EC/CS 528: Cloud Computing

Overview of Cloud Computing

Instructor: Alan Liu

BU Department of Electrical & Computer Engineering

BOSTON UNIVERSITY

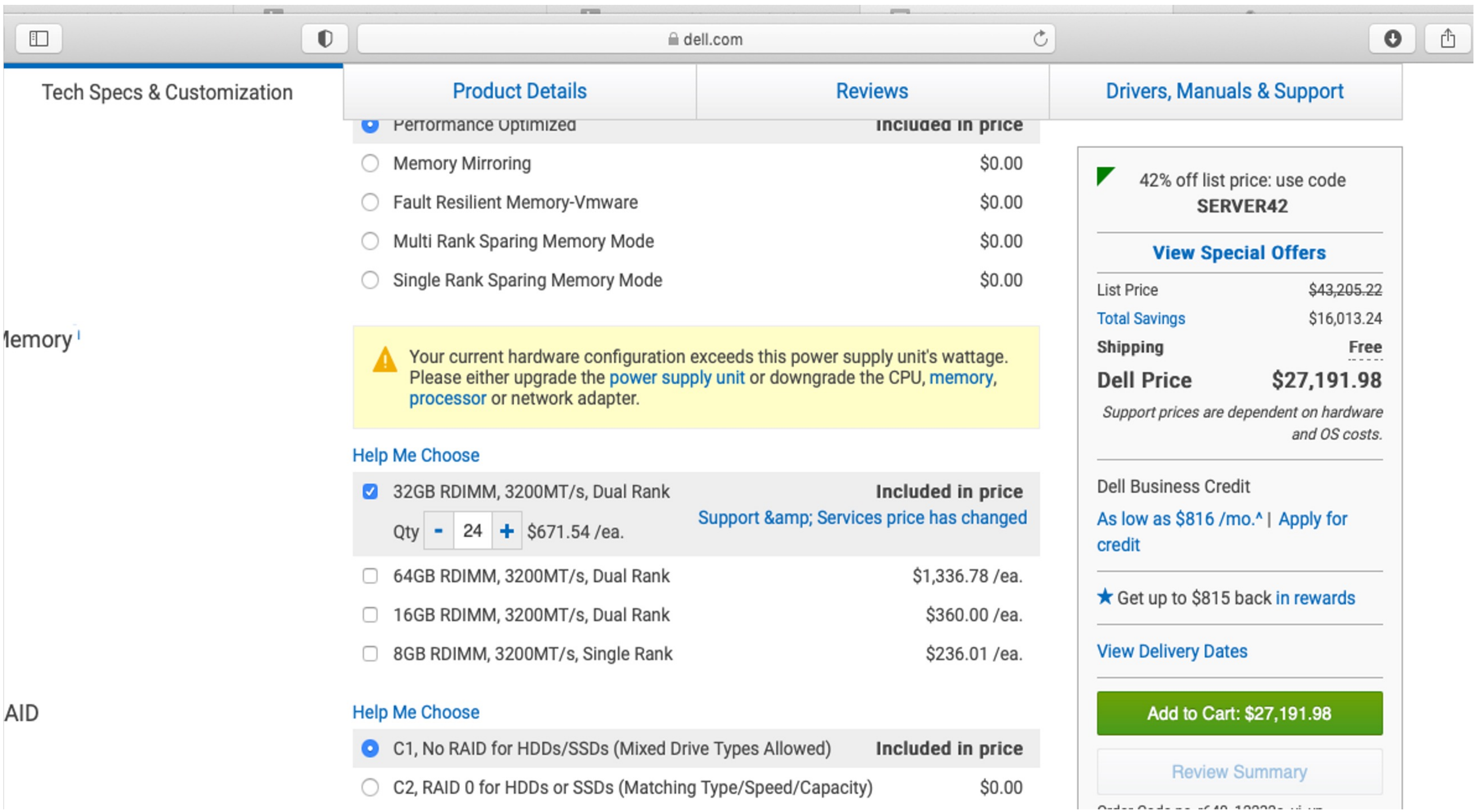
Announcements

- Piazza Discussion:
  - piazza.com/bu/fall2022/eccs528
  - Access code: cloudcomputing
- User account and skill survey:
  - Group and project matchings.
  - Prepare multiple project choices.
  - Q&A.
- New project update:
  - MLOps with Databricks in Public Clouds

How to start the project?

- Meet weekly with your mentor
  - Schedule a weekly meeting time
  - Record the meeting; mentors talk fast, being able to replay what they said can be super valuable
- Each person should say:
  - What have accomplished since last meeting?
  - What are you going to accomplish by next one?
  - Are they blocked?
- Don’t be blocked until weekly meeting:
  - Set up mechanism to ask quick questions to each other, and to mentor, e.g., slack
- Remember you are a team

Building the “cloud” from scratch - spec and buy



BOSTON UNIVERSITY

Then receive and assemble...

then you have to run it...

BOSTON UNIVERSITY

Issues?

[What do you think?]

- People and skills
  - N areas of expertise = O(N) people
- Scaling?

Why is Cloud Computing transformative?

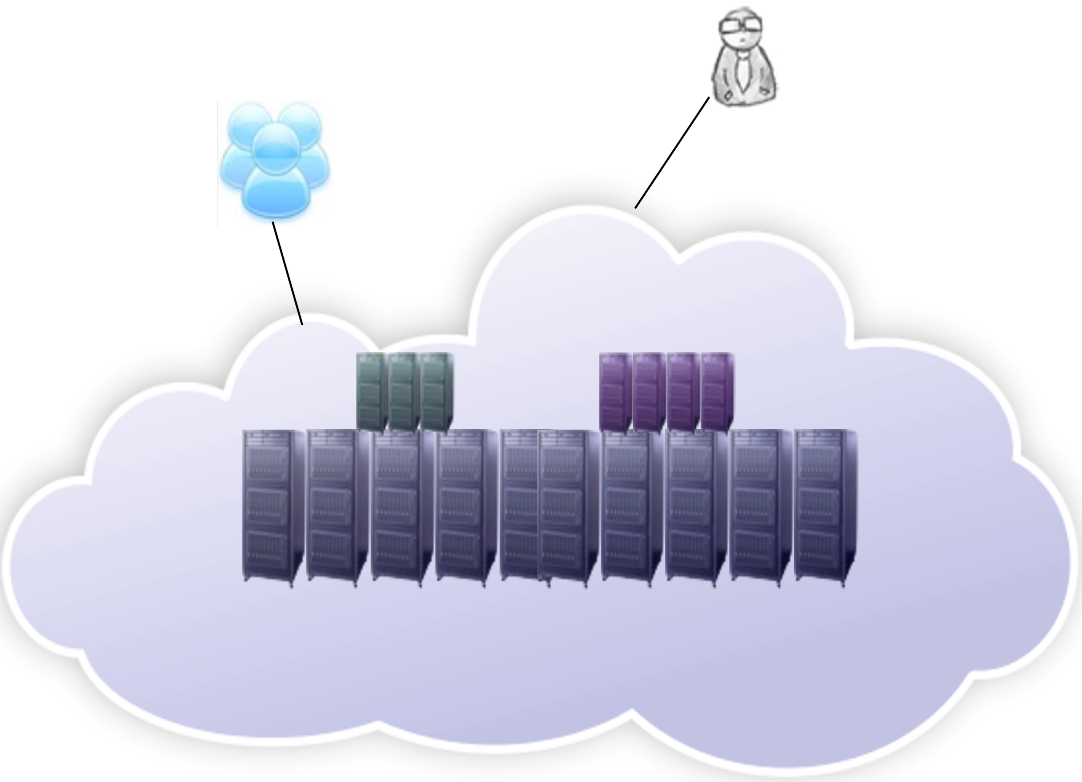
- Major change in computation is managed and used:
  - Economics of central utility: Price of computers, Operational efficiency, Location (e.g., cheap power, distribution), Co-location other customers, Utilization shared capacity, shared services (e.g., DR)
  - "As with the factory-owned generators that dominated electricity production a century ago, today's private IT plants will be supplanted by large-scale, centralized utilities." -- Nicholas Carr

