CS370 Operating Systems

Colorado State University

Yashwant K Malaiya

Fall 2021 L27

Containers & Data Centers

Slides based on

- Text by Silberschatz, Galvin, Gagne
- Various sources

Course Updates

- **Course Survey** is available on Canvas. Please fill the survey by coming Wednesday.
- **Project Slides** for both options need to be posted in Teams channel Project Slides and Videos by Dec 3.
- Research Project **Videos** (7-8 min) should also be posted there by Dec.3.
- Development Project **Demo schedule** will be available today. Each team should sign up for one 15-min slot (M,Tu,W). The link to the Signupgenius form will be posted on Teams today.

CS370 Operating Systems

Colorado State University

Yashwant K Malaiya

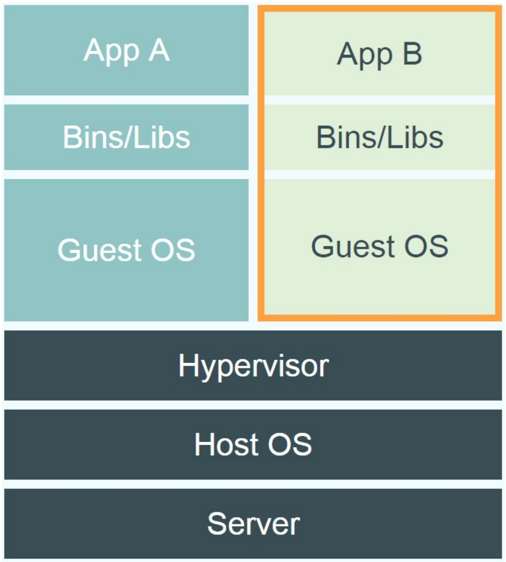
Fall 2021

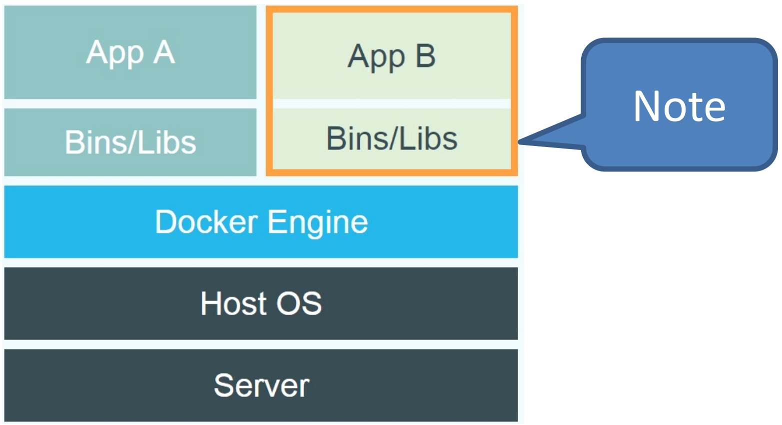
Containers

Slides based on

- Text by Silberschatz, Galvin, Gagne
- Various sources

- 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, even better than Type 1 VMM.

VMs vs Containers

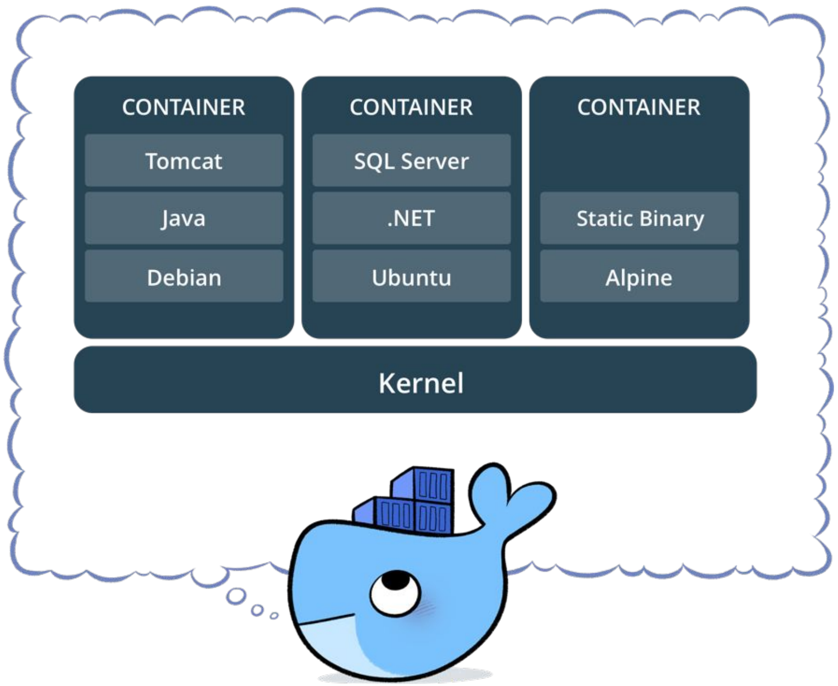
VMs	Containers
Heavyweight several GB	Lightweight tens of MB
Limited performance	Native performance
Each VM runs in its own OS	All containers share the host OS
Hardware-level virtualization	OS virtualization
Startup time in minutes	Startup time in milliseconds
Allocates required memory	Requires less memory space
Fully isolated and hence more secure	Process-level isolation, possibly less secure

