CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

INTRODUCTION TO PART 1

What this Course is about

- This course is about the internals of cloud computing
  - Not how to use cloud systems or write cloud applications (separate course in Cloud Specialization: *Cloud Applications*)
  - Not about networking (separate course in Cloud Specialization: *Cloud Networking*)
- We'll go underneath the hood and look at distributed systems that underlie today's cloud computing technologies

What this Course is about (2)

- We’ll discuss
  - Concepts
  - Techniques
  - Industry systems, including open source (from the inside)
- The course is a mix of
  - Distributed systems
  - Distributed algorithms
  - As applied to cloud computing

Syllabus for Part 1

- Introduction: Clouds, MapReduce, Key-value stores
- Classical precursors: Peer-to-peer systems, Grids
- Widely-used algorithms: Gossip, Membership, Paxos
- Classical algorithms: Time and Ordering, Snapshots, Multicast
- Fun: Interviews with leading managers and researchers, from both industry and academia

Exercises

- 2 Homeworks
- (Optional) 1 Programming Assignment (C++)
  - Implement a membership protocol inside an emulator
- 1 Exam

Onward!

- Cloud computing is an exciting area to be studying, very dynamic and continuously changing
- I’m looking forward to working with you!
- Come, let’s tour the landscape.

CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

AN ORIENTATION TO CLOUD COMPUTING

What this Lecture is about

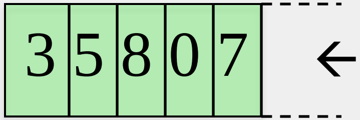
- Covers basic concepts in Computer Science that will be assumed in the Cloud Computing Concepts (C3) course
- For those of you already familiar, it's a refresher
- Use this as reference if you don't understand (during the course) how the basics are being used

What's in this Lecture

I. Basic datastructures
II. Processes
III. Computer architecture
IV. O() notation
V. Basic probability
VI. Miscellaneous

I. Basic Datastructures: Queue

- Queue: First-in First-out datastructure

Remove from head ← 

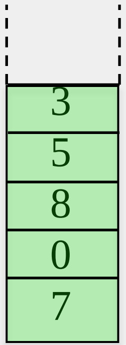
← **Insert** at tail

- Next item dequeued (removed) is 3.
  - Then 5
  - Then 8
  - And so on

Basic Datastructures: Stack

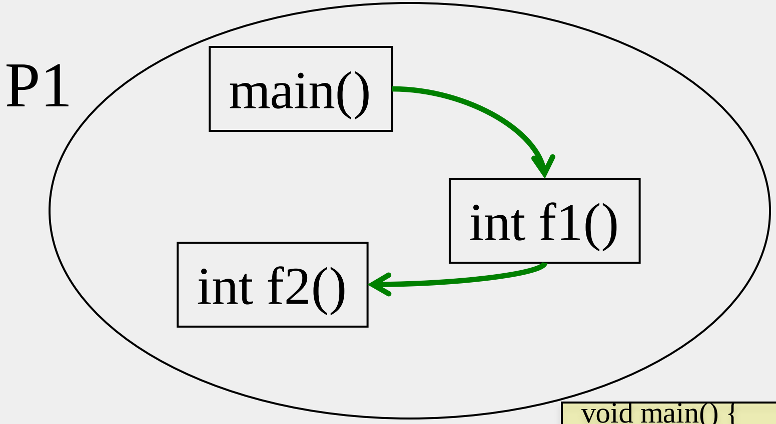
- Stack: First-in *Last-out* datastructure

Remove (Pop) from top ← ← **Insert (Push)** at top



- Insert (Push) 9: goes to top
- Remove (Pop): gets 9
- Pop: gets 3
- Next pop: gets 5 (and so on)

