EC/CS 528: Cloud Computing

Overview of Cloud Computing

Instructor: Alan Liu

BU Department of Electrical & Computer Engineering

BOSTON UNIVERSITY

Quiz

- Check Piazza for the link.
- 5-min.
- Nice stickers!!

BOSTON UNIVERSITY

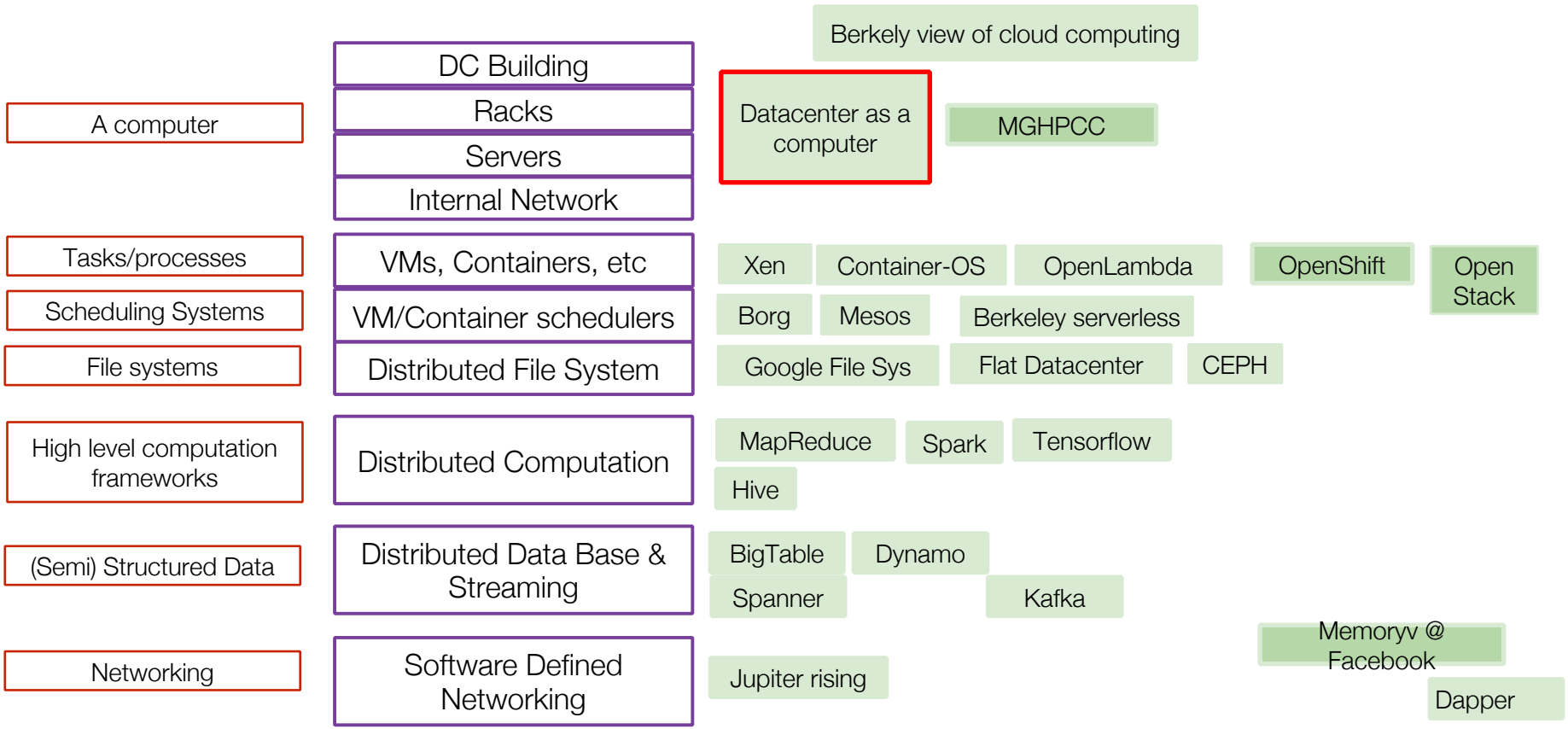
Project Description

- Work with your teammates and mentor(s).
- Due in a week.
- Using Markdown language (templated provided).

https://github.com/zaoxing/cloudcourse22/blob/main/Project_Description_Template.md

BOSTON UNIVERSITY

Top down view of the course



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

BOSTON UNIVERSITY

What is a warehouse scale computer (WSC)

- Some workloads require so much computing
  - better fit for a massive computing infrastructure rather than client-side computing
    - search, language translation, etc...
    - any more?

Machine learning workloads

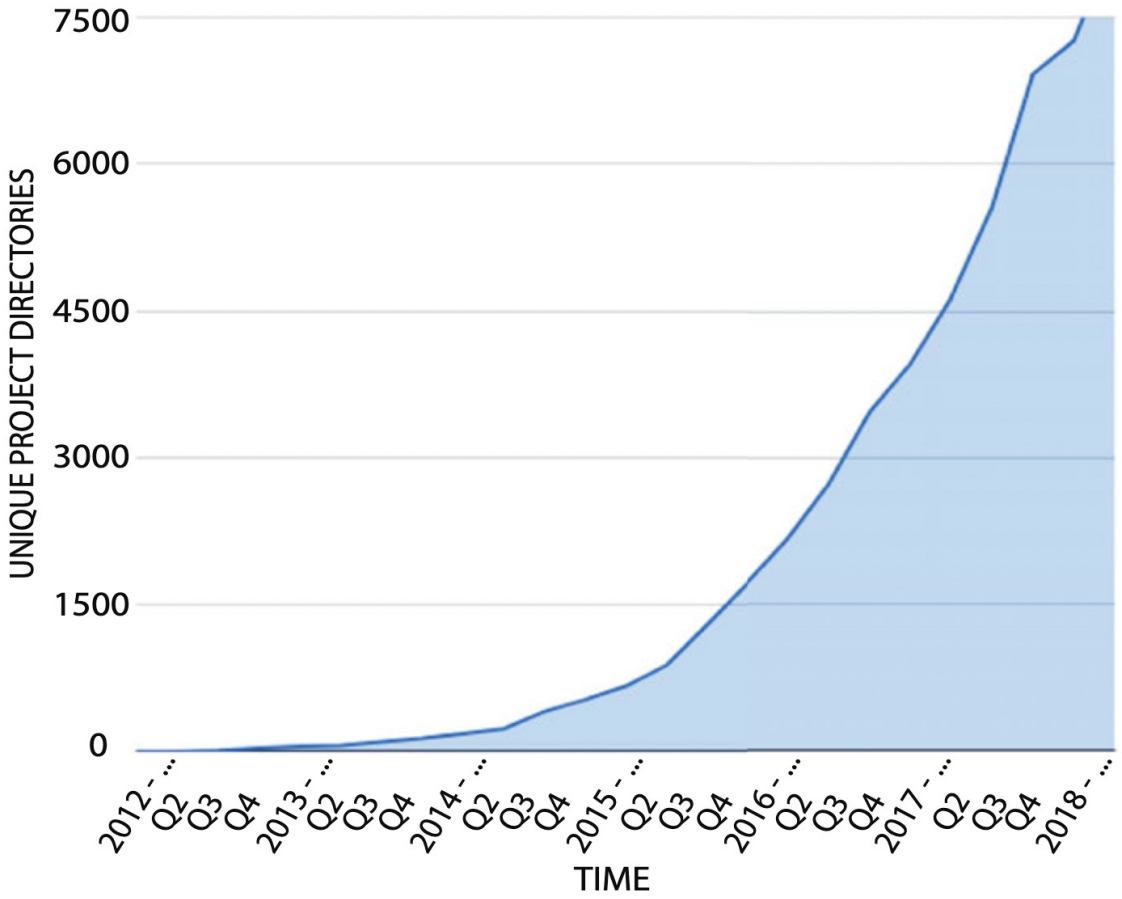


Figure 2.6: Growth of machine learning at Google.

What are some of the key reasons for the rise of WSC?

