COMP 6611B: Topics on Cloud Computing and Data Analytics Systems

---

Wei Wang
Department of Computer Science & Engineering
HKUST
Fall 2015

Above the Clouds

Utility Computing

- Applications and computing resources delivered as a service over the Internet
  - Pay-as-you-go
- Provided by the hardwares and system softwares in the datacenters

Visions

- The illusion of **infinite** computing resources available on demand
- The elimination of an up-front commitment by Cloud users
- The ability to pay for use of computing resources on a short-term basis as needed

amazon web services™ EC2

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
t2.micro	1	Variable	1	EBS Only	$0.013 per Hour
t2.small	1	Variable	2	EBS Only	$0.026 per Hour
t2.medium	2	Variable	4	EBS Only	$0.052 per Hour
t2.large	2	Variable	8	EBS Only	$0.104 per Hour
m4.large	2	6.5	8	EBS Only	$0.126 per Hour
m4.xlarge	4	13	16	EBS Only	$0.252 per Hour
m4.2xlarge	8	26	32	EBS Only	$0.504 per Hour
m4.4xlarge	16	53.5	64	EBS Only	$1.008 per Hour
m4.10xlarge	40	124.5	160	EBS Only	$2.52 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	$0.067 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	$0.133 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	$0.266 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	$0.532 per Hour

EC2

- Pay-as-you-go model
  - No upfront cost, no contract, no minimum usage commitment
  - Fixed hourly rate
  - Billing cycle rounded to nearest hour: 1.5 h = 2 h

1 instance for 1000 h = 1000 instances for 1 h

Cloud Economics: does it make sense?

Shall I move to the Cloud?

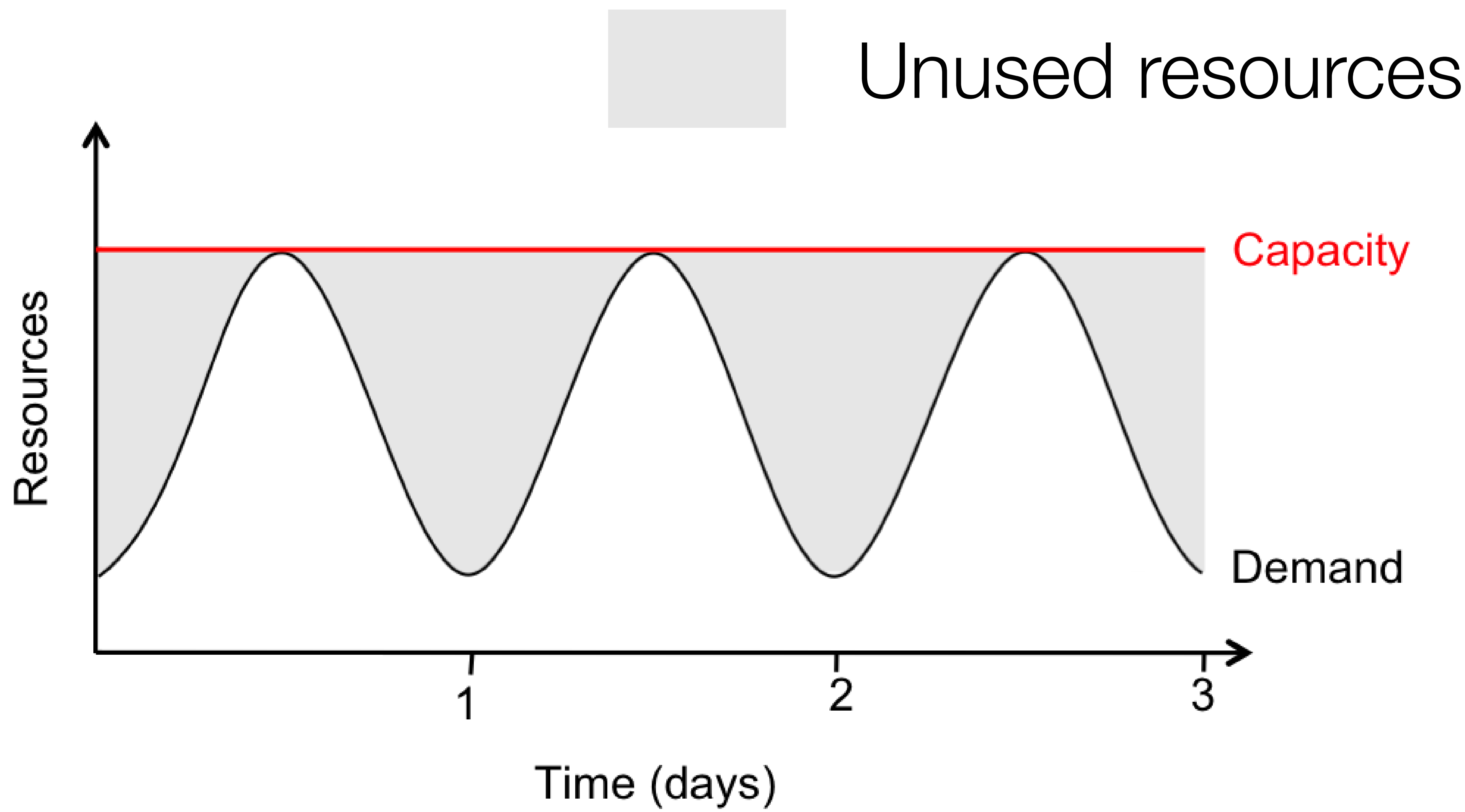
- Profit from cloud >= profit from in-house infrastructures

$$\text{UserHours}_{\text{cloud}} \times (\text{revenue} - \text{Cost}_{\text{cloud}})$$
$$\geq \text{UserHours}_{\text{datacenter}} \times (\text{revenue} - \frac{\text{Cost}_{\text{datacenter}}}{\text{Utilization}})$$

Source: Ambrust et al., “Above the clouds: A Berkeley’s view of Cloud Computing.”

