CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

Spring 2022 L25

RAIDs, Data Centers

Slides based on

- Text by Silberschatz, Galvin, Gagne
- Various sources

FAQ

- LAN: local area network
- WAN: wide area network consisting of many LANs
- Page$_{memory}$ vs blocks/sectors$_{disk}$
- Difference among a file, its inode, and inode number?
  - inode number is the index of the inode in the inode table
- Hard links vs symbolic links:
  - Hard links refer to the same inode
  - Symbol link file is a pointer

CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

Reliability & RAIDs

- Various sources

Reliability

- Storage is inherently unreliable. How can it be made more reliable?
- Redundancy
  - Complete mirrors of data: 2, 3 or more copies.
    - Use a good copy when there is failure,
    - Additional bits: Use parity bit/bits.
    - Use parity to reconstruct corrupted data
  - Rollback and retry
    - Go back to previously saved known good state and re-compute.

RAID Structure

- RAID – redundant array of inexpensive/independent disks. Multiple disk drives provides
  - **Higher reliability, repair capability**
  - **Higher performance /storage capacity**
  - **A combination**
- Increases the **mean time to failure**
- **Mean time to repair –** exposure time when another failure could cause data loss
- **Mean time to data loss** based on above factors

RAID Techniques

- **Striping** uses multiple disks in parallel by splitting data: higher performance, no redundancy (ex. RAID 0)
- **Mirroring** keeps duplicate of each disk: higher reliability (ex. RAID 1)
- **Block parity: One Disk hold** parity block for other disks. A failed disk can be rebuilt using parity. Wear leveling if interleaved (RAID 5, double parity RAID 6).
- Ideas that did not work: Bit or byte level level striping (RAID 2, 3) Bit level Coding theory (RAID 2), dedicated parity disk (RAID 4).
- Nested Combinations:
  - RAID 01: Mirror RAID 0
  - RAID 10: Multiple RAID 1, striping
  - RAID 50: Multiple RAID 5, striping
  - others

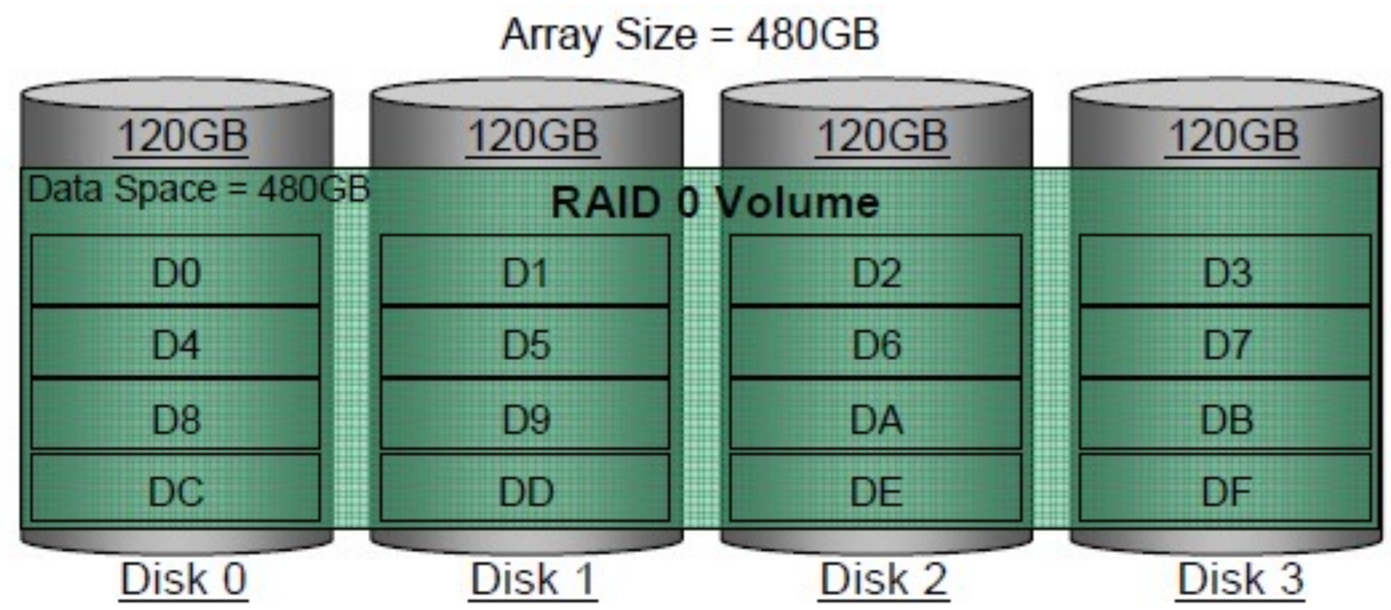
RAID

- Replicate data for availability
  - RAID 0: no replication
  - RAID 1: mirror data across two or more disks
    - Google File System replicated its data on three disks, spread across multiple racks
  - RAID 5: split data across disks, with redundancy to recover from a single disk failure
  - RAID 6: RAID 5, with extra redundancy to recover from two disk failures

Failures and repairs

- If a disk has *mean time to failure (MTTF) of* 100,000 hour.
  - Failure rate is 1/100,000 per hour.
- May be estimated using historical data
- If a disk has a bad data, it may be repaired
  - Copy data from a backup
  - Reconstruct data using available data and some invariant property.
- If data cannot be repaired, it is lost.

