

# UMD DATA605 - Big Data Systems

## Cloud Computing

GP Saggese  
[gsaggese@umd.edu](mailto:gsaggese@umd.edu)

with thanks to Alan Sussman,  
Amol Deshpande

# Outline

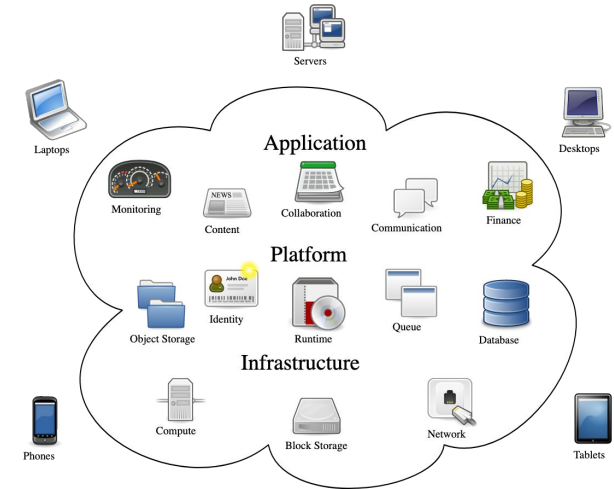
- Cloud computing
- Technologies behind cloud computing
  - Data centers
  - Virtualization
  - Programming frameworks
- Challenges and opportunities

# Outline

- **Cloud computing**
- Technologies behind cloud computing
  - Data centers
  - Virtualization
  - Programming frameworks
- Challenges and opportunities

# Cloud Computing

- **Computing as “service” rather than “product”**
  - Everything happens in the cloud: both storage and computing
  - Edge devices (e.g., phones, laptops, tablets) simply interact with the cloud
- **Advantages of cloud computing**
  - Device agnostic: computation can seamlessly move from one device to other
  - On demand
  - Efficiency / scalability
    - Programming frameworks (e.g., Hadoop, Spark, Dask) allow “easy” scalability
  - Reliability
  - Cost: “pay-as-you-go” allows renting computing resources as needed
    - Cheaper (?) than building your own systems
    - Computing as a commodity (like electricity)



# Building vs Renting

## - Building infrastructure

- Require time and capital investments (Capex)
  - Especially at the beginning when there aren't revenues
  - A smooth cash flow (constant \$/mo) is better than lumpy one (big one-time purchase)
- Buy hardware (e.g., computers, storage, network)
  - How do you estimate the size of your hardware for today?
  - Even more difficult to estimate future demands
- Update hardware when it becomes obsolete
- Cost of owning hardware (Opex)
  - E.g., data center, electricity, cooling, handling faulty
- Administering
  - Installing, updating, maintaining software stack

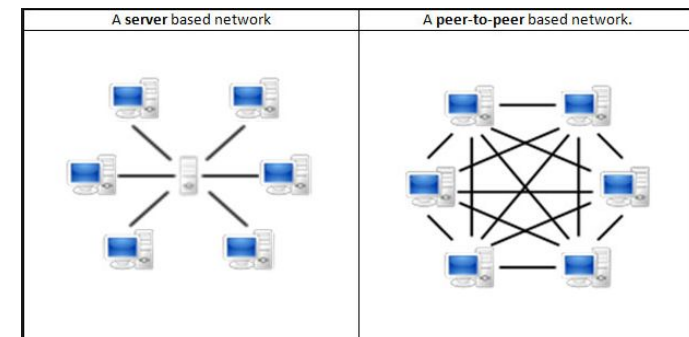
# Building vs Renting

- **Cloud computing**

- Pay for what you use
  - Low initial capital investment
- Get your systems ready with a click on a web-site
- No need to estimate a multi-year resource plan
- Pick machines that suit your application and data requirements

# Cloud Computing

- **Basic ideas of cloud computing around for a long time**
  - Mainframes + thin clients (1960s)
  - Personal computers (1980s)
  - Grid computing for supercomputers (1990s)
  - Peer-to-peer architecture (early 2000s)
  - Client-server model (Web 1.0 and Web 2.0)
  - Cloud computing (2010s)
- **It finally works**
- **Why now? A convergence of several key technologies over the last few years**
  - OS virtualization
  - Large data centers
  - Decreasing hardware costs
  - Big data frameworks