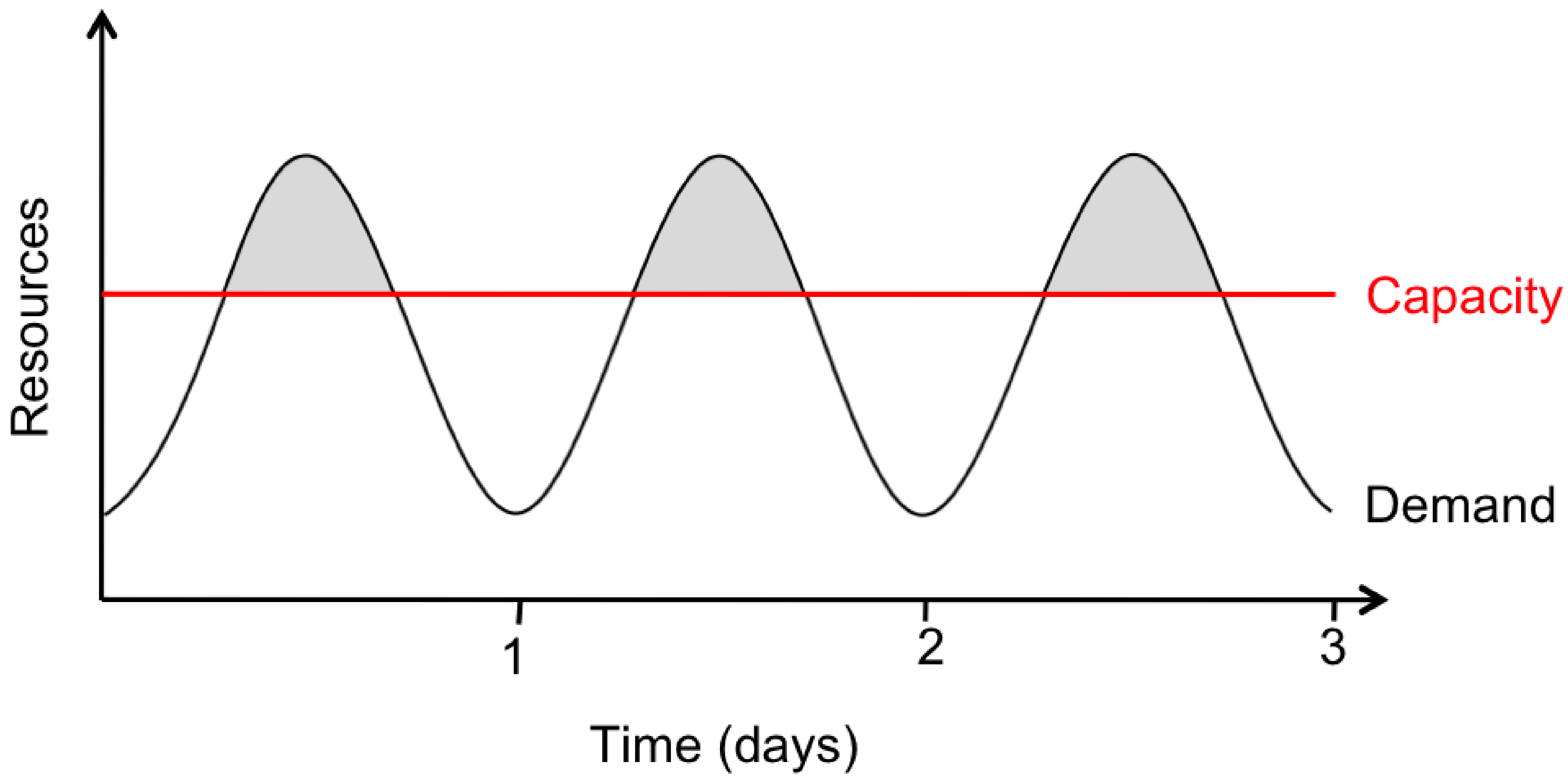
Provisioning for peak load

- Even if we can accurately predict the peak load



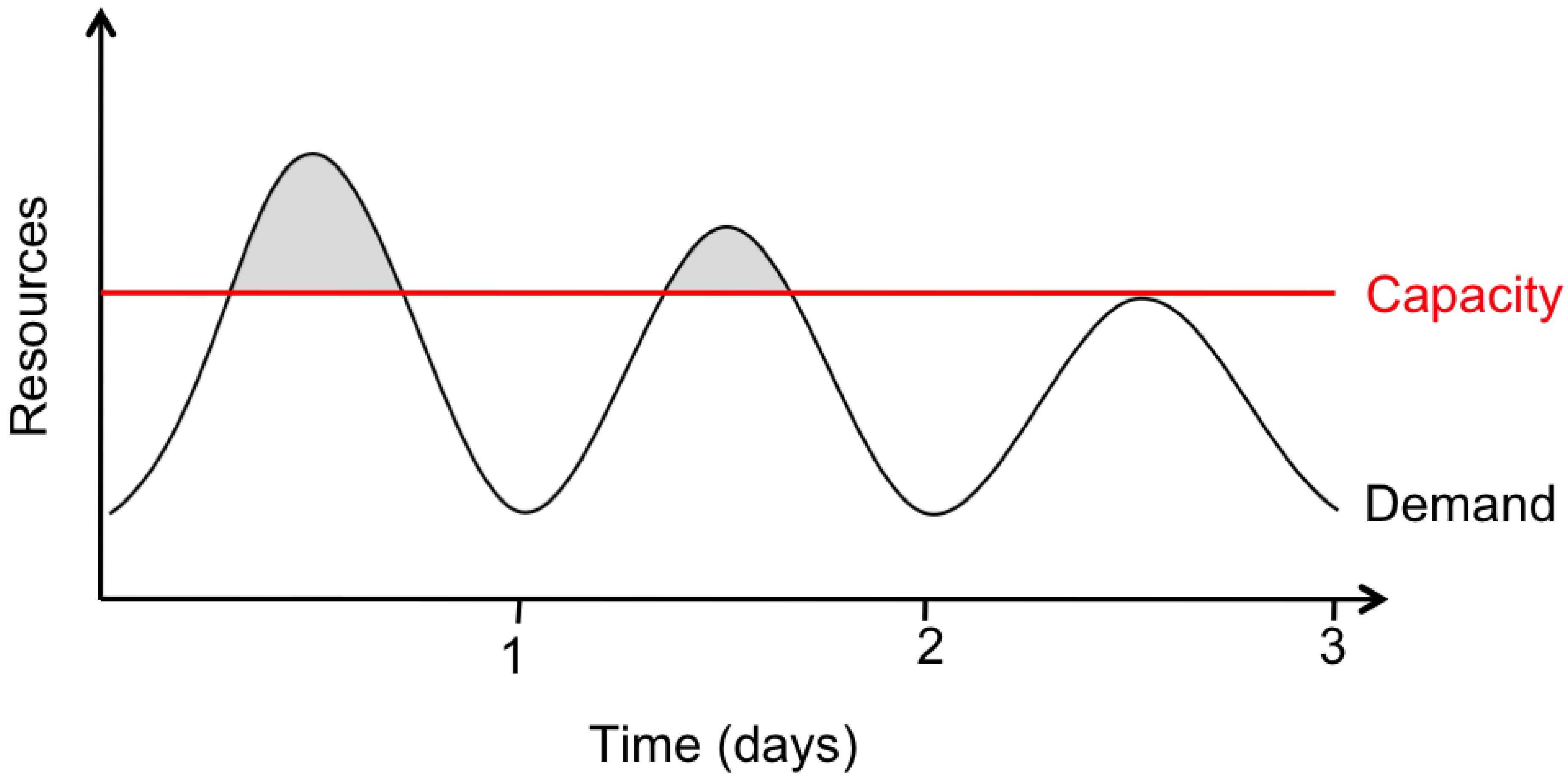
Source: Ambrust et al., “Above the clouds: A Berkeley’s view of Cloud Computing.”