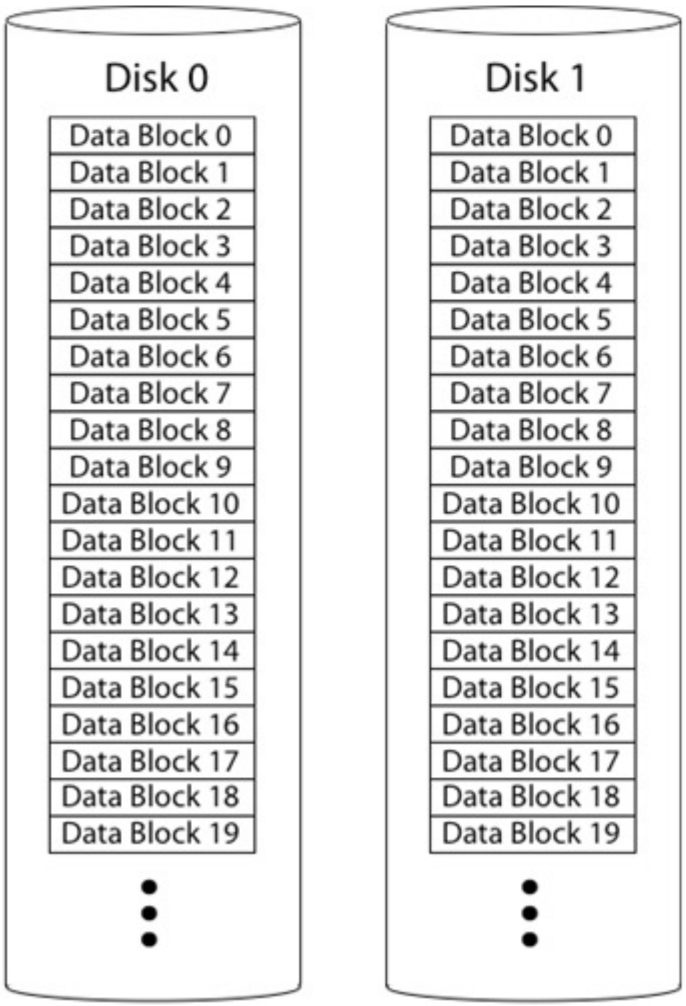
RAID 0: Striping



- Additional disks provide additional storage
- No redundancy

RAID 1: Mirroring

- Replicate writes to both disks
- Reads can go to either disk
- If they fail independently, consider disk with 100,000 hour *mean time to failure* and 10 hour *mean time to repair*
- One disk fails wile other is being repaired: data loss
  - probability that two will fail within 10 hours =
    $(2x10) \quad /100{,}000^2$
  - *Mean time to data loss* is
    $100{,}000^2/(2x10) = 500x10^6$
    hours, or 57,000 years!



Parity

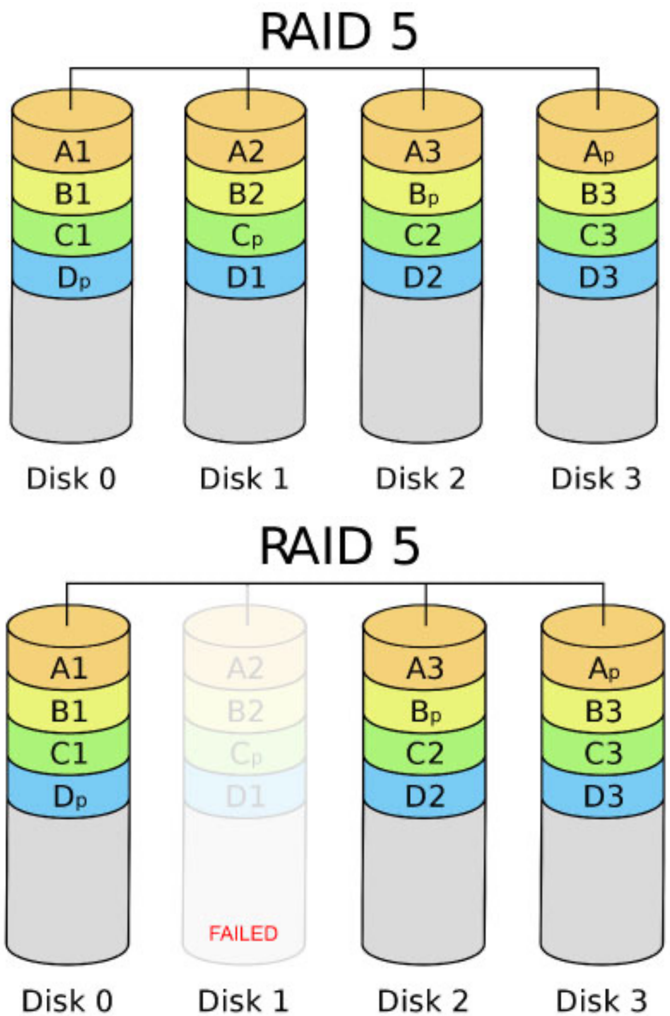
- Data blocks: Block1, block2, block3, ...
- Parity block: Block1 xor block2 xor block3 ...

