

CS5630

---

# Foundation (1): Cloud Computing Overview

**Prof. Supreeth Shastri**  
*Computer Science*  
*The University of Iowa*

# Results from the Classroom Survey

---

What is your primary expectation from the course?

to prepare for job/internship	22 respondents	71 %	
to use cloud in my research	4 respondents	13 %	
to whet my curiosity	4 respondents	13 %	
<b>something else</b>	1 respondent	3 %	 ✓

What is your experience with the cloud?

[as user] to store my data	22 respondents	71 %	
[as user] of software services	17 respondents	55 %	
[as programmer] of cloud hardware/platforms	9 respondents	29 %	
something else	4 respondents	13 %	

Which of the following cloud platforms have you used?

Amazon Web Services	12 respondents	39 %	
Google Cloud	12 respondents	39 %	
Microsoft Azure	9 respondents	29 %	
something else	8 respondents	26 %	

# Lecture goals

---

*A comprehensive overview of cloud computing*

- *What, why, and how of the cloud*
- *Technical and economic foundations*
- *Challenges and opportunities*

**Above the Clouds: A Berkeley View of Cloud Computing**



*Michael Armbrust  
Armando Fox  
Rean Griffith  
Anthony D. Joseph  
Randy H. Katz  
Andrew Konwinski  
Gunho Lee  
David A. Patterson  
Ariel Rabkin  
Ion Stoica  
Matei Zaharia*

Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2009-28  
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>

February 10, 2009

*Cloud computing refers to both **the computing resources delivered as services over the Internet** and **the hardware and software systems in datacenters** that provide those services*

lectures 6 - 9

lectures 3 - 5



(circa 2006)

## Elastic Cloud Compute (EC2)

Standard Instances	Linux/UNIX	Windows
Small (Default)	\$0.10 per hour	\$0.125 per hour
Large	\$0.40 per hour	\$0.50 per hour
Extra Large	\$0.80 per hour	\$1.00 per hour

- Small Instance (Default) 1.7 GB of memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB of instance storage, 32-bit platform
- Large Instance 7.5 GB of memory, 4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each), 850 GB of instance storage, 64-bit platform
- Extra Large Instance 15 GB of memory, 8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each), 1690 GB of instance storage, 64-bit platform

## Simple Storage Service (S3)

Storage
■ \$0.150 per GB – first 50 TB / month of storage used
■ \$0.140 per GB – next 50 TB / month of storage used
■ \$0.130 per GB – next 400 TB /month of storage used
■ \$0.120 per GB – storage used / month over 500 TB
Data Transfer
■ \$0.100 per GB – all data transfer in
■ \$0.170 per GB – first 10 TB / month data transfer out
■ \$0.130 per GB – next 40 TB / month data transfer out
■ \$0.110 per GB – next 100 TB / month data transfer out
■ \$0.100 per GB – data transfer out / month over 150 TB



**Explore Our Products**

Analytics	Application Integration	Blockchain	Business Applications	Cloud Financial Management
<b>Compute</b>	Containers	Customer Engagement	Database	Developer Tools
End User Computing	Front-End Web & Mobile	Game Tech	Internet of Things	Machine Learning
Management & Governance	Media Services	Migration & Transfer	Networking & Content Delivery	Quantum Technologies
Robotics	Satellite	Security, Identity & Compliance	Serverless	<b>Storage</b>

The screenshot displays the "Explore Our Products" section of the AWS website. It features a grid of 25 service icons, each with a corresponding name below it. The services are categorized into five main groups: Compute, Storage, Database, Machine Learning, and others. Two specific services, "Compute" and "Storage", are highlighted with orange circles around their respective icons.

