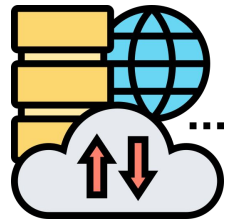
COEN 241
Introduction to Cloud Computing

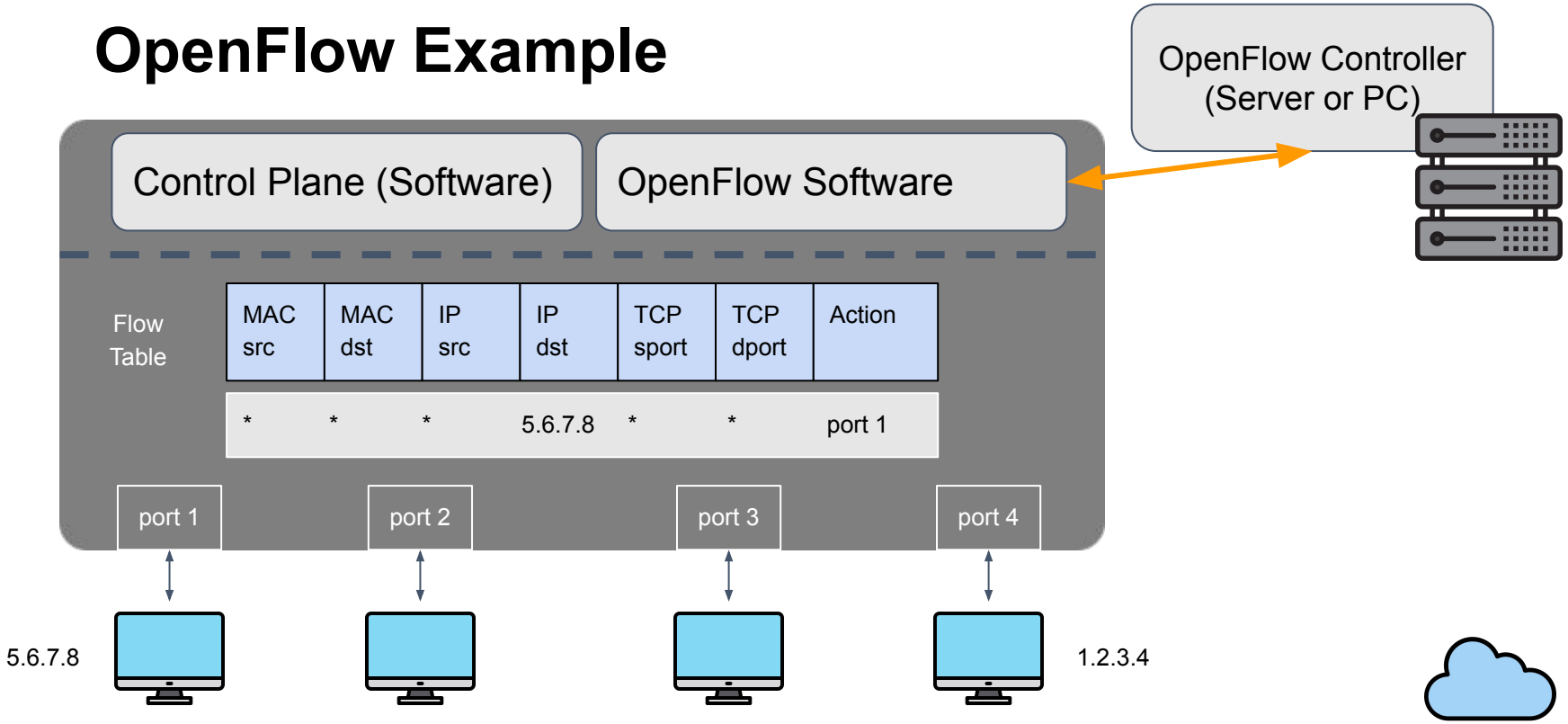
Lecture 14 - Storage Virtualization & Cloud Storage