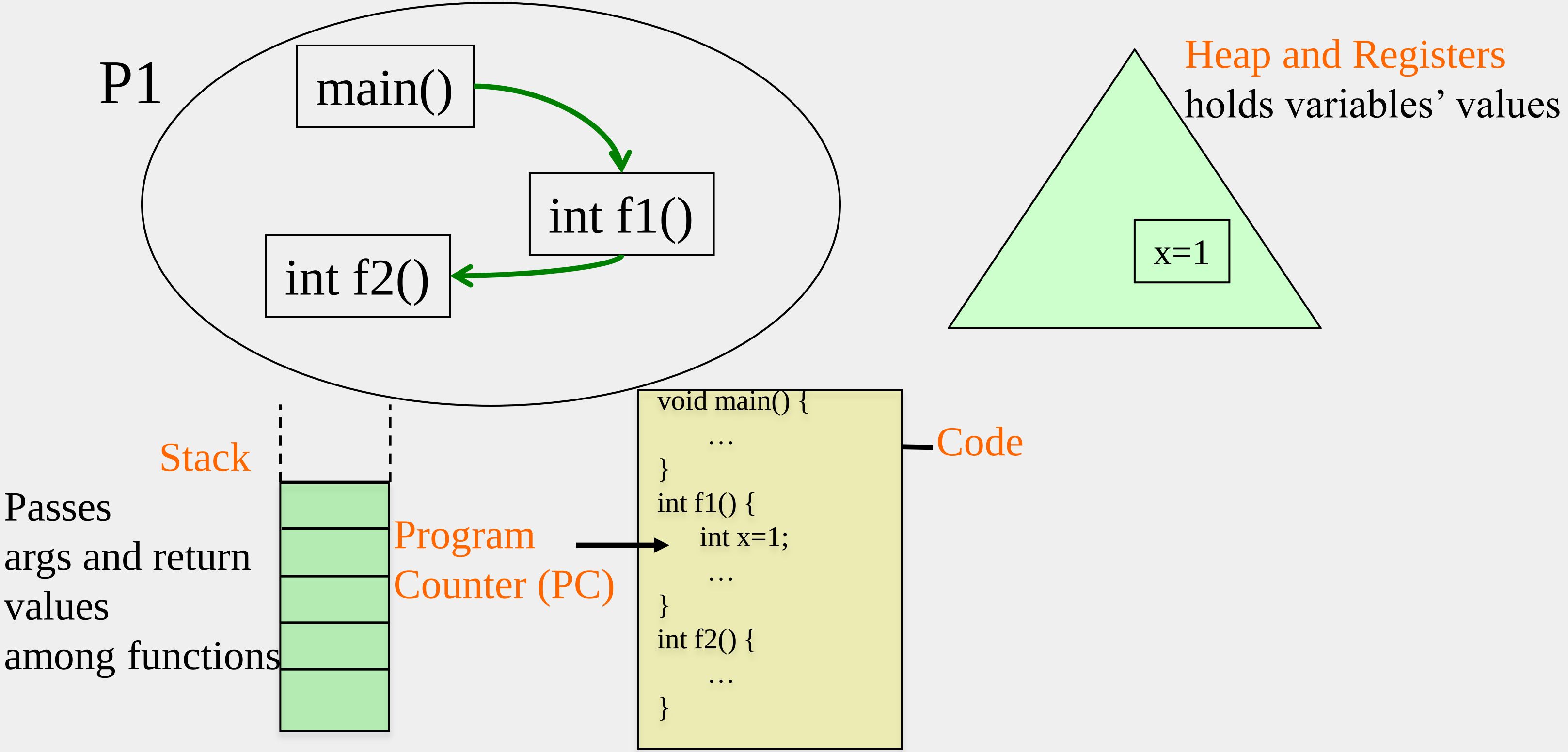
II. Process = a Program in Action



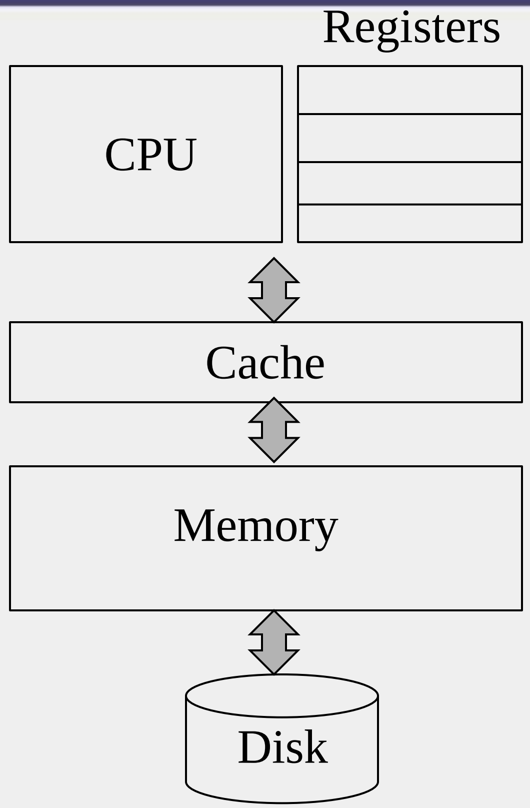

```
void main() {
      …
}
int f1() {
      int x=1;
      …
}
int f2() {
      …
}
```

Code

Inside a Process



III. Computer Architecture (Simplified)



Computer Architecture (2)

- A program you write (C++, Java, etc.) gets compiled to low-level machine instructions
  - Stored in file system on disk
- CPU loads instructions in batches into memory (and cache, and registers)
- As it executes each instruction, CPU loads data for instruction into memory (and cache, and registers)
  - And does any necessary stores into memory
- Memory can also be flushed to disk
- This is a highly simplified picture!
  - (but works for now)

IV. Big O() Notation

- One of the most basic ways of analyzing algorithms
- Describes *upper bound* on behavior of algorithm as some variable is scaled (increased) to infinity
- Analyzes run-time (or another performance metric)
- Worst-case performance

Big O() Notation: Informal Definition

- "An algorithm A is O(foo)"

  Means

- "Algorithm A takes < c * foo time to complete, for some constant c, beyond some input size N"
- Usually, foo is a function of input size N
  - –e.g., an algorithm is O(N)
  - –e.g., an algorithm is $O(N^2)$
- We don't state the constants in Big O() notation

Big O() Notation: Example 1

- "Searching for an element in an unsorted list is O(N), where N = size of list"
- Have to iterate through list
- Worst-case performance is when that element is not there in the list, or is the last one in the list
- Thus involves N operations
- Number of operations < c* N, where c=2.