Underprovisioning



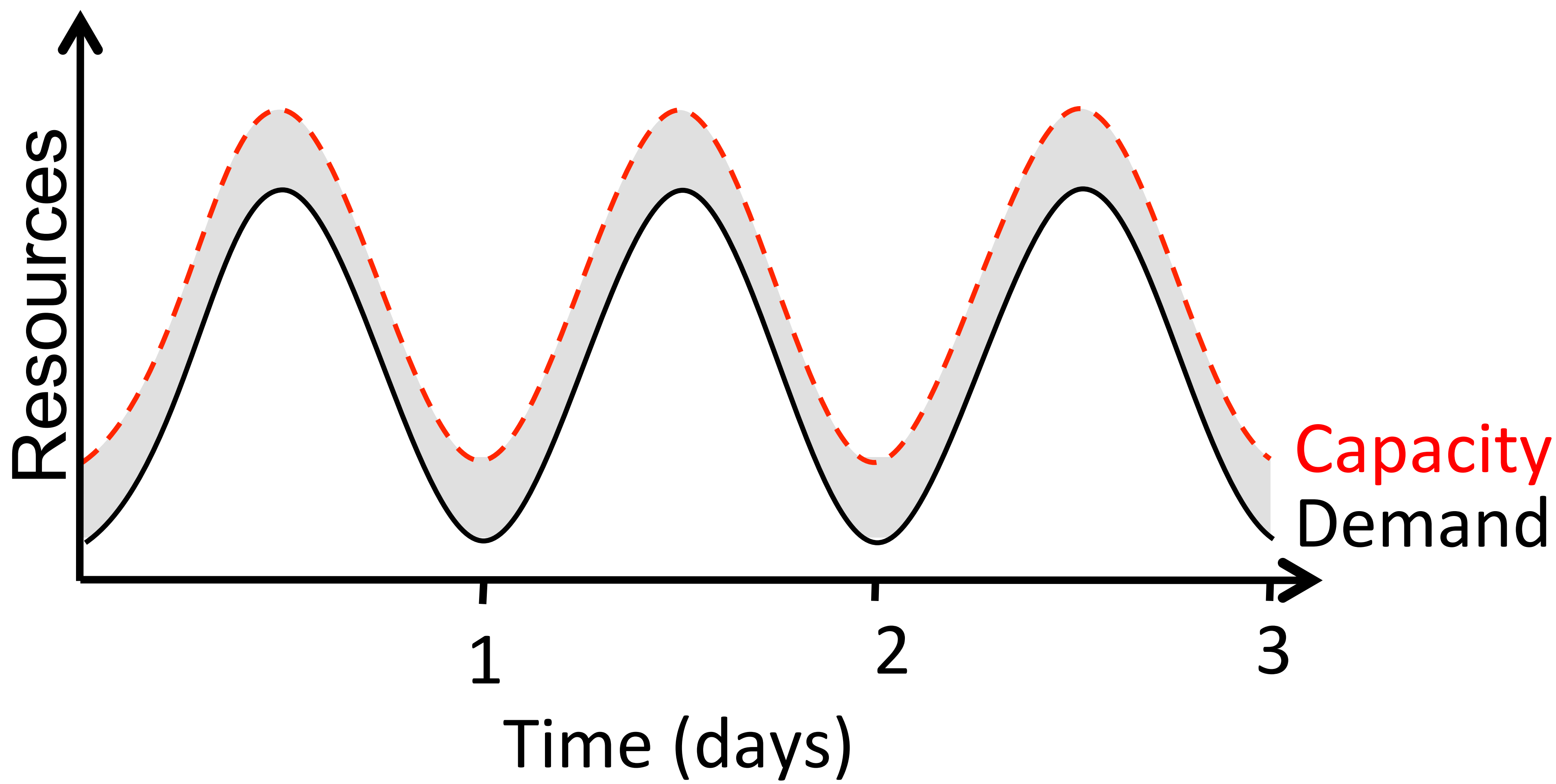
Source: Ambrust et al., “Above the clouds: A Berkeley’s view of Cloud Computing.”

Underprovisioning



Source: Ambrust et al., “Above the clouds: A Berkeley’s view of Cloud Computing.”