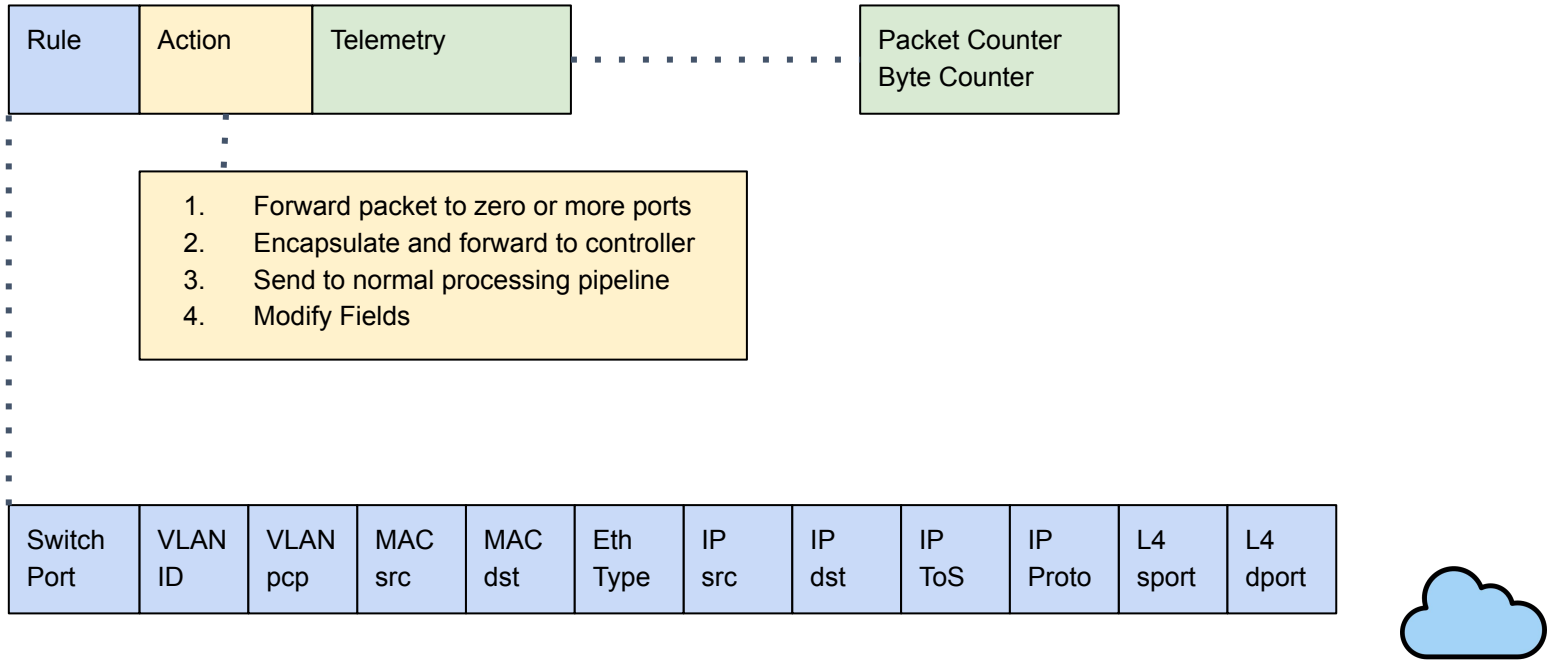
Lecture 13 Recap

- OpenFlow Part II
- Mininet
- NFV
- Readings
  - Recommended:
  - Optional: https://queue.acm.org/detail.cfm?id=2560327

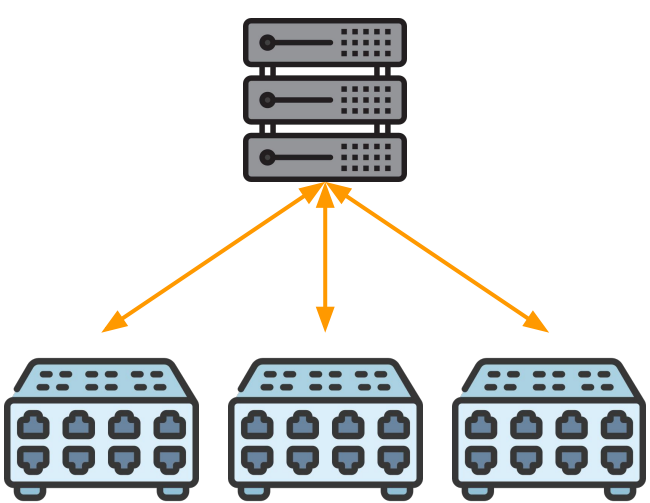
OpenFlow Example



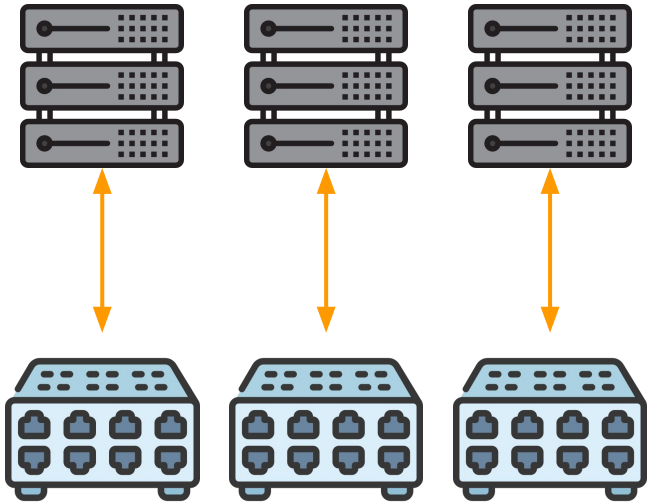
OpenFlow Basics



Centralized vs Distributed Controller



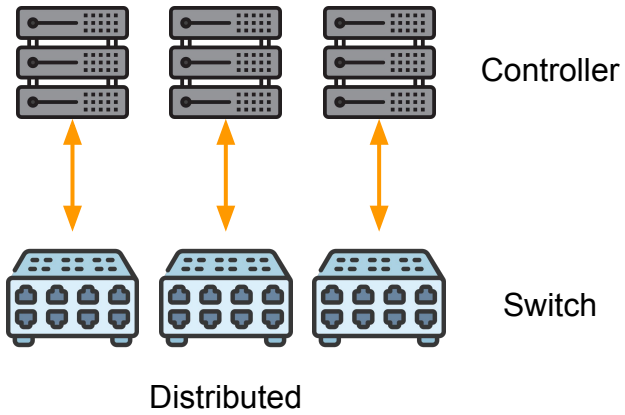
Centralized



Distributed

Centralized vs Distributed Controller

- Distributed Controller improves scalability
  - E.g., Many packets to capture
- Still requires lot of packets through the control channels
- How to solve this issue?
  - Modify the data plane
  - At the loss of visibility



Types of Flow Rules

- Flow-based
  - Every flow is individually set up by controller
  - Exact-match flow entries
  - Flow table contains one entry per flow
  - Good for fine grain control, e.g. campus networks

- Aggregated
  - One flow entry covers large groups of flows
  - Wildcard flow entries
  - Flow table contains one entry per category of flows
  - Good for large number of flows, e.g. backbone

Flow Rule Installation Methods

- Reactive
  - First packet of flow triggers controller to insert flow entries
  - Efficient use of flow table
  - Every flow incurs small additional flow setup time
  - If control connection lost, switch has limited utility

- Proactive
  - Controller pre-populates flow table in switch
  - Zero additional flow setup time
  - Loss of control connection does not disrupt traffic
  - Essentially requires aggregated (wildcard) rules

What is Mininet?

