Cloud Computing and Distributed Systems

Introduction

Raja Appuswamy

Eurecom

Cloud computing: The disruption

“The worldwide public cloud services market is projected to reach a total of $214.3 billion in 2019. Cloud Services Industry to grow exponentially to $331 billion by 2022.” – Gartner

“In 2018, AWS delivered most of Amazon's operating income” – Amazon

“80% of organizations will migrate toward the cloud by 2025.” – Gartner

“50% of all data will be held in the cloud by 2020. Cloud data centers will process 94% of workloads in 2021.” – IDC & Cisco

“Global data centers used roughly 416 terawatts (3% of the total electricity) last year, nearly 40% more than the entire United Kingdom.” - Forbes

“Big data solutions via cloud subscriptions will increase about 7.5 times faster than on-premise options.” - Forrester

“AI without the cloud is tough” – Information Age

This Course

- **What you will learn (roadmap)**
  - **Economic foundations**
    - Cloudonomics & Service models
  - **Infrastructure foundations**
    - Virtualization, containerization, serverless functions
  - **Systems foundations**
    - In-depth description of Hadoop & ecosystem
    - Architecture of Apache Spark
  - **Programming foundations**
    - Map—reduce and functional programming
    - Relational Algebra and High-Level Languages
  - **Algorithmic foundations**
    - Cluster scheduling with YARN, Mesos, Omega
    - CAP theorem, SQL and NoSQL
    - Coordination & Apache Zookeeper
    - Decentralization & blockchain (if time permits)

Who is this course for?

- **Cloud system and application engineers**
- **Data scientists**
- **Requirements**
  - Good knowledge of Python
  - Familiarity with operating systems concepts, and Linux
  - Good knowledge of git
  - Ideally, familiarity with distributed algorithms

How to make the most of this course?

- **Attend classes and the labs**
  - Many discussions in live classes, that are not on the slides
  - Laboratories can be hard for people with little CS background
- **Resources**
  - Lecture notes: https://raja-appuswamy.github.io/DISC-CLOUD-COURSE/

Grading

- **Final exam**
  - 50% of the grade
  - Generally divided in two parts
    - A series of questions
    - One or more problems to solve

- **Laboratory sessions**
  - Mainly Notebooks, some special labs
  - Question answering
  - Heuristic to map credits to grade

Introduction to the Cloud Computing

We live in a world of data

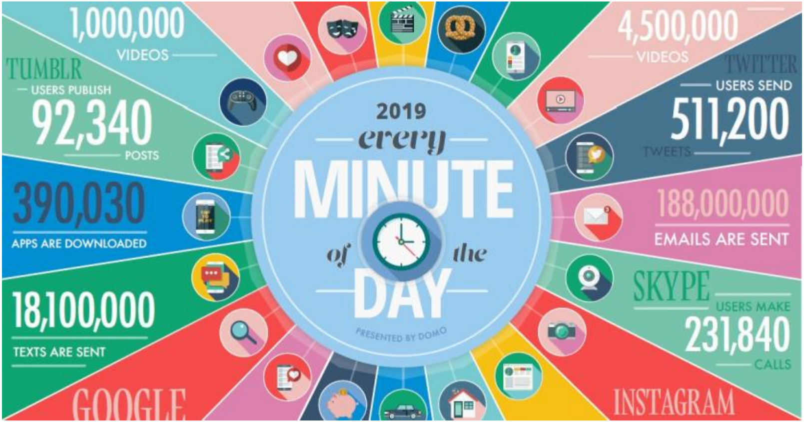


Figure: Data deluge.

Big Data

- **Big data is defined as large pools of data that can be captured, communicated, aggregated, stored, and analyzed.**
- **Data continues to grow**

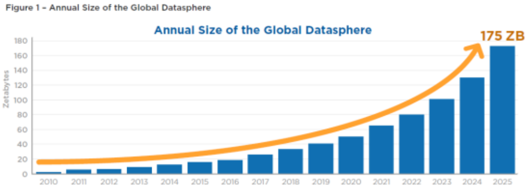


Figure: Global datasphere

- **Applications are becoming data intensive**
  - More data leads to better accuracy
  - With more data, accuracy of different algorithms converges

Let's look at your data.