BOSTON UNIVERSITY

Consumers move to mobile devices and the rise of Internet services

- trend towards server-side computing on consumer apps:
  - increased availability of high-speed connectivity
  - widespread use of internet services
  - move from PCs => mobile devices + internet services
    - services such as email, photo/video storage, office apps that resided on the client in the past moved to web apps

BOSTON UNIVERSITY

Enterprises move to DC

- SaaS allows faster app development:
  - Simpler to make changes
  - Don't have to deal with massive diversity client HW
- Statistical multiplexing
- De-duplication/sharing attachments
- Servers administered by professional staff
  - Reliability & Security
- Note – this is the case for everyone through IaaS clouds

BOSTON UNIVERSITY

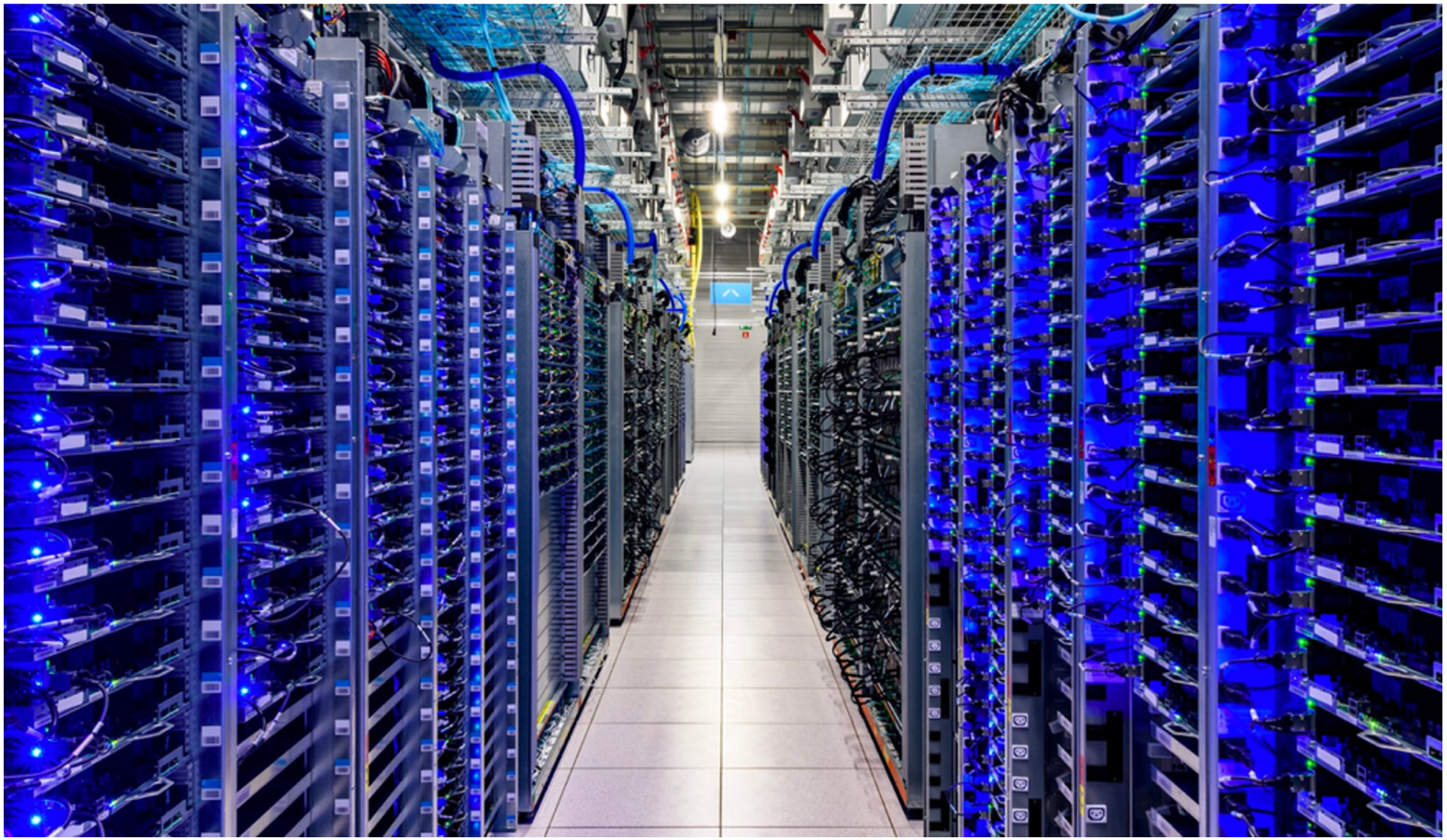
What are the differences between WSC and traditional data centers?

BOSTON UNIVERSITY

Enterprise “computer room”

BOSTON UNIVERSITY

Google Data Center



BOSTON UNIVERSITY

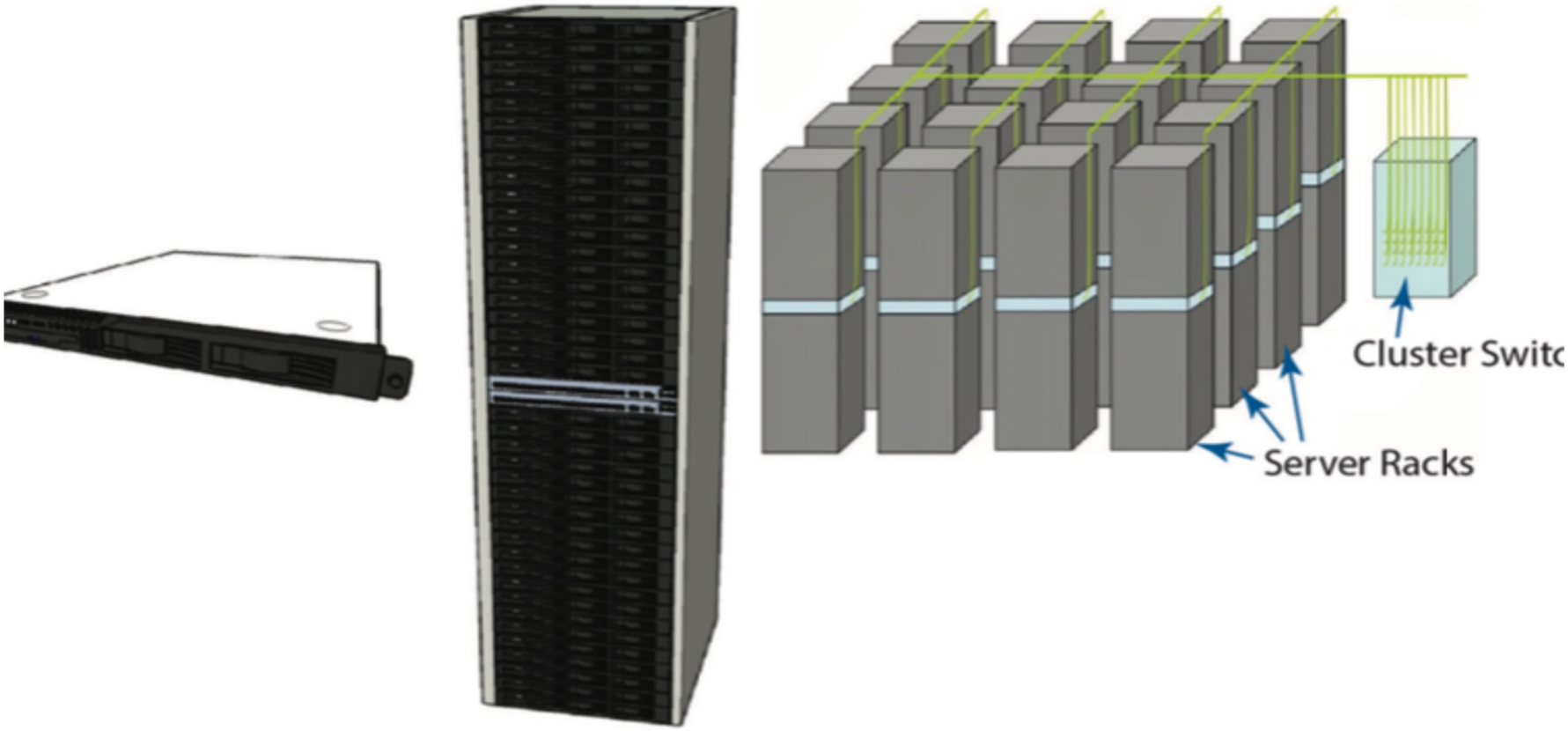
Traditional DC vs. WSC

- Large number of apps, each on dedicated isolated HW
- Multiple entities
- Each entity with own OS/SW/Middleware/control plane
- Reliable HW simplifies SW development