Container: basis

Linux kernel provides

- "control groups" (cgroups) functionality for a set of processes
  - allows allocation and prioritization of resources (CPU, memory, block I/O, network, etc.) without the need for starting any VM
- "namespace isolation" functionality
  - allows complete isolation of an applications' view of the operating environment including Process trees, networking, user IDs and mounted file systems.
- Managed by Docker etc.
  - Docker: build, share, run and orchestrate containerized apps.
  - Kubernetes: orchestration platform for managing, automating, and scaling containerized applications

What is a container?



- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works for all major Linux distributions
- Docker Desktop for Windows uses Windows-native Hyper-V virtualization (Win10)
- Containers native to Windows Server 2016
- Docker: a popular container management service technology

CS370 Students: Future dreams and nightmares

- Man on Mars
- Other dreams
  - AI with human rights
  - More powerful phones
  - natural language models being used to preserve some of the 50%-90% of human languages that will otherwise disappear by the year 2100
  - I expect to see hoverboards in the next 20 years
  - Vaccination created dynamically from AI
  - Flying cars
- Nightmares
  - Cloud gaming virtually everywhere
  - working class be taken over by robots
  - resource wars in my lifetime caused by human greed and the inability to adapt to growing problems as a species
  - future will have less progression than the past decades
  - implants of small operating systems in human brains - next 20-40 years. (2)
  - anything that does need human intellect to operate will be done by a computer

Some Docker vocabulary

- **Docker Image**
  - The basis of a Docker container. Represents a full application
- **Docker Container**
  - The standard unit in which the application service resides and executes
- **Docker Engine**
  - Creates, ships and runs Docker containers deployable on a physical or virtual, host locally, in a datacenter or cloud service provider
- **Registry Service (Docker Hub(Public) or Docker Trusted Registry(Private))**
  - Cloud or server-based storage and distribution service for images (can be **pull**ed or **push**ed)

9

Correspondence: excecutable:image   container:process

Some Docker vocabulary

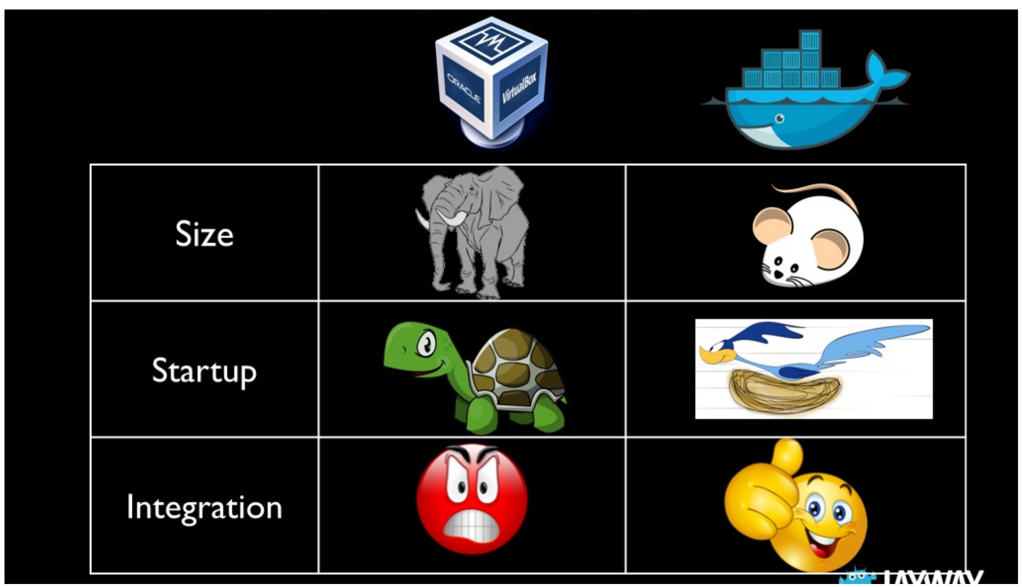
- **Dockerfile**: A text file with instructions to build image Automation of Docker Image Creation
  - Build dockerfile to create image
  - Run image to create container
- **Docker compose**: tool for defining & running multi-container docker applications
  - use yaml files to configure application services (docker-compose.yml)
  - can start all services with a single command: docker compose up
  - can stop all services with a single command: docker compose down
  - can scale up selected services when required
- Syntax, Commands, Examples: see documentation

