CLOUD COMPUTING

Storage Virtualization

Dr. Prafullata Kiran Auradkar

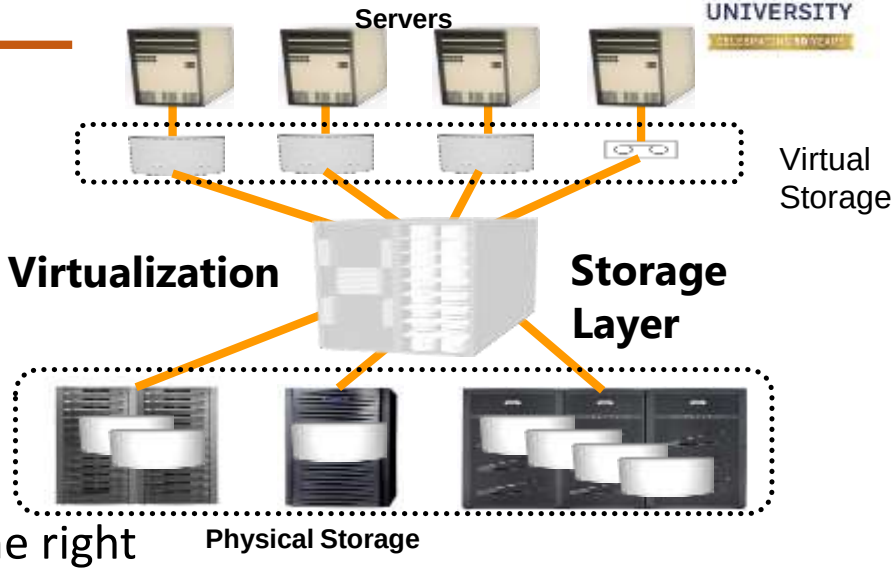
Department of Computer Science and Engineering

Acknowledgements:

Significant information in the slide deck presented through the Unit 3 of the course have been created by **Dr. H.L. Phalachandra** and would like to acknowledge and thank him for the same. There have been some information which I might have leveraged from the content of **Dr. K.V. Subramaniam's** lecture contents too. I may have supplemented the same with contents from books and other sources from Internet and would like to sincerely thank, acknowledge and reiterate that the credit/rights for the same remain with the original authors/publishers only. These are intended for classroom presentation only.

CLOUD COMPUTING

Storage – Virtualization



- Storage Virtualization is a means through which physical storage subsystems (disks, tapes ..) are abstracted from the user's application and presented as logical entities, hiding the underlying complexity of the storage subsystems and nature of access, network or changes to the physical devices.
- Its the process of aggregating the capacity of multiple storage devices into storage pools
- Aggregates multiple resources as one addressable entity (pool) or divides a resource to multiple addressable entities and enables easy provisioning of the right storage for performance or cost.
  - E.g. Single virtual large disks from multiple small disks or Many smaller virtual disks from a large disk
- Virtualization of storage helps achieve ***location independence*** by abstracting the physical location of the data. The virtualization system/layer presents to the user a logical space for data storage and handles the ***mapping*** to the actual physical location.
- Virtualization software is responsible for maintaining a consistent view of all of the mapping information and keeping it consistentt. This mapping information is often part of called **metadata**.
- Virtualization layer can be in H/W or in S/W