Cloud provisioning on demand



Case study

Animoto: a cloud-based video creation service

- Scale from 50 servers to 3500 servers in 3 days when making its services available via Facebook
- Scale back down to a level well below the peak afterwards

Highly profitable business for Cloud providers

Economy of scale

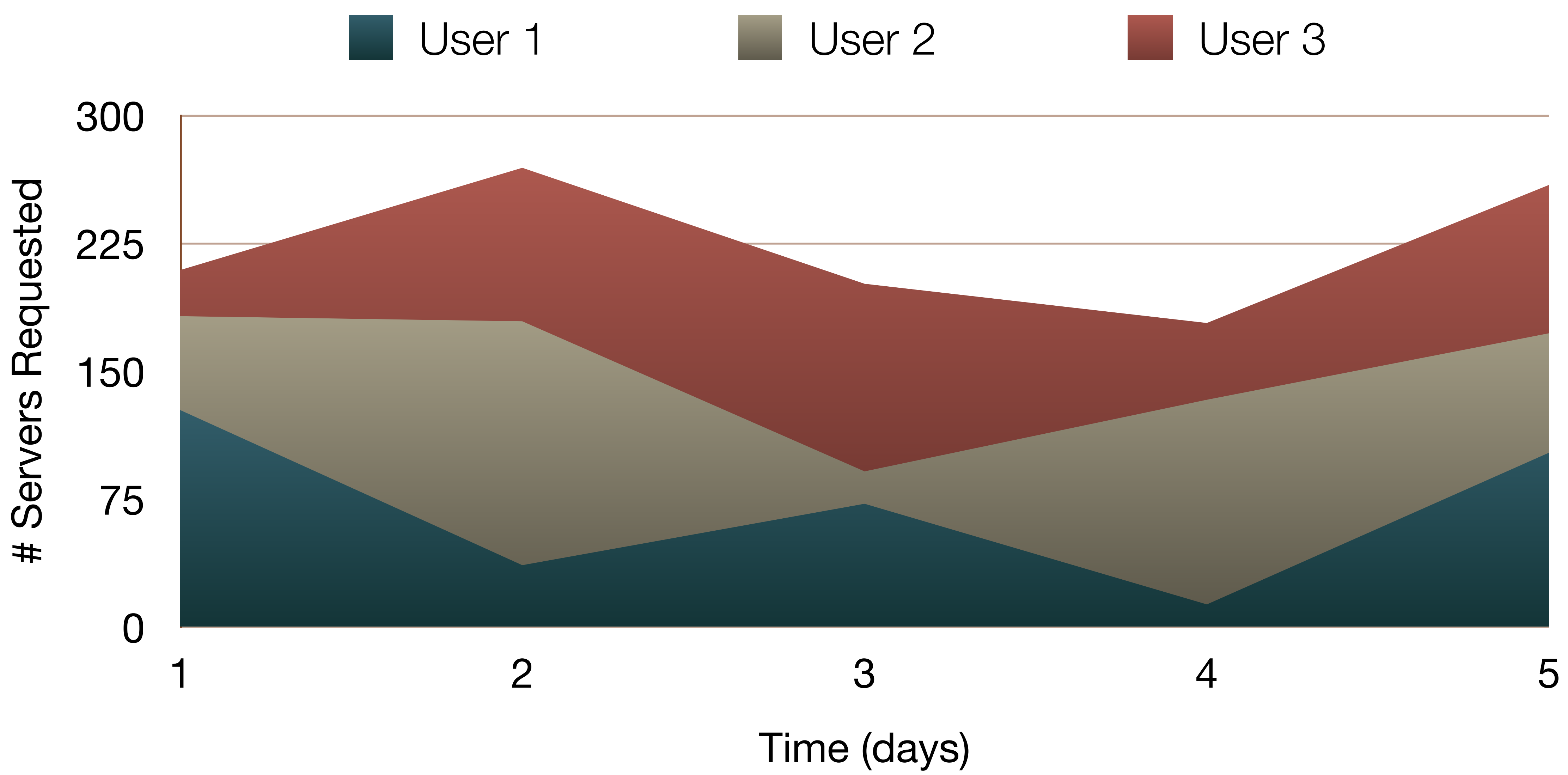
- A medium-sized datacenter (~1k servers) vs. a large datacenter (~50k servers) in 2006

Technology	Cost in Medium-sized DC	Cost in Very Large DC	Ratio
Network	$95 per Mbit/sec/month	$13 per Mbit/sec/month	7.1
Storage	$2.20 per GByte / month	$0.40 per GByte / month	5.7
Administration	≈140 Servers / Administrator	>1000 Servers / Administrator	7.1

5 - 7x decrease of cost!

Source: Ambrust et al., "Above the clouds: A Berkeley's view of Cloud Computing."