- Single organization
- Homogenous HW and system SW
- Common system management
- Often much SW in-house
- Small number of very large applications
- Applications need to be tolerant of failures
- Reliable platforms simplify SW development

BOSTON UNIVERSITY

Elements of data center



BOSTON UNIVERSITY

Elements of a data center

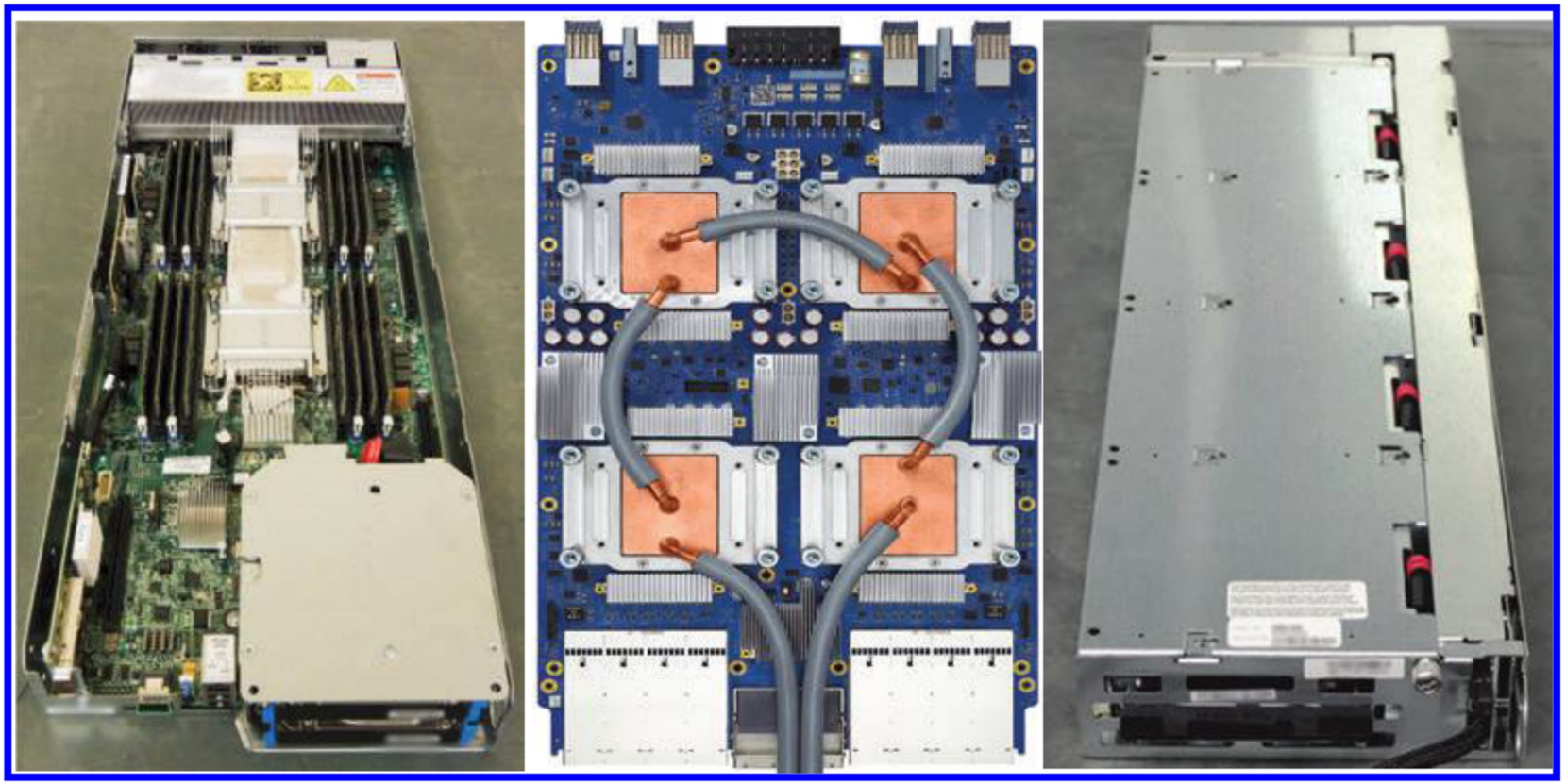


Figure 1.1: Example hardware building blocks for WSCs. Left to right: (a) a server board, (b) an accelerator board (Google's Tensor Processing Unit [TPU]), and (c) a disk tray.