10001101	block1
01101100	block2
11000110	block3

00100111	parity block (*ensures even number of 1s*)

- Can reconstruct any missing block from the others

RAID 5: Rotating Parity



Time to rebuild depends on disk capacity and data transfer rate

Read Errors and RAID recovery

- Example: RAID 5
  - Each bit has 10^{-15} probability of being bad.
  - 10 one-TB disks, and 1 disk fails
  - Read remaining disks to reconstruct missing data
- Probability of an error in reading 9 TB disks =

  10^{-15}*total bits =10^{-15}* (9 disks * 8 bits * 10^{12} bytes/disk)

  = 7.2% Thus recovery probability = 92.8%
- Even better:
  - RAID-6: two redundant disk blocks parity plus Reed-Solomon code
  - Can work even in presence of one bad disk, can recover from 2 disk failures
  - Scrubbing: read disk sectors in background to find and fix latent errors

CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

Big Data: HDFS and map-reduce

- Various sources, mostly external

Hadoop: Distributed Framework for Big Data

Big Data attributes:

- Large volume: TB -> PB varies with Kryder's law: disk density doubles / 13 months
- Geographically Distributed: minimize data movement
- Needs: reliability, analytic approaches

History:

- Google file system 2003 and Map Reduce 2004 programming lang
- Hadoop to support distribution for the Yahoo search engine project '05, given to Apache Software Foundation '06
- Hadoop ecosystem evolves with Yarn '13 resource management, Pig '10 scripting, Spark '14 distributed computing engine. etc.

- *The Google file system* by Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung (2003)
- *MapReduce: Simplified Data Processing on Large Clusters.* by Jeffrey Dean and Sanjay Ghemawat (2004)

Hadoop: Distributed Framework for Big Data

Recent development.

- Big data: multi-terabyte or more data for an app
- Distributed file system
  - Reliability through replication (Fault tolerance)
- Distributed execution
  - Parallel execution for higher performance

Hadoop: Core components

Hadoop (originally): HDFS + MapReduce

- HDFS: A **d**istributed **f**ile **s**ystem designed to efficiently allocate data across multiple commodity machines, and provide self-healing functions when some of them go down
- MapReduce: A programming framework for processing parallelizable problems across huge datasets using a large number of commodity machines.

- Commodity machines: lower performance per machine, lower cost, perhaps lower reliability compared with special high-performance machines.

Challenges in Distributed Big Data

Common Challenges in Distributed Systems

- **Node Failure**: Individual computer nodes may overheat, crash, have hard drive failures, or run out of memory or disk space.
- **Network issues**: Congestion/delays (large data volumes), Communication Failures.
- **Bad data**: Data may be corrupted, or maliciously or improperly transmitted.
- **Other issues**: Multiple versions of client software may use slightly different protocols from one another.
- **Security**

HDFS Architecture

Hadoop Distributed File System (HDFS):

- HDFS Block size: 64-128 MB ext4: 4KB
- HDFS file size: “Big”
- Single HDFS FS cluster can span many nodes possibly geographically distributed. datacenters-racks-blades
- Node: system with CPU and memory