- Availability of massive capacity on demand; elastically scale up and down:
  - Startups don't need to be acquired by Google or MS: a startup won't get money today to buy HW.
  - What happens when massive HPC becomes available to everyone?

- Gets rid of key impediments for developing & distributing SW
  - Avoids need for broad HCL, OS support, … many highly specialized software products…

BOSTON UNIVERSITY

Cloud in a nutshell



- On-demand access
- Economies of scale

All computing will move to the cloud

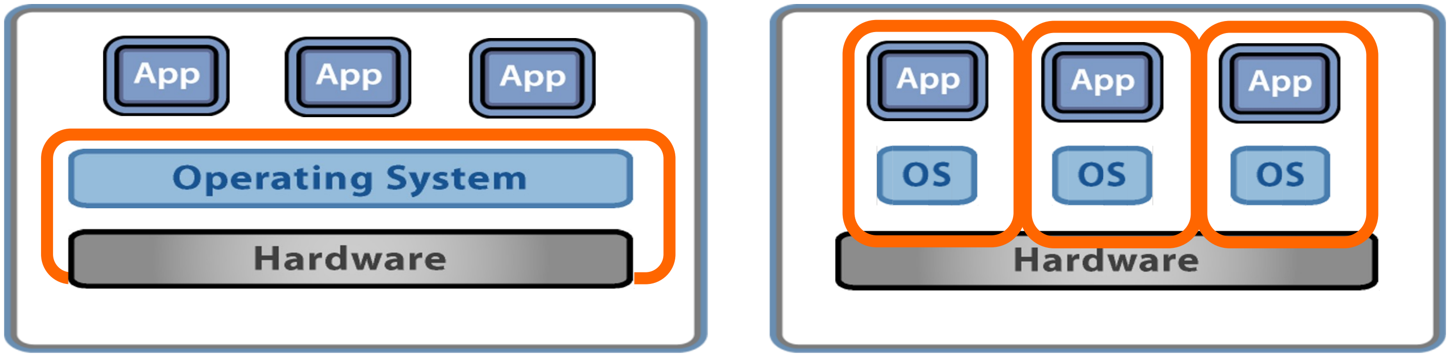
This is really nothing new…

Original vision of Utility/grid computing:

"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry."

When was this statement from?

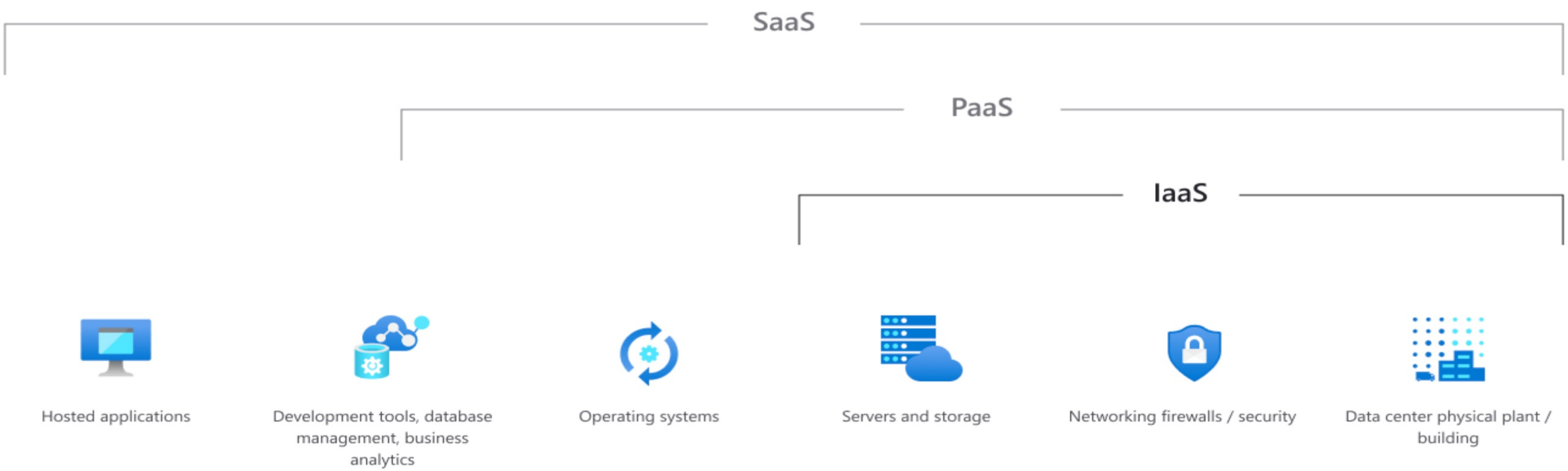
Why now?



BOSTON UNIVERSITY

Layers of Cloud

- Infrastructure as a Service (IaaS): AWS, Azure, OpenStack, MOC…
- Platform as a Service (PaaS): Salesforce's Force.com, Google App engine, AWS, MSFT Azure
- Software as a Service (SaaS): Hosted applications: Gmail, Facebook, Google docs, eBay



Motivation for using cloud

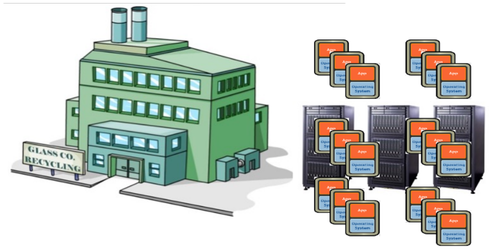
- Cloud is not inexpensive today
  - 2-20x more expensive than local
- Administrators do not come in fractional units; if you are small cheaper
- Offers elasticity: can deal with massive fluctuations on demand
- Offers huge variety of services:
  - cloud provider can afford to amortize cost over a huge number of customers

BOSTON UNIVERSITY

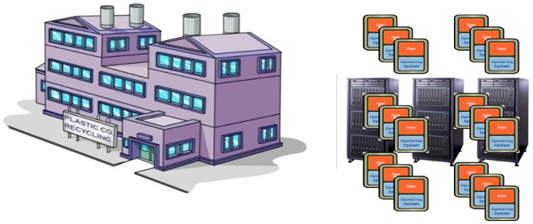
Examples

- Microsoft's Azure
- Amazon's AWS
- Google's Cloud Services

Remember this?