- Software for development and testing of network tools and protocols
- Creates virtual network on any type of machine (VM or native)
- Enables the following features:
  - Fast prototyping for new networking protocols
  - Simplified testing for complex topologies without buying expensive hardware
  - Realistic execution as it runs real code on the Unix and Linux kernels
  - Large open source community
- Designed for experiment in SDN

Mininet Topology

- Default topology is two hosts and one switch
- `sudo mn --topo single,3` gives single switch and three hosts
- `sudo mn --topo linear,3` gives three switch and three hosts
- `sudo mn --topo tree,fanout=2,depth=2` gives a tree topology with depth of two and fanout of two
- You can also pass in a custom topology
  - http://mininet.org/walkthrough/#custom-topologies

What is Network Function Virtualization?

- Move network control functions from switch firmware into software
  - I.e., Virtualize the network function
- Common network functions include:
  - **DHCP**: dynamic host configuration protocol (assign IPs)
  - **Firewalls**: filter and modify traffic to secure networks
  - **DPI**: deep packet inspection: scans packet data
  - **IDS**: intrusion detection systems scan network for attacks
  - **NTP**: network time protocol

Agenda for Today

- Final Presentation and Report Guideline
- Storage Virtualization
- Cloud Storage
- Cloud Databases
- Readings
  - Recommended:
    - CCSA Chapter 5,6,7
    - https://www.enterprisestorageforum.com/hardware/storage-virtualization/
  - Optional: None

Final Presentation and Report

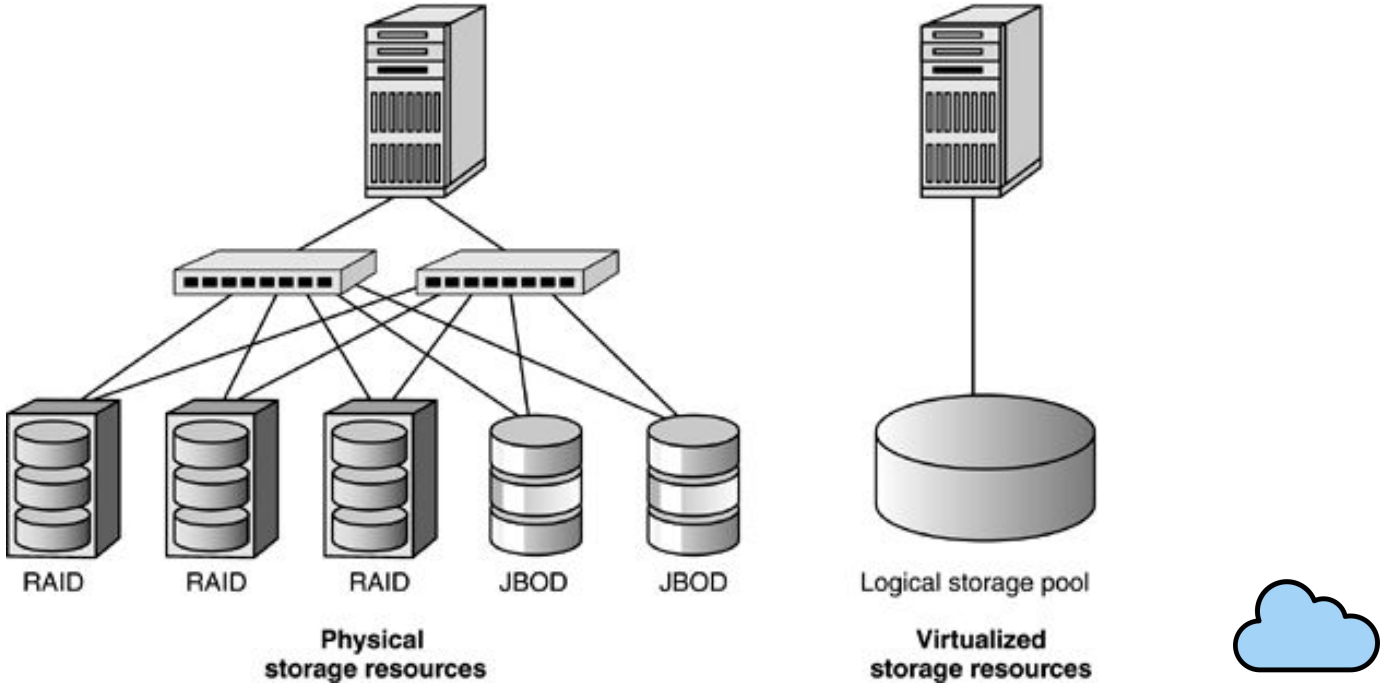
- Instructions in Camino
- Provide a set of sections based on the instruction

Storage Virtualization

What is Storage Virtualization?

- Pooling of physical storage from multiple storage devices into what appears to be a single storage device
  - Managed from a central console
- Relies on software to identify available storage capacity from physical devices and then aggregate that capacity as a pool of storage
  - Metadata stores the location of the data
- A concept that existed even before the Cloud!

What is Storage Virtualization?