Docker Volumes

- Volumes mount a directory on the host into the container at a specific location
- Can be used to share (and persist) data between containers
  - Directory persists after the container is deleted
    - Unless you explicitly delete it
- Can be created in a Dockerfile or via CLI

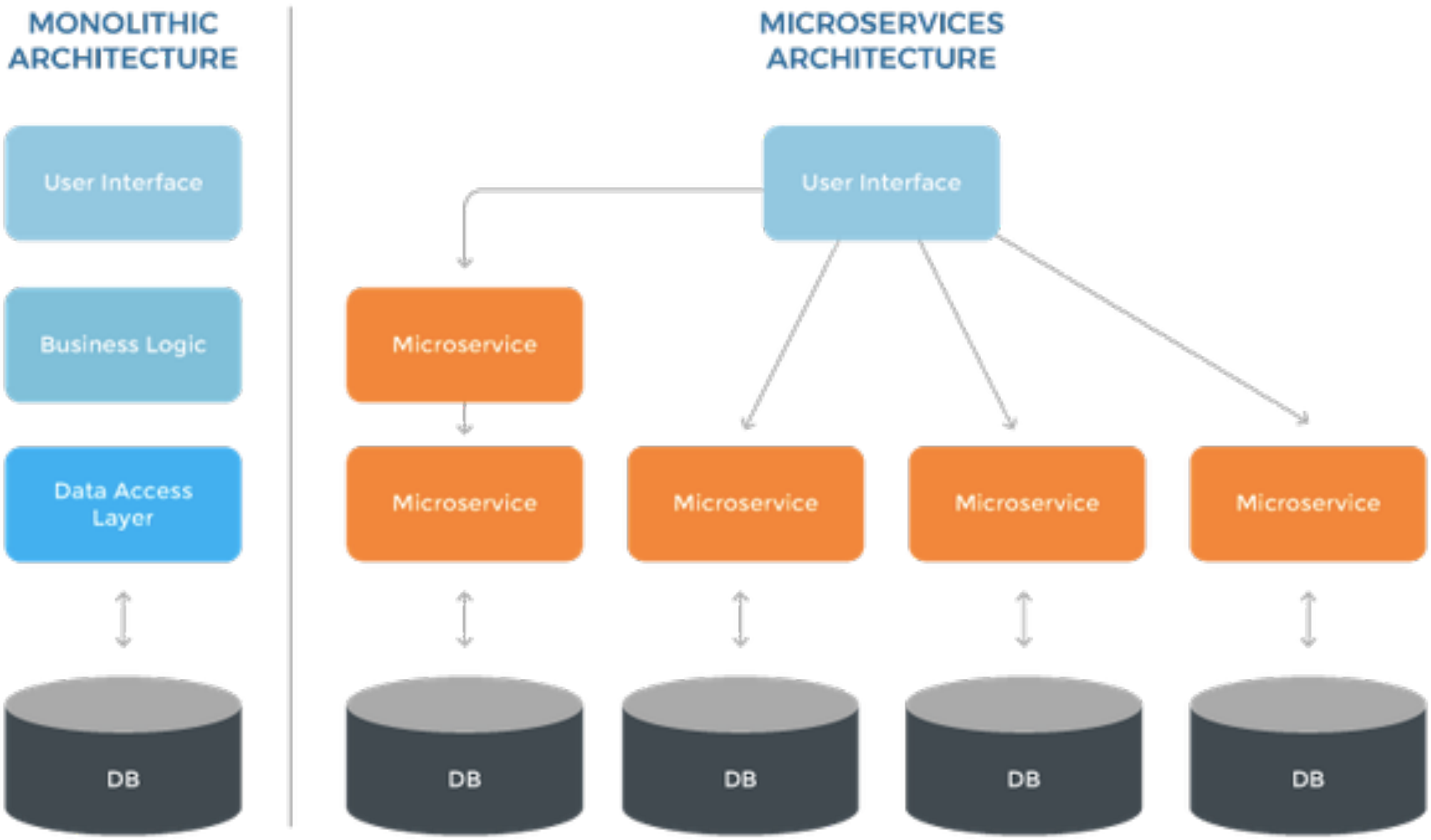
Containers and microservices

- Containers run in the user space
- Each container has it own: process space, network interface, booting mechanism with configuration
- Share kernel with the host
- Can be packaged as Docker images to provide *microservices*.