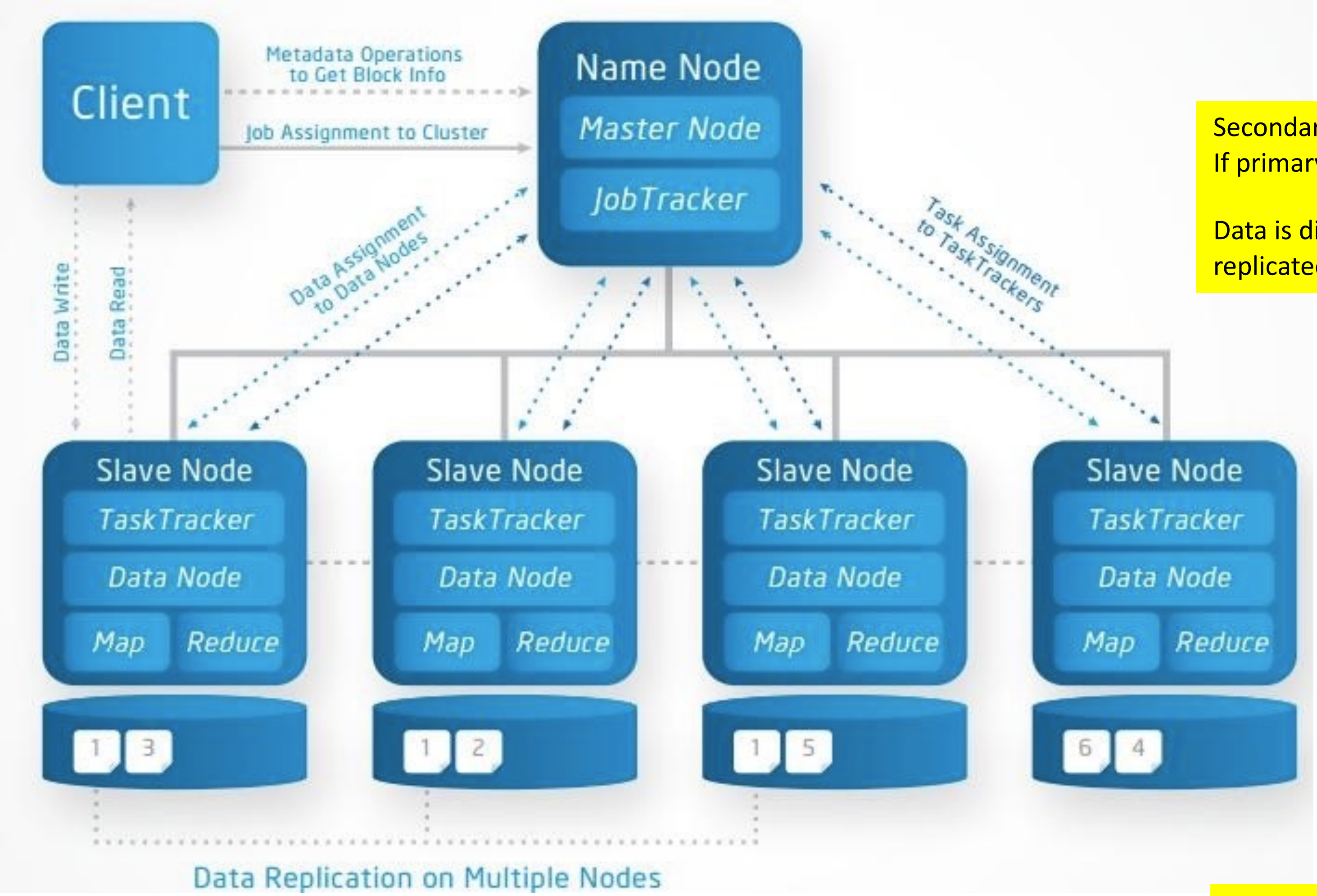
Metadata (corresponding to superblocks, Inodes)

- Name Node: metadata giving where blocks are physically located

Data (files blocks)

- Data Nodes: hold blocks of files (files are distributed)

HDFS Architecture



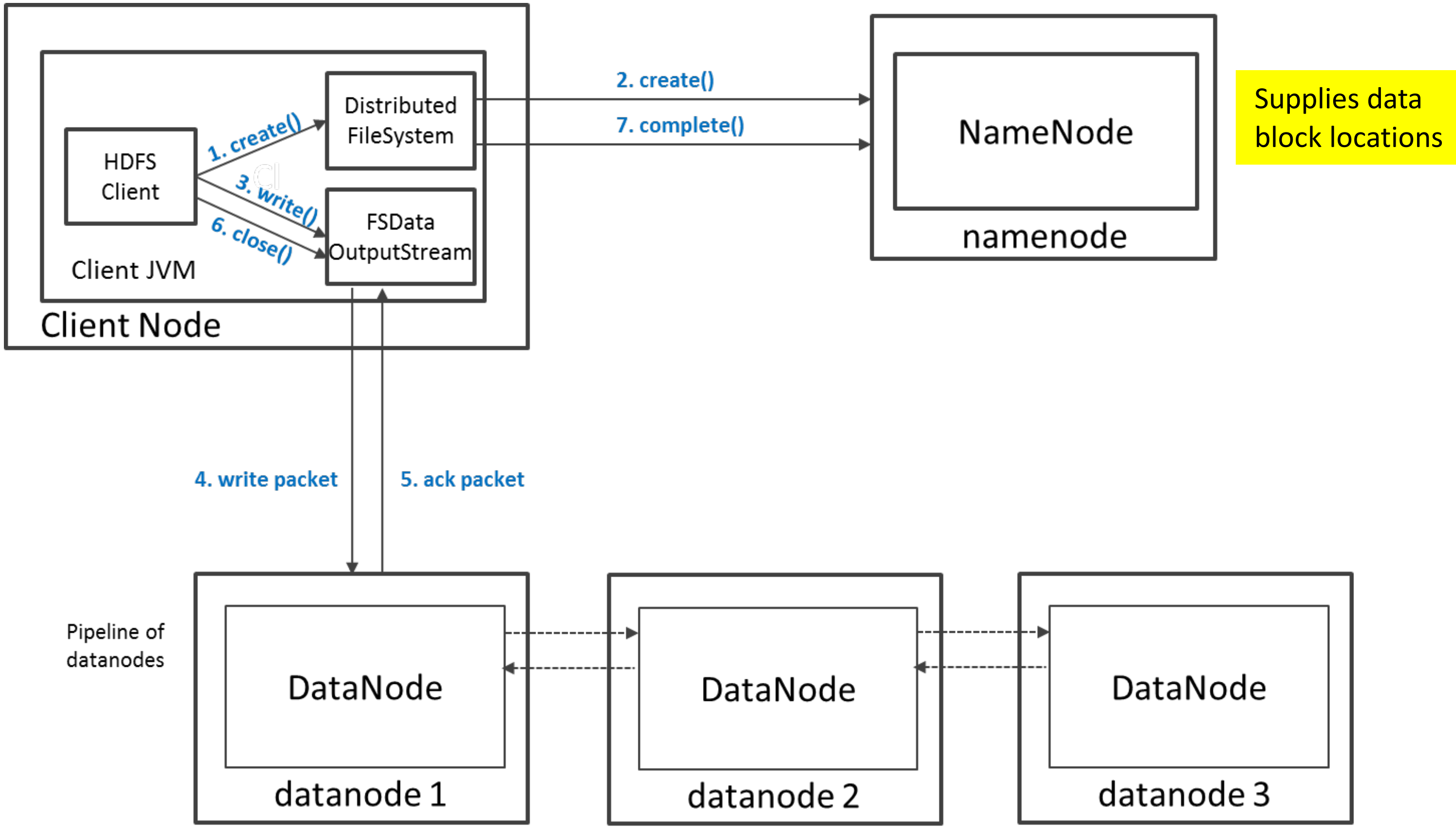
Secondary Name node If primary fails.