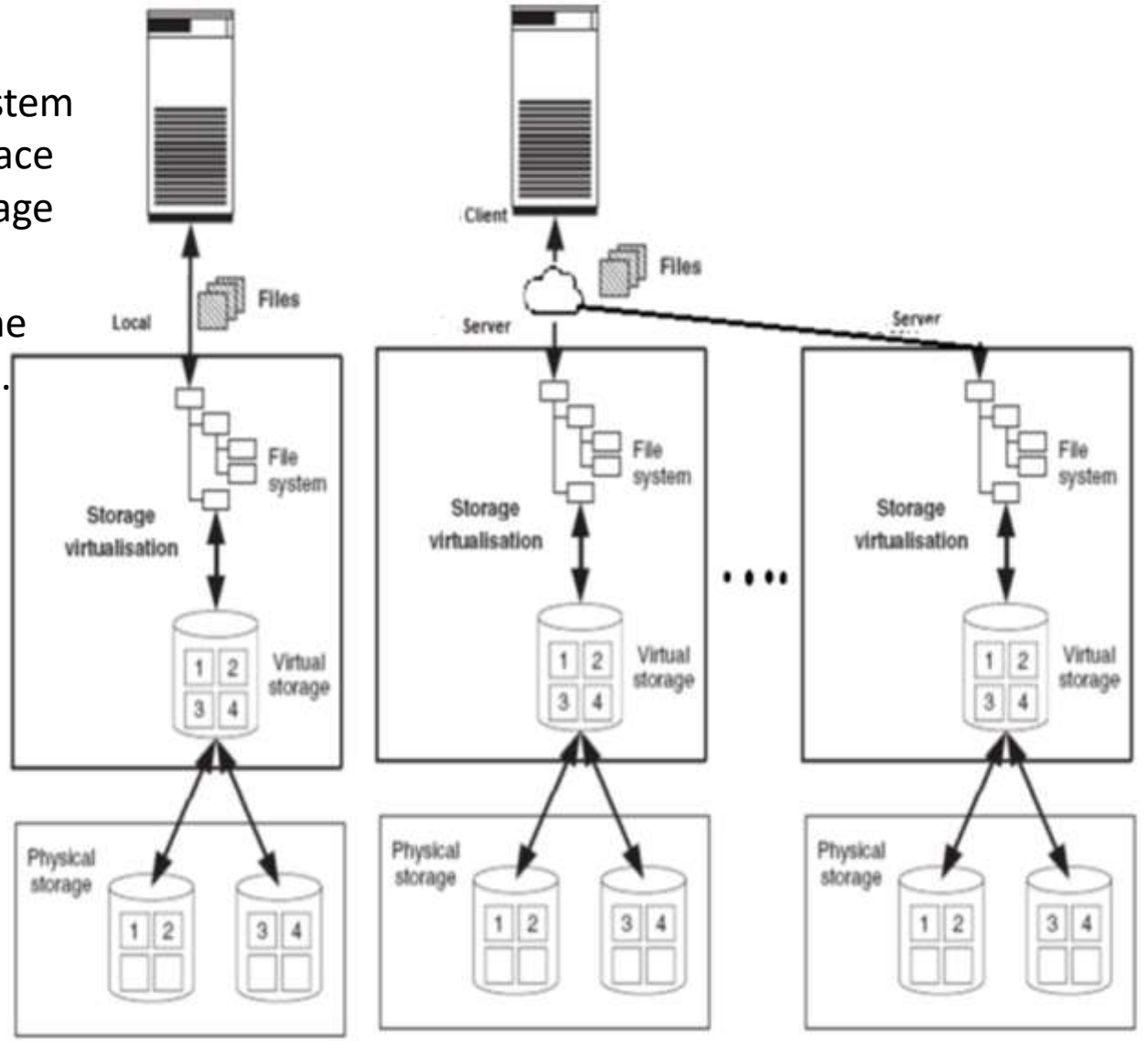
CLOUD COMPUTING

Storage – Categories of Storage Virtualization

1. File level Virtualization

- A file system virtualization provides an abstraction of a file system to the application (with a standard file-serving protocol interface such as NFS or CIFS) and manages changes to distributed storage hardware underneath the file system implementation.
- Eliminates the dependencies between the data accessed at the file level and the location where the files are physically stored.
- Virtualization layer manages files, directories or file systems across multiple servers and allows administrators to present users with a single logical file system.
- A typical implementation of a virtualized file system is as a network file system that supports sharing of files over a standard protocol with one or more file servers enabling access to individual files.
- **File-serving protocols** that are typically employed are NFS, CIFS and Web interfaces such as HTTP or WebDAV

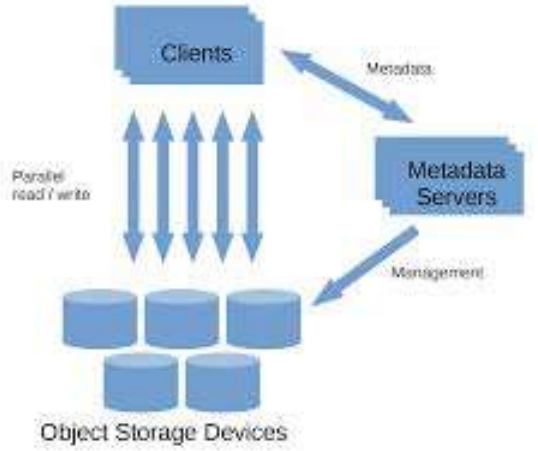
  Adv: This provides opportunities to optimize storage usage, server consolidation & non-disruptive file migrations.