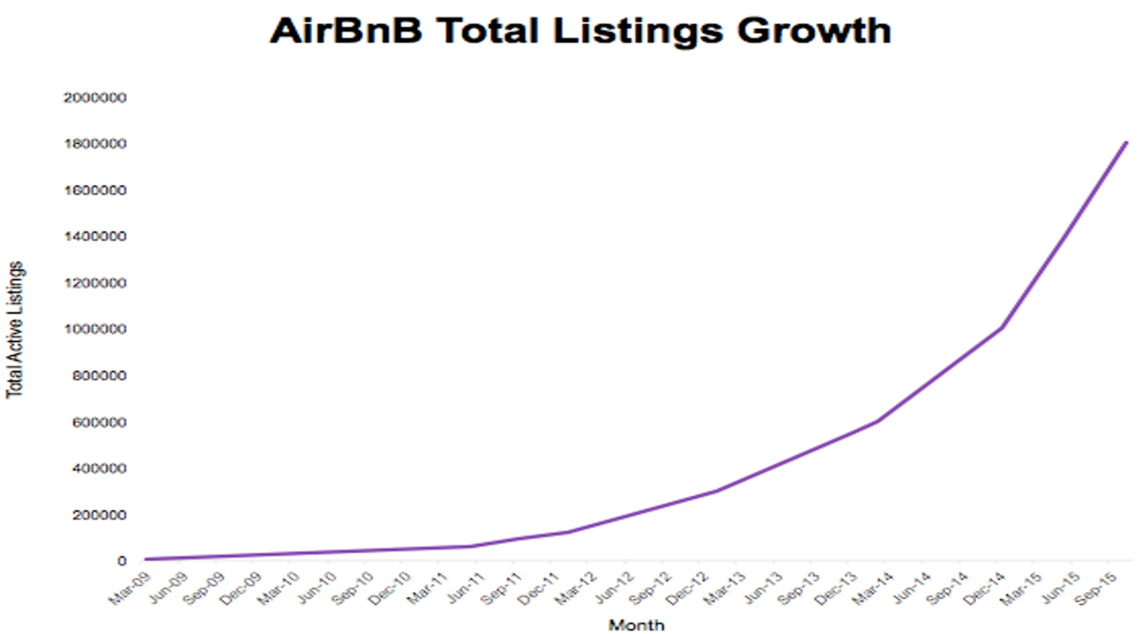
Host it R us.



Host 4 Less

AirBnB Example

- Success of market depends on network of renters and landlords;
  - —starts really small



BOSTON UNIVERSITY

AirBnB

https://aws.amazon.com/solutions/case-studies/airbnb/

- 2010 – 24 EC2 instances, 300 GB of data
- 2015 – 1000 EC2 instances, 50 TBytes data

- Grew up entirely on AWS, no data center, no capital purchases, no racking/stacking, no acquisition networking…
  – 5-person operations team
  – Piggyback on AWS for external network, availability zones

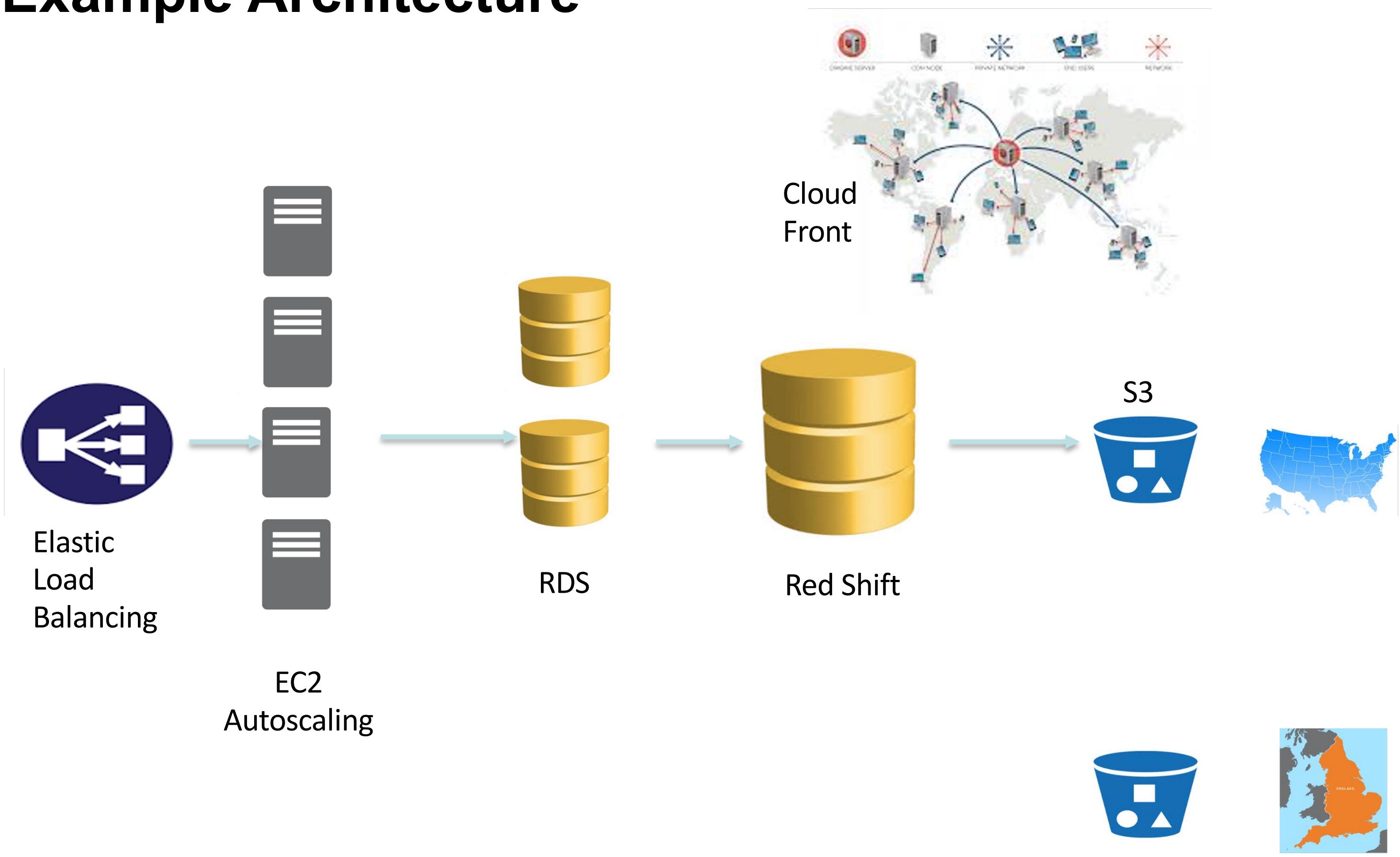
- Rapid growth easily accommodated.