**A collection of 200+ services**

**Three models of computing:**  
Infrastructure as a Service (IaaS),  
Platform as a Service (PaaS), and  
Software as a Service (SaaS)

**Four different access methods:**  
Web-based management console;  
Command line tools; Software  
development kits, or RESTful APIs

# Characterizing Amazon's public cloud

## On-demand

*users can procure resources if/when needed; no need for making commitments a priori*

## Scalability

*offers an illusion of infinite scalability; allows users to scale their resources up/down in real-time*

## Billing model

*allows users to trade capital expense for operating expense; fine-grained billing proportional to the time and size of resources used*

## Strong guarantees

*services come with high levels of availability and reliability (three to four nines)*

## Ease of administration

*hardware (and low-level system software) are virtualized, so users don't have to maintain any infrastructure*

## Global deployment

*users have the ability to select the geographical regions in which their data/compute will reside*

# Public vs. Private Clouds

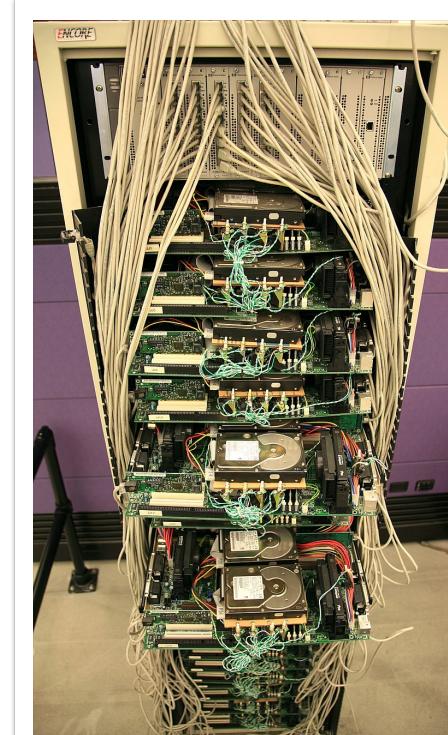
*If the cloud platform and its services are made available to the public or if it is restricted for the internal use of one or more private organizations*

*Google built one of the first global scale clouds (out of necessity) to support its search and other applications.*

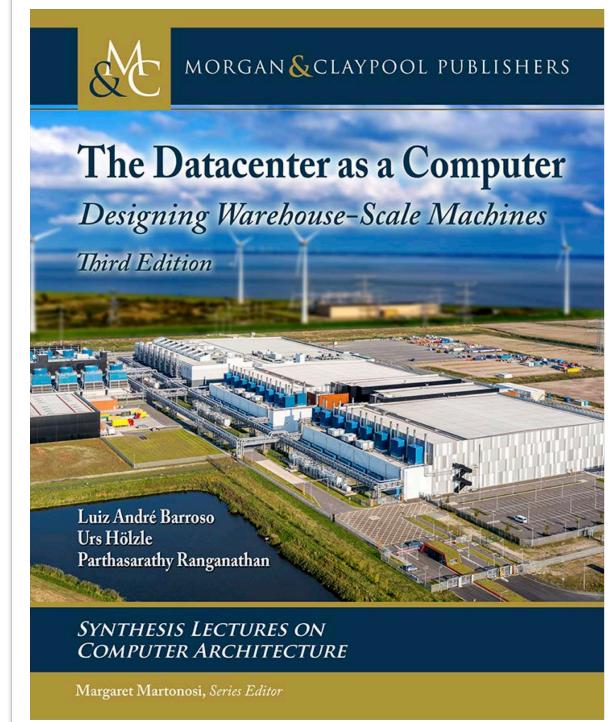
*In 2011, Google became a full-fledged public cloud provider.*



(circa 1998)



(circa 2009)



# Why now? Key enablers of cloud computing

---

## Inevitable rise of distributed systems/infrastructure

- ▶ In early 2000, companies realized that vertical scaling of servers has hit a ceiling (i.e., one cannot buy powerful enough servers to keep up with increasing load)
- ▶ This resulted in an increased focus on horizontal scaling a.k.a. building a distributed infrastructure using 1000s of commodity servers
- ▶ Companies such as Google and Amazon developed expertise in building and operating massive datacenters; designed software systems to make them work reliably