You want to access, shared, process your data from all your devices, anytime, anywhere.

How will we manage all this data?

- **Manage it ourselves?**
  - How do we store it?
  - How do we share it?
  - How can we enable access to it from any place?
  - How do we process all of it?
  - How do we secure it?
  - ....
- **What if it is managed by someone else?**
  - Someone provides a management “service”
  - You pay a subscription for this “service”

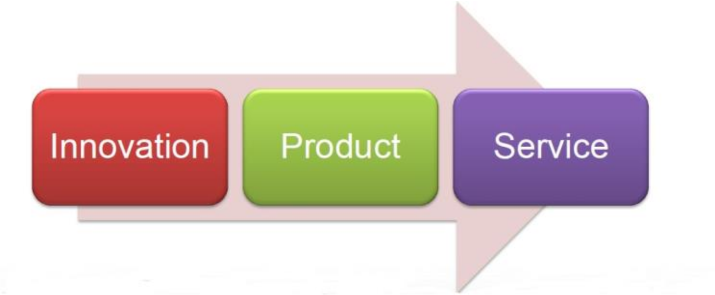
Cloud computing: The prophecy

- In 1965, MIT's Fernando Corbató and the other designers of the Multics operating system envisioned a **computer facility operating "like a power company or water company"**.
- Plug your thin client into the computing Utility and Play your favorite Intensive Compute & Communicate Application

Utility–Product–Service lifecycle: Water

Utility–Product–Service lifecycle: Electricity

Generalizing the lifecycle



Cloud Computing

Transformation of IT from a product to a service

How did IT transformation happen?

- **Requirements to transform IT**
  - Connectivity to move data
  - Interactivity for seamless interface
  - Reliability against failures
  - Acceptable performance
  - Ease of programmability for developing new services
  - Manageability for Big Data
  - Pay-as-you-go to avoid capital investment
  - Scalability and elasticity for changing needs<

Supporting technologies

- Cloud computing is a combination of technologies
    - Connectivity to move data => **Networked systems**
    - Interactivity for seamless interface => **Web 2.0 and HCI**
    - Reliability against failures => **Dependable systems**
    - Acceptable performance => **Parallel and distributed systems**
    - Ease of programmability for developing new services => **Programming languages**
    - Manageability for Big Data => **Storage systems**
    - Pay-as-you-go to avoid capital investment => **Utility computing & economics**
    - Scalability and elasticity for changing needs => **Virtualization**

Formal definition

Cloud Computing is the delivery of computing as a **service** rather than a **product**,

whereby **shared resources, software**, and **information** are provided to computers and other devices,

as a **metered service** over a **network.**

Why Cloud Computing?

Applications enabled by cloud computing

- **High-growth applications**
  - When you startup gains traction, can you keep up?
  - Friendster(2001): Could not keep up with user growth
  - Facebook (2006): $Billion company today
  - Airbnb, Uber, Expedia, ...
- **Aperiodic applications**
  - How do you deal with sudden load peaks?
    - Amazon Prime Day: Aurora cloud database processed 148 billion transactions, stored 609 terabytes of data, and transferred 306 terabytes of data
    - Flipkart: Website crashed on their “Big Billion Day” sale
  - If you design for peak, how do you deal with low loads?
    - Amazon normal day: 1.3 billion transactions

Applications enabled by cloud computing(2)

- **On-off applications**
  - Scientific simulation using 1000s of computers
    - DNA Nexus and Baylor college of medicine analyzed DNA of more than 14,000 individuals
    - 2.4 million core-hours of computational time, 440 TB of results, 1PB of storage
  - Why not rent computing time to run such one-off experiments?
- **Periodic applications**
  - Stock market analysis
    - Mine market data during day
    - Analyze data during night
    - Different computational requirements at different times
  - Dynamic, flexible infrastructure can reduce costs, improve performance

"A Cloudy History of Time"