Data is distributed and replicated.

http://a4academics.com/images/hadoop/Hadoop-Architecture-Read-Write.jpg

Slave nodes have been renamed worker nodes.

HDFS Write operation



https://indico.cern.ch/event/404527/contributions/968835/attachments/1123385/1603232/Introduction_to_HDFS.pdf

CERN

HDFS Fault-tolerance

- Disks use error detecting codes to detect corruption.
- Individual node/rack may fail.
- Data Nodes (on slave nodes):
  - data is replicated. Default is 3 times. Keep a copy far away.
  - Send periodic heartbeat (I'm OK) to Name Nodes. Perhaps once every 10 minutes.
  - Name node creates another copy if no heartbeat.

HDFS Fault-tolerance

Name Node (on master node) Protection:

- Transaction log for file deletes/adds, etc. Creation of more replica blocks, when necessary, after a Data Node failure
- Standby name node: namespace backup
  - In the event of a failover, the Standby will ensure that it has read all of the edits from the Journal Nodes and then promotes itself to the Active state
  - Implementation/delay version dependent

Name Node metadata is in RAM as well as checkpointed on disk.
On disk the state is stored in two files:

- fsimage: Snapshot of file system metadata
- editlog: Changes since last snapshot

HDFS Command line interface

- hadoop fs –help
- hadoop fs –ls : List a directory
- hadoop fs mkdir : makes a directory in HDFS
- hadoop fs –rm : Deletes a file in HDFS
- copyFromLocal : Copies data to HDFS from local filesystem
- copyToLocal : Copies data to local filesystem
- Java code can read or write HDFS files (URI) directly

HDFS is on top of a local file system

https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

Distributing Tasks

MapReduce Engine:

- JobTracker splits up the job into smaller tasks(“Map”) and sends it to the TaskTracker process in each node.
- TaskTracker reports back to the JobTracker node and reports on job progress, sends partial results (“Reduce”) or requests new jobs.
- Tasks are run on local data, thus avoiding movement of bulk data.
- Originally developed for search engine implementation.

Hadoop Ecosystem Evolution

- Hadoop YARN: A framework for job scheduling and cluster resource management, can run on top of Windows Azure or Amazon S3.
- Apache spark is more general, faster and easier to program than MapReduce.
    - Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing, Berkeley, 2012

tate University

CS370 Operating Systems

Colorado State University
Yashwant K Malaiya
Spring 2022

Data Centers & Cloud Computing

Slides based on

- Text by Silberschatz, Galvin, Gagne
- Various sources

Data Centers

- Large server and storage farms
  - 1000s-100,000 of servers
  - Many PBs of data
- Used by
  - Enterprises for server applications
  - Internet companies
  - Some of the biggest DCs are owned by Google, Facebook, etc
- Used for
  - Data processing
  - Web sites
  - Business apps

Data Center architecture

Traditional - static

- Applications run on physical servers
- System administrators monitor and manually manage servers
- Storage Array Networks (SAN) or Network Attached Storage (NAS) to hold data

Modern – dynamic with larger scale

- Run applications inside virtual machines
- Flexible mapping from virtual to physical resources
- Increased automation, larger scale

Data Center architecture

Giant warehouses with:

- Racks of servers
- Storage arrays
- Cooling infrastructure
- Power converters
- Backup generators

Or with containers

- Each container filled with thousands of servers
- Can easily add new containers
- "Plug and play"
- Pre-assembled, cheaper, easily expanded

Server Virtualization

Allows a server to be "sliced" into Virtual Machines

- VM has own OS/applications
- Rapidly adjust resource allocations
- VM migration within a LAN

- Virtual Servers
  - Consolidate servers
  - Faster deployment
  - Easier maintenance
- Virtual Desktops
  - Host employee desktops in VMs
  - Remote access with thin clients
  - Desktop is available anywhere
  - • Easier to manage and maintain