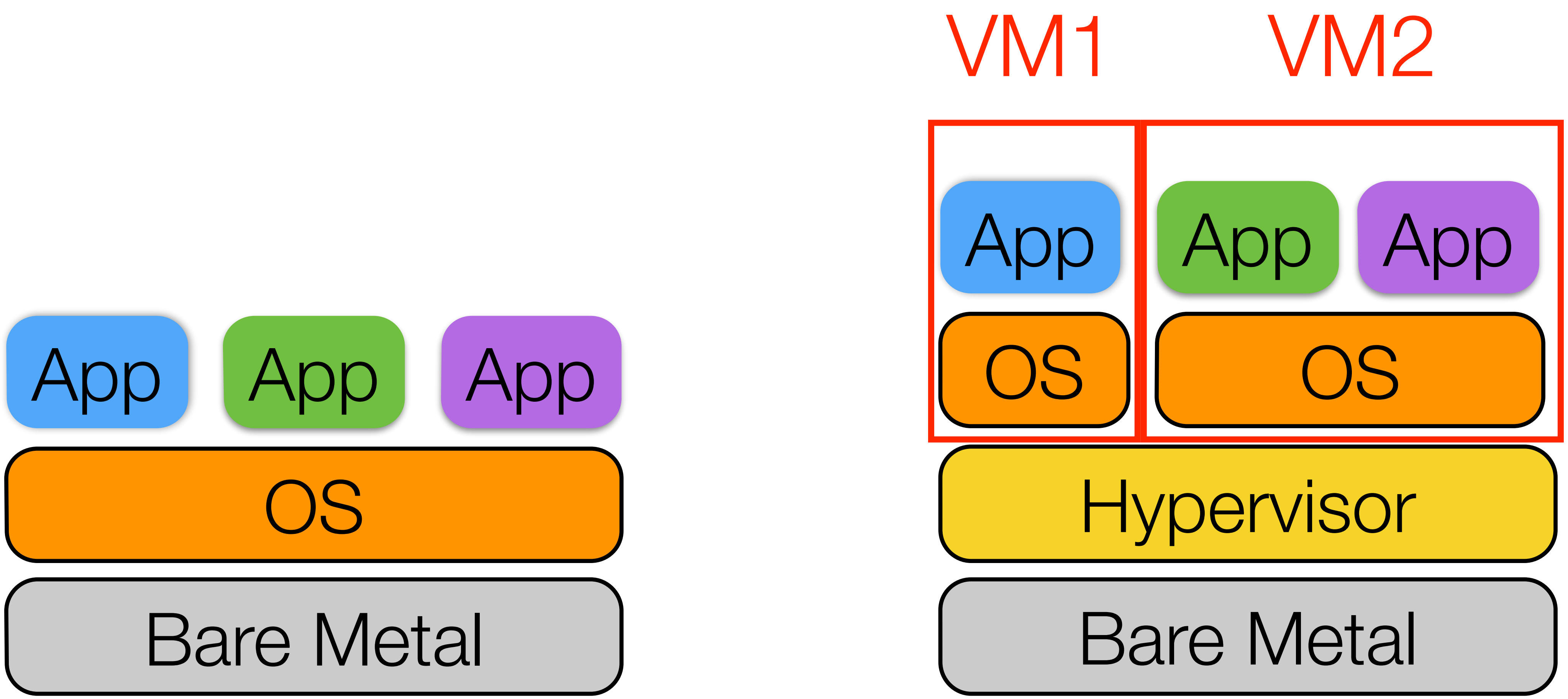
Statistical multiplexing



Plus…

- **Leverage existing investment**, e.g., Amazon
- **Defend a franchise**, e.g., Microsoft Azure
- **Attack an incumbent**, e.g., Google AppEngine
- **Leverage customer relationships**, e.g., IBM
- **Become a platform**, e.g., Facebook, Apple, etc.

Enabling technology: Virtualization



Traditional stack

Virtualized stack

What kind of Cloud services do I expect?

Infrastructure-as-a-Service

- Processing, storage, networks, and other computing resources, typically in a form of virtual machines
- Full control of OS, storage, applications, and some networking components (e.g., firewalls)

Platform-as-a-Service

- Deploy onto the cloud infrastructure the applications created by programming languages, libraries, services, and tools supported by the provider