# X-as-a-Service

## Infrastructure-as-a-service (IaaS)

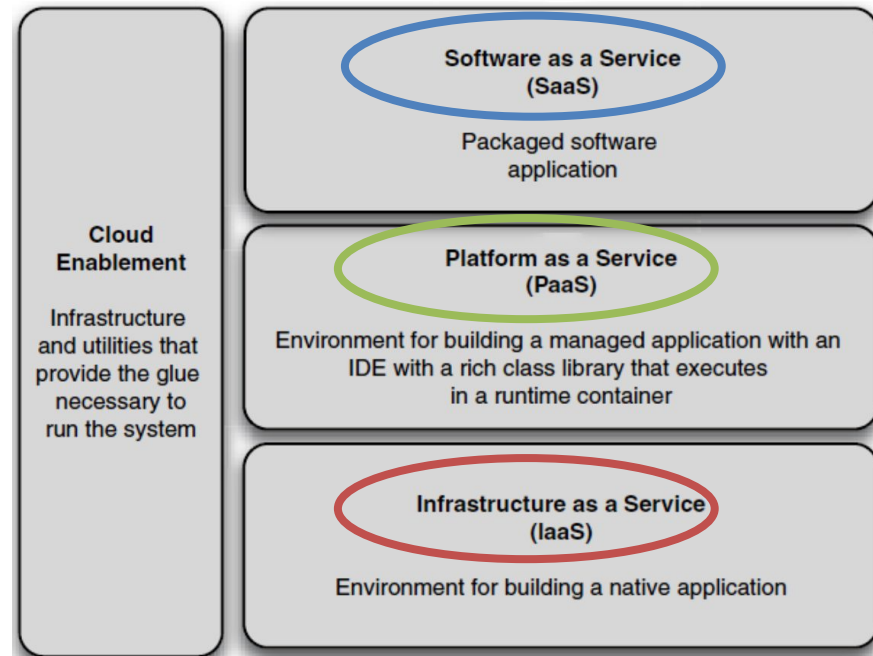
- Cloud provides low-level resources
- You install and maintain OS and applications
- E.g., AWS EC2

## Platform-as-a-service (PaaS)

- Cloud provides OS and programming languages
- You build application on top of it
- E.g., Google App engine, managed Hadoop

## Software-as-a-service (SaaS)

- Cloud provides the application
- You use it
- No need to install anything on your machine
- E.g., Dropbox, Salesforce, any app running in a browser





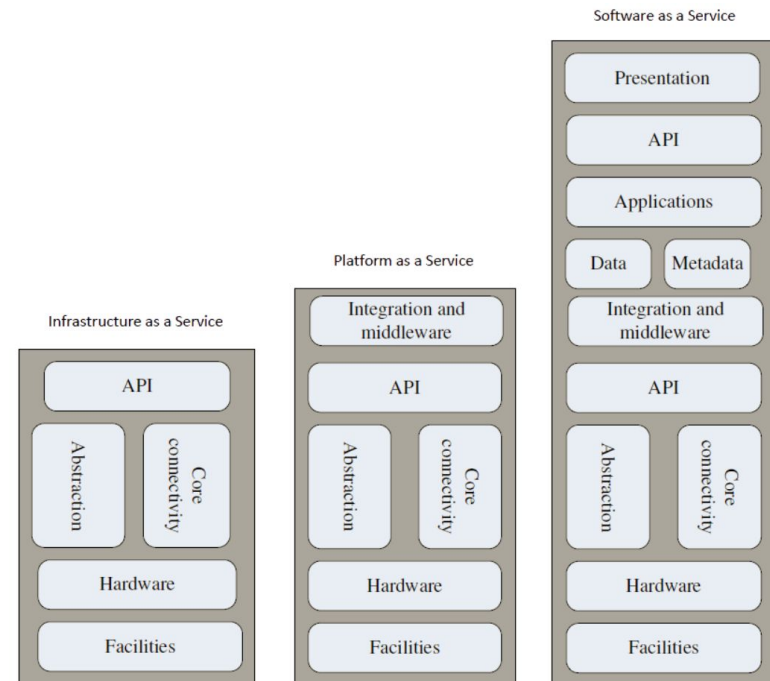
# X-as-a-Service

## ... XaaS

- Business model of “Anything-as-a-service” (2010s)
- Desktop-as-a-service (e.g., AWS app)
- Mobility-as-a-service (e.g., Uber)
- Games-as-a-service (e.g., Google Stadia)
- Storage-as-a-service (e.g., S3, Google Drive)
- Marketing-as-a-service
- Banking-as-a-service, ...

# Platform-as-a-Service

- **Problem: assembling your own software stack requires work**
  - Install
  - Configure
  - Manage dependencies
  - Incompatible versions
- **Solution: get a pre-built software stack**
  - Pre-installed OS
  - Libraries
  - Application software
  - As a virtualization solution
    - E.g., VMware or Docker
    - The software stack comes as a large file with the image of the system
- **Business model built around this**
  - E.g., pre-built images for Hadoop
    - E.g., Hortonworks, Cloudera
  - E.g., pre-built distributions for Linux
    - RedHat, Gentoo, CentOS