First large datacenters: ENIAC, ORDVAC, ILLIAC
Many used vacuum tubes and mechanical relays

1940

1950

1960

1970

1980

1990

2000

2010

Personal Computers
Berkeley NOW Project
Supercomputers
Server Farms (e.g., Oceano)

P2P Systems (90s-00s)
•Many Millions of users
•Many GB per day

Cloud

Data Processing Industry
- 1968: $70 M. 1978: $3.15 Billion

Timesharing Industry (1975):
•Market Share: Honeywell 34%, IBM 15%,
•Xerox 10%, CDC 10%, DEC 10%, UNIVAC 10%
•Honeywell 6000 & 635, IBM 370/168,
Xerox 940 & Sigma 9, DEC PDP-10, UNIVAC 1108

Open Science Grid

Grids (1980s-2000s):
•GriPhyN (1970s-80s)
•Open Science Grid and Lambda Rail (2000s)
•Globus & other standards (1990s-2000s)

Cloud computing: Full circle back to time sharing

New features of Cloud Computing

- **Massive scale.**

- **On-demand access:** Pay-as-you-go, no upfront commitment.
    - And anyone can access it

- **Data-intensive Nature:** What was MBs has now become TBs, PBs and XBs.
    - Daily logs, forensics, Web data, etc.

- **New Cloud Programming Paradigms:** MapReduce/Hadoop, Spark, NoSQL, NewSQL,… and many others.
    - High in accessibility and ease of programmability
    - Lots of open-source

Cloud Infrastructure

What is a server?

- **Servers are computers that provide "services" to "clients"**
    - Typically designed for reliability and to service a large number of requests
    - Dual-socket servers are the fundamental building block of cloud infrastructure
- **Organizations typically require many physical servers to provide various services**
    - Web server, database server, mail server, ...
- **Server hardware is becoming more compact**
    - conserving floor space
    - improving manageability
    - power and cooling

What is a rack?

- Servers are grouped, placed, and organized in racks
- Equipment are designed in a modular fashion to fit into rack units (1RU = 4.45cm)
- A single rack (6 ft or 180cms) can hold up to 42 1U servers

Figure: Global datasphere

What is a data center?

- **Facility used to house a large number of computer systems and associated components**
  - Air conditioning
  - Power supply
  - Hazard protection
  - Security and monitoring systems
  - Networking and connectivity
- Let's take a look at a special Microsoft datacenter (https://www.youtube.com/watch?v=L2oJw1a_qEM)

Trivia: World's largest datacenter

(2018) China Telecom. 10.7 Million sq. ft.

(2017) "The Citadel" Nevada. 7.2 Million sq. ft.

(2015) In Chicago!

- 350 East Cermak, Chicago, 1.1 MILLION sq. ft.
- Shared by many different "carriers"
- Critical to Chicago Mercantile Exchange

See:

https://www.gigabitmagazine.com/top10/top-10-biggest-data-centres-world

https://www.racksolutions.com/news/data-center-news/top-10-largest-data-centers-world/

Problems with privately owned data centers

- **Expensive to setup (High capital expenses or CAPEX)**
  - Real estate, server and peripherals, ...
- **Expensive to operate (High operational expenses or OPEX)**
  - Energy costs (Good data centers have efficiency of 1.7, 0.7 Watts lost for each 1W delivered to the servers)
  - Administration costs
- **Difficult for applications to grow/shrink**
  - How do we map applications to servers?
  - What if we over/under provision?
- **Low utilization (30% server usage considered good)**
  - Throw money at the performance problem (peak provisioning)
  - Uneven application fit: each server has CPU, memory, and disk: most applications exhaust one resource, stranding the others
  - Uncertainty in demand: Demand for a new service can spike quickly

What if

- **Turn the servers into a single large resource pool and let services dynamically expand and contract their footprint as needed?**
- **Two main requirements:**
    - Means for rapidly and dynamically satisfying application fluctuating resource needs
        - Provided by virtualization
    - Means for servers to quickly and reliably access shared and persistent data
        - Provided by programming models and distributed file/storage/database systems

What is a cloud then?