Big O() Notation: Example 2

- "Insertion sorting of an unsorted list is $O(N^2)$, where N = size of list"
- Insertion sort Algorithm:
  - Create new empty list
  - For each element in unsorted list
    - Insert element into sorted list at appropriate position
- First element takes 1 operation to insert
- Second element takes (in worst case) 2 operations to insert
- i-th element takes i operations to insert
- Total time = $1+2+3+\ldots+N=N(N+1)/2 < 1*N^2$

V. Basic Probability

- Set=collection of things
  - S="Set of all humans who live in the world"
- Subset=collection of things that is part of a larger set
  - S2="Set of all humans who live in Europe"
  - S2 is a subset of S

Basic Probability

- Any event has a probability of happening
- If you wake up at a random hour of the day, what is the probability of the event that the time is between 10 am and 11 am?
- There are 24 hours in a day
  - Set of hours contains 24 elements: 12 am, 1 am, 2 am, … 10 am, 11 am, …11 pm
- You pick one hour at random
- Probability you pick 10 am = 1/24

Multiplying Probabilities

- E1 is an event
- E2 is an event
- E1 and E2 are <u>independent</u> of each other
- Then: Prob(E1 AND E2) = Prob(E1) * Prob(E2)
- You have three shirts: blue, green, red
- You wake up at a random hour and blindly pick a shirt
- Prob(You woke up between 10 am and 11 am AND that you're wearing a green shirt) = (1/24) * (1/3) = 1/72
- But beware: can't multiply probabilities if events are dependent (i.e., influence each other)!

Adding Probabilities

- E1 is an event
- E2 is an event
- Then:

Prob(E1 OR E2) = Prob(E1) + Prob(E2) – Prob(E1 AND E2)

- If you don't know Prob(E1 AND E2), then you can write

Prob(E1 OR E2) ≤ Prob(E1) + Prob(E2)

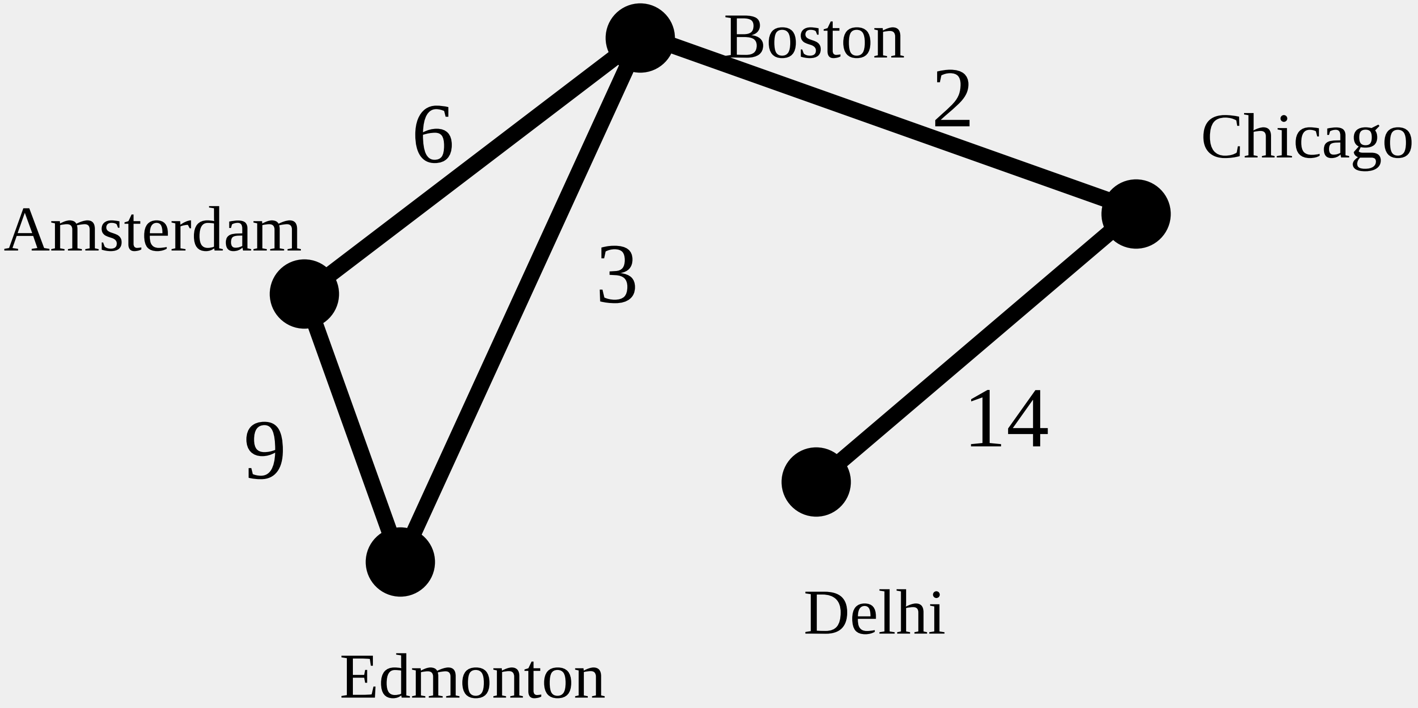
VI. DNS

- DNS = Domain Name System
- Collection of servers, throughout the world
- Input to DNS: a URL, e.g., coursera.org
  - URL is a name, a human-readable string that uniquely identifies the object