### Q: who is Google cloud's first customer?

- Google operates eight global-scale applications each with billion+ user base
- Google applications run with average uptime of 99.99%

# Why now? Key enablers of cloud computing

---

## Advances in resource virtualization

- ▶ First commercial virtualization (IBM Mainframes in **1970**)
- ▶ First virtualization of x86 architecture (VMware ESX in **1998**)
- ▶ Significant reduction in virtualization overheads (Linux containers in **2008**)

## Increasing broadband speeds

- ▶ Faster and better quality access to the Internet over the last two decades; average broadband speed in the US ~100 Mbps

## Emergence of applications that benefited from the cloud model

- ▶ Large-scale data analytics. For e.g., election campaigns
- ▶ Interactive mobile applications. For e.g., Pokemon Go
- ▶ Video streaming. For e.g., YouTube

# **Cloud Economics**

*Economic perspectives on being a cloud user/provider*

# Why would anyone be a cloud provider?

*Building, provisioning, and maintaining datacenters required to power cloud platforms takes hundreds of million dollars. Is selling computing for cents-an-hour worth it?*

---

- ➡ **Leveraging existing investments:** clouds of Amazon, Google, and Microsoft were initially built to serve their internal computing needs.
- ➡ **Economies of scale:** Large-scale datacenters (>10K servers) can purchase hardware, network bandwidth, and electricity for 5-10x lower price than regular businesses.
- ➡ **At scale, cents add up to millions:** statistical multiplexing allows cloud providers to achieve higher cost-efficiency, which can then be passed on to customers to attract volume business.
- ➡ **Become a de-facto computing platform:** cloud providers have gradually introduced more profitable higher level offerings such as ML, security, analytics etc

# Why would anyone be a cloud user?

*This is more obvious than the previous question. Yet, is it always economical?*

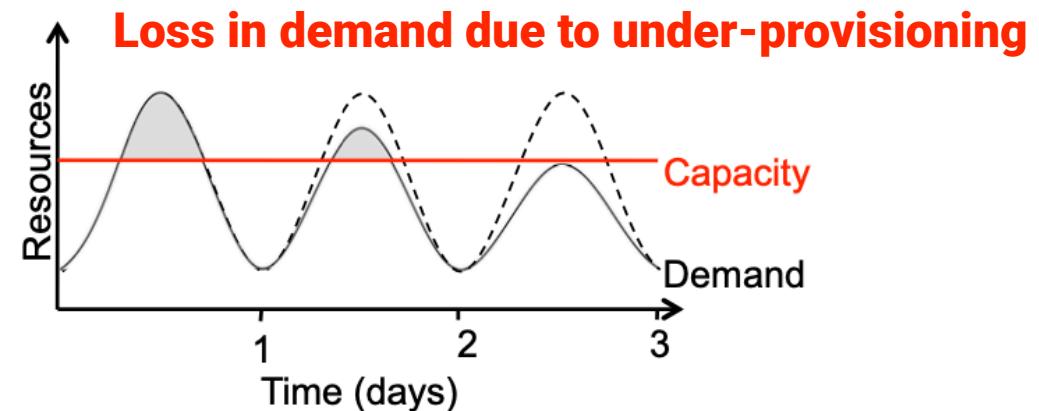
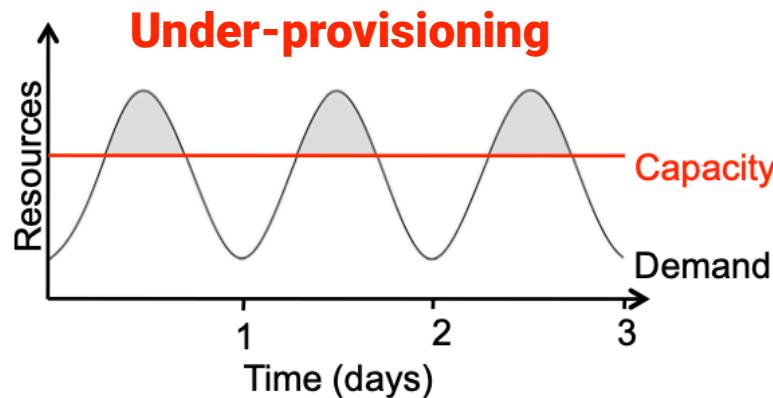
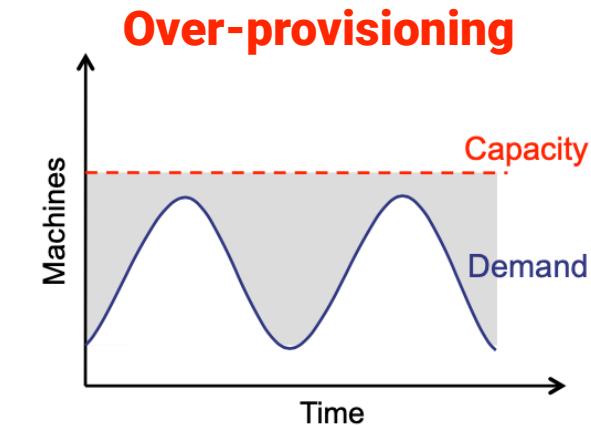
---