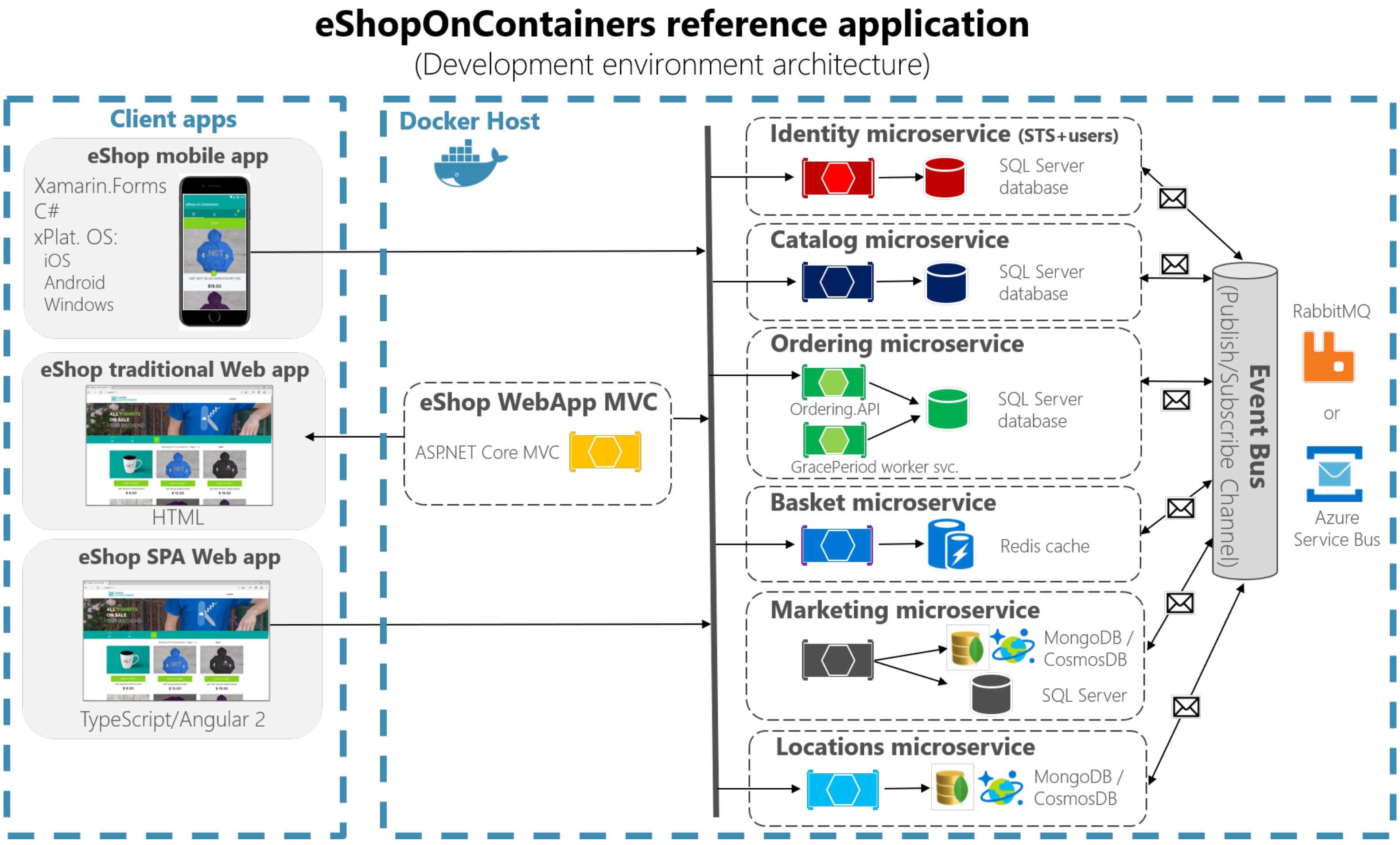
Colorado State University

Monolithic architecture vs microservices



Colorado State University

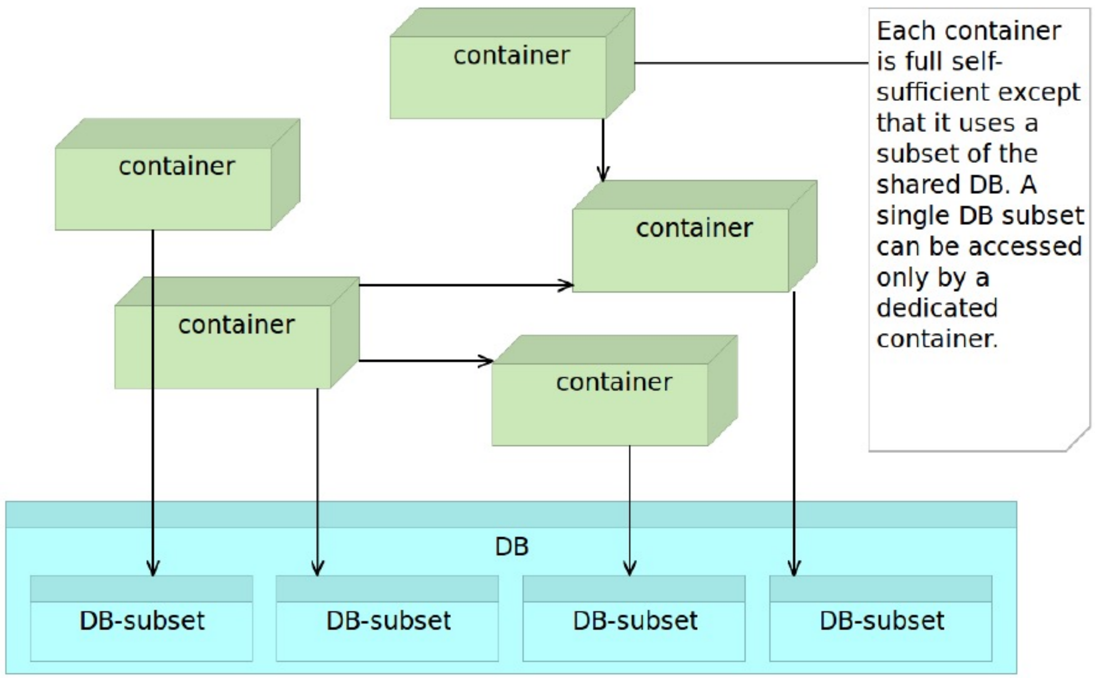
Microservices



https://docs.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/implement-api-gateways-with-ocelot