BOSTON UNIVERSITY

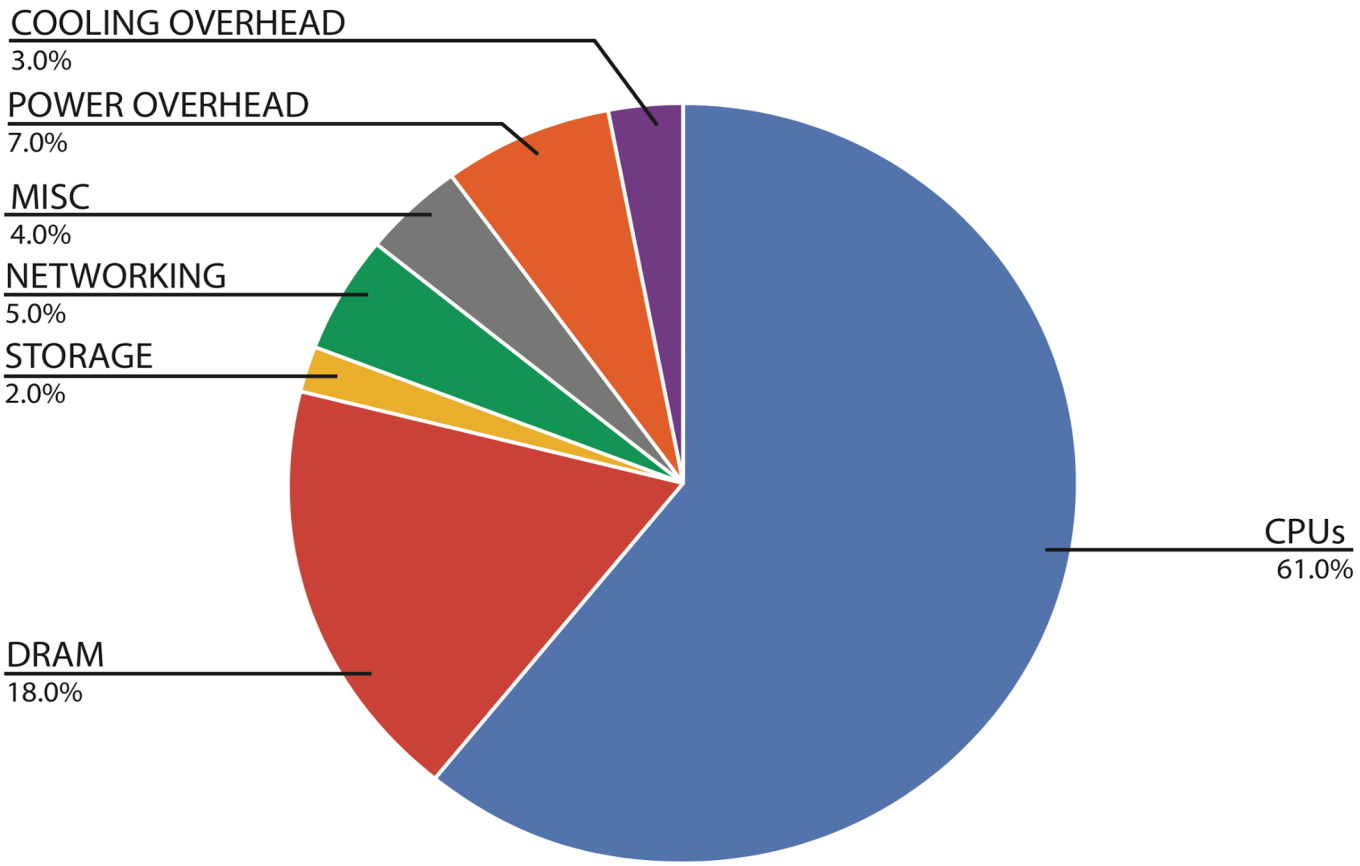
Building and Infrastructure

Power distribution

Datacenter cooling

BOSTON UNIVERSITY

Power Usage

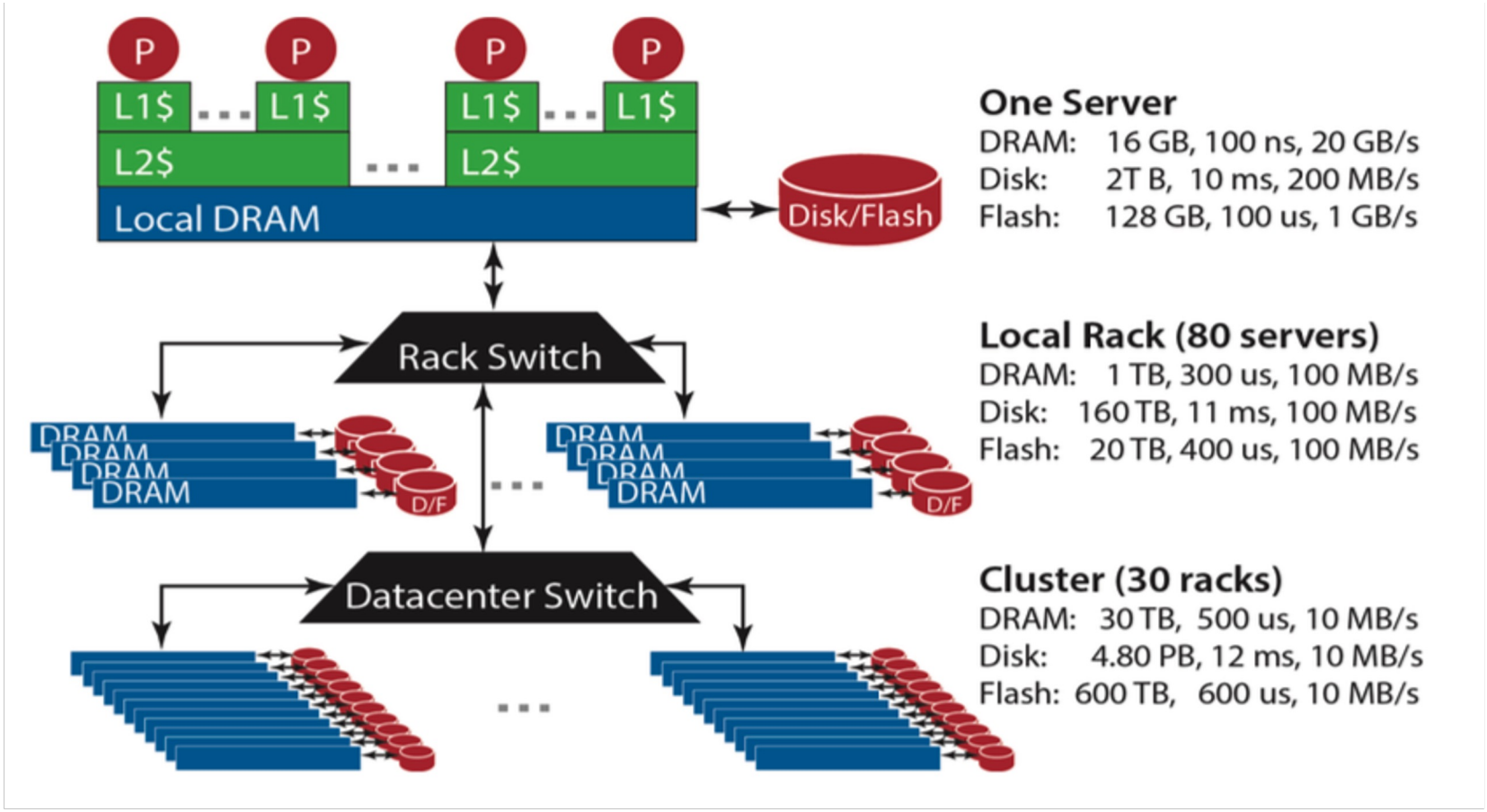


BOSTON UNIVERSITY

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., 40x40Gig links nodes, 4x100Gig Links up

Storage Hierarchy (2013)



Storage Hierarchy (2018)

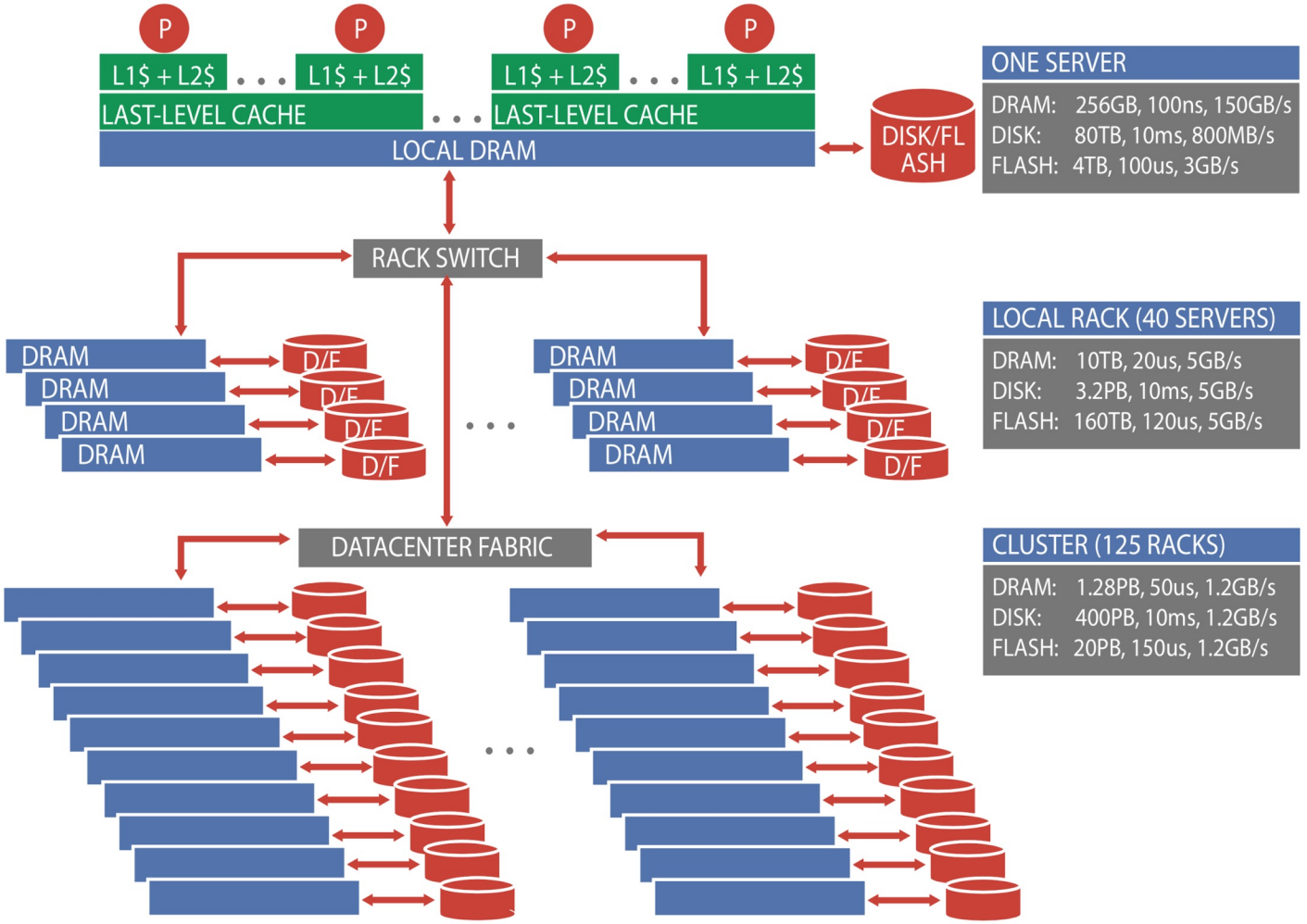
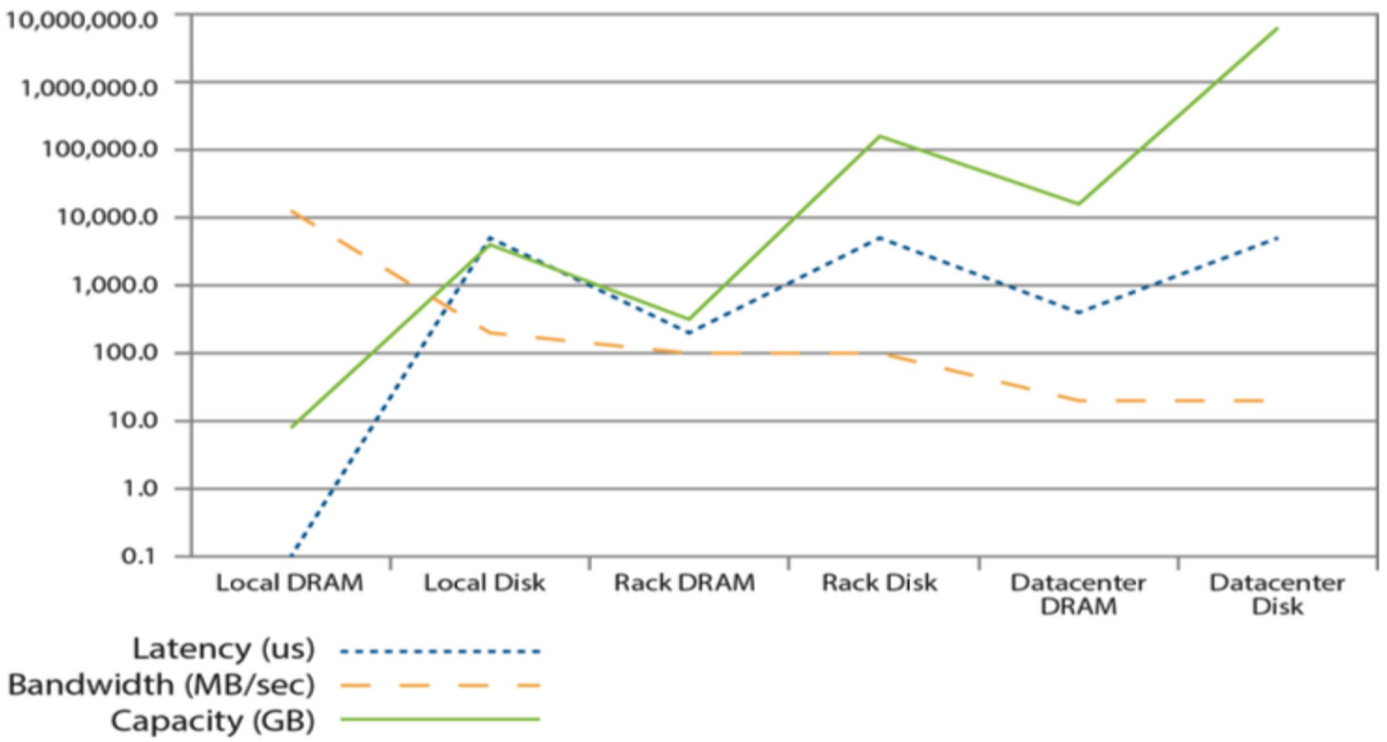


