Redis, Riak, FairCom c-treeACE, Aerospike, OrientDB
- **Graph:** Allegro, Neo4J, InfiniteGraph, OrientDB, Virtuoso, Stardog
- **Multi-model:** OrientDB, FoundationDB, ArangoDB, Alchemy Database, CortexDB.

http://en.wikipedia.org/wiki/NoSQL

תודה רבה
Hebrew

Ευχαριστώ
Greek

Спасибо
Russian

Danke
German

धन्यवादः
Sanskrit

Merci
French

ধন্যবাদ
Bangla

நன்றி
Tamil

شكراً
Arabic

ಧನ್ಯವಾದಗಳು
Kannada

Thank You
English

നന്ദി
Malayalam

多謝
Traditional Chinese

Grazie
Italian

ధన్యవాదాలు
Telugu

આભાર
Gujarati

ਧੰਨਵਾਦ
Punjabi

धन्यवाद
Hindi & Marathi

多谢
Simplified Chinese

Gracias
Spanish

https://sites.google.com/site/animeshchaturvedi07

Obrigado
Portuguese

ありがとうございました
Japanese

ขอบคุณ
Thai

감사합니다
Korean