Colorado State University

Microservices Accessing the Shared Database



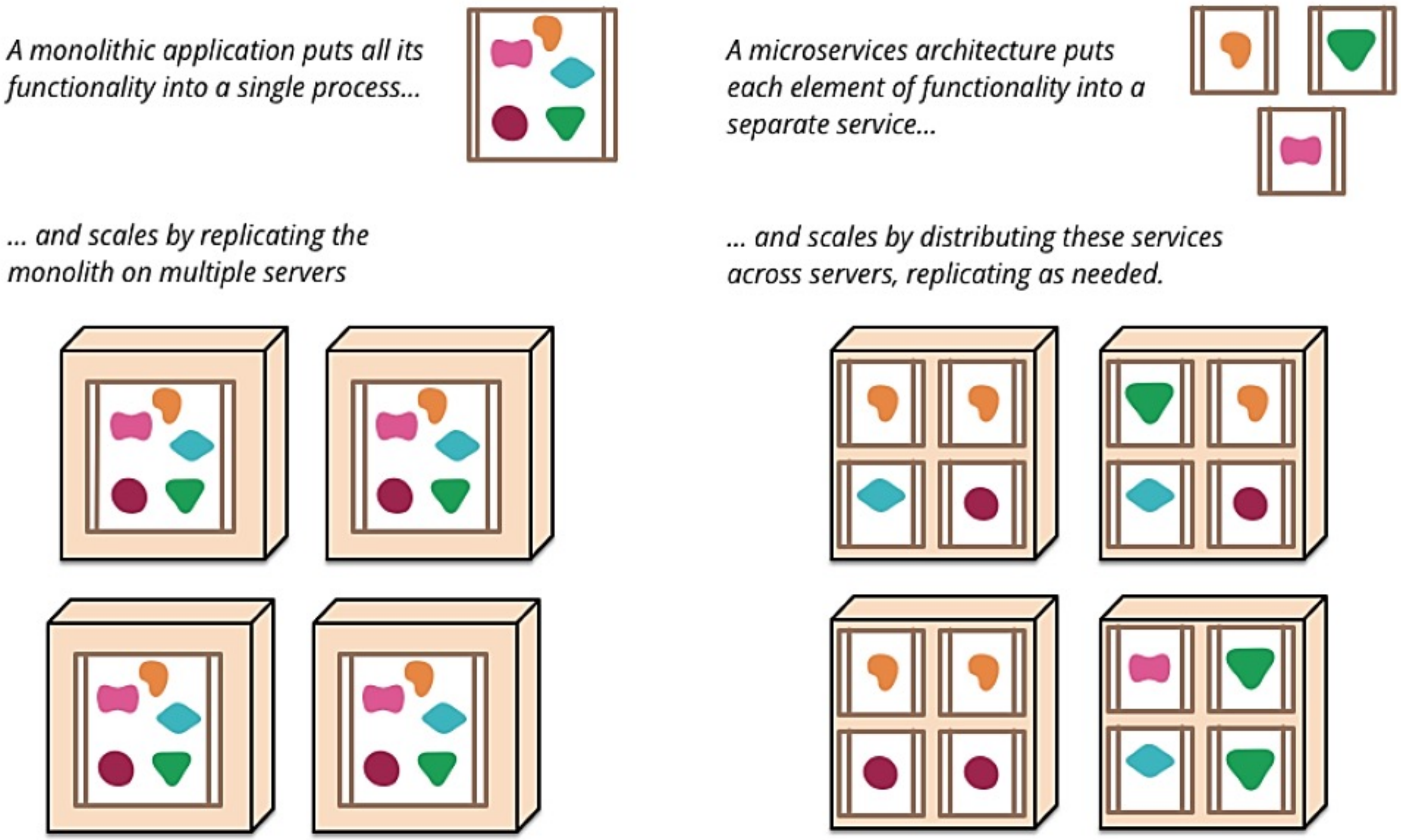
Colorado State University

15

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

Microservices. Scalability



CS370 Operating Systems

Colorado State University

Yashwant K Malaiya

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 term has a different meaning here!

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

Colorado State University