Coursera

https://aws.amazon.com/solutions/case-studies/coursera/

- Massive on-line courses from Stanford, Duke…
- Went from 0 to 3.2 million users in first year
- Accessed from around the world
- Spikes common, e.g., 75% increase in load in 5 minutes

Example Architecture



BOSTON UNIVERSITY

Technology discussed

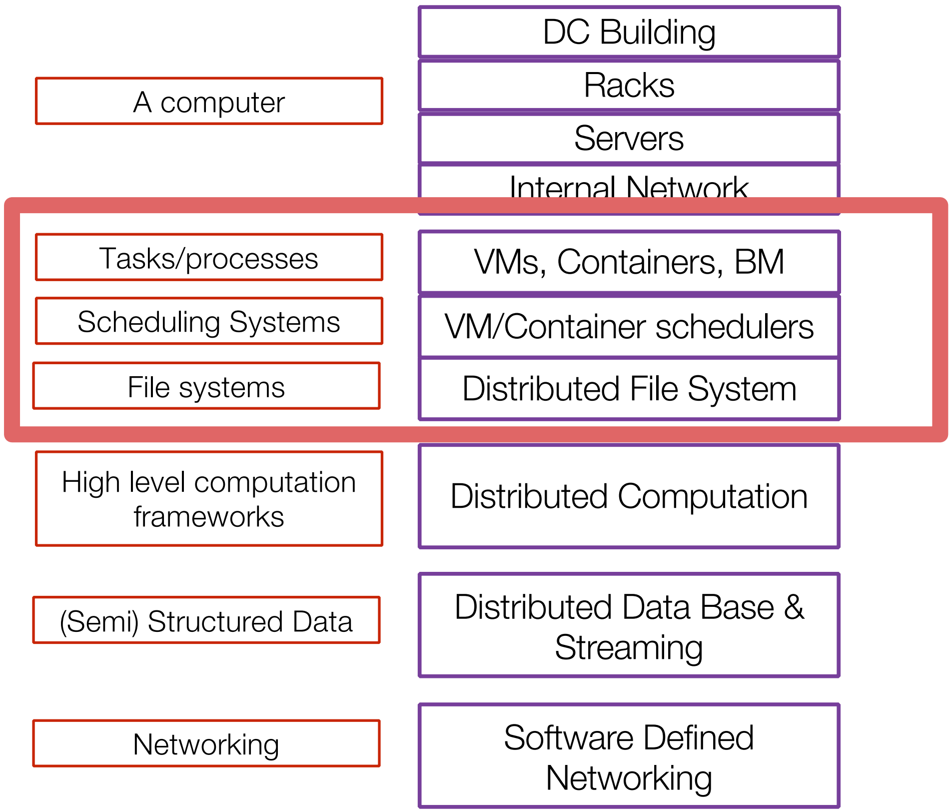
- EC2 & Elastic Load Balancing & EC2 Autoscaling – increase/decrease number of servers as needed.
- Relational Database Service (RDS) – managed service set up DB, patching, read-only replicas, across regions, backups automatically, snapshots
- Cloud Front – CDN, moved from 500 msec to 50msec average latency
- Red Shift – Data warehouse

Layers of data center

A computer	DC Building
	Racks
	Servers
	Internal Network
Tasks/processes	VMs, Containers, BM
Scheduling Systems	VM/Container schedulers
File systems	Distributed File System
High level computation frameworks	Distributed Computation
(Semi) Structured Data	Distributed Data Base & Streaming
Networking	Software Defined Networking

Hardware level:
How do you build cloud-scale systems?

Layers of data centers



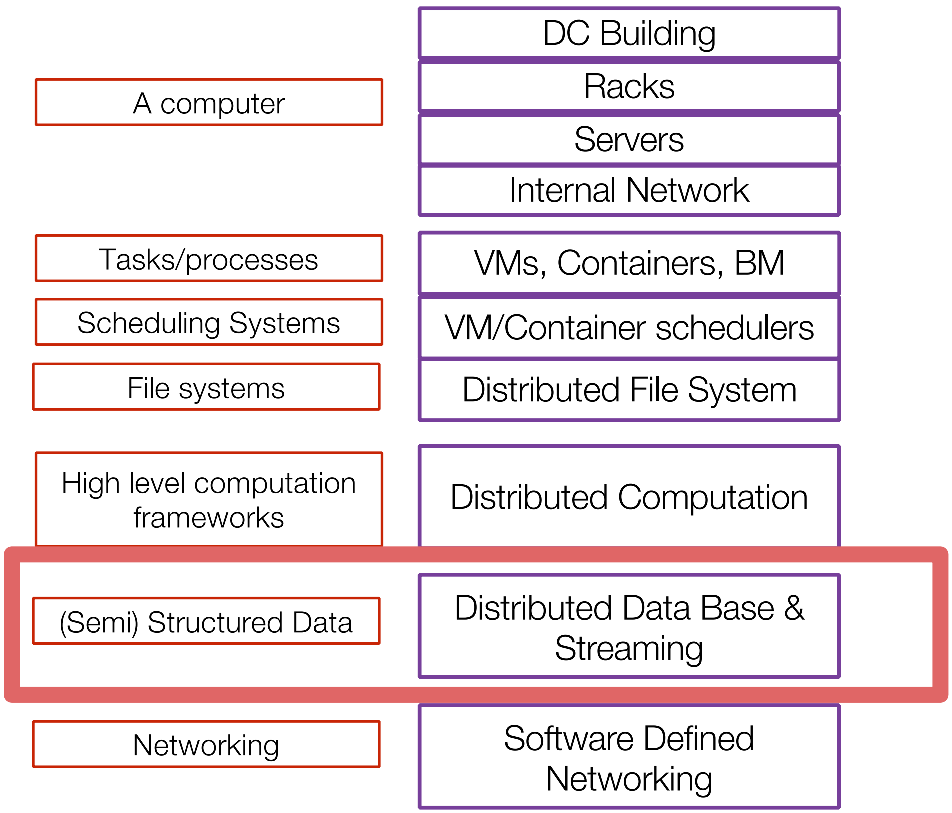
“Operating system”:
How do you manage and run cloud applications? What about file systems?

Layers

	DC Building
A computer	Racks
	Servers
	Internal Network
Tasks/processes	VMs, Containers, BM
Scheduling Systems	VM/Container schedulers
File systems	Distributed File System
High level computation frameworks	Distributed Computation
(Semi) Structured Data	Distributed Data Base & Streaming
Networking	Software Defined Networking

Frameworks:
How do you write a distributed application?

Layers



Above the file system:
How do you manage and work with structured data?

Layers

	DC Building
A computer	Racks
	Servers
	Internal Network
Tasks/processes	VMs, Containers, BM
Scheduling Systems	VM/Container schedulers
File systems	Distributed File System
High level computation frameworks	Distributed Computation
(Semi) Structured Data	Distributed Data Base & Streaming
Networking	Software Defined Networking

Networking:
How do the parts of a cloud-scale system talk to each other?