- **Single-site cloud**
  - A data center hardware and software that the vendors use to offer the computing resources and services
- **Geographically distributed cloud**
  - Multiple such sites, with each site perhaps having different structure and services

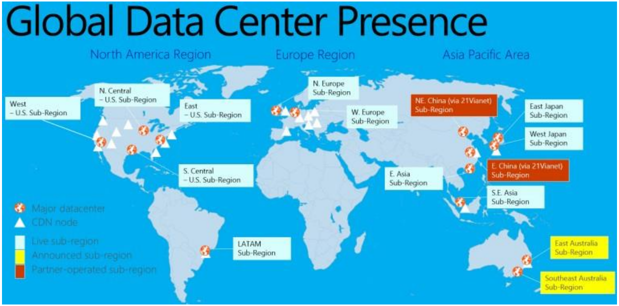


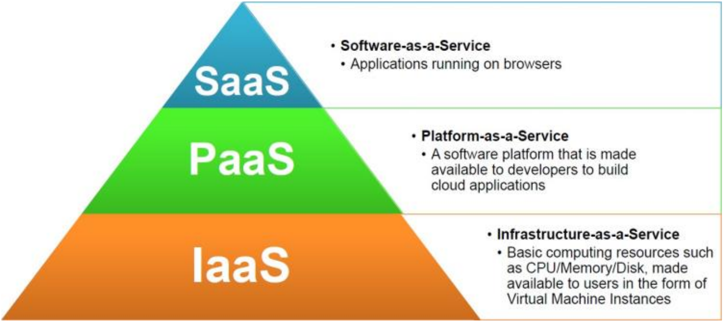
Figure: Azure: 1 million servers, 100 data centers across 90 countries.

Cloud Computing

IT as a service

- How do we offer IT as a service?
- Different users have different needs
    - Average end user
    - Mobile app developer
    - Enterprise systems architect
- Let us look at some service models

Basic cloud service models



SaaS

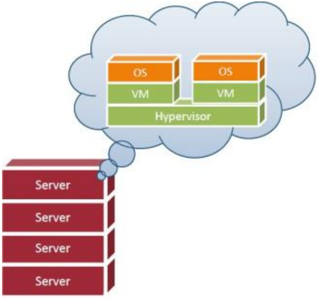
- Software is delivered as a service over the Internet, eliminating the need to install and run the application on the customer's own computer
- Simplifies maintenance and support
- You use SaaS products everyday
    - Gmail, Google docs, Youtube, ...
- Salesforce.com is a popular commercial pioneer (ERP, CRM, ...)

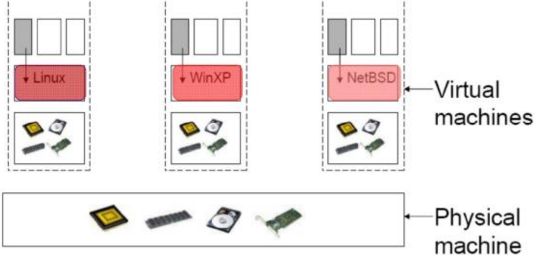
PaaS

- The Cloud provider exposes a set of tools (a platform) and APIs which allows users to create SaaS applications
- The SaaS application runs on the provider's infrastructure
- The cloud provider manages the underlying hardware and requirements
- Examples: Google App Engine, Windows Azure Web App service

IaaS

- The cloud provider leases to users Virtual Machine Instances (i.e., computer infrastructure) using the virtualization technology
- The user has access to a standard Operating System environment and can install and configure all the layers above it
- Ex: AWS EC2, Rackspace, Google Compute Engine





Other services models

- Hardware-as-a-service (HaaS)
  - You get access to barebones hardware machines, do whatever you want with them, Ex: Your own cluster
  - Not always a good idea because of security risks
- X-as-a-service, where X can be
  - Backend (BaaS), Desktop (DaaS), ...