- No control of OS, storage, or network, but can control the deployed applications and host environment

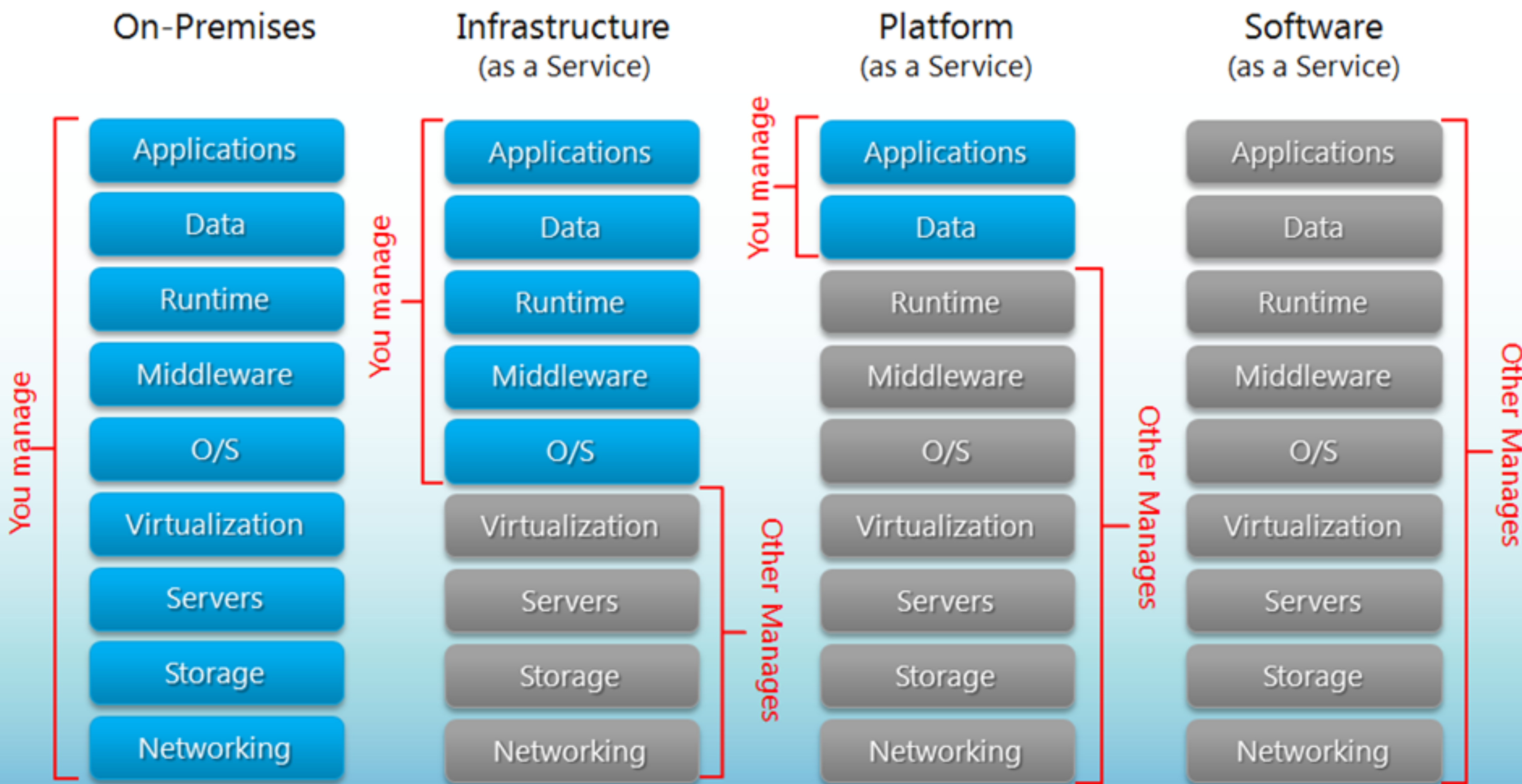
Google App Engine

Microsoft Azure

Software-as-a-Service

- Use the provider's applications running on a cloud infrastructure
- No control of network, OS, storage, and application capabilities, except limited user-specific configuration settings

Separation of Responsibilities



Source: K. Remde, “SaaS, PaaS, and IaaS.. Oh my!” TechNet Blog, 2011

Infrastructure (as a Service) | **Platform (as a Service)** | **Software (as a Service)**

Lower-level,
General-purpose,
Less managed

Higher-level,
Application-specific,
More managed

We shall focus on IaaS in this course

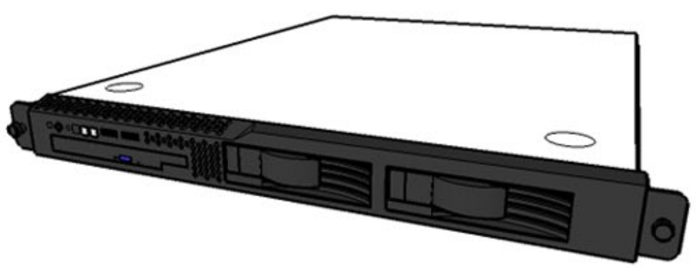
How can the Cloud services be provisioned?