Storage : 1. File Level Virtualization

Distributed File System

- A distributed file system (DFS) is a network file system wherein the file system is distributed across multiple servers.
- DFS enables location transparency and file directory replication as well as tolerance to faults.
- Some implementations may also cache recently accessed disk blocks for improved performance. Though distribution of file content increases performance considerably, efficient management of metadata is crucial for overall file system performance
- Two important techniques for managing metadata for highly scalable file virtualization:
  a. **Separate data from metadata** with a **centralized metadata** server (used in ***Lustre***)
  b. **Distribute data and metadata** on multiple servers (used in ***Gluster***)



Storage : 1. File Level Virtualization – Distributed File System

Distributed File Systems with Centralized Metadata

- A centralized metadata management scheme achieves scalable DFS with a dedicated metadata server to which all metadata operations performed by clients are directed.
- Lock-based synchronization is used in every read or write operation from the clients.
- In centralized metadata systems, the metadata server can become a bottleneck if there are too many metadata operations.
- For workloads with large files, centralized metadata systems perform and scale very well

Storage : 1. File Level Virtualization – Distributed File System

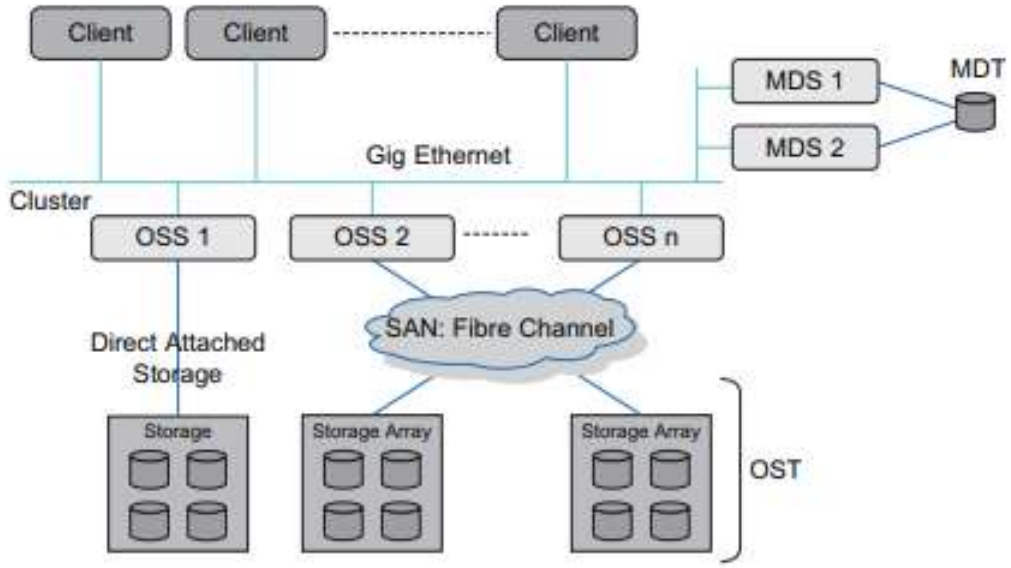
Lustre - Distributed File Systems with Centralized Metadata

- Lustre is a massively parallel, scalable distributed file system for Linux which employs a cluster-based architecture with centralized metadata.
- This is a software solution with an ability to scale over thousands of clients for a storage capacity of petabytes with high performance I/O throughput.
- The architecture of Lustre includes the following three main functional components, which can either be on the same nodes or distributed on separate nodes communicating over a network
    1. Object storage servers (OSSes), which store file data on object storage targets (OSTs).
    2. A single metadata target (MDT) that stores metadata on one or more Metadata servers (MDS)
    3. Lustre Clients that access the data over the network using a POSIX interface

Storage : 1. File Level Virtualization – Distributed File System

Lustre architecture (Functioning)

- When a client accesses a file, it does a filename lookup on a MDS.
- Then, MDS creates a metadata file on behalf of the client or returns the layout of an existing file.
- The client then passes the layout to a logical object volume (LOV) for read or write operations.
- The LOV maps the offset and size to one or more objects, each residing on a separate OST.
- The client then locks the file range being operated on and executes one or more parallel read or write operations directly to the OSTs.