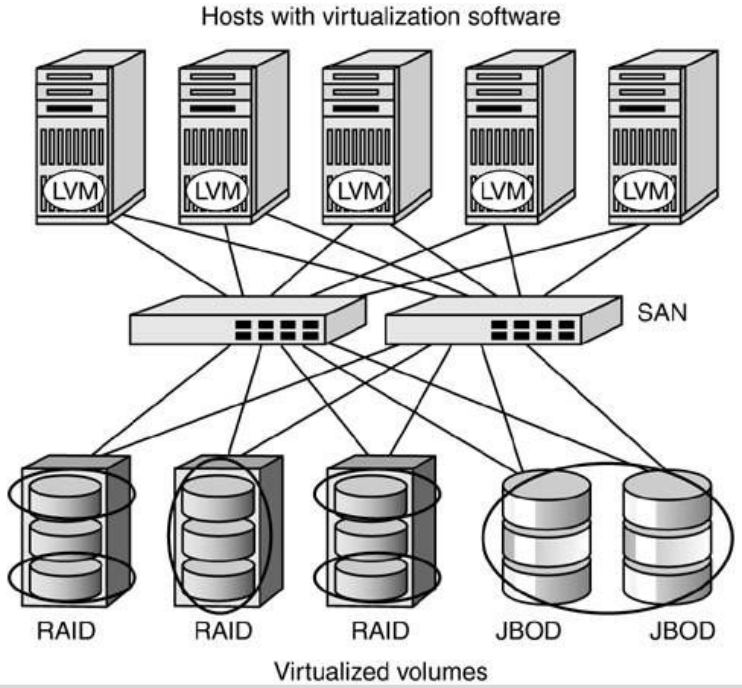
Example / Types of Storage Virtualization

- Host based
  - Running additional software on each host
- Storage device based
  - Creating disk arrays of storages for virtualization
- Network based
  - Combining storage over a computer network

Host-based Storage Virtualization

- Implemented through a logical volume management (LVM) application installed on each server
- Pro: No splitting of control and data paths, thus no bottlenecks and performance issues associated with retrieving the metadata
- Con: virtualization on each host must be configured and managed separately
  - Hard to scale

Host-based Storage Virtualization

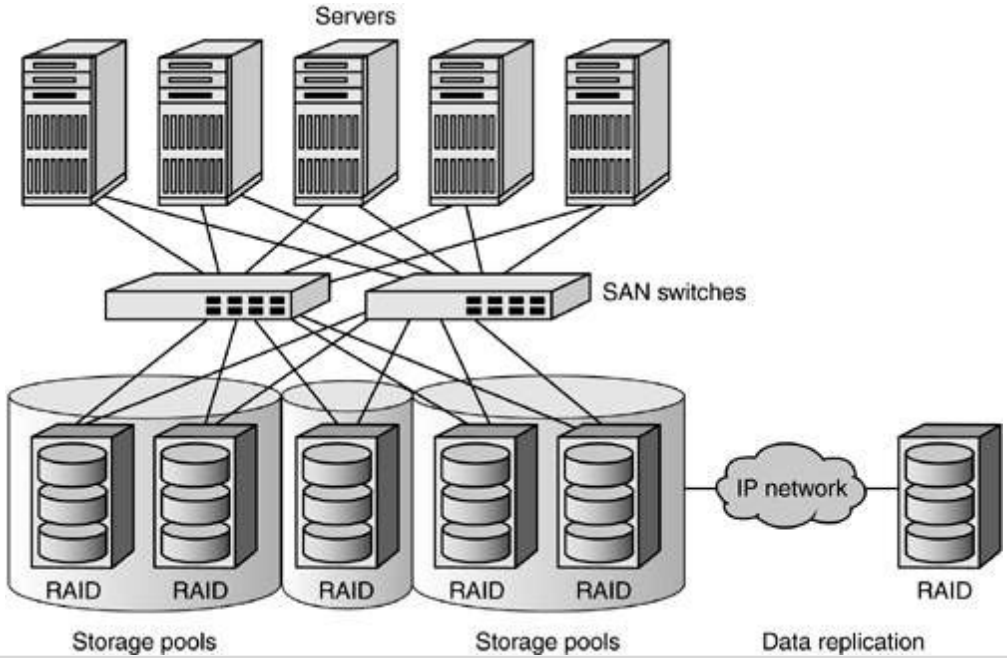


*SAN: Storage Area Network

Storage Device Based Storage Virtualization

- Enterprise-class storage arrays provide low-level virtualization via RAID and mirroring
  - Also support higher-level functions such as storage pooling
- Also provides direct disk-to-disk data replication
- Pro: Less likely to generate SAN-wide bottlenecks to performance
- Con: Proprietary to each vendor, and so they can be implemented only in homogeneous storage environments