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

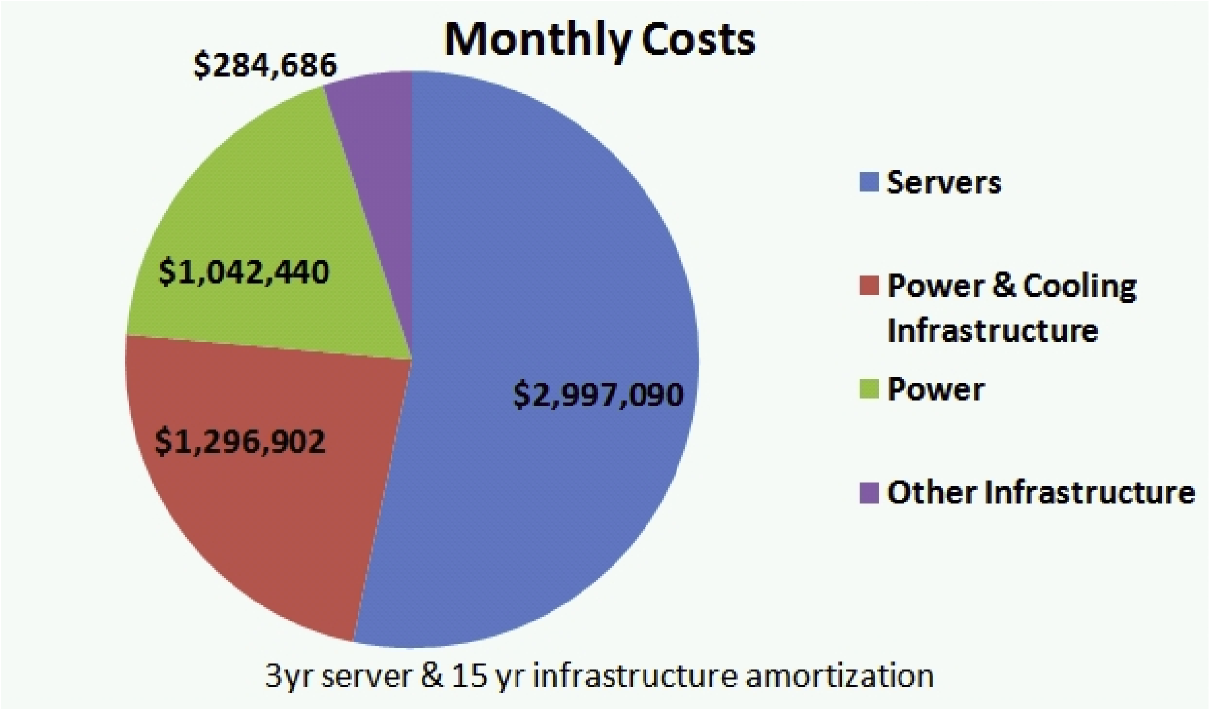
Resource management

- 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



Efficiency captured as *Power Usage Effectiveness*

- Total Powe/ IT 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

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 etc

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

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)