Storage : 1. File Level Virtualization – Distributed File System

Distributed File Systems with Distributed Metadata

- In distributed metadata management, metadata is distributed across all nodes in the system, rather than using centralized metadata servers.
- Such systems have greater complexity than centralized metadata systems, since the metadata management is spread over all the nodes in the system.

Storage : 1. File Level Virtualization – Distributed File System

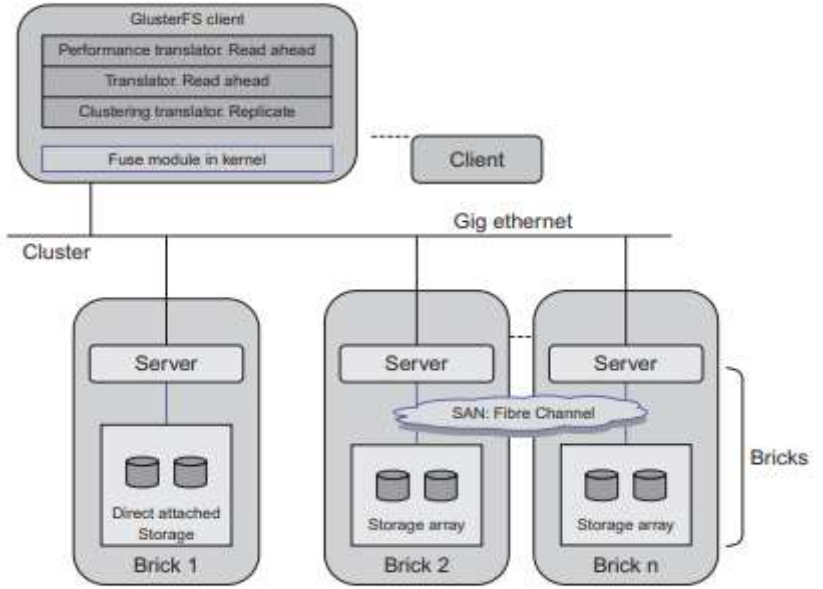
GlusterFS - Distributed File Systems with Distributed Metadata

- GlusterFS is an open-source, distributed cluster file system without a centralized metadata server.
- It is also capable of scaling to thousands of clients, Petabytes of capacity and is optimized for high performance.
- GlusterFS employs a modular architecture with a stackable user-space design. It aggregates multiple storage bricks on a network (over Infiniband RDMA or TCP/IP interconnects) and delivers as a network file system with a global name space.
- It consists of just two major components: a Client and a Server.
  - The Gluster server clusters all the physical storage servers and exports the combined diskspace of all servers as a Gluster File System.
  - The Gluster client is actually optional and can be used to implement highly available, massively parallel access to every storage node and handles failure of any single node transparently

Storage : 1. File Level Virtualization – Distributed File System

Gluster FS architecture

- GlusterFS uses the concept of a storage brick consisting of a server that is attached to storage directly (DAS) or through a SAN.
- Local file systems (ext3, ext4) are created on this storage.
- Gluster employs a mechanism called translators to implement the file system capabilities.
- Translators are programs (like filters) inserted between the actual content of a file and the user accessing the file as a basic file system interface
- Each translator implements a particular feature of GlusterFS.