Figure 3.15: Storage hierarchy of a WSC.

Latency, bandwidth & capacity (2013)



Latency, bandwidth & capacity (2018)

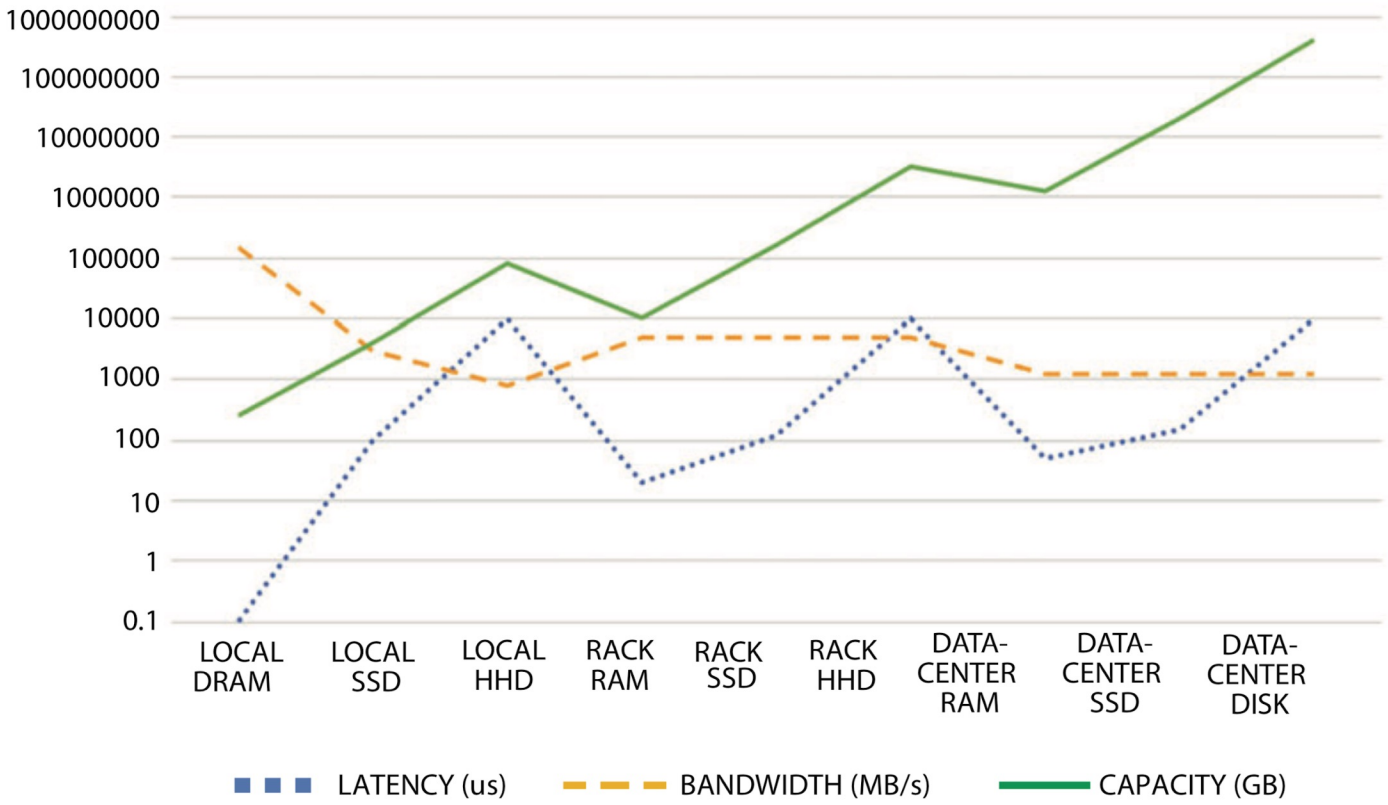


Figure 3.16: Latency, bandwidth, and capacity of WSC storage hierarchy levels.

Latency Numbers

Table 2.3: Latency numbers that every WSC engineer should know. (Updated version of table from [Dea09].)

Operation	Time
L1 cache reference	1.5 ns
L2 cache reference	5 ns
Branch misprediction	6 ns
Uncontended mutex lock/unlock	20 ns
L3 cache reference	25 ns
Main memory reference	100 ns
Decompress 1 KB with Snappy [Sna]	500 ns
"Far memory"/Fast NVM reference	1,000 ns (1us)
Compress 1 KB with Snappy [Sna]	2,000 ns (2us)
Read 1 MB sequentially from memory	12,000 ns (12 us)
SSD Random Read	100,000 ns (100 us)
Read 1 MB bytes sequentially from SSD	500,000 ns (500 us)
Read 1 MB sequentially from 10Gbps network	1,000,000 ns (1 ms)
Read 1 MB sequentially from disk	10,000,000 ns (10 ms)
Disk seek	10,000,000 ns (10 ms)
Send packet California→Netherlands→California	150,000,000 ns (150 ms)

Local vs geographic networking

98PPJ

US $2,180.00

Description
DELL 98PPJ Networking Z9100-on 32 X 100gbe + 2 X Sfp+
Stock.