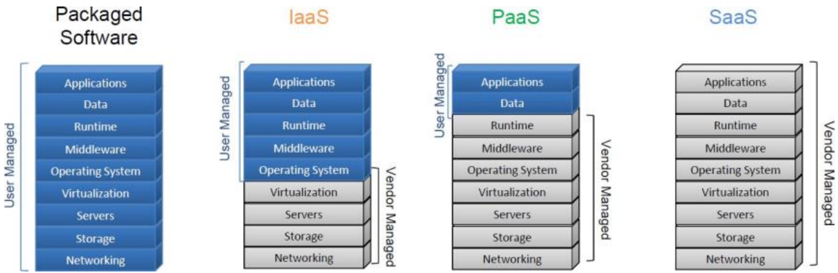
The Cloud Stack

- **Applications**
  - Cloud applications can range from Web applications to scientific computational jobs
- **Data**
  - Old SQL systems (Oracle, SQLServer)
  - NoSQL systems (MongoDB, Cassandra)
  - NewSQL systems (TimesTen, Impala, Hekaton)
- **Runtime environment**
  - Runtime platforms to support cloud programming models
  - Example: Hadoop, Spark

The Cloud Stack

- **Middleware**
  - Platforms for Resource Management, Monitoring, Provisioning, Identity Management and Security
- **Operating systems**
  - Standard Operating Systems used in Personal Computing
  - Packaged with libraries and software for quick deployment and provisioning
  - E.g., Amazon Machine Images (AMI) contain OS as well as required software packages as a "snapshot" for instant deployment
- **Virtualization (serverse, storage, networking)**
  - Key enabler of cloud computing
  - Providers resource virtualization, multitenancy
  - Ex: Amazon EC2 is based on the Xen virtualization platform, Azure based on HyperV

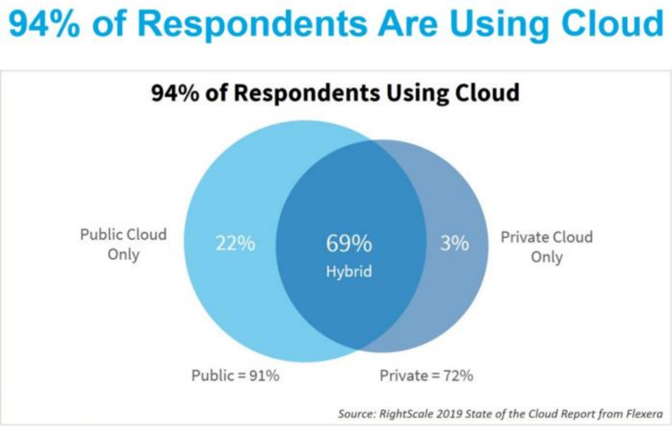
Cloud service models and the cloud stack



Types of clouds

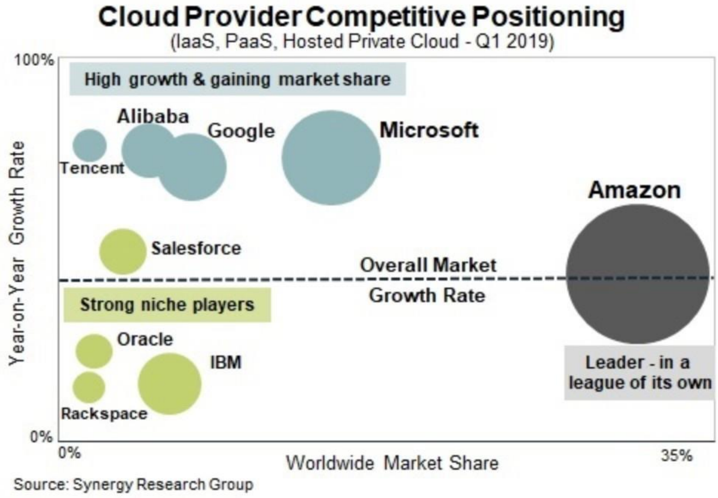
- **Public (external) cloud**
    - Open market for on demand computing and IT resources
    - Concerns: Limited SLA, reliability, availability, security, and trust
- **Private (internal) cloud**
    - For large enterprises with the budget and large-scale IT
- **Hybrid cloud**
    - Extend the private cloud(s) by connecting it to other public cloud vendors to make use of their available cloud services
    - Use the local cloud, and when you need more resources, burst into the public cloud