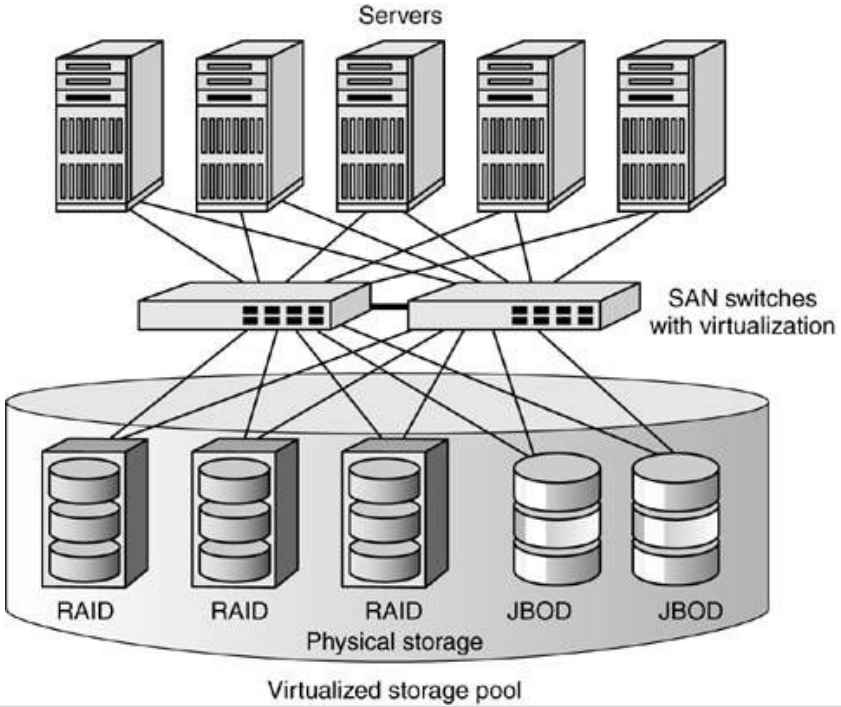
Storage Device Based Storage Virtualization



Network Based Storage Virtualization

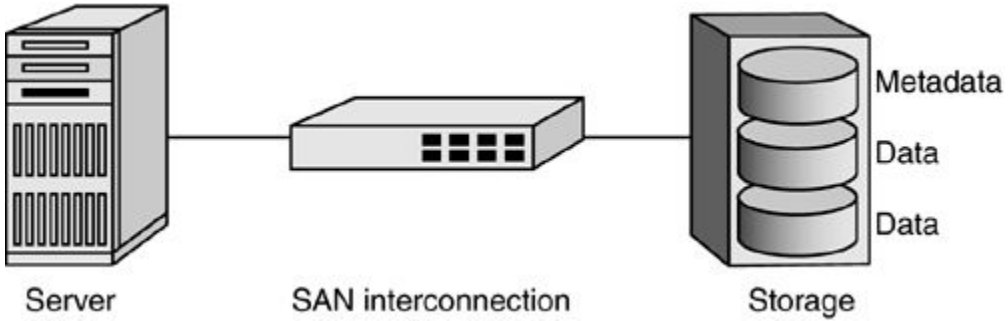
- Moving the virtualization intelligence into the network
  - No special agents are required on the host systems, and any mix of storage targets can be supported
  - Require sufficient power on the network
- Pro: Completely transparent to servers and can mix storage targets
- Con: Huge bottlenecks and complication of switches

Network Based Storage Virtualization



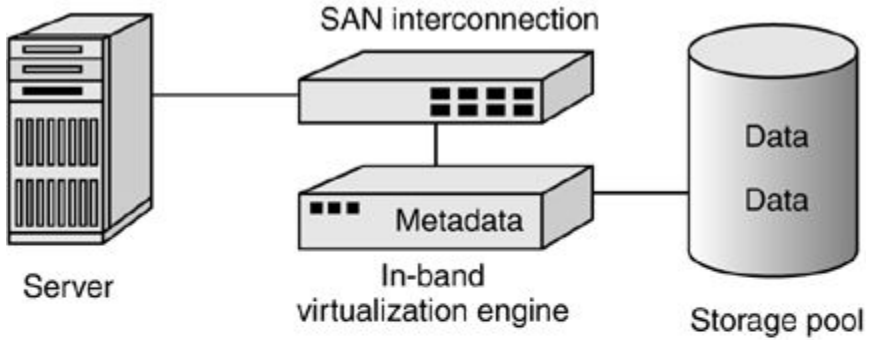
In-Band & Out-of-Band Storage Virtualization

- In storage virtualization, you need to have both storage and metadata of where the files are
- Two ways storing the metadata
  - In-Band
  - Out-of-Band



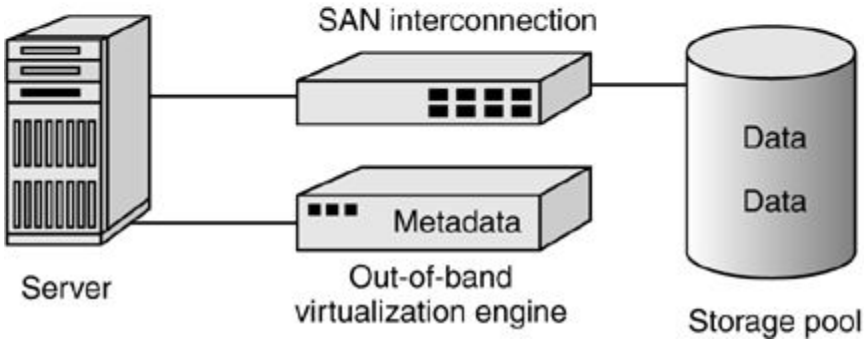
In-Band Storage Virtualization

- The metadata storage sits in the path from the client and the data
- Pro: Least intrusive and easier to implement
- Con: Metadata storage can be a bottleneck and a point of failure



Out-of-Band Storage Virtualization

- Putting metadata storage outside the data transport path
- Pro: Allows scaling of out-of-band metadata storage
- Con: Harder to configure



Benefits of Storage Virtualization

- Non-disruptive data migration
  - Users gain ability to migrate data while maintaining concurrent I/O access
  - When the data has been copied or moved, the meta-data can simply be updated to point to the new location
- Improved Utilization
  - Storage can be assigned where it is needed at that point in time, reducing the need to guess how much a given host will need in the future
- Simpler Management
  - Multiple disks are seen as a single block of storage

Disadvantages of Storage Virtualization

- Backing out a failed implementation
  - Once the abstraction layer is in place, only the virtualizer knows where the data actually resides on the physical medium
  - Backing out of a virtual storage environment therefore requires the reconstruction of the logical disks
- Interoperability and vendor support
- Added Complexity
  - Infrastructure design
  - Management of infrastructure
  - Harder to code the management layer
  - Metadata Management
- Performance and scalability
  - Caching can be harder to implement

Cloud Storage

What is Cloud Storage?