DNS (2)

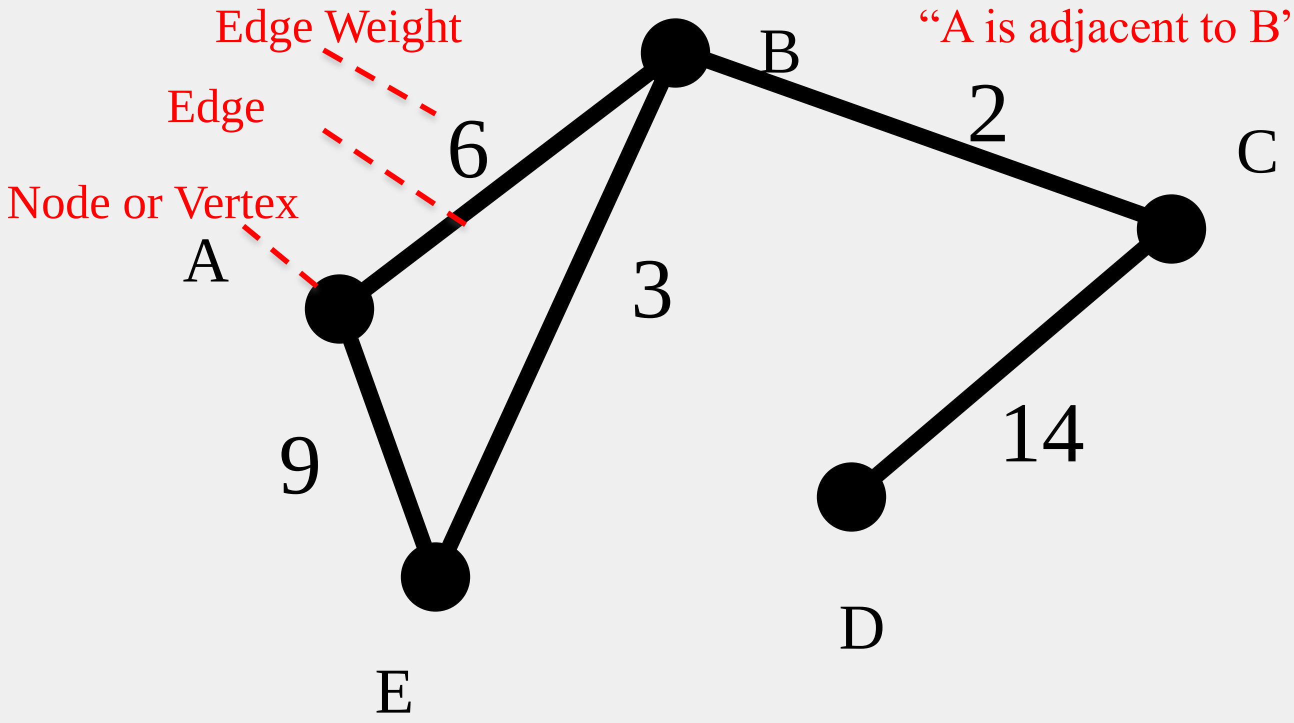
- Output from DNS: IP address of a web server that hosts that content
  - IP address is an ID, a unique string pointing to the object. May not be human readable.
- IP address may refer to either
  - Web server actually hosting that content, or
  - An indirect server, e.g., a CDN (content distribution network) server, e.g., from Akamai

VII. Graphs



GRAPHS (2)

“A is adjacent to B” = There is an A-B edge



What We Covered

I. Basic datastructures
II. Processes
III. Computer architecture
IV. O() notation
V. Basic probability
VI. Miscellaneous

CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

INTRODUCTION TO CLOUDS

Lecture A

WHY CLOUDS?

The Hype!

- Gartner in 2009 – Cloud computing revenue will soar faster than expected and will exceed $150 billion by 2013. It will represent 19% of IT spending by 2015.
- IDC in 2009: "Spending on IT cloud services will triple in the next 5 years, reaching $42 billion."
- Forrester in 2010 – Cloud computing will go from $40.7 billion in 2010 to $241 billion in 2020.
- Companies and even federal/state governments using cloud computing now: fedbizopps.gov

Many Cloud Providers

- AWS: Amazon Web Services
  - EC2: Elastic Compute Cloud
  - S3: Simple Storage Service
  - EBS: Elastic Block Storage
- Microsoft Azure
- Google Compute Engine
- Rightscale, Salesforce, EMC, Gigaspaces, 10gen, Datastax, Oracle, VMWare, Yahoo, Cloudera
- And many, many more!

Two Categories of Clouds

- Can be either a (i) public cloud, or (ii) private cloud
- Private clouds are accessible only to company employees
- Public clouds provide service to any paying customer:
  - Amazon S3 (Simple Storage Service): store arbitrary datasets, pay per GB-month stored
  - Amazon EC2 (Elastic Compute Cloud): upload and run arbitrary OS images, pay per CPU hour used
  - Google App Engine/Compute Engine: develop applications within their App Engine framework, upload data that will be imported into their format, and run

Customers Save Time and $$$

- Dave Power, Associate Information Consultant at Eli Lilly and Company: "With AWS, a new server can be up and running in three minutes (it used to take Eli Lilly seven and a half weeks to deploy a server internally) and a 64-node Linux cluster can be online in five minutes (compared with three months internally). … It's just shy of instantaneous."