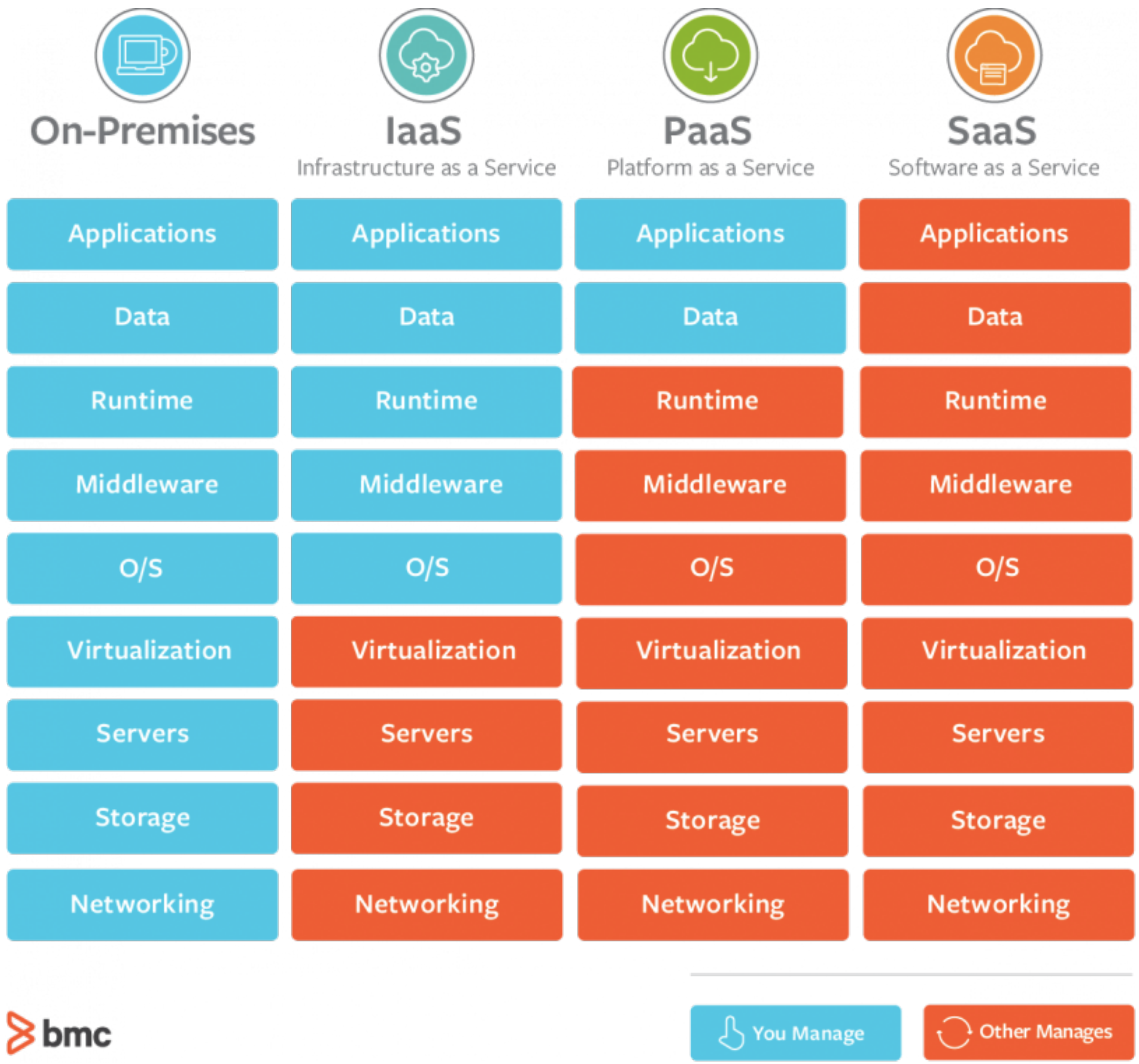
instance	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
t2.nano	1	Variable	0.5	EBS Only	$0.0058 per Hour
t2.small	1	Variable	2	EBS Only	$0.023 per Hour
t2.medium	2	Variable	4	EBS Only	$0.0464 per Hour
m5.4xlarge	16	61	64	EBS Only	$0.768 per Hour
m4.16xlarge	64	188	256	EBS Only	$3.2 per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