- Translators can be loaded both in client and server side appropriately to improve or achieve new functionalities
- Gluster performs very good load balancing of operations using the I/O Scheduler translators
- GlusterFS also supports file replication with the Automatic File Replication (AFR) translator, which keeps identical copies of a file/directory on all its subvolumes

Storage : 2. Block Virtualization

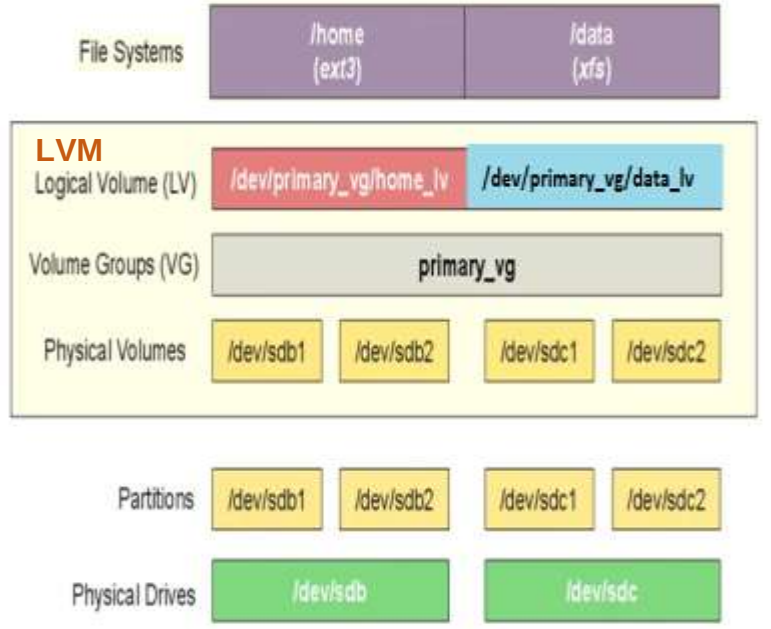
- Block level Virtualization Virtualizes multiple physical disks and presents the same as a single logical disk.
- The data blocks are mapped to one or more physical disks sub-systems.
- These block addresses may reside on multiple storage sub-systems, appearing however as a single storage (logical) storage device.
- Block level storage virtualization can be performed at three levels:
  a. Host-Based
  b. Storage Level
  c. Network level

CLOUD COMPUTING

Storage : 2. Block Virtualization

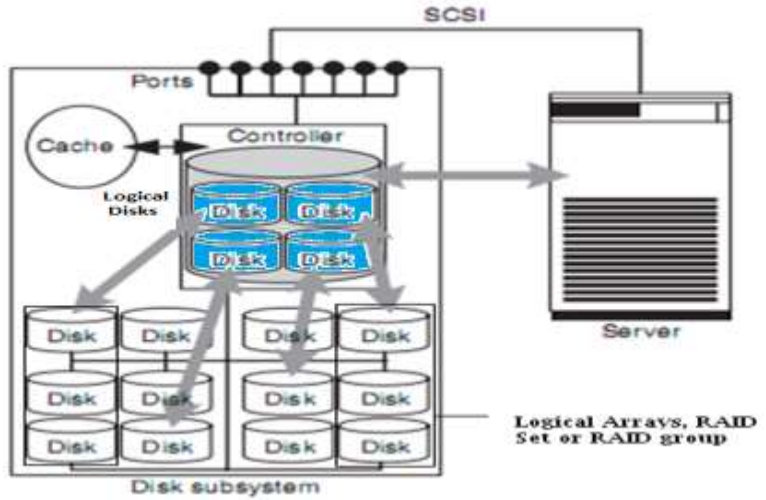
Host-Based block virtualization

- Uses a **Logical Volume Manager** (LVM), a virtualization layer that supports creation of a Storage pool by combining multiple disks (as in the picture) that is greater than the size of a physical disk from where the logical storage is created.
- This also allows transparent allocation and management of disk space for file systems or raw data with capabilities to dynamically shrink or increase physical volumes.