Cloud adoption



- All major cloud providers are extending their offering to private and hybrid markets
    - Example: Google Anthos, Microsoft AzureStack

Know the leaders



Cloud Economics

Economics of cloud computing

- What is the value proposition for cloud computing?
- How did Cloud Computing emerge from business / industry rather than from Academia?
- How did software service models evolve?

Cost of IT

- **When you are using IT there are three primary costs associated with it:**
  - Software cost (Media + License cost/user)
  - Support cost (vendor support, updates, ...)
  - Management cost (Manpower, IT infrastructure, ...)

Traditional model

- a.k.a Classic model
- Software provider develops software and charges a license fee per user for the client
- The provider may charge a support fee /user
- The management of the software is the clients responsibility
    - Up to 4x the cost of the actual software per year!
    - Infrastructure, Manpower, software maintenance
- Traditional Software example: Oracle, SQL Server, Outlook, ...

Software service models

Open Source Model

- a.k.a "Free" model
- Software provider packages Open Source Software and provides it at little or no cost to the client
- The provider makes money on support, charges a higher fee than traditional model
- The cost of Managing the software remains the same as Traditional Model
    - Up to 4x the cost of the actual software per year
    - Infrastructure, Manpower, software maintenance
- Traditional Software example: Oracle, SQL Server, Outlook, ...

Software service models

	Traditional	Open Source
Software Cost	$4000 /user (one-time)	$0 /user
Support Cost	$800 /user /year	$1600 /user /year
Management Cost	Up to 4x the cost of Software!	
Deployment Location	Client Side	

Outsourcing Model

- Primary cost of Software Management is in Manpower
- Why not delegate the management of software to a country with cheaper labor costs?
- Outsource the management of software for a flat fee – keep IT management costs under control

Software service models

	Traditional	Open Source	Outsourcing
Software Cost	$4000 /user (one-time)	$0 /user	$4000 /user (one-time)
Support Cost	$800 /user /year	$1600 /user /year	$800 /user /year
Management Cost	Up to 4x the cost of Software!		< 1300 /user /month
Deployment Location	Client Side		Client or Provider Side

Hybrid and Hybrid+ models

- Business Software Requirements do not change often.
  - ERP, Financials, CRM etc.
- Why reinvent the wheel? Standardize, Specialize and Repeat
  - Create a flexible version of the Software that can be quickly configured and deployed.
  - Automate support through remote access.
- Sell easy to deploy software to many clients.
  - Decrease the Margin
  - Increase the Customers
- Hybrid+ is more advanced – charge a flat monthly fee for the software, support and management

Software service models

	Traditional	Open Source	Outsourcing	Hybrid	Hybrid+
Software Cost	$4000 /user (one-time)	$0 /user	$4000 /user (one-time)	$4000 /user (one-time)	$300 / user month
Support Cost	$800 /user /year	$1600 /user /year	$800 /user /year	$800 /user /year	
Management Cost	Up to 4x the cost of Software!		Bid < 1300 /user /month	$150 /user /month	
Deployment Location	Client Side		Client or Provider Side		

Software-as-a-service and cloud computing

- Develop Web Application
- Offer to customers over Internet
- No deployment costs
- Amortize Management and Support costs over many clients

Software service models

	Traditional	Open Source	Outsourcing	Hybrid	Hybrid+	SaaS
Software Cost	$4000 /user (one-time)	$0 /user	$4000 /user (one-time)	$4000 /user (one-time)	$300 / user month	< $100 /user /month
Support Cost	$800 /user /year	$1600 /user /year	$800 /user /year	$800 /user /year		
Management Cost	Up to 4x the cost of Software!		Bid < 1300 /user /month	$150 /user /month		
Deployment Location	Client Side		Client or Provider Side			Provider Side