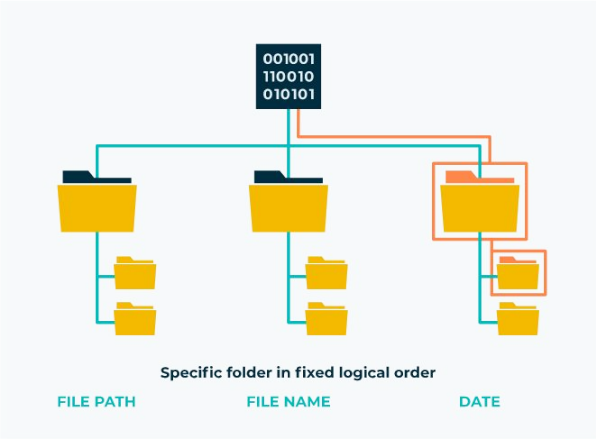
- Model of computer data storage in which the digital data is stored in logical pools over multiple servers in the internet (in the “cloud”)
- Servers are mostly managed by storage / cloud providers
  - Responsible for making storage available and accessible
  - Provides security to the data
- Data is accessed via APIs or web-based content management system

Types of Cloud Storage

- File Storage
  - Data stored as files
- Block Storage
  - Data stored as blocks in separate pieces
- Object Storage
  - Data is broken into pieces and spread out among hardware

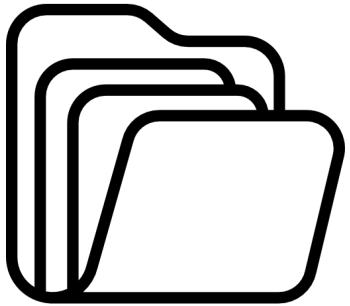
File Storage

- The traditional, old-school, approach to storage.
  - Each file gets a name
  - Store files in folders/directories and sub-directories
- Use Cases
  - File sharing / Collaboration
  - Backup and Recovery
- Pros
  - Easy to understand and manage at small scale
  - Users can manage their own files
- Cons
  - Hard and expensive to manage at larger scale
  - Hard to work with unstructured data



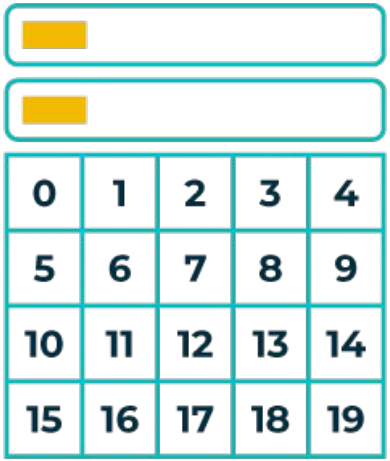
File Storage Examples

- Amazon Elastic File System
- Azure Files
- Google Cloud Filestore



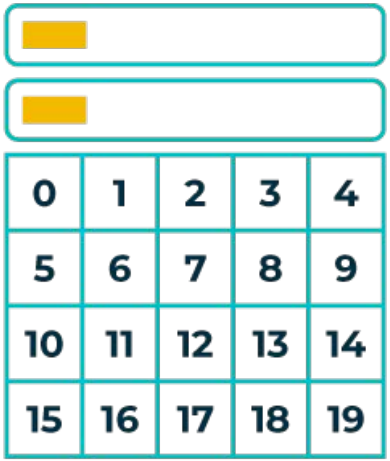
Block Storage

- Data split into fixed blocks of data and store separately
  - Blocks are given an ID
  - Reassembled at retrieval
- Use Cases
  - Databases
  - Virtual machine file system
- Pros
  - Fast and reliable
  - Easy to modify per block
- Cons
  - Lack of metadata, less usable for unstructured data
  - Not searchable and expensive



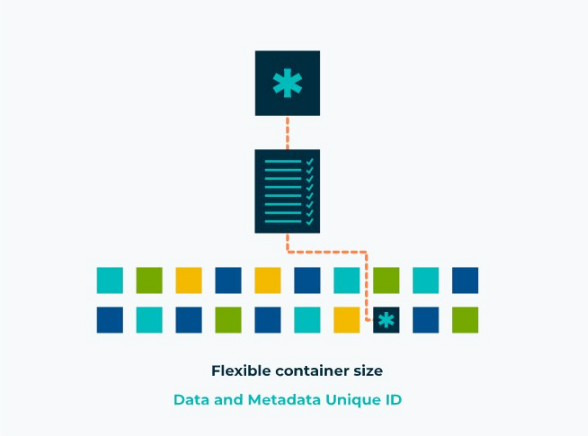
Block Storage Examples

- Amazon Elastic Block Store
  - Used to attach the volume to a virtual machine
  - Deploying databases
  - Varying types of volumes based on access patterns
- Azure Disks
- Google Persistent Disks
- Rackspace Cloud Block Storage