BOSTON UNIVERSITY

Top-down view of the course

Berkely view of cloud computing

	DC Building
A computer	Racks
	Servers
	Internal Network
Tasks/processes	VMs, Containers, etc
Scheduling Systems	VM/Container schedulers
File systems	Distributed File System
High level computation frameworks	Distributed Computation
(Semi) Structured Data	Distributed Data Base & Streaming
Networking	Software Defined Networking

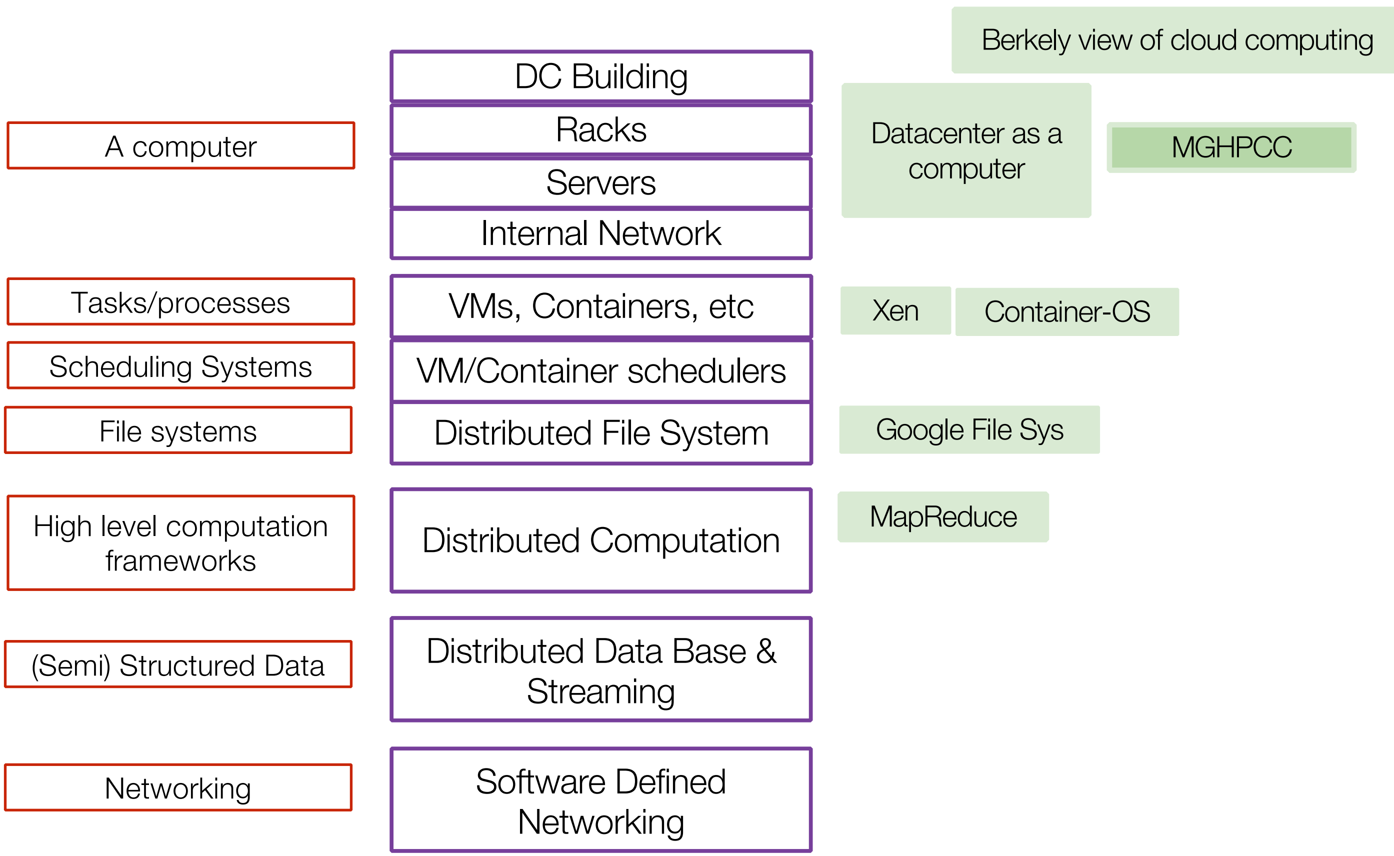
Datacenter as a computer

Top-down view of the course

		Berkely view of cloud computing
	DC Building	Datacenter as a computer
A computer	Racks	
	Servers	
	Internal Network	
Tasks/processes	VMs, Containers, etc	Xen; Container-OS
Scheduling Systems	VM/Container schedulers	
File systems	Distributed File System	
High level computation frameworks	Distributed Computation	
(Semi) Structured Data	Distributed Data Base & Streaming	
Networking	Software Defined Networking	

BOSTON UNIVERSITY

Top-down view of the course



Top-down view of the course

A computer	DC Building Racks Servers Internal Network	Berkely view of cloud computing Datacenter as a computer MGHPCC
Tasks/processes	VMs, Containers, etc	Xen Container-OS OpenLambda
Scheduling Systems	VM/Container schedulers	Borg Berkeley serverless
File systems	Distributed File System	Google File Sys
High level computation frameworks	Distributed Computation	MapReduce
(Semi) Structured Data	Distributed Data Base & Streaming	
Networking	Software Defined Networking	

BOSTON UNIVERSITY

Top-down view of the course

		Berkely view of cloud computing
A computer	DC Building Racks Servers Internal Network	Datacenter as a computer; MGHPCC
Tasks/processes	VMs, Containers, etc	Xen; Container-OS; OpenLambda; OpenShift
Scheduling Systems	VM/Container schedulers	Borg; Mesos; Berkeley serverless
File systems	Distributed File System	Google File Sys
High level computation frameworks	Distributed Computation	MapReduce
(Semi) Structured Data	Distributed Data Base & Streaming	
Networking	Software Defined Networking	Dapper

BOSTON UNIVERSITY

Top-down view of the course

	DC Building	Berkely view of cloud computing
A computer	Racks	Datacenter as a computer; MGHPCC
	Servers	
	Internal Network	
Tasks/processes	VMs, Containers, etc	Xen; Container-OS; OpenLambda; OpenShift
Scheduling Systems	VM/Container schedulers	Borg; Mesos; Berkeley serverless
File systems	Distributed File System	Google File Sys
High level computation frameworks	Distributed Computation	MapReduce
(Semi) Structured Data	Distributed Data Base & Streaming	BigTable; Dynamo
Networking	Software Defined Networking	Dapper

Top-down view of the course

		Berkely view of cloud computing
A computer	DC Building Racks Servers Internal Network	Datacenter as a computer MGHPCC
Tasks/processes	VMs, Containers, etc	Xen Container-OS OpenLambda OpenShift
Scheduling Systems	VM/Container schedulers	Borg Mesos Berkeley serverless
File systems	Distributed File System	Google File Sys
High level computation frameworks	Distributed Computation	MapReduce Spark Hive
(Semi) Structured Data	Distributed Data Base & Streaming	BigTable Dynamo
Networking	Software Defined Networking	Dapper

BOSTON UNIVERSITY

Top-down view of the course

		Berkely view of cloud computing
A computer	DC Building Racks Servers Internal Network	Datacenter as a computer; MGHPCC
Tasks/processes	VMs, Containers, etc	Xen; Container-OS; OpenLambda; OpenShift
Scheduling Systems	VM/Container schedulers	Borg; Mesos; Berkeley serverless
File systems	Distributed File System	Google File Sys; Flat Datacenter; CEPH
High level computation frameworks	Distributed Computation	MapReduce; Spark; Hive
(Semi) Structured Data	Distributed Data Base & Streaming	BigTable; Dynamo; Kafka
Networking	Software Defined Networking	Dapper