# Different Cloud Computing Layers

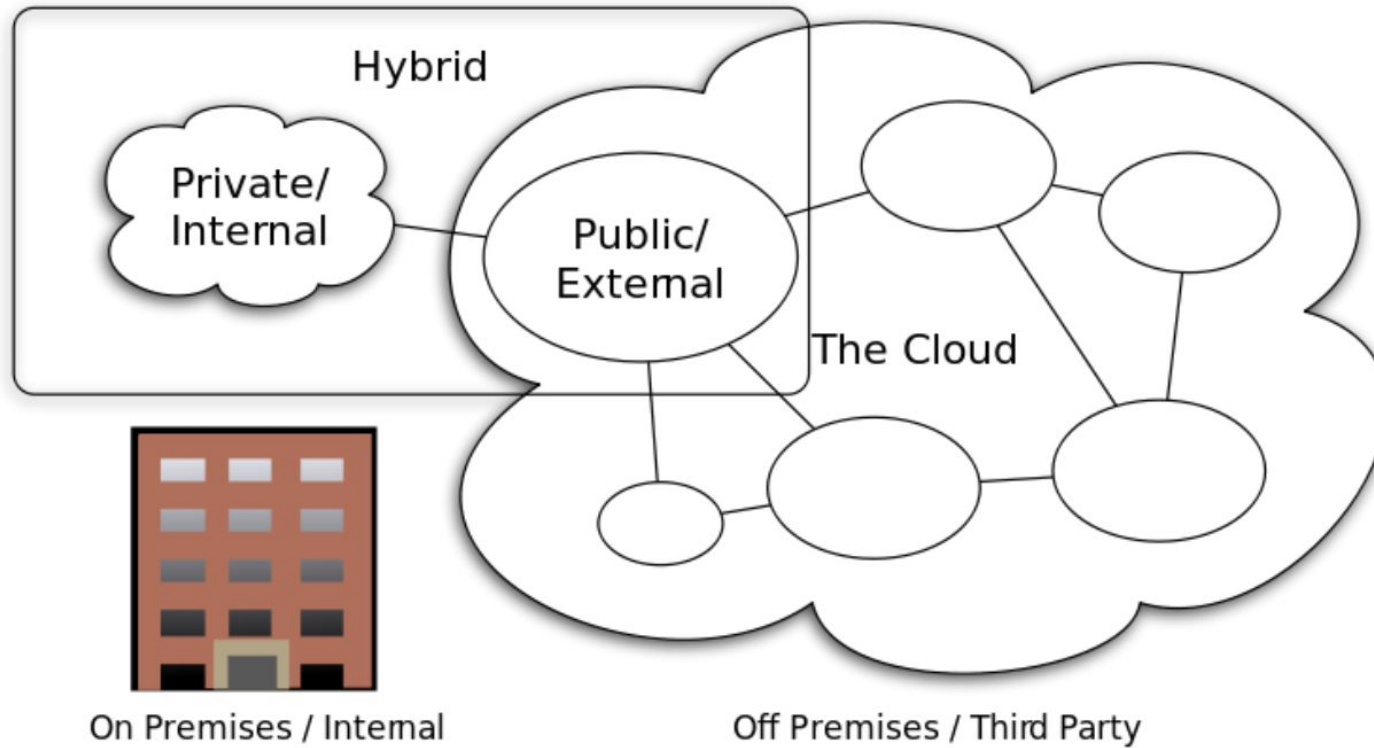
<b>Application Service (SaaS)</b>	Google Docs, Salesforce, Quicken Online, Zoho, ... <i>Pretty much everything on Internet</i>
<b>Application Platform</b>	Google App Engine, Heroku, AWS, Facebook
<b>Server Platform</b>	AWS EC2, Google Compute, Digital Ocean, Linode
<b>Storage Platform</b>	Amazon S3, Apple iCloud, Google Drive, DropBox, Box

# Cloud Computing Service Layers

		Services	Description
Application Focused		Services	Services – Complete business services (e.g., PayPal, OpenID, OAuth, Google Maps, Alexa)
		Application	Application – Cloud based software that eliminates the need for local installation (e.g., Google Apps, Microsoft Office 365)
		Development	Development – Software development platforms used to build custom cloud based applications (PAAS & SAAS), e.g., Salesforce
Infrastructure Focused		Platform	Platform – Cloud based platforms, typically provided using virtualization (e.g., Amazon EC2, Google Compute)
		Storage	Storage – Data storage or cloud based NAS (e.g., AWS S3, Apple iCloud, CloudNAS, CTERA)
		Hosting	Hosting – Physical data centers (e.g., IBM, HP, DigitalOcean)

From Dr. Mehmet Gunes, U. Nevada Reno

# Cloud Types



## Cloud Computing Types

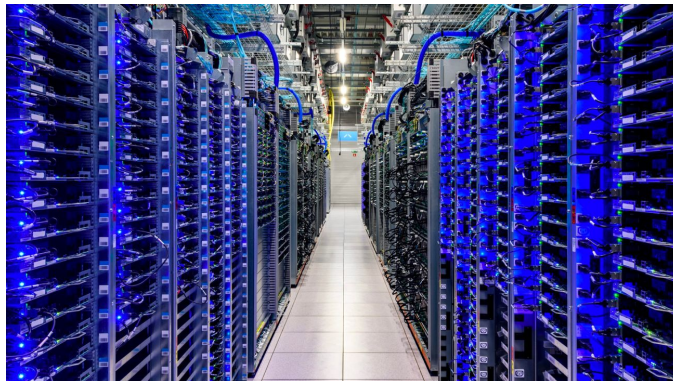
CC-BY-SA 3.0 by Sam Johnston

# Outline

- Cloud Computing
- Technologies behind cloud computing
  - **Data centers**
  - Virtualization
  - Programming Frameworks
- Challenges and opportunities

# Data Centers

- Data centers are key infrastructure piece that enables cloud computing
  - Every large company (e.g., AWS, Apple, Google, Facebook) is building data centers around the world
- Huge amount of work on deciding how to build / design them
- Research on how to save energy to power and cool data centers