Data Center Challenges

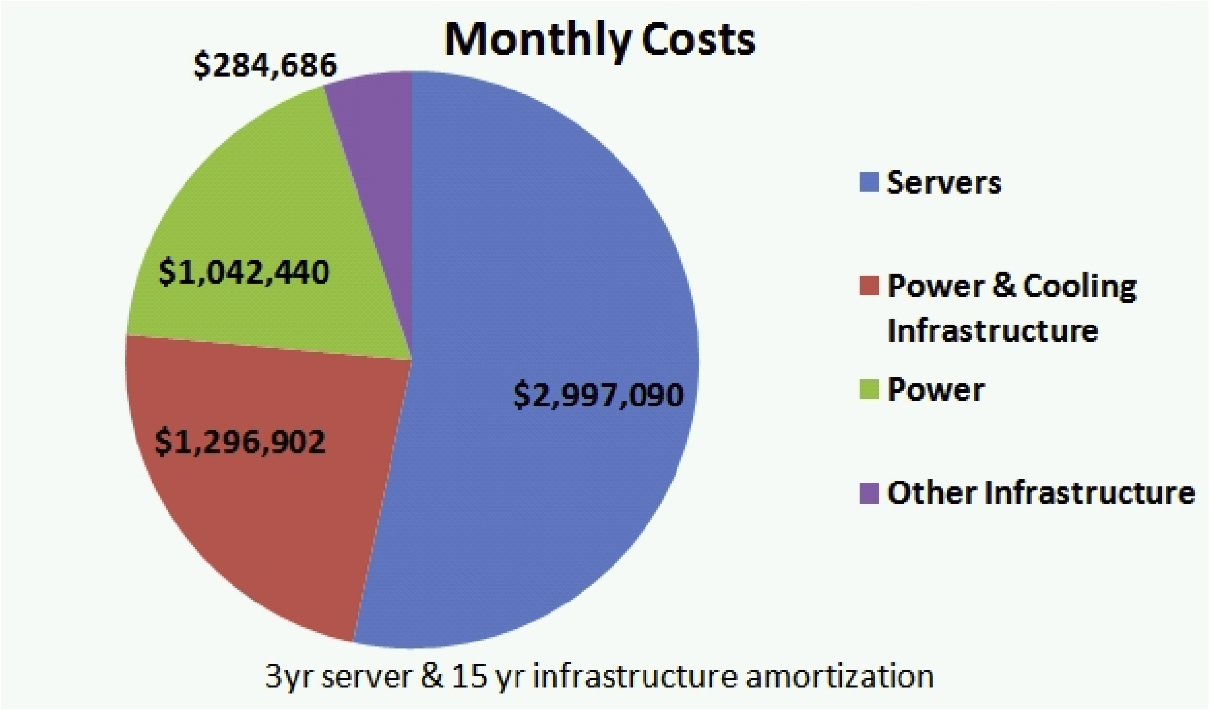
Resource management

- How to efficiently use server and storage resources?
- Many apps have variable, unpredictable workloads
- Want high performance and low cost
- Automated resource management
- Performance profiling and prediction

Energy Efficiency

- Servers consume huge amounts of energy
- Want to be "green"
- Want to save money

Data Center Challenges



Power Efficiency measured as *Power Usage Effectiveness*

- *Power Usage Effectiveness* = Ratio of IT Power / Total Power
- typical: 1.7, Google PUE ~ 1.1)

http://perspectives.mvdirona.com/2008/11/28/CostOfPowerInLargeScaleDataCenters.aspx

Economy of Scale

Larger data centers can be cheaper to buy and run than smaller ones

- Lower prices for buying equipment in bulk
- Cheaper energy rates
- Automation allows small number of sys admins to manage thousands of servers
- General trend is towards larger mega data centers
- 100,000s of servers
- Has helped grow the popularity of cloud computing

Economy of Scale

Resource	Cost in Medium DC	Cost in Very Large DC	Ratio
CPU cycle cost	2 picocents	< 0.5 picocents	
Network	$95 / Mbps / month	$13 / Mbps / month	7.1x
Storage	$2.20 / GB / month	$0.40 / GB / month	5.7x
Administration	≈140 servers/admin	>1000 servers/admin	7.1x

Pico = 10^{-3} nano = 10^{-12}

Data Center Challenges

Reliability Challenges

Typical failures in a year of a Google data center:

- 20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
- 3 router failures (have to immediately pull traffic for an hour)
- 1000 individual machine failures
- thousands of hard drive failures

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/people/jeff/stanford-295-talk.pdf

CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

ICQ 4/27/21

Utilization

What's the typical utilization rate for a non-virtualized server?

A. 2% to 3%
B. 5% to 15%
C. 25% to 40%
D. 50% to 80%

Moving

2. Moving virtual workloads from one physical server to another with no downtime is called:

A. Server provisioning
B. Disaster recovery
C. High availability
D. Live migration

Answers

CS370 Operating Systems

Colorado State University

Yashwant K Malaiya

Back from ICQ

Capacity provisioning

User has a variable need for capacity. User can choose among

Fixed resources: Private data center

- Under-provisioning when demand is too high, or
- Provisioning for peak

Variable resources:

- Use more or less depending on demand
- Public Cloud has elastic capacity (i.e. way more than what the user needs)
- User can get exactly the capacity from the Cloud that is actually needed

Why does this work for the provider?