BOSTON UNIVERSITY

Top-down view of the course

		Berkely view of cloud computing
A computer	DC Building Racks Servers Internal Network	Datacenter as a computer; MGHPCC
Tasks/processes	VMs, Containers, etc	Xen; Container-OS; OpenLambda; OpenShift
Scheduling Systems	VM/Container schedulers	Borg; Mesos; Berkeley serverless
File systems	Distributed File System	Google File Sys; Flat Datacenter; CEPH
High level computation frameworks	Distributed Computation	MapReduce; Spark; Hive
(Semi) Structured Data	Distributed Data Base & Streaming	BigTable; Dynamo; Spanner; Kafka
Networking	Software Defined Networking	Memoryv @ Facebook; Dapper

BOSTON UNIVERSITY

Top-down view of the course

Topic	Layer	Examples
		Berkely view of cloud computing
A computer	DC Building Racks Servers Internal Network	Datacenter as a computer; MGHPCC
Tasks/processes	VMs, Containers, etc	Xen; Container-OS; OpenLambda; OpenShift; Open Stack
Scheduling Systems	VM/Container schedulers	Borg; Mesos; Berkeley serverless
File systems	Distributed File System	Google File Sys; Flat Datacenter; CEPH
High level computation frameworks	Distributed Computation	MapReduce; Spark; Tensorflow; Hive
(Semi) Structured Data	Distributed Data Base & Streaming	BigTable; Dynamo; Spanner; Kafka
Networking	Software Defined Networking	Memoryv @ Facebook; Dapper

BOSTON UNIVERSITY

Top-down view of the course

		Berkely view of cloud computing
A computer	DC Building Racks Servers Internal Network	Datacenter as a computer MGHPCC
Tasks/processes	VMs, Containers, etc	Xen Container-OS OpenLambda OpenShift Open Stack
Scheduling Systems	VM/Container schedulers	Borg Mesos Berkeley serverless
File systems	Distributed File System	Google File Sys Flat Datacenter CEPH
High level computation frameworks	Distributed Computation	MapReduce Spark Tensorflow Hive
(Semi) Structured Data	Distributed Data Base & Streaming	BigTable Dynamo Spanner Kafka
Networking	Software Defined Networking	Jupiter rising Memoryv @ Facebook Dapper

Transformation

- Transformed how SW is developed:
  - continuous deployment; changes tested with real customers
  - example Facebook failure last year
  - massive advantage over waterfall
- It's all about distributed applications
  - change from pets to cattle
  - care about 99th% tail latency
  - stateless servers
  - huge set of higher level services: Containers as a Service, Functions as a Service, Analytics as a Service...

BOSTON UNIVERSITY

The challenges

- Monoculture from security perspective
- Emerging oligopoly:
  - Lack of competition limits sources innovation
  - Price is outrageously expensive
- Effort to lock in users: e.g., networking
- Big brother…, or perhaps just Giants whose incentives are not aligned with privacy and marketplace; Consider Facebook