Brand: DELL

Condition: Refurbished

Quantity: 1

local: 32x 100gbit = $2K

remote:
fiber (x24) $10-15K/mile
bandwidth: 64x 100gbit per fiber

BOSTON UNIVERSITY

This is interesting, but… what's the implication?

BOSTON UNIVERSITY

What does this mean…

- Rack locality super important in this world
- Important to understand the tradeoffs in designing distributed application; but these tradeoffs are changing all the time:
- Sanity check: Things change
  - Was 2013 a long time ago?
    - See VL2 and Flat Data Center Storage papers with designs for changing assumptions
    - It changed with Flash dropping in price…
  - Oops, now we have persistent memory...
  - Oops it is changing again with 100Gig cards
- At the end; most programmers can't design to these tradeoffs; programming models needed
- We better isolate application developer from detailed tradeoffs...

How are WSC applications different?

Implications of applications

- Lots of parallelism – request and data
  - Often trivially parallelizable
- Workload churn
  - Lots of small changes on daily basis
    - core Google search re-implemented effectively every 3 years
  - Lots of new applications all the time
  - Have ability to innovate HW, SW will adapt rapidly

Implications on SW development

- Agile development
  - Continuously refine often with real users
- Design of services using Service Oriented Architecture: app == service composed of multiple services
  - Each maintained by its own team
  - Each scaled independently based on demand
  - Small two-pizza sized team
- Continuous Integration versus six month release
- Move to Dev/Ops

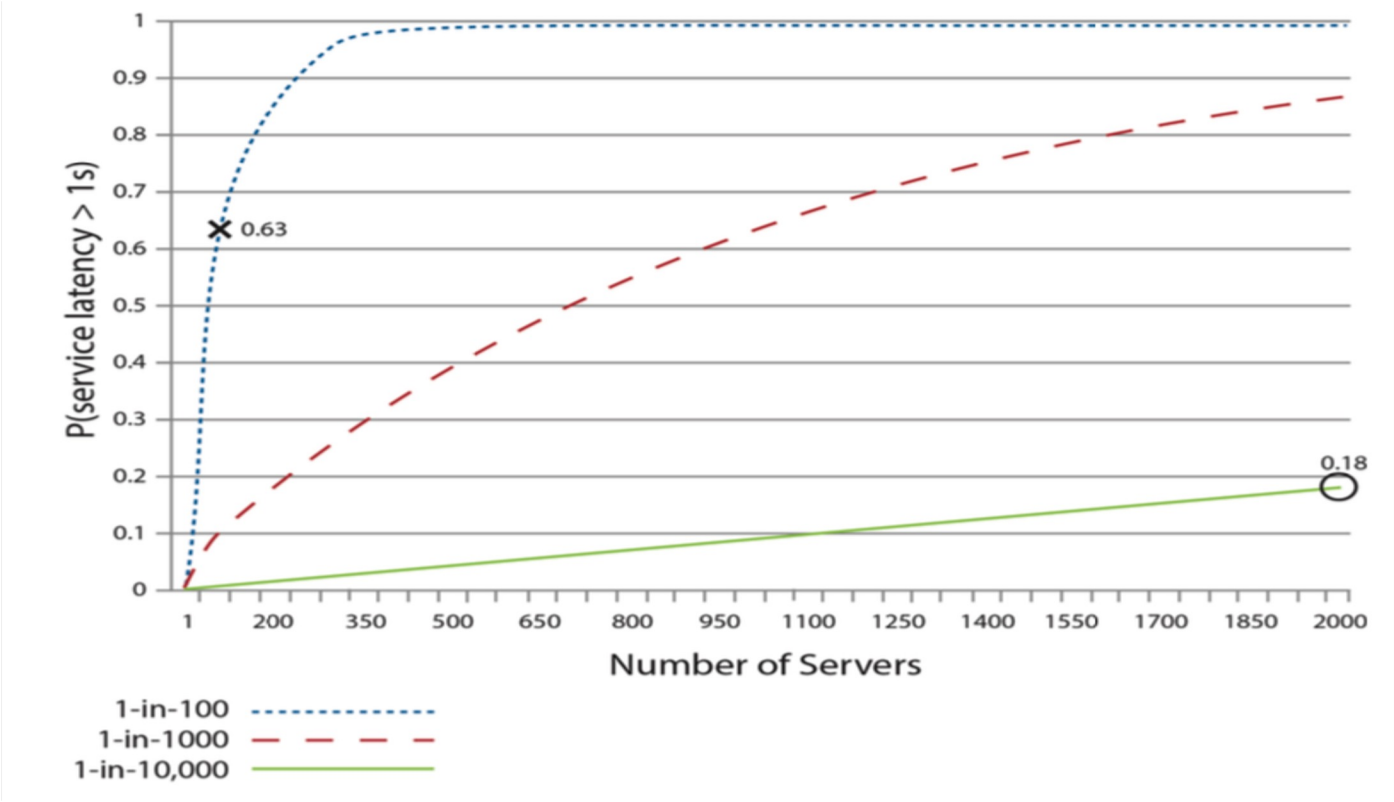
What programming concepts are important/interesting?

BOSTON UNIVERSITY

Key programming concepts

- Eventual Consistency
- Centralized control
- Re-use of sophisticated platforms/services:
  - GFS, MapReduce, Dynamo…
- Automation (e.g., puppet), monitoring everything...
- Tail latency critical
  - Even low probability delays have disproportionate impact
  - Requires redundant execution
  - Major change in how we optimize our systems; focus on worst case rather than average

Probability of 1 s response



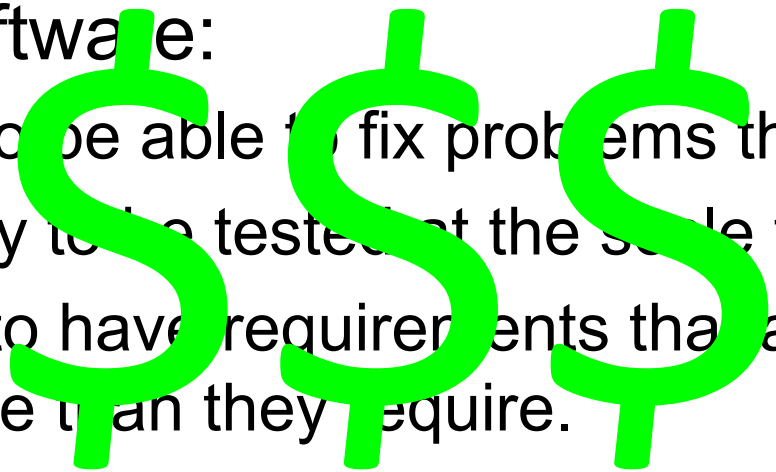
Build vs. Buy?

Build vs. Buy

- Big provider's allergic to using proprietary third party software:
  - Need to be able to fix problems themselves
  - Unlikely to be tested at the scale they need
  - Likely to have requirements that are more general purpose than they require.
- This is much broader industry trend

Build vs. Buy

- Big provider's allergic to using proprietary third party software:
  - Need to be able to fix problems themselves
  - Unlikely to be tested at the scale they need
  - Likely to have requirements that are more general purpose than they require.
- This is much broader industry trend



What did you get out of Chapter 3?