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

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/

Colorado State University

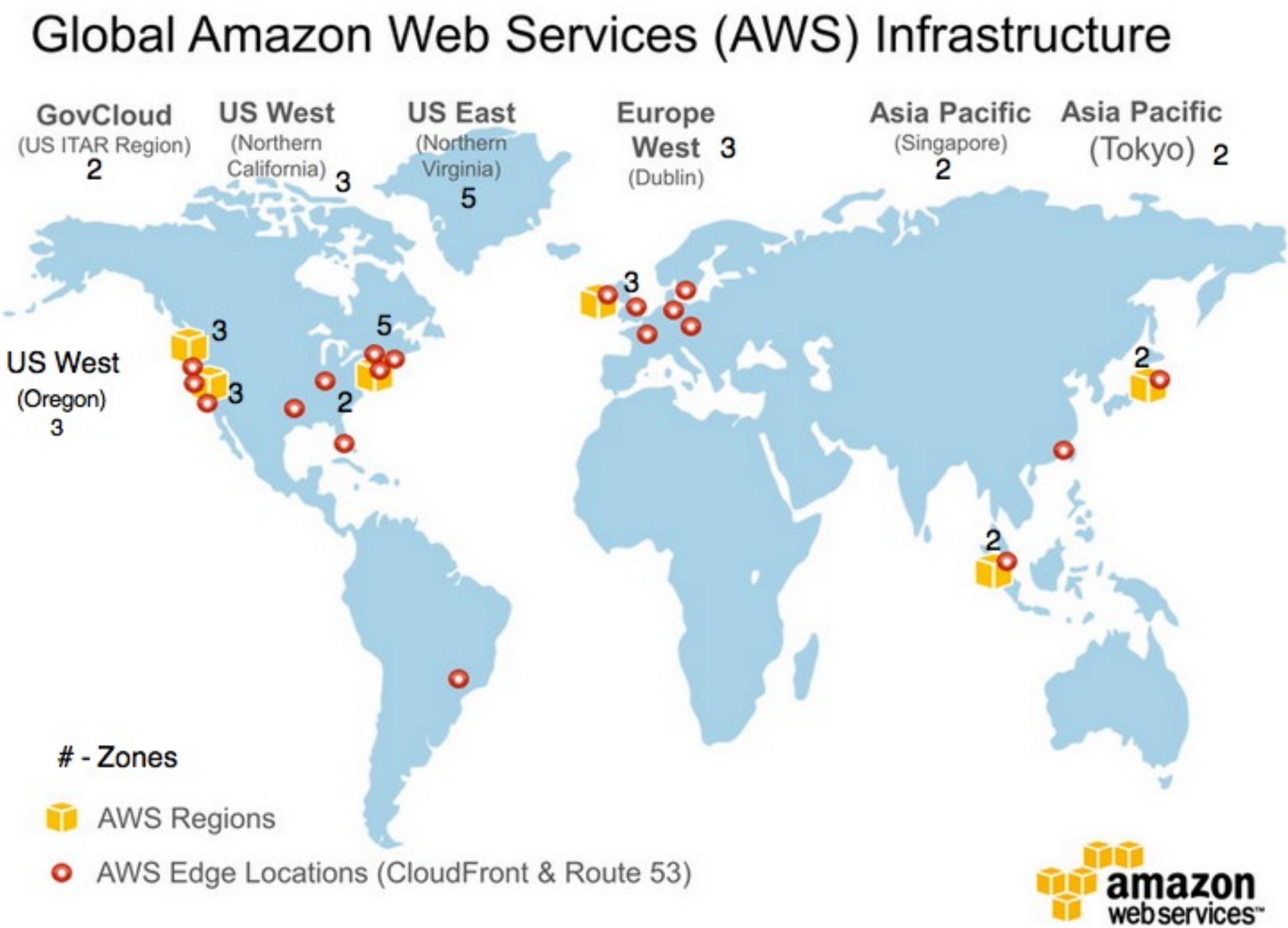
Cloud Management models

- **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

Reflecting on Part 1

- System structure and program compilation/execution
- Processes & Threads:
  - creation
  - scheduling
  - termination
- Inter-process communication
  - Synchronization
  - Deadlocks (included in Part 2)

Part 2

- We will review these on next Thursday.
- Virtual and physical address spaces
  - Pages and frames
    - Translation using page tables and TLBs
    - Effective access time
  - Virtual memory
    - Demand paging, page replacement algorithms
  - File systems
    - Disk organization, block allocation, scheduling
    - RAIDs
  - Virtual machines and containers
  - Data centers and cloud