  - Varying demand is statistically smoothed out over many users, their peaks may occur at different times
  - Prices set low for low overall demand periods

Amazon EC2 Instance types

On-Demand instances

- Users that prefer the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted

Spot Instances (cheap)

- request spare Amazon EC2 computing capacity for up to 90% off
- Applications that have flexible start and end times

Reserved Instances (expensive)

- Applications with steady state usage
- Applications that may require reserved capacity

Dedicated Hosts

- physical EC2 server dedicated for your use.
- server-bound software licenses, or meet compliance requirements

Amazon EC2 Prices (samples from their site)

General Purpose - Current Generation Region: US East (Ohio)

instance	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
t2.nano	1	Variable	0.5	EBS Only	$0.0058 per Hour
t2.small	1	Variable	2	EBS Only	$0.023 per Hour
t2.medium	2	Variable	4	EBS Only	$0.0464 per Hour
m5.4xlarge	16	61	64	EBS Only	$0.768 per Hour
m4.16xlarge	64	188	256	EBS Only	$3.2 per Hour

ECU = EC2 Compute Unit (perf), EBS: elastic block store (storage) , automatically replicated

Host OS answer

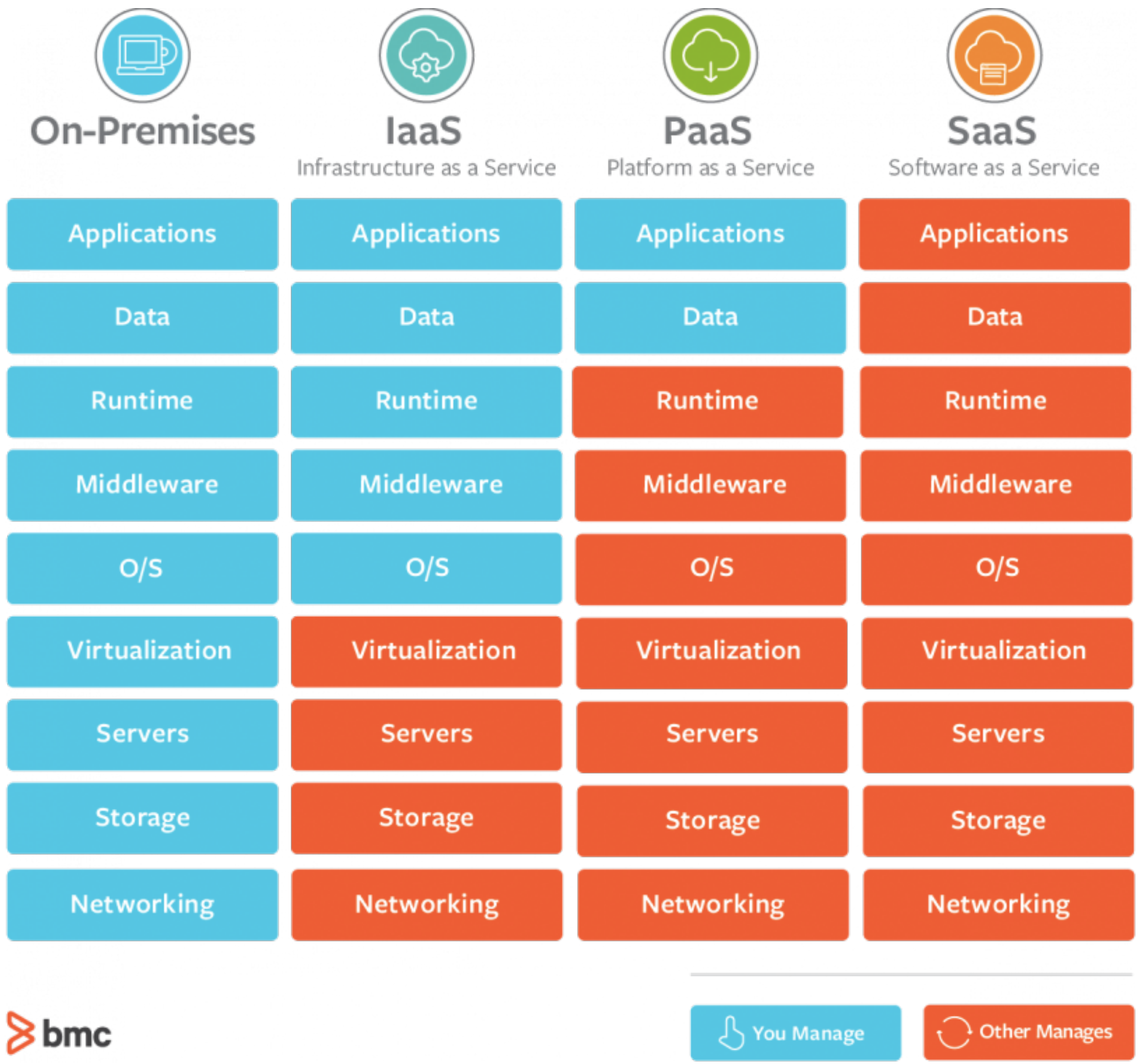
1. **In Type 1 VMM, is there a host OS?** **No.** Hypervisor services the guest Oss.