Chapter 3: Hardware building blocks

- Smorgasbord of concepts…
  - Large-scale SMP versus commodity
    - If your program is not distributed, little value
  - Brawny versus Wimpy nodes (note James Hamilton)
- Technologies like GFS, BigTable, Dynamo:
  - We will discuss...
- Accelerators, GPUs, TPUs,
- Argument for commodity networks
- Discussion of SDN, bisectional BW, VM migration

BOSTON UNIVERSITY

Small SMPs vs. Large SMPs

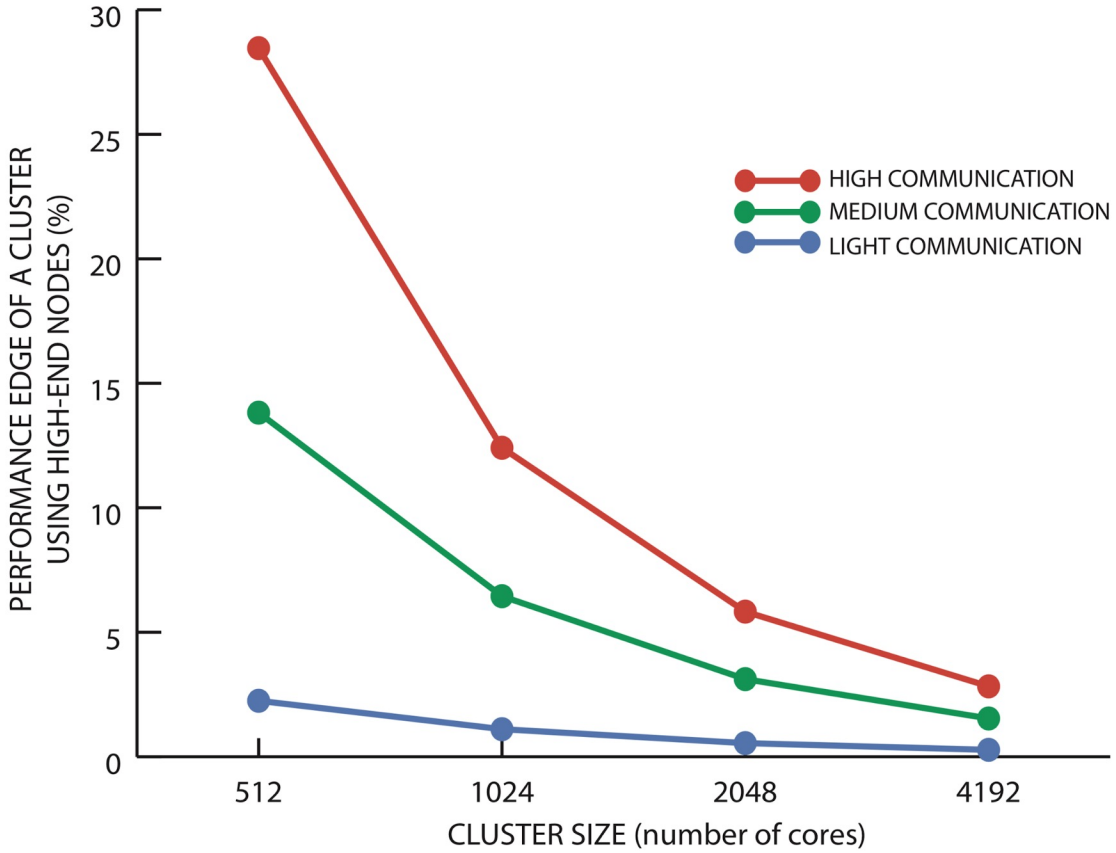
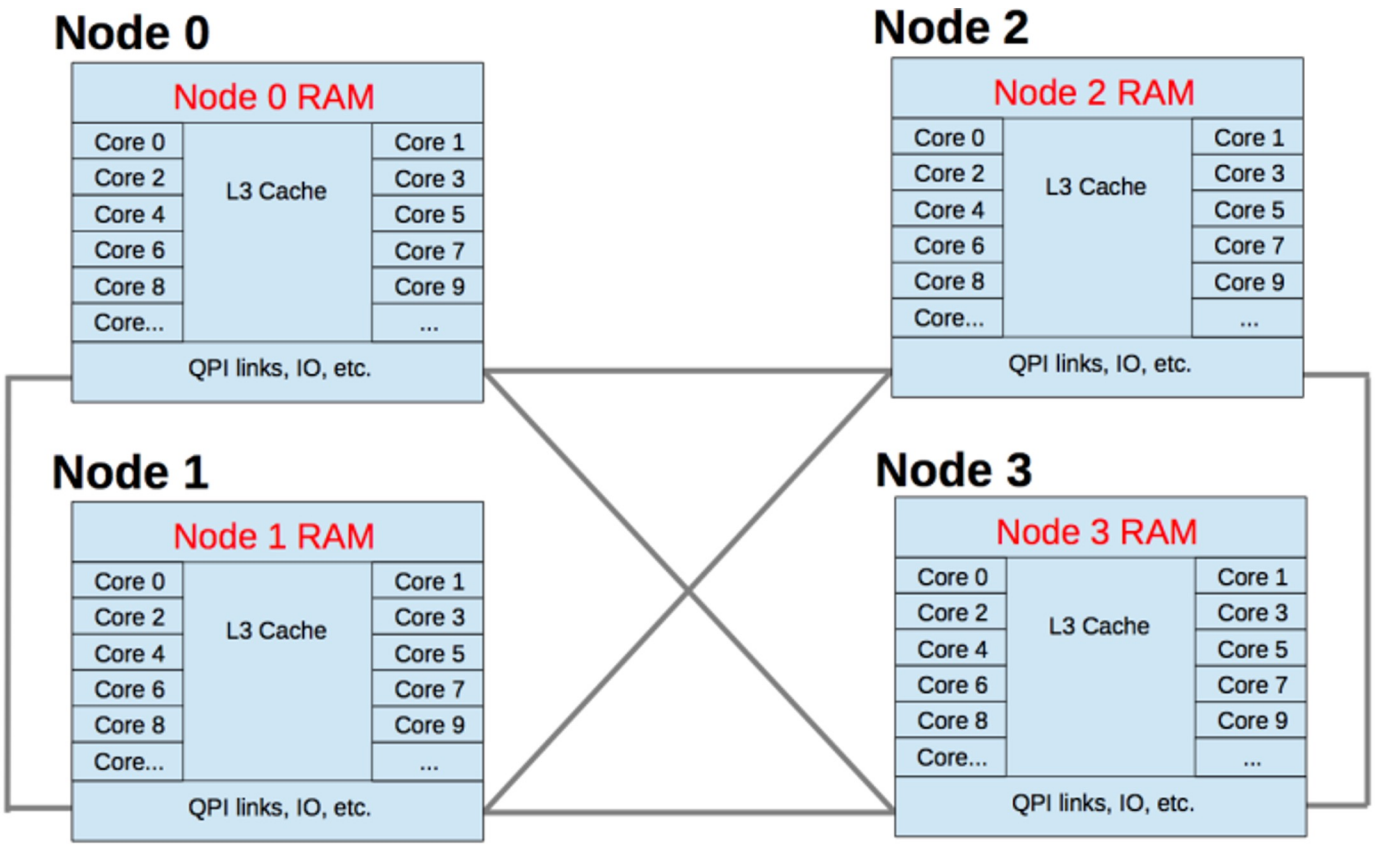


Figure 3.5: Deteriorating performance advantage of a cluster built with large SMP server nodes (128-core SMP) over a cluster with the same number of processor cores built with low-end server nodes (four-core SMP), for clusters of varying size.

Compare to big SMP:



BOSTON UNIVERSITY

Big old SMPs...

SGI Origin 2000

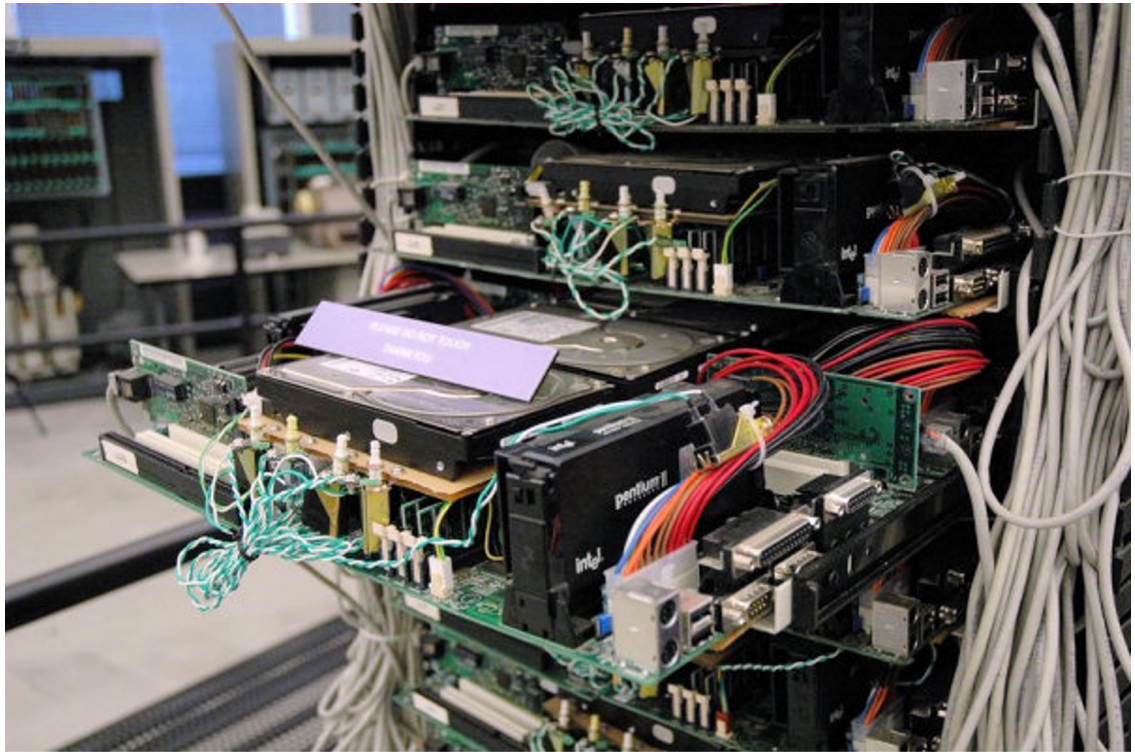
And not so old...