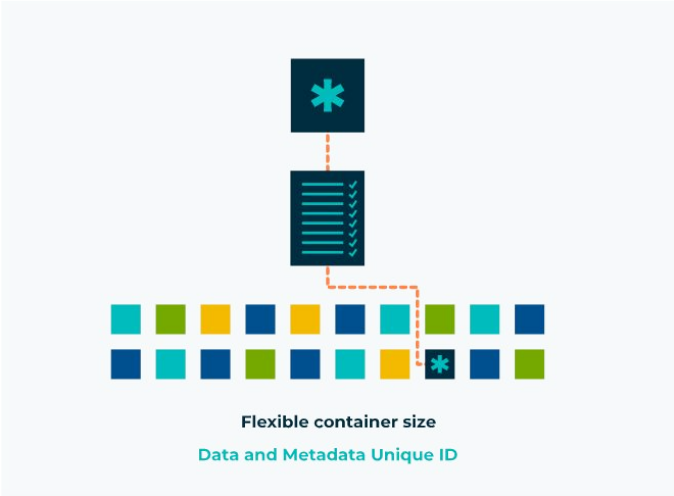
Object Storage

- Divides data into self-contained units stored in flat environment
  - All objects are at the same level, no sub-directories
  - Data is split up and requires metadata to access
- Use Cases
  - Store unstructured data (even large data set)
  - Store database dumps and logs
- Pros
  - Unlimited Scalability (Best for Machine Learning)
  - Easy to search
- Cons
  - Cannot change the object once created
  - Slow performance



Object Storage Example

- Amazon S3
  - Widely used!
- Azure
- Google Cloud Store

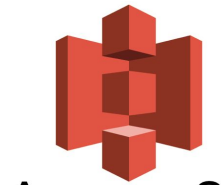


Amazon S3

- Amazon Simple Storage Service
  - Started in 2006
- Uses terms that are widely used
  - Buckets: A storage collection
  - Objects: A sequence of bytes in buckets
    - Can be up to 5TB
    - Referred to by user selected keys
- Provides security to the files

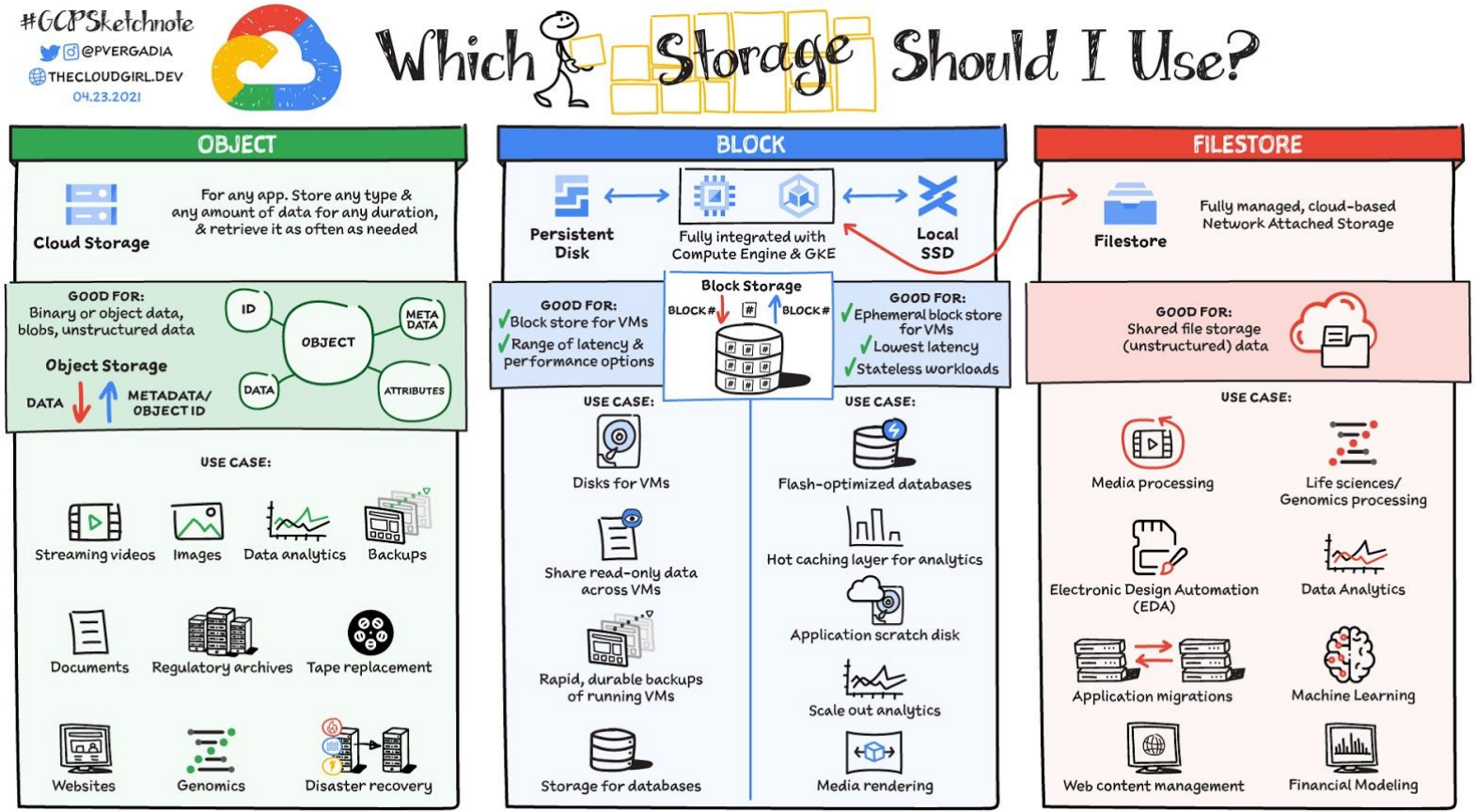
Amazon S3

- S3 Objects have their on URI (Uniform Resource Identifier)
  - https://my-bucket.s3.us-west-2.amazonaws.com/puppy.png
  - https://s3.us-west-2.amazonaws.com/mybucket/puppy.jpg
  - s3://mybucket/puppy.jpg
- All the URIs are in a flat structure
- Buckets provide top-level namespace
  - Charged based on each bucket
  - Reporting on each bucket
  - Access control for each bucket



Which Type of Storage Would You Use? Why?