- Ingo Elfering, Vice President of Information Technology Strategy, GlaxoSmithKline: "With Online Services, we are able to reduce our IT operational costs by roughly 30% of what we' re spending."

- Jim Swartz, CIO, Sybase: "At Sybase, a private cloud of virtual servers inside its datacenter has saved nearly $US2 million annually since 2006, because the company can share computing power and storage resources across servers."

- Hundreds of startups in Silicon Valley can harness large computing resources without buying their own machines.

But what exactly IS a cloud?

- Next lecture!

CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

INTRODUCTION TO CLOUDS

Lecture B

WHAT IS A CLOUD?

What is a Cloud?

- It' s a cluster!
- It' s a supercomputer!
- It' s a datastore!
- It' s Superman!

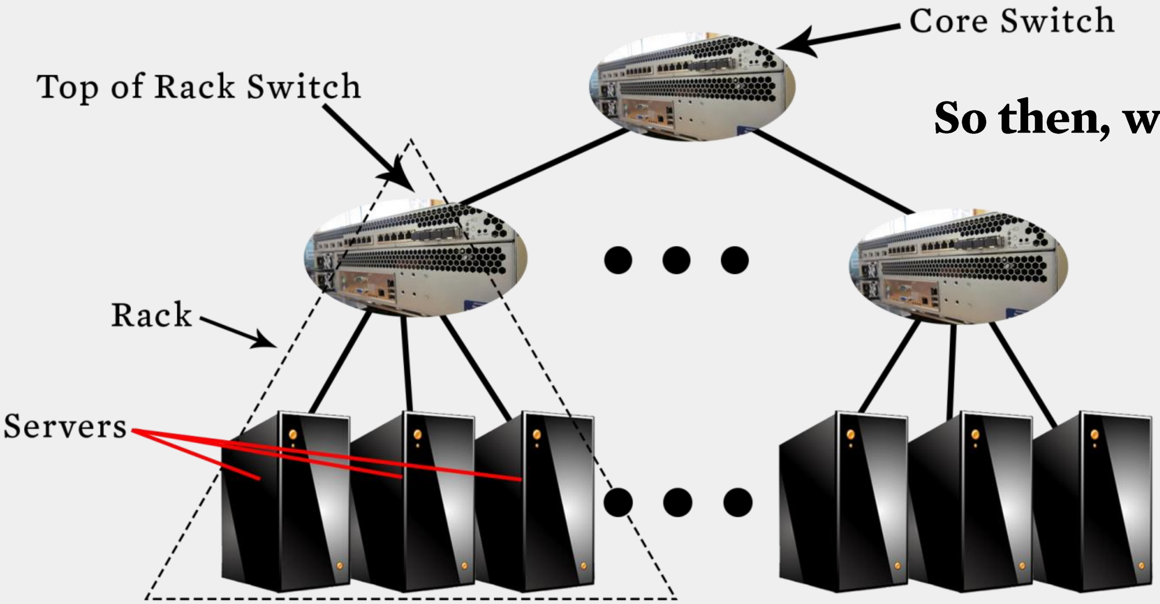
- None of the above
- All of the above

- Cloud = Lots of storage + compute cycles nearby

What is a Cloud?

- A single-site cloud (aka “datacenter”) consists of
  - Compute nodes (grouped into racks)
  - Switches, connecting the racks
  - A network topology, e.g., hierarchical
  - Storage (backend) nodes connected to the network
  - Front-end for submitting jobs and receiving client requests
  - Software services
- A geographically distributed cloud consists of
  - Multiple such sites
  - Each site perhaps with a different structure and services

A Sample Cloud Topology



So then, what is a cluster?