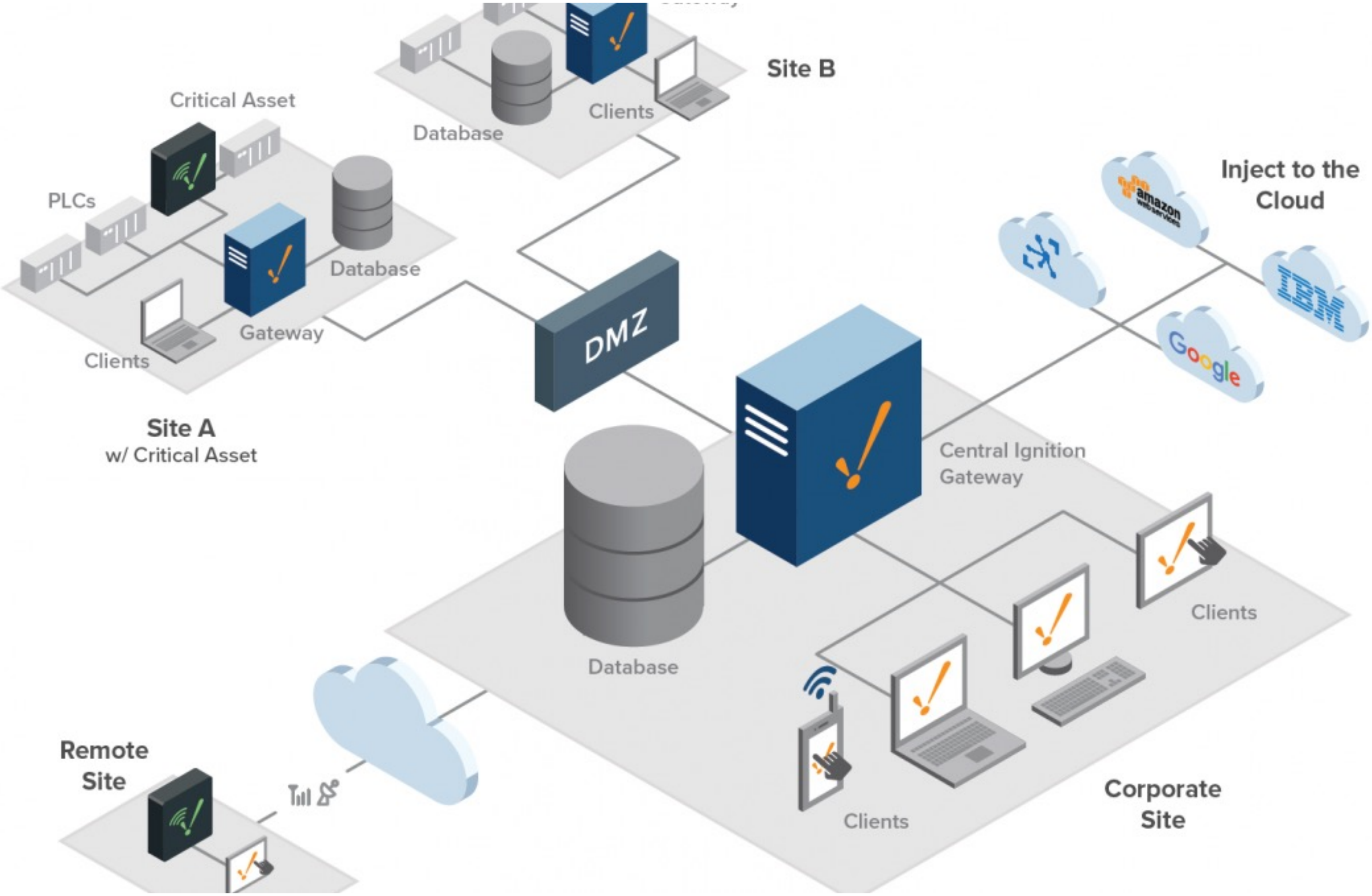
BOSTON UNIVERSITY

The Datacenter as a Computer

An Introduction to the Design of Warehouse-Scale Machines – 2nd Edition

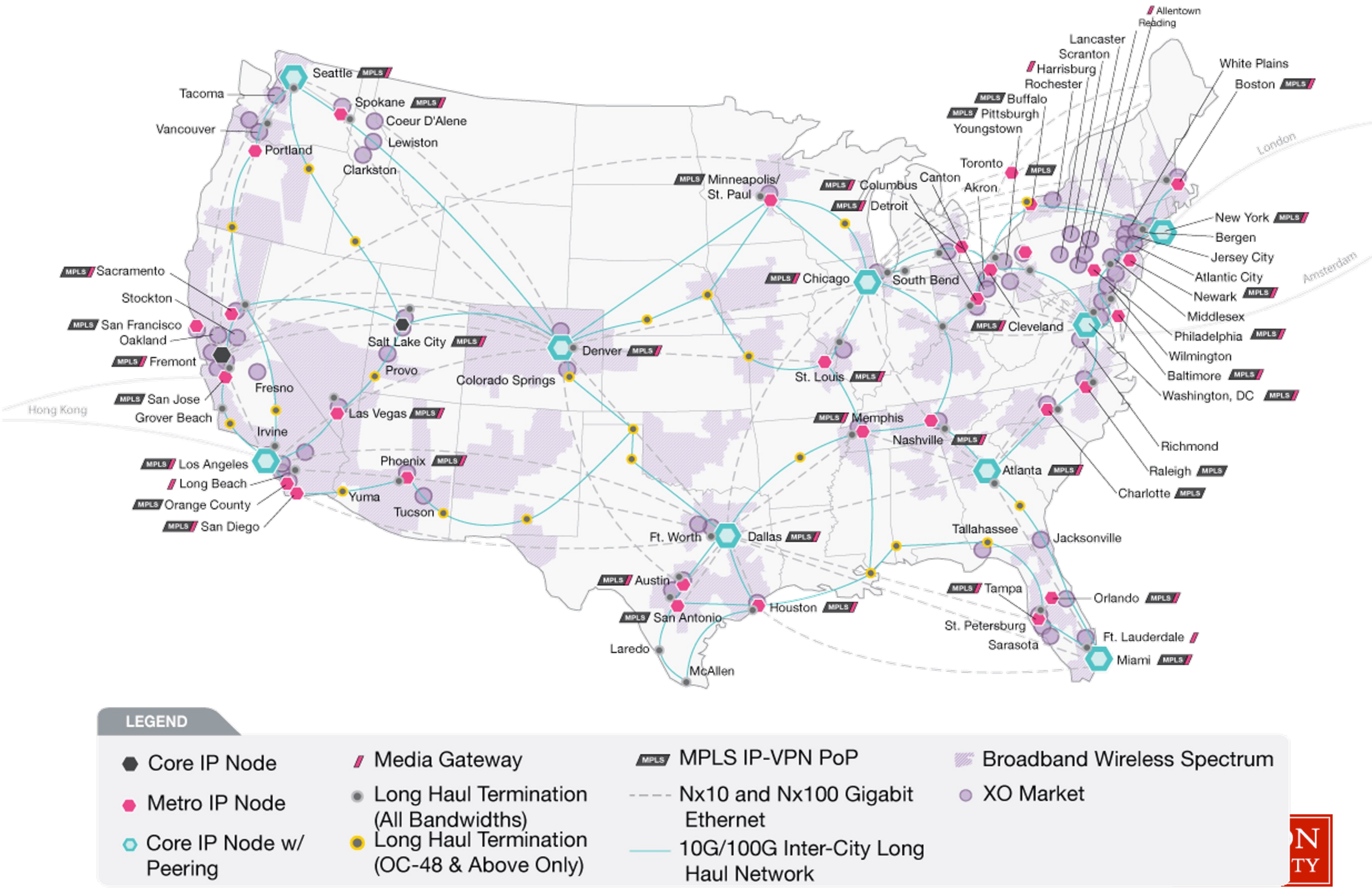
Luiz André Barroso, Jimmy Clidaras, Urs Hölzle