Learn | Partner Network | AWS Marketplace | Customer Enable

Instance Type	vCPU	Mem (GiB)
u-6tb1.56xlarge	224	6,144
u-6tb1.112xlarge	448	6,144
u-9tb1.112xlarge	448	9,216
u-12tb1.112xlarge	448	12,288

BOSTON UNIVERSITY

“Commodity hardware” - Gen 1

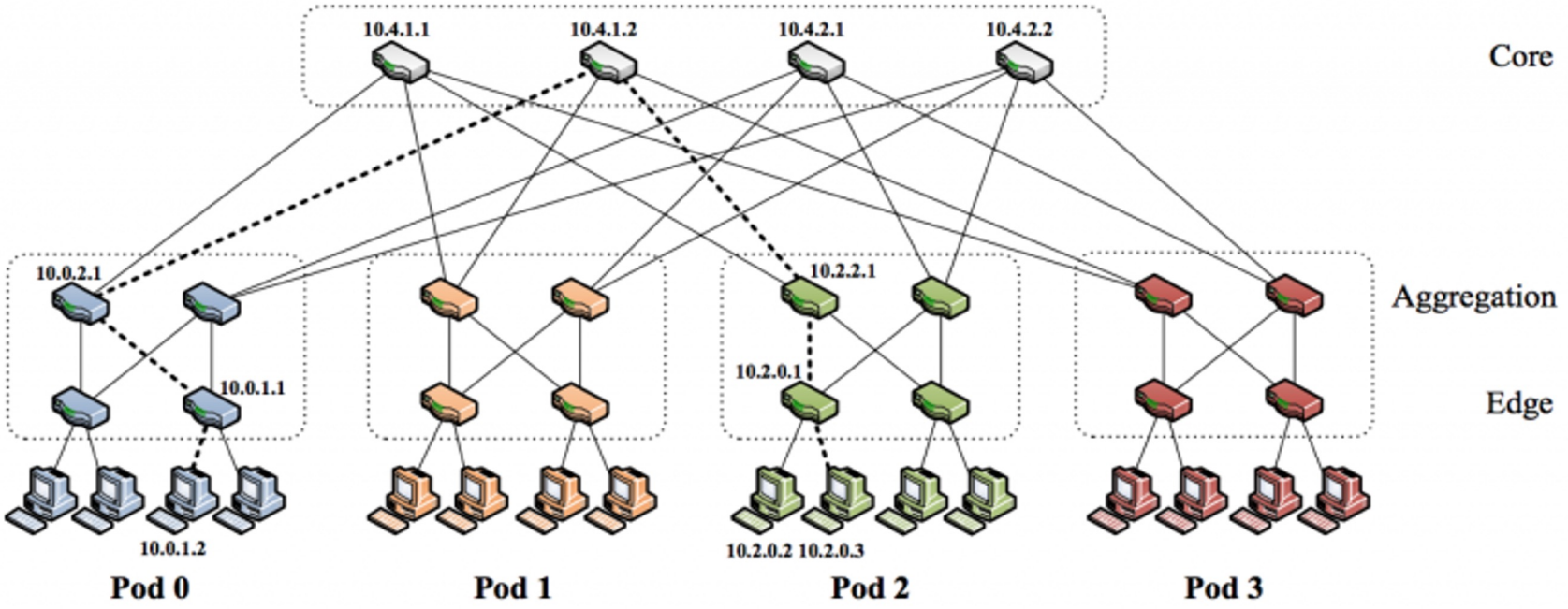


BOSTON UNIVERSITY

Commodity networking

A “switch chip”.

Clos Topologies



What criticisms do you have for this book?

Criticisms

- Lack of understanding of other spaces:
  - BlueGene radically different design point for massive scale system designed as a single system
  - Much larger space of problems can be solved with Wimpy processors (see project Kittyhawk)
  - VMware and networking partners solved many of the problems with VM migration
- Focus of authors is on closed systems; designed by a single company.
- Note, Amazon is now massively beating Google in IaaS market
- Can any data center truly be considered as having a single entity?
- The paper constantly confuses price with cost.
- Historically, open systems/marketplaces eventually dominate…

Concluding remarks

- This book is fascinating, discusses internal thinking of the most successful massive provider:
  - This is a practitioner paper, this is what they evolved and built incrementally
  - It doesn't mean that they are right.
- WSC is a fundamental change in what we think of as the system
- Transformative change moving from client to cloud.
- Drives fundamental changes in our SW development practices and new concerns:
  - E.g., concern for tail latency
- New programming paradigms critical to dealing with complex distributed systems
- Everyone is concerned today with cluster rather than individual system performance.
- Cloud is the new commodity; the focus on using commodity parts misses the point

Extra Materials

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
UC Berkeley EECS Tech Report UCB/EECS-2009-28

BU Department of Electrical & Computer Engineering

BOSTON UNIVERSITY

What is “cloud”?

- Applications delivered as services over the Internet
  - Hardware and software delivering these applications

When made available on a pay-as-you-go basis:

Public cloud

Some people are confused

”*We’ve redefined Cloud Computing to include everything that we already do*"

Larry Ellison, Oracle

"It’s a marketing hype campaign"

disgraced free software guru

(note - Oracle **didn’t have a cloud** in 2008)

BOSTON UNIVERSITY

Introduces 3 levels in the cloud

1. “User”
   uses a service

   me - I use Dropbox, GitHub
2. “Service provider”
   provides a service
   uses a platform

   Dropbox, Github themselves
3. “Platform provider”

   AWS, Azure, etc.

Classes of “Utility computing”

- Bare Virtual Machine (EC2).
  what is it? a bare computer
- General-purpose platform (Azure)
  what is it? a windows / .NET environment
- App-specific platform
  (Google appengine [3-tier web], force.com [CRM])

What can you run on it?

- EC2:
  pretty much anything
- Azure
  Any Windows or .NET application
- AppEngine:
  Any 3-tier web server written in python using Google's AppEngine-specific APIs

Objections to cloud

- Confidentiality - “I’ll never trust X to the cloud”
- Data lock-in - yup.
- Data transfer bottlenecks

And a lot of concerns that have us scratching our heads nowadays.

Some key insights

- “pay-as-you-go” ≠ “renting”
  (or at least: ≠ “leasing”)

Cloud provides pooling of risk (of over / under-utilization) and assumption of risk by the cloud provider, in return for profit

Kind of like an insurance company...

BOSTON UNIVERSITY

Some key insights

- Cloud is not a technology

It:

- is enabled by some techologies (e.g. virt)
- enables some other technologies

but that's different

What did they get wrong?

- User / Service Provider / Utility provider distinction?
- Naive view of future applications?
- anything else?

Privacy issues

- Cloud service providers have already collected petabytes of sensitive personal information stored in data centers around the world. The acceptance of Cloud Computing therefore will be determined by privacy issues addressed by these companies and the countries where the data centers are located.

- Privacy is affected by cultural differences; some cultures favor privacy, others emphasize community. This leads to an ambivalent attitude towards privacy in the Internet which is a global system.

Q&A

BOSTON UNIVERSITY