CLOUD COMPUTING CONCEPTS

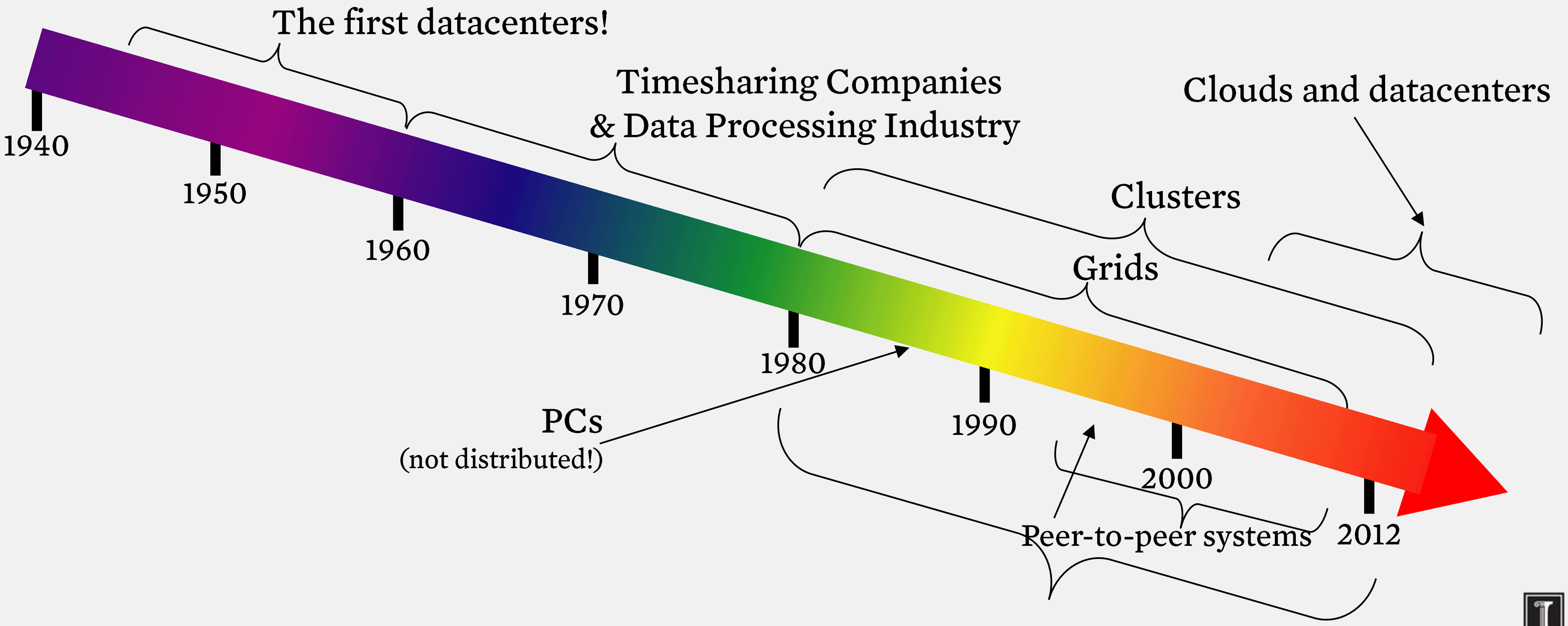
with Indranil Gupta (Indy)

INTRODUCTION TO CLOUDS

Lecture C

HISTORY

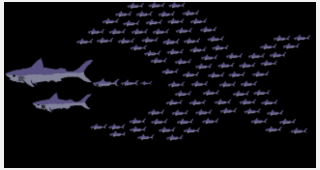
"A Cloudy History of Time"



"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
2012



Berkeley NOW Project
Supercomputers
Server farms (e.g., Oceano)

P2P Systems (90s–00s)
- Many millions of users
- Many GB per day

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

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

Clouds

Trends: Technology

- Doubling periods – storage: 12 months, bandwidth: 9 months, and (what law is this?) CPU compute capacity: 18 months
- Then and Now
  - Bandwidth
    - 1985: mostly 56Kbps links nationwide
    - 2012: Tbps links widespread
  - Disk capacity
    - Today' s PCs have TBs, far more than a 1990 supercomputer

Trends: Users

- Then and Now
  - Biologists:
    - 1990: were running small single-molecule simulations
    - 2012: CERN′ s Large Hadron Collider producing many PB/year

Prophecies

- 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
  - Have today' s clouds brought us closer to this reality? Think about it.

CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

INTRODUCTION TO CLOUDS

Lecture D

WHAT'S NEW IN TODAY'S CLOUDS

Four Features New in Today's Clouds

I. Massive scale.

I. On-demand access: Pay-as-you-go, no upfront commitment.
- Anyone can access it

II. Data-intensive Nature: What was MBs has now become TBs, PBs and XBs.
- Daily logs, forensics, Web data, etc.
- Humans have data numbness: Wikipedia (large) compress is only about 10 GB!

III. New Cloud Programming Paradigms: MapReduce/Hadoop, NoSQL/Cassandra/MongoDB and many others.
- High in accessibility and ease of programmability
- Lots of open-source

Combination of one or more of these gives rise to novel and unsolved distributed computing problems in cloud computing.

I. Massive Scale

- Facebook [GigaOm, 2012]
  - 30K in 2009 -> 60K in 2010 -> 180K in 2012
- Microsoft [NYTimes, 2008]
  - 150Kmachines
  - Growth rate of 10K per month
  - 80K total running Bing
- Yahoo! [2009]:
  - 100K
  - Split into clusters of 4000
- AWS EC2 [Randy Bias, 2009]
  - 40,000 machines
  - 8 cores/machine
- eBay [2012]: 50K machines
- HP [2012]: 380K in 180 DCs
- Google: A lot