Enterprise IT

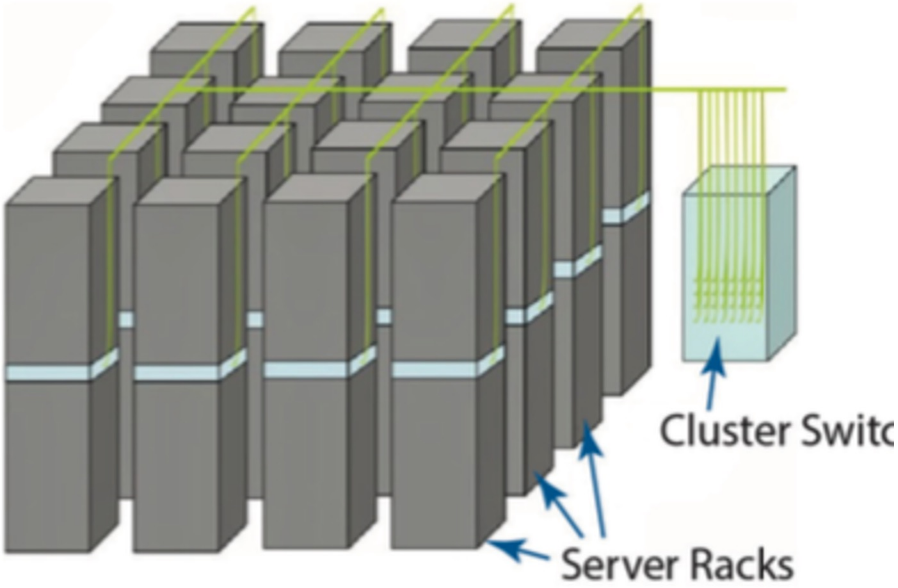


BOSTON UNIVERSITY

Traditional “wide-area” networks



Elements of data center

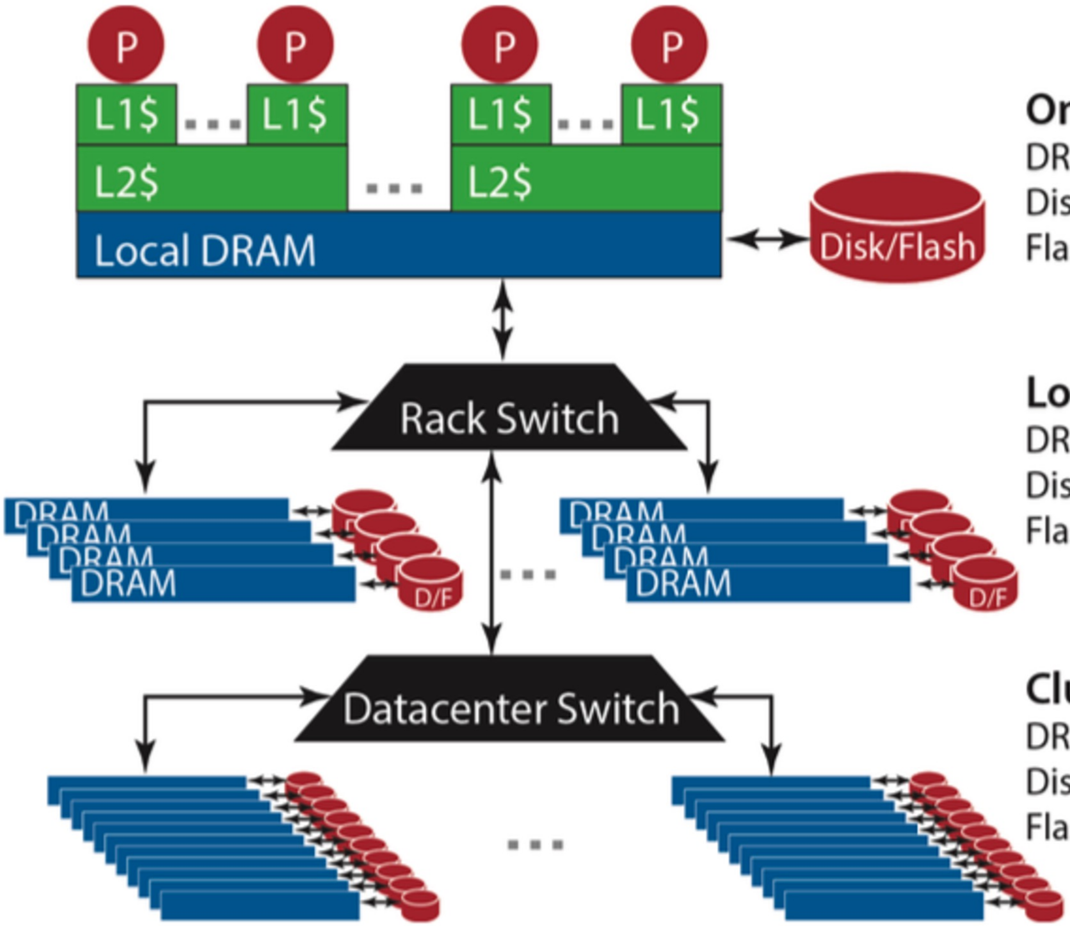


BOSTON UNIVERSITY

Storage assumptions

- Storage distributed across all machines
- Software like GFS distributes, versus NAS appliance
  - Redundancy even if rack level failure
  - Multiplex server resources (NIC/enclosure/power)
  - Exploits cheap desktop disks
- Typically network oversubscribed
  - E.g., 32 * 40Gig links nodes, 4 *100Gig Links up

Storage Hierarchy

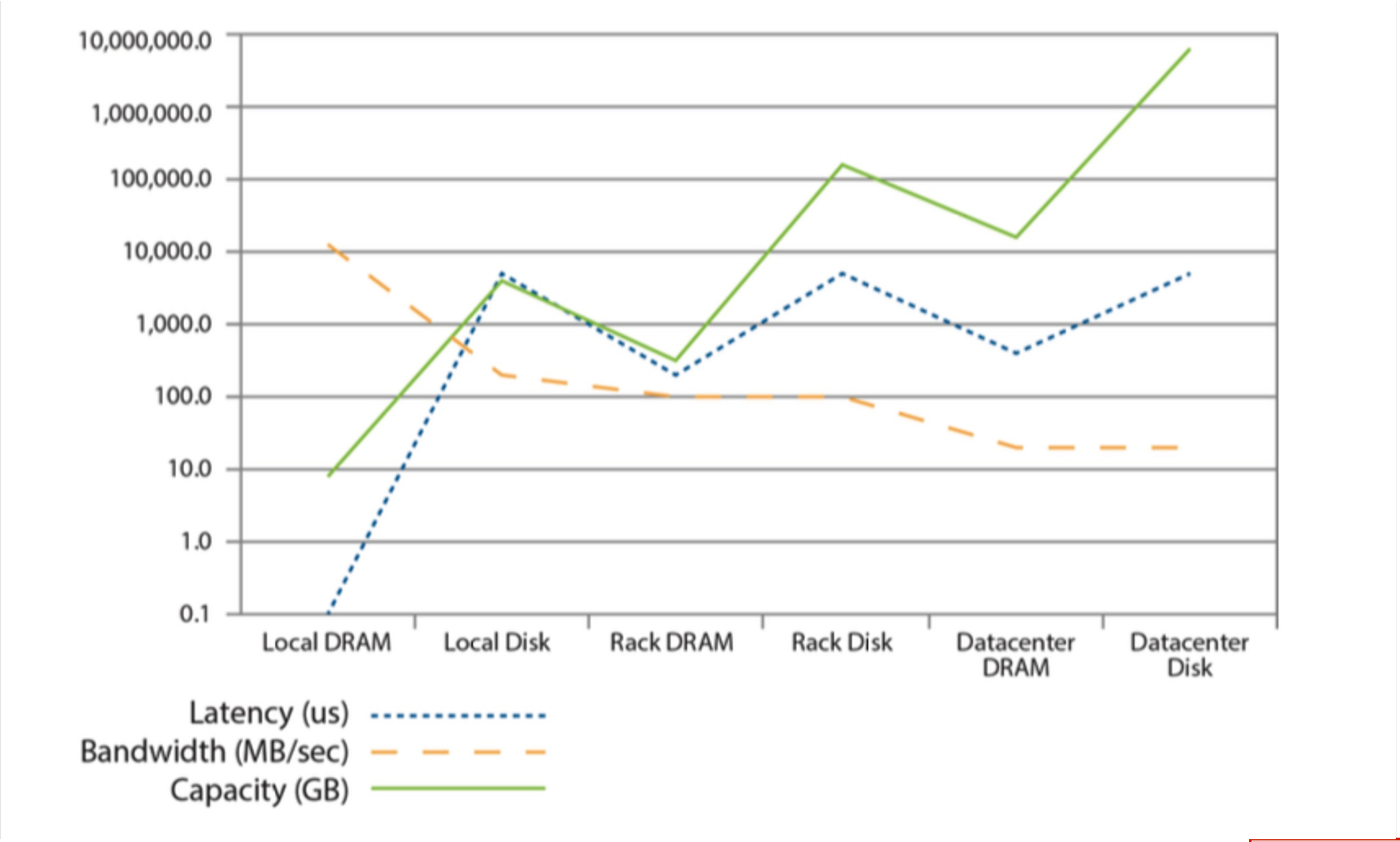


One Server
DRAM: 16 GB, 100 ns, 20 GB/s
Disk: 2T B, 10 ms, 200 MB/s
Flash: 128 GB, 100 us, 1 GB/s

Local Rack (80 servers)
DRAM: 1 TB, 300 us, 100 MB/s
Disk: 160 TB, 11 ms, 100 MB/s
Flash: 20 TB, 400 us, 100 MB/s

Cluster (30 racks)
DRAM: 30 TB, 500 us, 10 MB/s
Disk: 4.80 PB, 12 ms, 10 MB/s
Flash: 600 TB, 600 us, 10 MB/s

Latency, bandwidth & capacity



BOSTON UNIVERSITY

Inside a data center

Self-introduction

Q&A