Storage-Device level virtualization

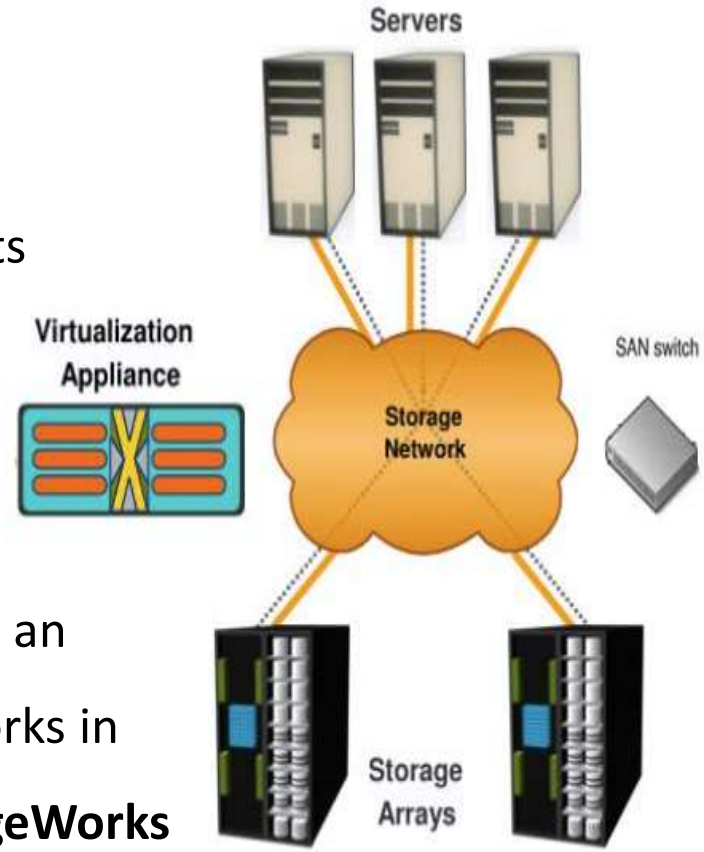
- Creates **Virtual Volumes** over the physical storage space of the specific storage subsystem.
- Storage disk arrays provide this form of virtualization using RAID techniques. Array controllers create **Logical UNits** (LUNs) spanning across multiple disks in the array in RAID Groups. Some disk arrays also virtualize third-party external storage devices attached to the array.
- This technique is generally host-agnostic and has low latency since the virtualization is a part of the storage device itself and in the firmware of the device.



Storage : 2. Block Virtualization

Network-Based block virtualization

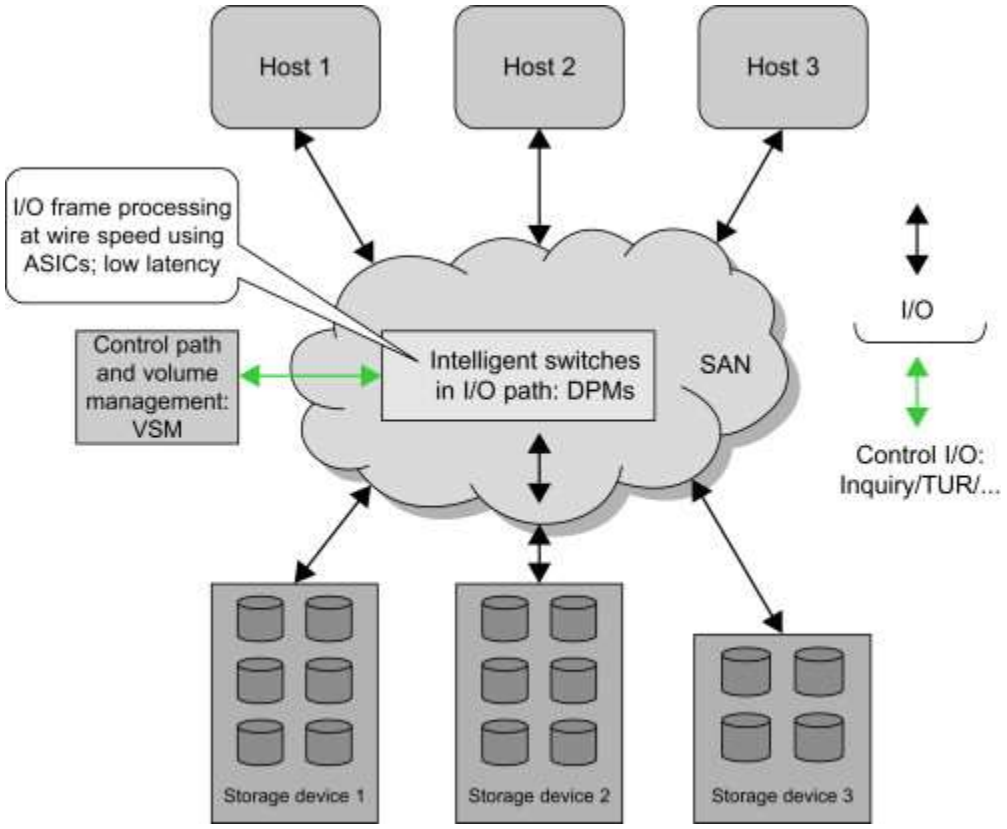
- This is the most commonly implemented form of scalable virtualization.
- Virtualization functionality is implemented within the network connecting hosts and storage, say a Fibre Channel Storage Area Network (SAN).
- There are broadly two categories **based on where the virtualization functions are implemented**: either in **switches** (routers) or in **appliances** (servers).
  - In a switch-based network virtualization, the actual virtualization occurs in an intelligent switch in the fabric and the functionality is achieved when it works in conjunction with a metadata manager in the network. Example**: HP StorageWorks SAN Virtualization Services Platform**
  - In an appliance-based approach, the I/O flows through an appliance that controls the virtualization layer. Example: **IBM SAN Volume Controller**