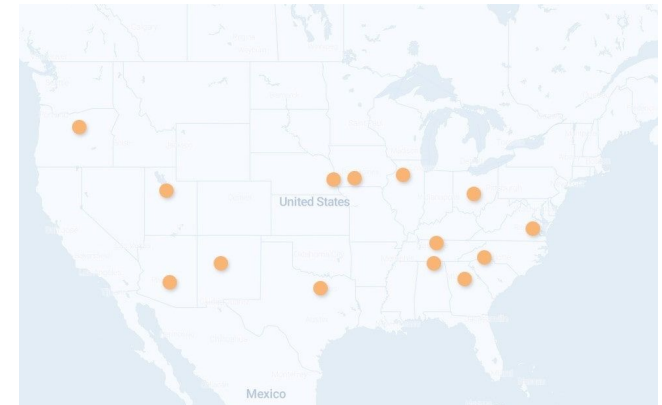


# Data Centers

- **Equipment cost**
  - E.g., computing, memory, storage, networking
  - A data center costs around \$1B
  - Very expensive, but prices keep dropping
- **Powering / cooling cost**
  - Cost of running the equipment
  - Cost of cooling is quite high
    - Both have led to focus on “energy-efficient computing”
  - Appropriate placement of vents is a key issue
    - Thermal hotspots often appear and need to be worked around
  - PUE (Power Usage Effectiveness)
    - Measure of how much power is converted in computation, the rest is overhead
  - Hard to optimize in small data centers
    - Ideally PUE should be 1, currently numbers are around 1.07-1.22
    - May lead to very large data centers in near future (already happening)

Data Center	Online	Buildings	SqFt (m)	Investment (\$bn)
Dekalb, Illinois	2022	2	0.9	\$0.8
Altoona, Iowa	2014	10	4.1	\$2.0
Papillion (Sarpy), Nebraska	2019	8	3.6	\$1.5
New Albany, Ohio	2020	5	2.5	\$1.0
Huntsville, Alabama	2021	4	2.5	\$1.0
Newton, Georgia	2023	5	2.5	\$1.0
Forest City, North Carolina	2012	4	1.3	\$0.8
Gallatin, Tennessee	2023	2	1.0	\$0.8
Henrico, Virginia	2020	7	2.5	\$1.0
Mesa, Arizona	Q4 2023	2	1.0	\$0.8
Los Lunas, New Mexico	2019	6	2.8	\$1.0
Fort Worth, Texas	2017	5	2.6	\$1.5
Prineville, Oregon	2011	11	4.6	\$2.0
Eagle Mountain, Utah	2021	5	2.4	\$1.0
Odense, Denmark	2019	2	0.9	\$1.6
Clonee, Ireland	2018	3	1.6	\$0.4
Luleå, Sweden	2013	3	1.0	\$1.0
Tanjong Kling, Singapore	2022	1	1.8	\$1.0
<b>Total</b>		<b>85</b>	<b>39.6</b>	<b>\$20.1</b>

