CS370 Operating Systems

Colorado State University
Yashwant K Malaiya
Spring 2021  L27  Part B
Final Review

Slides based on
• Text by Silberschatz, Galvin, Gagne
• Various sources
Peer Review Needed

• You need to review
  – Two research projects
  – Two development projects
  – Members of your team
  – Identify best research and best development project

• Review form due May 8.
• Please finish course survey on Canvas by May 8.
• No ICQ this week, no quiz this coming weekend.
Study/Resources

• Terms, concepts, implementations, algorithms, problems
• Lecture slides
  – Also see Midterm Review Slides on website
  – Possible questions not limited to Review Slides
• Quizzes, assignments
• Textbook
• Extra:
  – Study Guide, Final Review Quiz (Under Quizzes)
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

Parity: allows rebuilding of a disk

Not common: RAID 2, 3, 4
Most common: RAID 5
Parity

- Parity block: Block1 xor block2 xor block3 ...

  10001101  block1
  01101100  block2
  \textcolor{red}{11000110}  block3

  \textcolor{red}{------------}

  00100111  parity block (\textit{ensures even number of 1s})

- Can reconstruct any bad block using all others
Read Errors and RAID recovery

• Example: RAID 5
  – 10 one-TB disks, and 1 fails
  – Read remaining disks to reconstruct missing data

• Probability of an error in reading 9 TB disks =
  \[10^{-15} \times (9 \text{ disks} \times 8 \text{ bits} \times 10^{12} \text{ bytes/disk})\]
  = 7.2% Thus recovery probability = 92.8%

• Even better:
  – RAID-6: two redundant disk blocks
  – Can work even in presence of one bad disk
  – Scrubbing: read disk sectors in background to find and fix latent errors
Hadoop: Core components

- **Hadoop (originally):** MapReduce + HDFS
- **For Big Data** applications.
- **MapReduce:** A programming framework for processing parallelizable problems across huge datasets using a large number of commodity machines.
- **HDFS:** A distributed file system designed to efficiently allocate data across multiple machines, and provide self-healing functions when some of them go down.
Name Node: metadata, where blocks are physically located
Data Nodes: hold blocks of files (files are distributed)

HDFS Block size: 64-128 MB
ext4: 4KB
HDFS is on top of a local file system.

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

- Individual node/rack may fail.
  - Disks use error detecting codes to detect corruption.
- **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.
- **Name Node (on master node) Protection:**
  - Transaction log for file deletes/adds, etc (only metadata recorded).
  - Creation of more replica blocks when necessary after a DataNode 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 of VMMs

- **Type 1 hypervisors** - Operating-system-like software built to provide virtualization. Runs on ‘bare metal’.
  - Including VMware ESX, Joyent SmartOS, and Citrix XenServer
- Also includes general-purpose operating systems that provide standard functions as well as VMM functions
  - Including Microsoft Windows Server with HyperV and RedHat Linux with KVM
- **Type 2 hypervisors** - Applications that run on standard operating systems but provide VMM features to guest operating systems
  - Including VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox
Memory Mapping:

- On a bare metal machine:
  - VPN -> PPN

- VMM: Real physical memory (machine memory) is shared by the OSs. Need to map PPN of each VM to MPN (Shadow page table).
  - PPN -> MPN

- Where is this done?
  - In Full virtualization?
Live Migration

- Migration from source VMM to target VMM
  - Source establishes a connection with the target
  - Target creates a new guest
  - Source sends all read-only memory pages to target
  - Source starts sending all read-write pages
  - Source VMM freezes guest, sends final stuff,
  - Once target acknowledge
Linux Containers and Docker

• Linux containers (LXC) are “lightweight” VMs
• Comparison between LXC/docker and VM

• Containers provide “OS-level Virtualization” vs “hardware level”.
• Containers can be deployed in seconds.
• Very little overhead during execution, just like Type 1.
Microservices Characteristics

• Many smaller (fine grained), clearly scoped services
  – Single Responsibility Principle
  – Independently Managed

• Clear ownership for each service
  – Typically need/adopt the “DevOps” model

• 100s of MicroServices
  – Need a Service Metadata Registry
    (Discovery Service)

• May be replicated as needed

• A microservice can be updated without interruption
Cloud 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
Cloud Instance types/Service/Management models

Instance types
- On-Demand instances
- Spot Instances
- Reserved Instances
- Dedicated Hosts

Service models
- IaaS: Infrastructure as a Service
- PaaS: Platform as a Service
- SaaS: Software as a Service

Cloud Management models
- Public clouds
- Private clouds
- Hybrid clouds:
Part B: HW6 Review

• Discussion of HW6 answers.
  – Annotated word document.

• They will not be posted according to our usual policy.