Source: Google

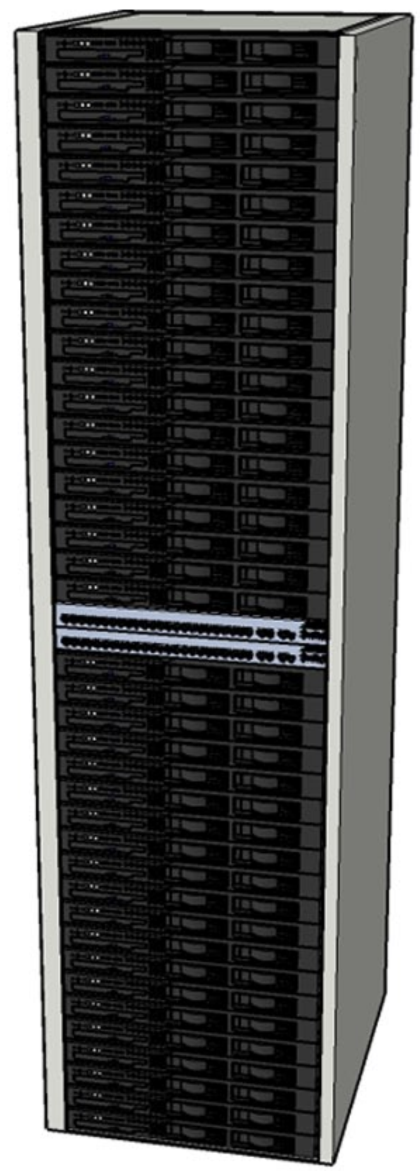

Source: Google


Source: Google

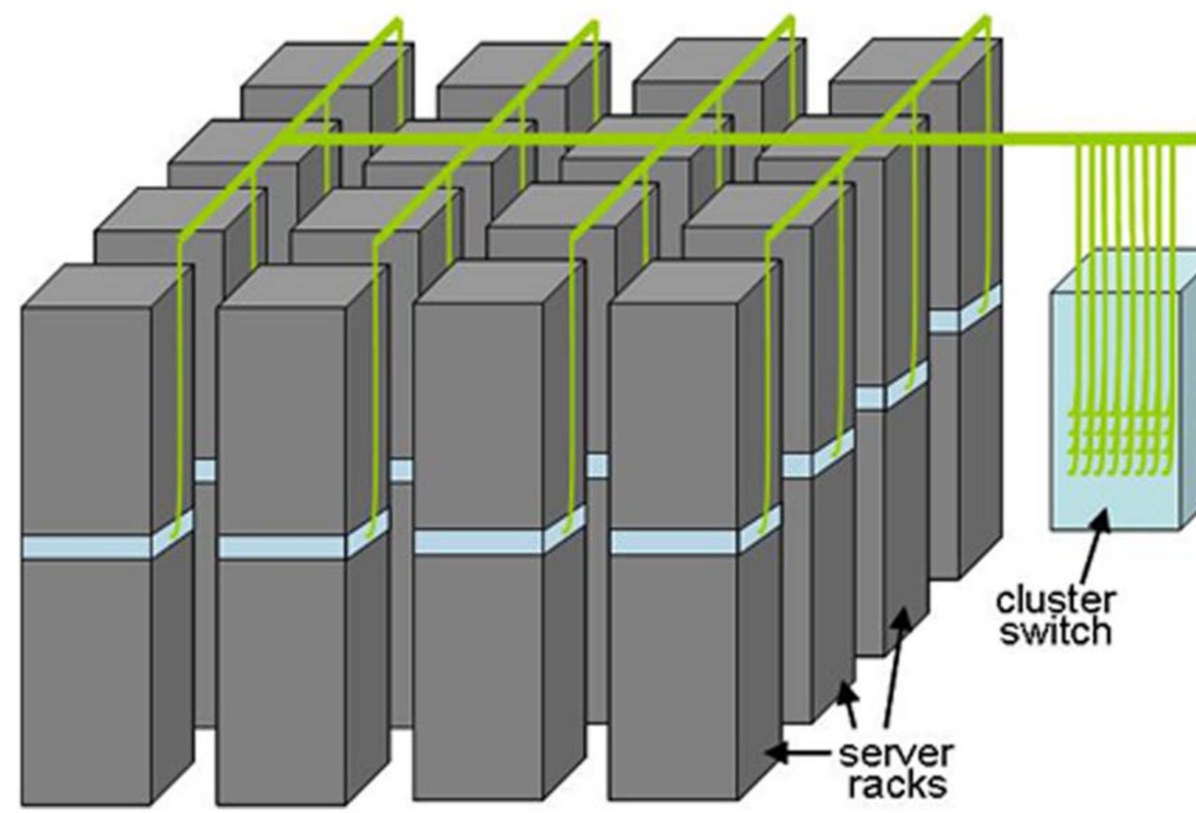
A look into the datacenter



Commodity Server



Rack

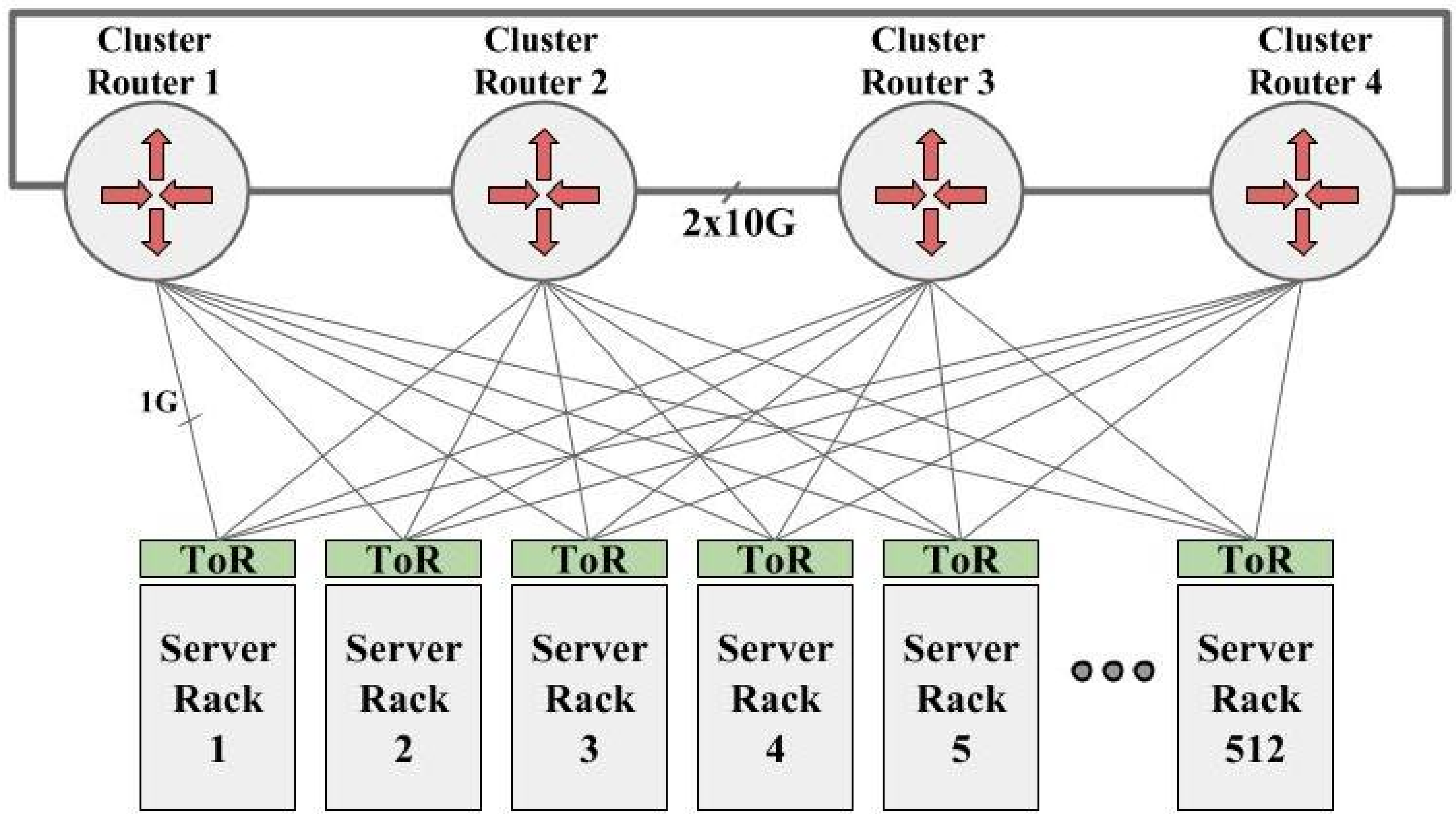


Cell

Source: L. Barroso et al., “The datacenter as a computer: An introduction to the design of warehouse-scale machines.”