## Meta investment in 18 data centers





# Data Centers

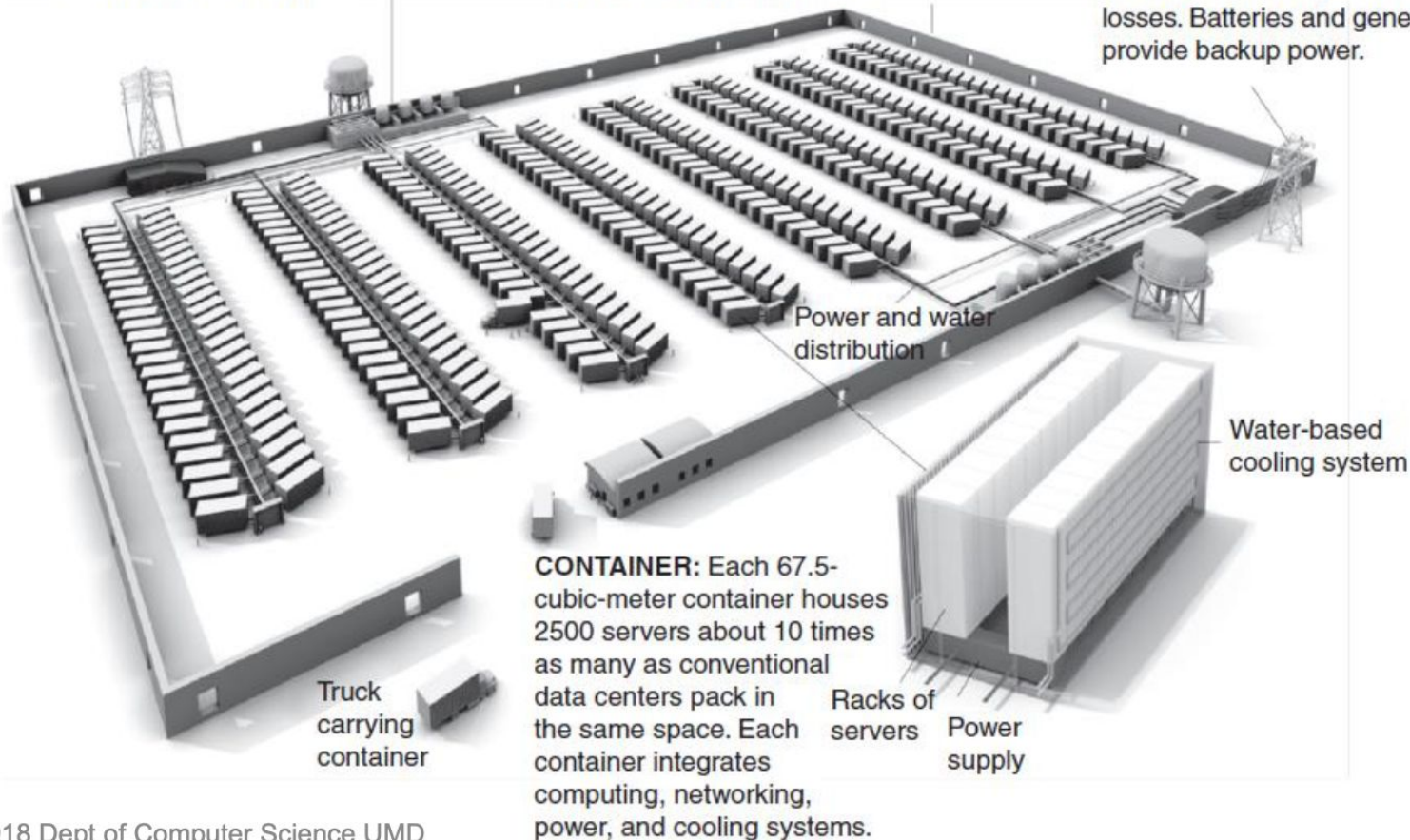


# (Modular) Data Centers

**COOLING:** High-efficiency water-based cooling systems—less energy-intensive than traditional chillers—circulate cold water through the containers to remove heat, eliminating the need for air-conditioned rooms.

**STRUCTURE:** A 24 000-square-meter facility houses 400 containers. Delivered by trucks, the containers attach to a spine infrastructure that feeds network connectivity, power, and water. The data center has no conventional raised floors.

**POWER:** Two power substations feed a total of 300 megawatts to the data center, with 200 MW used for computing equipment and 100 MW for cooling and electrical losses. Batteries and generators provide backup power.



**CONTAINER:** Each 67.5-cubic-meter container houses 2500 servers about 10 times as many as conventional data centers pack in the same space. Each container integrates computing, networking, power, and cooling systems.

018 Dent of Computer Science UMD

From [James Hamilton Presentation](#)

# Amazon Web Services

## As of 2022, 28 geographical regions





# Amazon Web Services (EC2)

- The most widely used current solution to cloud computing
  - However alternatives are attractive depending on your needs
  - Current prices are very “low” and likely to remain that way due to competition
  - See <https://aws.amazon.com/ec2/pricing/on-demand/> for current on-demand pricing

## Small Instance – default\*

1.7 GB memory  
1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit)  
160 GB instance storage  
32-bit platform  
I/O Performance: Moderate  
API name: m1.small

## Large Instance

7.5 GB memory  
4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)  
850 GB instance storage  
64-bit platform  
I/O Performance: High  
API name: m1.large

## Extra Large Instance

15 GB memory  
8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)  
1,690 GB instance storage  
64-bit platform  
I/O Performance: High  
API name: m1.xlarge

Viewing 564 of 564 available instances

< 1 2 3 4 5 6 7 ... 29 >

Instance name ▲	On-Demand hourly rate ▼	vCPU ▼	Memory ▼	Storage ▼	Network performance ▼
a1.medium	\$0.0255	1	2 GiB	EBS Only	Up to 10 Gigabit
a1.large	\$0.051	2	4 GiB	EBS Only	Up to 10 Gigabit
a1.xlarge	\$0.102	4	8 GiB	EBS Only	Up to 10 Gigabit
a1.2xlarge	\$0.204	8	16 GiB	EBS Only	Up to 10 Gigabit
a1.4xlarge	\$0.408	16	32 GiB	EBS Only	Up to 10 Gigabit
a1.metal	\$0.408	16	32 GiB	EBS Only	Up to 10 Gigabit
t4g.nano	\$0.0042	2	0.5 GiB	EBS Only	Up to 5 Gigabit
t4g.micro	\$0.0084	2	1 GiB	EBS Only	Up to 5 Gigabit
t4g.small	\$0.0168	2	2 GiB	EBS Only	Up to 5 Gigabit
inf1.xlarge	\$0.228	4	8 GiB	EBS Only	Up to 25 Gigabit
inf1.2xlarge	\$0.362	8	16 GiB	EBS Only	Up to 25 Gigabit
inf1.6xlarge	\$1.18	24	48 GiB	EBS Only	25 Gigabit
inf1.24xlarge	\$4.721	96	192 GiB	EBS Only	100 Gigabit

# Amazon S3

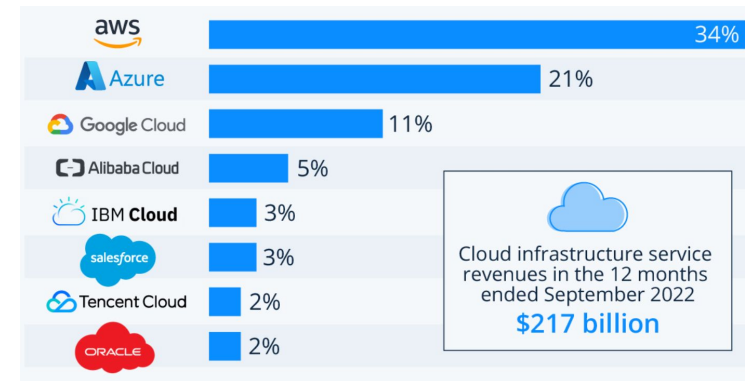
- Amazon storage services (Simple Storage Solution)
  - Pay for what you use