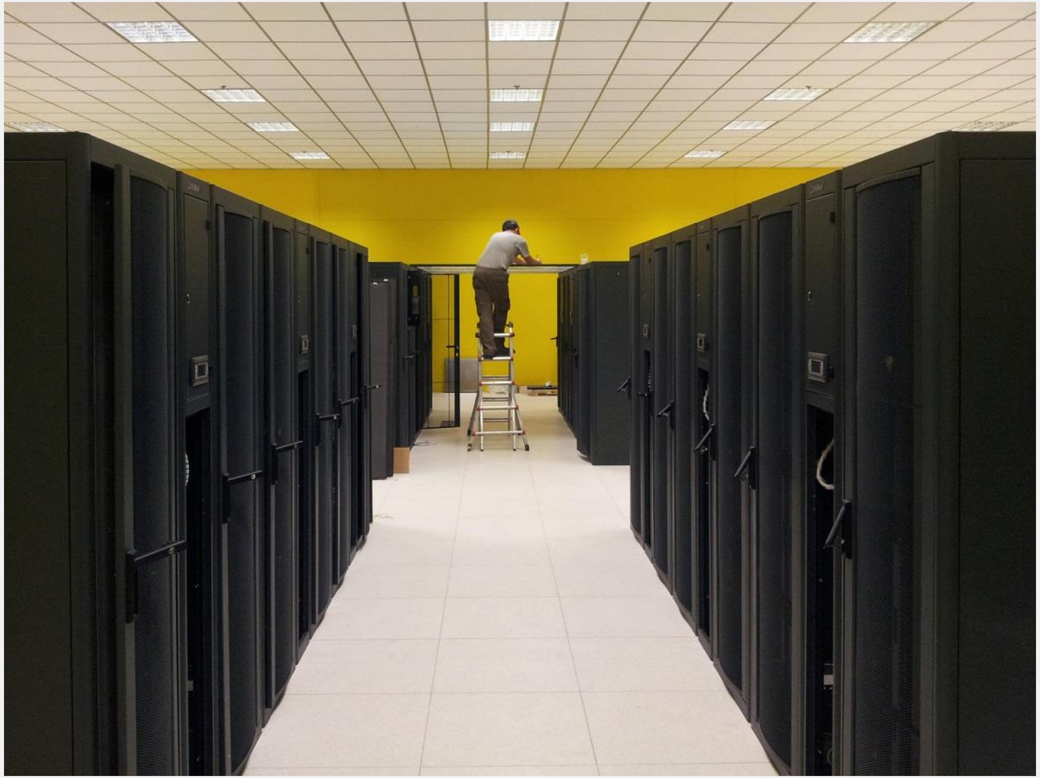
What does a datacenter look like from inside?

- A virtual walk through a datacenter

- Additional reference:

http://gigaom.com/cleantech/a-rare-look-inside-facebooks-oregon-data-center-photos-video/

Servers

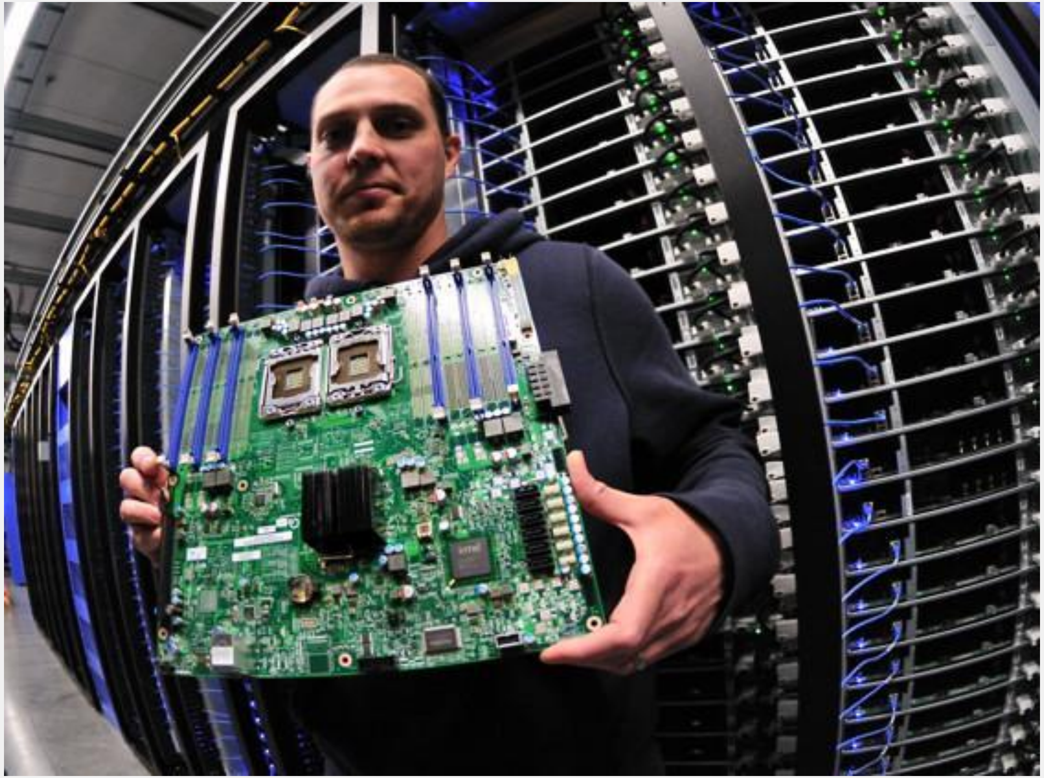


Front

Servers

Back

Servers



Inside

Intel Free Press | CC BY 2.0

Servers

Some highly secure
(e.g., financial info)