- ▶ **Convert CapEx to OpEx:** cloud's pay-as-you-go model lets users eliminate capital expenses and simply focus on operational expenses.
- ▶ **Cost associativity saves time:** only on cloud does it costs the same money to use 1000 servers for 1 hour as 1 server for 1000 hours.
- ▶ **Elasticity in usage:** cloud resources can be consumed non-uniformly over time (for e.g., use 100 server hours today, 10 server hours tomorrow, and nothing for day-after).
- ▶ **Transfer provisioning risk:** incorrect provisioning of resources results in either poor application performance (thereby dissatisfied customers), or wasted resources (thereby loss in revenue). Users can automatically transfer this risk to cloud providers.

# Elasticity and Provisioning Risk

Static provisioning, as is common in self-managed infrastructure, results in either over-provisioning (figure to the right) or under-provisioning (figures to the bottom)

Cloud elasticity enables a fine-grained matching of demand and consumption



# Why would anyone be a cloud user?

*This is more obvious than the previous question. Yet, is it always economical?*

---

*Consider a web service hosted on cloud vs. a private datacenter. Assuming that revenue is proportional to user-hours (i.e., total time a customer spends on the service), the following equations approximate the expected profit:*

$$\text{Expected-profit}_{\text{cloud}} = \text{User-hours}_{\text{cloud}} \bullet (\text{Revenue} - \text{Cost}_{\text{cloud}})$$

$$\text{Expected-profit}_{\text{private-dc}} = \text{User-hours}_{\text{private-dc}} \bullet (\text{Revenue} - \frac{\text{Cost}_{\text{private-dc}}}{\text{Utilization}})$$

# **Reflections on 15-years of Cloud**

## Obstacles & Opportunities

---

*Solved or  
became irrelevant*

1	Service availability	Use multiple clouds
4	Data transfer bottlenecks	FedExing disks; Better networking
6	Scalable storage	Invent scalable stores
8	Scaling quickly	Invert auto scaler
9	Reputation fate sharing	Offer reputation-guarding services
10	Software licensing	Pay-per-use licenses

*Yet to be solved*

2	Data lock-in	Standardize APIs
3	Data confidentiality	Integrate security and privacy
5	Performance unpredictability	Improved VM, scheduling
7	Bugs in distributed systems	Invert debuggers for distributed VMs

“One reason you should not use cloud is that you lose control. You're putty in the hands of whoever developed that software.”



– Richard Stallman  
In The Guardian (11/29/2008)

# **Spot Quiz (ICON)**