## S3 Storage Types

	S3 Default	S3 RRS	S3 IA	Glacier
Durability	99.999999999%	<b>99.99%</b>	99.999999999%	99.999999999%
Availability	99.99%	99.99%	<b>99.9%</b>	99.99%
Extra Fees *	None	None	<b>Retrieval</b>	<b>Retrieval</b>
Real-Time Access?	Yes	Yes	Yes	<b>No (mins/hours)</b>
Frequently Accessed?	Yes	Yes	<b>No</b>	<b>No</b>

# Google App Engine

- Google Compute Engine (IaaS)
  - Directly compete with AWS EC2
- Google infrastructure (PaaS)
  - Run Docker container on Google resources
  - Managed services (e.g., SQL and NoSQL DBs)
- Google Docs (SaaS)
  - Share documents in the cloud
  - E.g., word processor, spreadsheet, presentations
- Google cloud computing market share
  - Built software infrastructure / data centers before Amazon
  - Invented many cloud technologies (e.g., Google File System, MapReduce, BigTable)
  - Market share is 3x smaller than AWS
  - Issues:
    - Developer / customer unfriendliness
    - (Often) lack of commitment ([Killed by google](#))
    - No (or horrible) customer service



# Outline

- Cloud Computing
- Technologies behind cloud computing
  - Data centers
  - **Virtualization**
  - Programming frameworks
- Challenges and opportunities

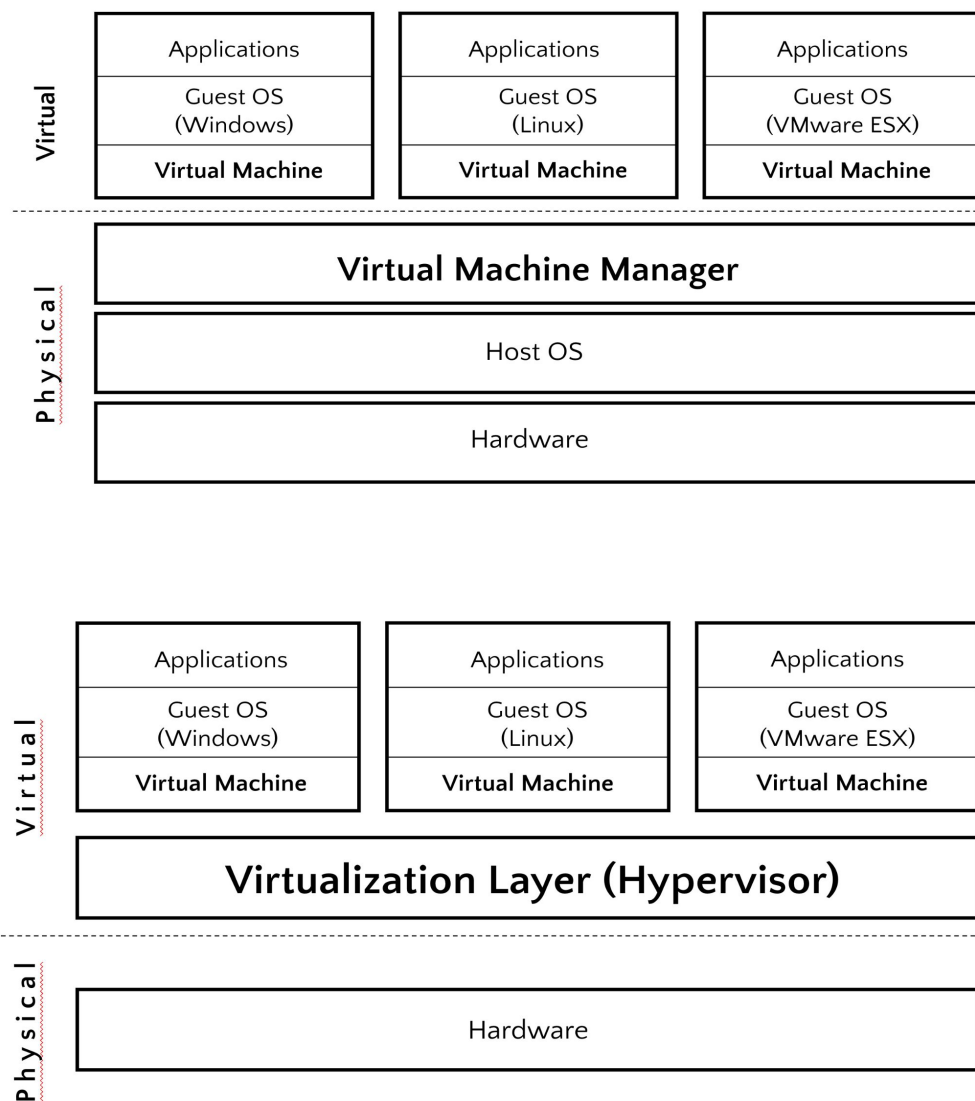
# Virtualization

- **Virtual machines have been around for a long time**
  - E.g., running Windows inside a Mac
  - Used to be very slow (e.g., QEMU)
- **Only recently (~2000s) became efficient enough for cloud computing**
- **Basic idea of cloud computing**
  - Run virtual machines on your servers and sell time on them
  - E.g., Amazon EC2, Microsoft Azure, Google Cloud
- **Many advantages**
  - Security: virtual machines have almost impenetrable boundary
  - Multi-tenancy: can have multiple VMs on the same server
  - Efficiency: replace many underpowered machines with fewer high-powered machines