- Email storage
- Video surveillance
- Creating document sharing system in your organization
- Store a large financial data
- Running a MySQL databases


#GCPSketchnote
@PVERGADIA
THECLOUDGIRL.DEV
04.23.2021
Which Storage Should I Use?
OBJECT
Cloud Storage
For any app. Store any type & any amount of data for any duration, & retrieve it as often as needed
GOOD FOR:
Binary or object data, blobs, unstructured data
Object Storage
DATA
METADATA/ OBJECT ID
ID
META DATA
OBJECT
DATA
ATTRIBUTES
USE CASE:
Streaming videos
Images
Data analytics
Backups
Documents
Regulatory archives
Tape replacement
Websites
Genomics
Disaster recovery
BLOCK
Persistent Disk
Fully integrated with Compute Engine & GKE
Local SSD
Block Storage
BLOCK#
BLOCK#
GOOD FOR:
Block store for VMs
Range of latency & performance options
GOOD FOR:
Ephemeral block store for VMs
Lowest latency
Stateless workloads
USE CASE:
Disks for VMs
Share read-only data across VMs
Rapid, durable backups of running VMs
Storage for databases
USE CASE:
Flash-optimized databases
Hot caching layer for analytics
Application scratch disk
Scale out analytics
Media rendering
FILESTORE
Filestore
Fully managed, cloud-based Network Attached Storage
GOOD FOR:
Shared file storage (unstructured) data
USE CASE:
Media processing
Life sciences/ Genomics processing
Electronic Design Automation (EDA)
Data Analytics
Application migrations
Machine Learning
Web content management
Financial Modeling

Cost of Storage

- Which one costs less?
  - Depends!
- Cost can vary based on access patterns
  - Object storage charges for number of requests
  - Block storage charges for IOPS
  - File storage pays for exactly what you use, but more expensive

Digression: Amazon Snowball

- Hard-disk shipping service
- Hardened storage appliances are shipped from Amazon
  - up to 100 petabytes (how much terabytes?)
- Client transfers data on/off and Prepaid courier service returns the device (& data) to Amazon
- Why do you want to do this?

Digression: Amazon Snowball

- Time to transfer is long for large amounts of data!

Available Internet Connection	Theoretical Min. Number of Days to Transfer 100TB at 80% Network Utilization	When to Consider AWS Snowball?
T3 (44.736Mbps)	269 days	2TB or more
100Mbps	120 days	5TB or more
1000Mbps	12 days	60TB or more

Cloud Databases

Types of Databases

- SQL
  - Traditional vs Cloud
  - Self-managed vs Fully Managed
- NoSQL
  - Key-value store
  - Document
  - Columnar
  - Graph

SQL Database

- Also called a relational database
- Uses a language called SQL to query data
  - `SELECT * FROM users;`
- Examples: MySQL, Oracle, PostgreSQL
- What is SQL Database good for?
  - Normalized/structured Data
  - Transactions
  - High data integrity
  - Complex queries

SQL Database on Cloud

- SQL databases are hard to setup and manage
- Not built to easily scale horizontally and vertically
- Hard to set up security
- Hard to maintain reliability and availability
- Managed services to the rescue!

SQL Database on Cloud

- Amazon RDS, Cloud SQL, Azure SQL
  - Fully-managed Relational Database Services
  - Scales horizontally and vertically automatically
- Amazon Redshift
  - Columnar database
  - Automatically scales to petabytes
  - Supports SQL for analytical queries
- Amazon Athena, Google BigQuery (?)
  - Runs SQL over Object Storage
  - Used for queries over large data
  - Serverless

NoSQL Database

- Databases that do not support SQL
- Various ways to read data
- Examples: MongoDB, Cassandra, Neo4J, HBase
- What is NoSQL Database good for?
  - Unstructured Data
  - Hierarchical data storage
  - Fast insertion, large amount of data
  - Simple queries

NoSQL Database on Cloud

- Using NoSQL often require speed and scalability
  - Often require dynamic scalability
- Hard to set up security
- Hard to maintain reliability and availability
- Require replication at low cost

NoSQL Database on Cloud

- Amazon DynamoDB, Google BigTable, Azure CosmoDB, MongoDB
  - Key-value
  - Document store
    - Data stored in loosely structured format
- Cassandra and Hbase (Managed on EMR)
  - Wide column store
  - Schema free
- AWS Neptune
  - Graph Database
  - Best for storing relations

Choosing the Correct Database On Cloud

- Many factors to consider
  - Cost
  - Amount of Storage
  - Speed
  - Use cases
- Often require more than one solution with data migration tools
  - E.g., SQL for transactional data and NoSQL for fast inserts

Agenda for Today

- Final Presentation and Report Guideline
- Storage Virtualization
- Cloud Storage
- Cloud Databases
- Readings
  - Recommended:
    - CCSA Chapter 5,6,7
    - https://www.enterprisestorageforum.com/hardware/storage-virtualization/
  - Optional: None

TODOs!

- HW 3
- Final Project
- Quiz 4 coming out soon!

Questions?