POWER

U.S. Army | CC BY 2.0

Greg Goebel | CC BY-SA 2.0

David Goehring | CC BY 2.0

- WUE = Annual Water Usage / IT Equipment Energy (L/kWh)
  (low is good)

- PUE = Total Facility Power / IT Equipment Power
  (low is good - e.g., Google = 1.11)

Off-site

On-site

COOLING


Leslie Hawthorn | CC BY-SA 2.0


Tim Dorr | CC BY-SA 2.0

ChrisDag| CC BY 2.0

- Air sucked in
- Combined with purified water
- Moves cool air through system

Extra - Fun Videos to Watch

- Microsoft GFS Datacenter Tour (Youtube)
  - http://www.youtube.com/watch?v=hOxA1l1pQIw

- Timelapse of a Datacenter Construction on the Inside (Fortune 500 company)
  - http://www.youtube.com/watch?v=ujO-xNvXj3g

CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

INTRODUCTION TO CLOUDS

Lecture E

NEW ASPECTS OF CLOUDS

II. On-demand access: *aaS Classification

On-demand: renting a cab vs. (previously) renting a car, or buying one. Ex.:

- AWS Elastic Compute Cloud (EC2): a few cents to a few $ per CPU hour
- AWS Simple Storage Service (S3): a few cents to a few $ per GB-month

- HaaS: Hardware as a Service
  - 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
- IaaS: Infrastructure as a Service
  - You get access to flexible computing and storage infrastructure. Virtualization is one way of achieving this (what's another way, e.g., using Linux). Often said to subsume HaaS.
  - Ex: Amazon Web Services (AWS: EC2 and S3), Eucalyptus, Rightscale, Microsoft Azure

II. On-demand access: *aaS Classification

- PaaS: Platform as a Service
  - You get access to flexible computing and storage infrastructure, coupled with a software platform (often tightly)
  - Ex: Google′ s AppEngine/Compute Engine (Python, Java, Go)
- SaaS: Software as a Service
  - You get access to software services, when you need them. Often said to subsume SOA (Service-Oriented Architectures).
  - Ex: Google docs, MS Office on demand

III. Data-intensive Computing

- Computation-Intensive Computing
  - Example areas: MPI-based, high-performance computing, grids
  - Typically run on supercomputers (e.g., NCSA Blue Waters)
- Data-Intensive
  - Typically store data at datacenters
  - Use compute nodes nearby
  - Compute nodes run computation services
- In data-intensive computing, the focus shifts from computation to the data: CPU utilization no longer the most important resource metric, instead I/O is (disk and/or network)

IV. New Cloud Programming Paradigms

- Easy to write and run highly parallel programs in new cloud programming paradigms:
  - Google: MapReduce and Sawzall
  - Amazon: Elastic MapReduce service (pay-as-you-go)
  - Google (MapReduce)
    - Indexing: a chain of 24 MapReduce jobs
    - ~200K jobs processing 50PB/month (in 2006)
  - Yahoo! (Hadoop + Pig)
    - WebMap: a chain of 100 MapReduce jobs
    - 280 TB of data, 2500 nodes, 73 hours
  - Facebook (Hadoop + Hive)
    - ~300TB total, adding 2TB/day (in 2008)
    - 3K jobs processing 55TB/day
  - Similar numbers from other companies, e.g., Yieldex, eharmony.com, etc.
  - NoSQL: MySQL is an industry standard, but Cassandra is 2400 times faster!

CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

INTRODUCTION TO CLOUDS

Lecture F

ECONOMICS OF CLOUDS

Two Categories of Clouds

- Can be either a (i) public cloud, or (ii) private cloud
- Private clouds are accessible only to company employees
- Public clouds provide service to any paying customer

- You′ re starting a new service/company: should you use a public cloud or purchase your own private cloud?

Single site Cloud: to Outsource or Own?

- Medium-sized organization: wishes to run a service for *M* months
  - Service requires 128 servers (1024 cores) and 524 TB
  - Same as UIUC CCT cloud site
- Outsource (e.g., via AWS): monthly cost
  - S3 costs: $0.12 per GB month. EC2 costs: $0.10 per CPU hour (costs from 2009)
  - Storage = $ 0.12 X 524 X 1000 ~ $62 K
  - Total = Storage + CPUs = $62 K + $0.10 X 1024 X 24 X 30 ~ $136 K
- Own: monthly cost
  - Storage ~ $349 K / *M*
  - Total ~ $ 1555 K / *M* + 7.5 K (includes 1 sysadmin / 100 nodes)
    - using 0.45:0.4:0.15 split for hardware:power:network and 3 year lifetime of hardware

Single site Cloud: to Outsource or Own?

- Breakeven analysis: more preferable to own if:
  - $349 K / M < $62 K (storage)
  - $1555 K / M + 7.5 K < $136 K (overall)

  Breakeven points
  - M > 5.55 months (storage)
  - M > 12 months (overall)

- As a result
  - Startups use clouds a lot
  - Cloud providers benefit monetarily most from storage

Summary

- Clouds build on many previous generations of distributed systems
- Especially the timesharing and data processing industry of the 1960–70s.
- Need to identify unique aspects of a problem to classify it as a new cloud computing problem
  - Scale, On-demand access, data-intensive, new programming
- Otherwise, the solutions to your problem may already exist!