# Virtualization

- **Consumer / desktop virtualization**
  - E.g., VMWare, Xen, VirtualBox
  - Run on top of host OS
  - Hypervisor / VM supports guest OS in virtual environment
- **Server virtualization**
  - Run hypervisor directly on top of hardware
  - Good for server farms, like in cloud computing
  - Amazon used Xen on RedHat machines
  - Now custom hardware AWS Nitro
  - Support Windows and Linux Virtual Machines

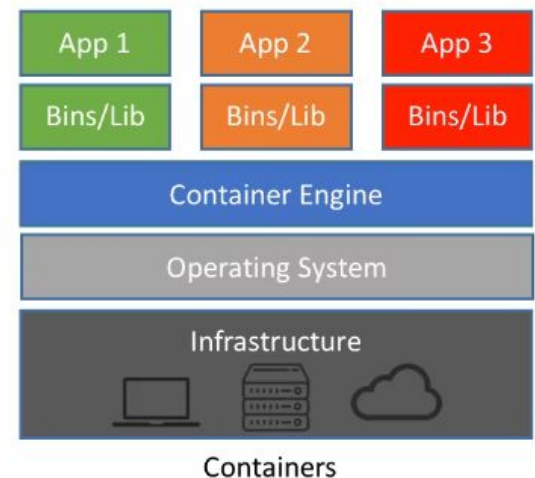
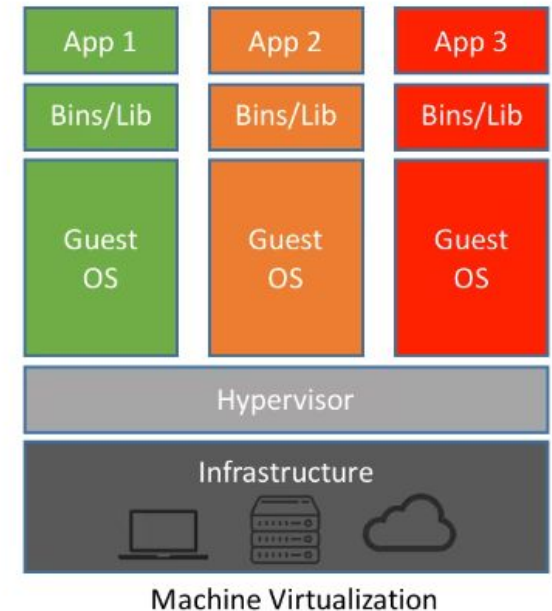


# Virtualization

- Performance and tricky things to keep in mind
  - Hard to reason about performance (if you care)
  - Identical VMs may deliver somewhat different performance
    - E.g., multi-tenancy, different hardware
- “Bare-metal” compute to improve performance

# Docker

- Enable true independence between application devs and IT ops
- Package all the dependencies
- Advantages
  - Containers are fast and portable
  - Reduce overhead of virtualization
  - Containers don't require full-blown OS
  - All containers run on a single host
  - Reduce OS licencing cost
  - Reduce overhead of OS maintenance



# Outline

- Cloud Computing
- **Start**
- Technologies behind cloud computing
  - Data centers
  - Virtualization
  - **Programming frameworks**
- Challenges and opportunities

# Programming Frameworks

- **Programming frameworks emerged from efforts to “scale out” workloads**
  - Distribute work over large numbers of machines (thousands of machines)
- Parallelism has been around for a long time
  - Both in a single machine and as a cluster of computers
- Parallelism is very hard for programmers
  - Many things to keep track of:
    - How to parallelize application
    - How to distribute the data
    - How to handle failures
    - Debugging
    - Race conditions/Heisenbugs
    - ...
- **The difference is the user interface**
  - Google developed MapReduce and BigTable frameworks and started a new era
    - Hadoop, Spark
  - AWS services

# MapReduce Framework

- Provide a fairly restricted, but still powerful abstraction for programming distributed workloads
- Programmers only write functions called *map* and *reduce*
  - **map** programs
    - inputs: a list of “records” (e.g., images, genomes)
    - output: for each record, produce a set of **(key, value)** pairs
  - **reduce** programs
    - input: a list of **(key, [values])** grouped together from the mapper
    - output: whatever
  - Both can do arbitrary computations on the input data as long as the basic structure is followed
- Everything else (e.g., task scheduling, fault tolerance) is taken care of by the framework
- **Batch processing of data**
  - MapReduce and other analytics frameworks are great
- **Streaming data**
  - Very large-scale applications that need real-time access with few ms latencies
  - Trade-off between “consistency” and “performance”