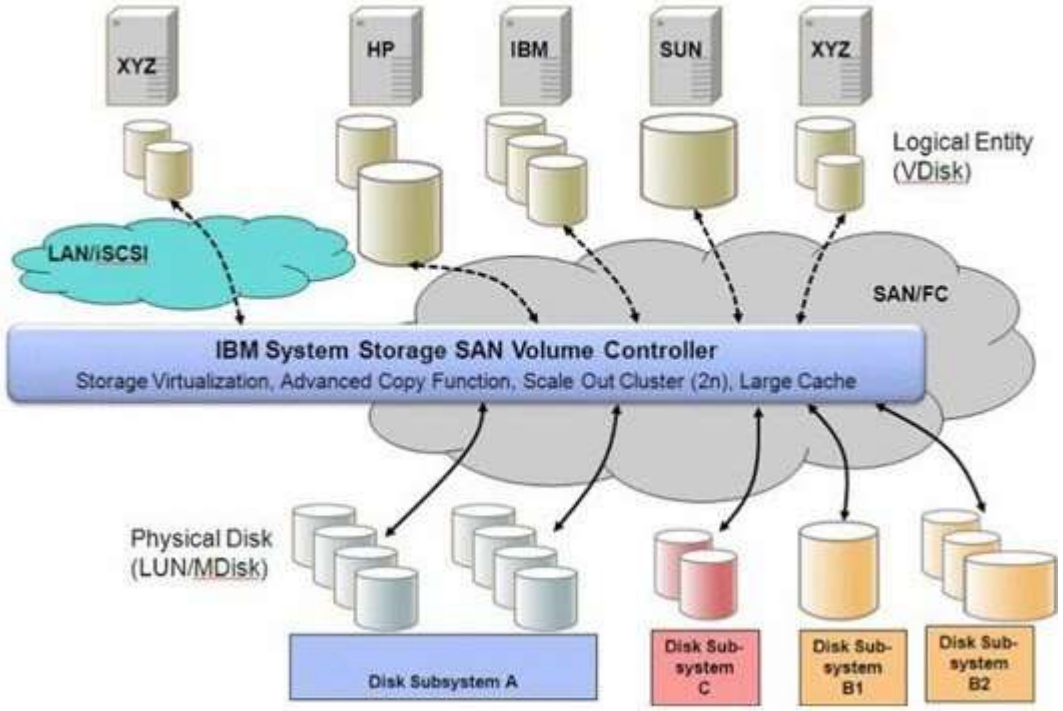
CLOUD COMPUTING

Storage : 2. Block Virtualization

Switch-based network virtualization



Appliance-based network virtualization



Storage : 2. Block Virtualization

Network-Based block virtualization

- There are broadly two variations of an appliance-based implementation.
- The appliance can either be **in-band** or **out-of-band**.
- In **in-band**, all I/O requests and their data pass through the virtualization device and the clients do not interact with the storage device at all. All I/O is performed by the appliance on behalf of the clients
- In **out-of-band** usage, the appliance only comes in between for metadata management (control path), while the data (I/O) path is directly from the client to each host (with agents on each host/client).

THANK YOU

Prafullata Kiran Auradkar

Department of Computer Science and Engineering

prafullatak@pes.edu