2. **Can a single hypervisor manage VMs with different OSs, win, linux, MacOS?** **Yes**

Service models

- IaaS: Infrastructure as a Service
  - infrastructure components traditionally present in an on-premises data center, including servers, storage and networking hardware
  - e.g., Amazon EC2, Microsoft Azure, Google Compute Engine
- PaaS: Platform as a Service
  - supplies an environment on which users can install applications and data sets
  - e.g., Google AppEngine, Heroku, Apache Stratos
- SaaS: Software as a Service
  - a software distribution model with provider hosted applications
  - Microsoft Office365, Amazon DynamoDB, Gmail

The Service Models



https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/

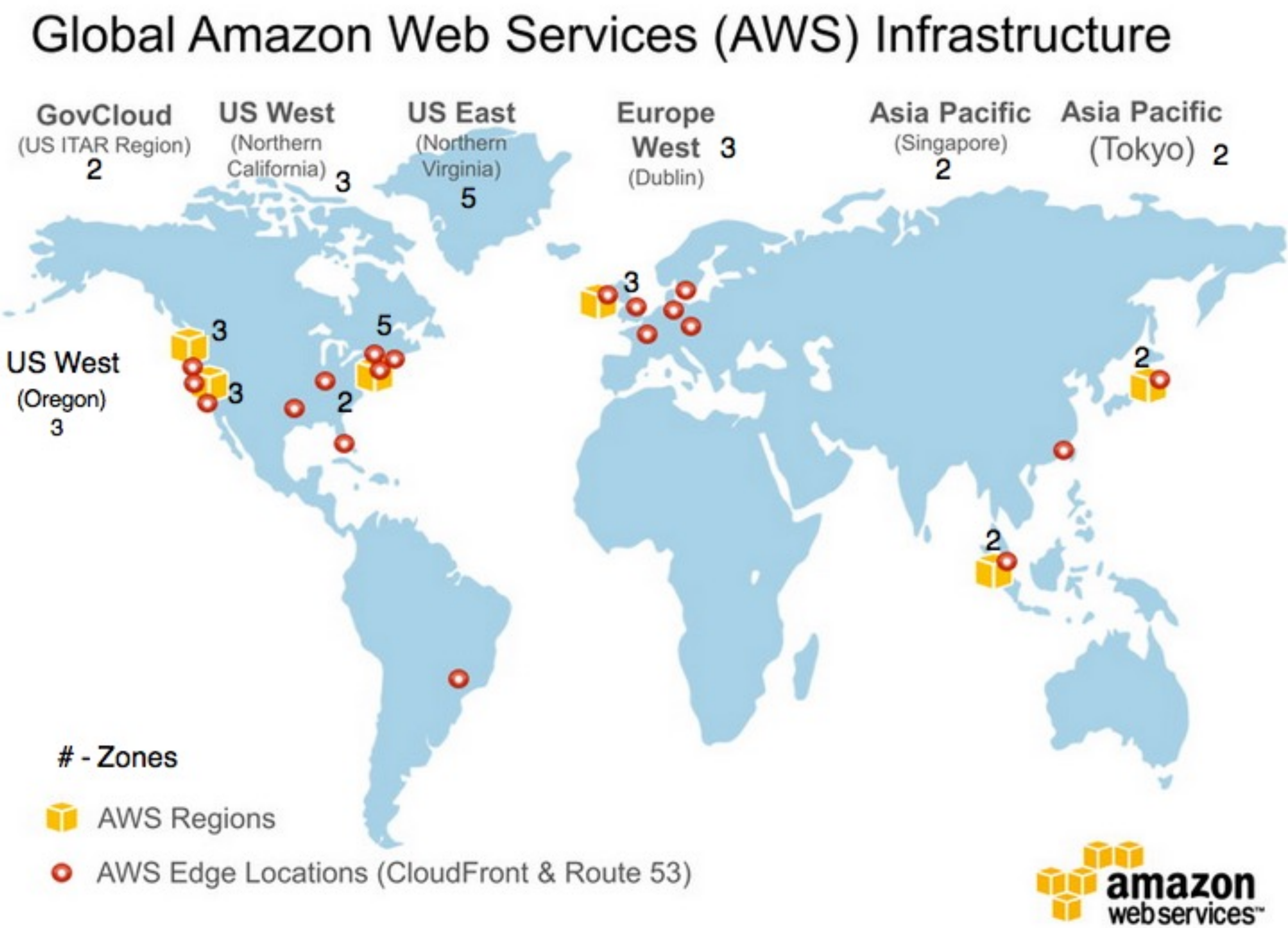
Cloud Management models

- **Public clouds**
  - Utility model
  - Shared hardware, no control of hardware,
  - Self-managed (e.g., AWS, Azure)
- **Private clouds**:
  - More isolated (secure?)
  - Federal compliance friendly
  - Customizable hardware and hardware sharing
- **Hybrid clouds**:
  - a mix of on-premises, private cloud and third-party, public cloud services.
  - Allows workloads to move between private and public clouds as computing needs and costs change.

Different Regions to Achieve HA

- AWS datacenters is divided into regions and zones,
  - that aid in achieving availability and disaster recovery capability.
- Provide option to create point-in-time snapshots to back up and restore data to achieve DR capabilities.
- The snapshot copy feature allows you to copy data to a different AWS region.
  - This is very helpful if your current region is unreachable or there is a need to create an instance in another region
  - You can then make your application highly available by setting the failover to another region.

Different Regions to Achieve HA



Colorado State University