# Other Programming Frameworks

- Many different programming frameworks suitable for different applications
  - Address limitations of MapReduce-based platforms
- **High-performance Computing (HPC) Systems**
  - Cluster of supercomputers (instead of cheap computers)
  - E.g., GridRPC, MPI
  - More expressive and complex, but a lot more efficient
- **Spark**
  - Based on Resilient Distributed Data (RDD)
  - All in-memory, much more efficient
  - Uses Scala, Python, Java for programming
- **Apache Hive**
  - SQL-like interface on top of Hadoop / HDFS, originally from Facebook
- **Apache HBase**
  - NoSQL column oriented DB
  - Random read/write very large tables (like SQL) on top of Hadoop / HDFS
  - Modeled after Google BigTable
- **Apache Storm, Spark Streaming**
  - Focused on handling real-time streaming data
- **Giraph, GraphLab, GraphX**
  - Graph processing systems

# Outline

- Cloud computing
- Technologies behind cloud computing
  - Data centers
  - Virtualization
  - Programming frameworks
- **Challenges and opportunities**



# Advantages of Cloud Computing

- **Lower edge computer costs**

- Since applications run in the cloud, desktop does not need processing power / memory / disk demanded by traditional desktop software

- **Improved performance of desktop**

- Better performance / faster system boot for your desktop due to large programs hogging your computer's memory

- **Device independence**

- You are no longer tethered to a single computer
- Applications and documents follow you through the cloud
- Move to another (maybe portable) device and your applications and documents are still available

- **Reduced software costs**

- Instead of purchasing expensive software applications, you can get most of what you need for free(-ish)
  - Google Docs suite vs Microsoft
  - Most applications are cloud based today
- Rent (monthly payments) instead of buying

# Advantages of Cloud Computing

- **Instant software updates**

- No longer faced with choosing between obsolete software and high upgrade costs
- When the application is web-based, updates happen automatically
  - Available the next time you log into the cloud
  - You get the latest version without needing to pay for or download an upgrade
- Docker or VMs can package software to run on any platform that supports virtualization

- **Improved document format compatibility**

- Do not have to worry about the documents on your machine being compatible with other users' applications or OSes
- Potentially no format incompatibilities when everyone is sharing documents in the (same) cloud

# Advantages of Cloud Computing

- **“Unlimited” storage capacity**
  - Cloud computing offers virtually limitless storage
  - Your computer has ~1TB hard drive
  - Hundreds of PBs available in the cloud (elastic and on-demand)
- **Increased data reliability**
  - Hard disk can crash and destroy all your valuable data of your desktop computer
  - Few individual users backup their data on a regular basis
  - A computer crashing in the cloud does not affect the storage of your data
    - All your data is still out there in the cloud, still accessible
    - Cloud computing is a data-safe computing platform

# Advantages of Cloud Computing

- **Universal document access**

- Documents stay in the cloud
- Documents are instantly available from wherever you are
- As long as you have an Internet connection

- **Latest version availability**

- When you edit a document at home, that edited version is what you see when you access the document at work
- The cloud always hosts the latest version of your documents
- Built-in revision control system

- **Easier collaboration**

- Sharing documents leads directly to better collaboration
- Multiple users can collaborate easily on documents and projects

# Disadvantages of Cloud Computing

- **Using cloud computing means dependence on Big Tech**
  - Big Internet companies like AWS, Google, and Microsoft monopolise the market
  - Can possibly limit flexibility and innovation
  - Lots of competition in cloud computing
  - High barriers to break into the cloud computing market
- **Security is an issue**
  - Unclear how safe outsourced data is
  - Ownership of data is not always clear
- **Issues relating to policy and access**
  - If your data is stored abroad whose policy do you adhere to?
    - To deal with government restrictions (e.g., export restrictions), cloud providers ensure your data is stored within your country's borders
    - E.g., TikTok saga
  - What happens if the remote server goes down?
    - How will you then access files?
  - Cases of users being locked out of accounts and losing access to data

# Disadvantages of Cloud Computing

- **Requires a constant Internet connection**
  - Without an Internet connection you cannot access anything, applications or your own documents
  - No Internet means no work: this could be a deal-breaker
    - Many applications offer off-line capabilities
- **Does not work well with low-speed / spotty connections**
  - Low-speed Internet connection makes cloud computing painful or impossible
  - Web-based applications require a lot of bandwidth to download
- **Web-interface can be slow**
  - Even with a fast connection, web-based applications are often slower than your desktop
  - Interface (not data) is sent back and forth from your computer to the computers in the cloud
  - Latency (more than throughput) matters

# Disadvantages of Cloud Computing

- **Features might be limited**

- Many web-based applications simply are not as full-featured as their desktop-based applications
- For example, you can still do a lot more with Microsoft PowerPoint than with Google Slides
- This situation is changing rapidly

- **Stored data might not be secure**

- With cloud computing, all your data is stored on the cloud
- How secure is the cloud?
  - Probably more secure than your laptop
- Can unauthorized users gain access to your confidential data?
  - News are data leaks are constant

- **Stored data can be lost**

- Data stored in the cloud is safe, replicated across multiple machines (theoretically)
- Doing local backups might still be a good idea

# Disadvantages of Cloud Computing

- **HPC systems**

- Unclear if you can run compute-intensive HPC applications that use MPI or OpenMP
- Need low-latency, high-bandwidth interconnect
- Scheduling is important with this type of application
- All nodes should be co-located to minimize communication latency
- This has changed (e.g., MapReduce, AWS EC2) and is still changing

- **Interoperability**

- Each cloud systems uses different protocols and different APIs
- May not be possible to run applications across different cloud-based systems
- Solution: indirection layer (Terraform, Ansible)