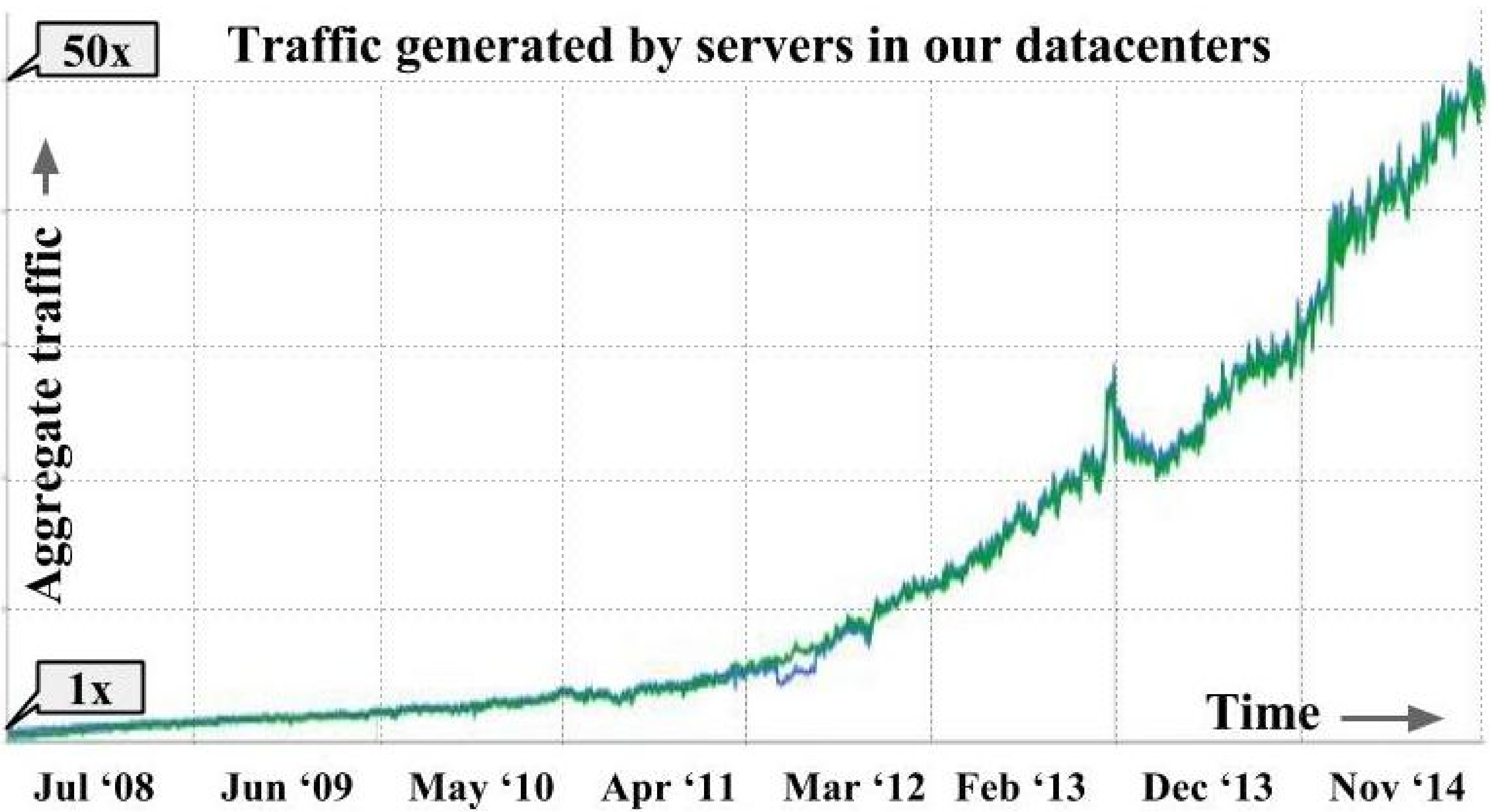
Network infrastructure

- Back to 2004 when Google has only 20k servers in a datacenter



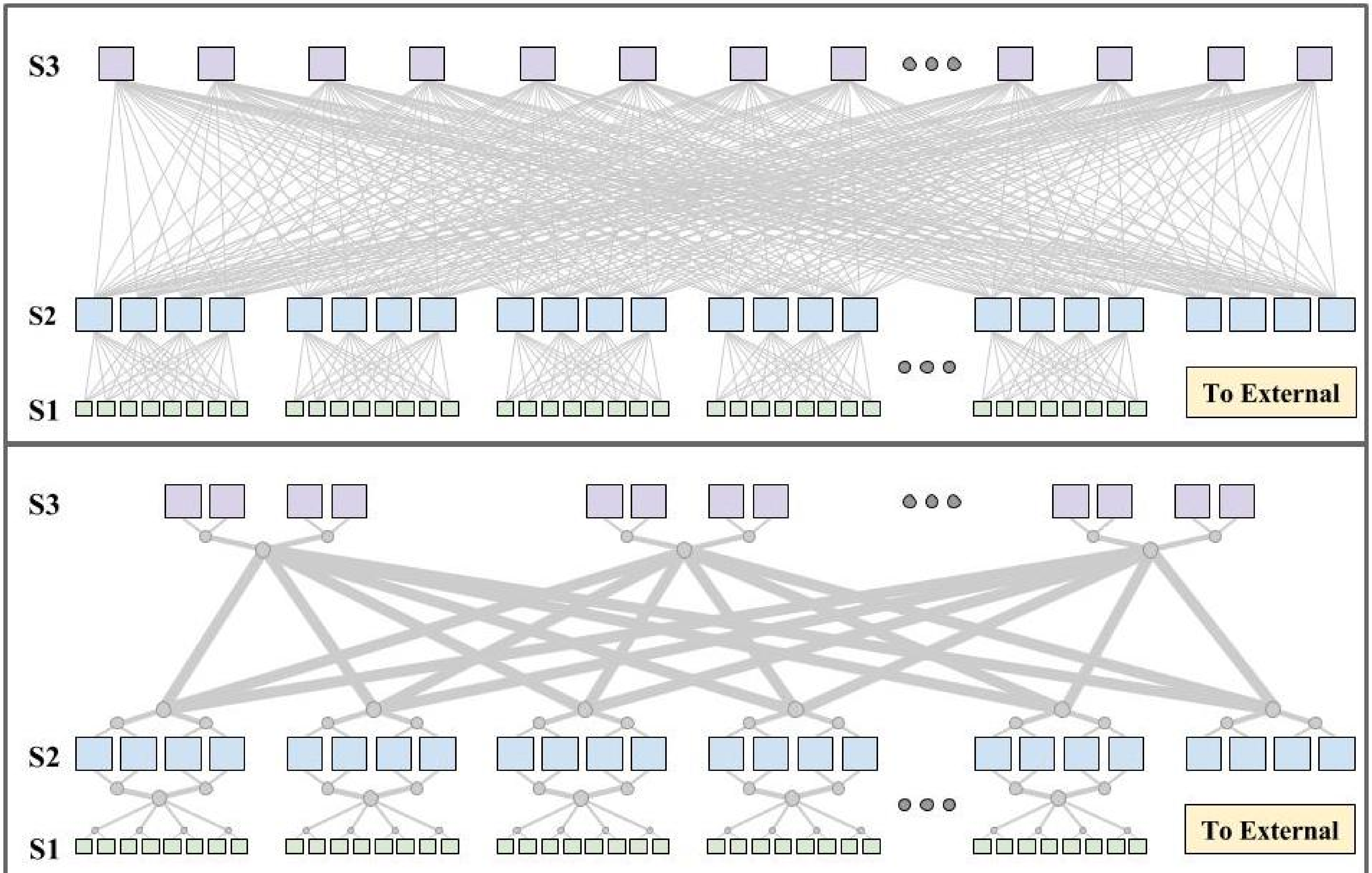
Source: A. Singh et al., "Jupiter rising: A decade of Clos topologies and centralized control in Google's datacenter network," ACM SIGCOMM'15.

Things have changed quite a lot



Source: A. Singh et al., "Jupiter rising: A decade of Clos topologies and centralized control in Google's datacenter network," ACM SIGCOMM'15.

Challenge: network



Source: A. Singh et al., "Jupiter rising: A decade of Clos topologies and centralized control in Google's datacenter network," ACM SIGCOMM'15.

Challenge: storage

- Large dataset cannot fit into a local storage
- Persistent storage must be **distributed**
  - GFS, BigTable, HDFS, Cassandra, S3, etc.
- Local storage goes **volatile**
  - Cache for data being served
  - local logging and async copy to persistent storage

Challenge: scale

- Large cluster: able to host petabytes of data
- Extremely large cluster: at Google, the storage system pages a user *if there is only a few petabytes of spaces left available*!
  - A 10k-node cluster is considered small- to medium-sized

Challenge: faults

>1%	DRAM errors per year
2-10%	Annual failure rate of disk drive
2	# crashes per machine-year
2-6	# OS upgrades per machine-year
>1	Power utility events per year

Failure is a norm, not an exception!

- A 2000-node cluster will have >10 machines crashing per day

— Luiz Barroso

Source: J. Wilkes, "Cluster management at Google."

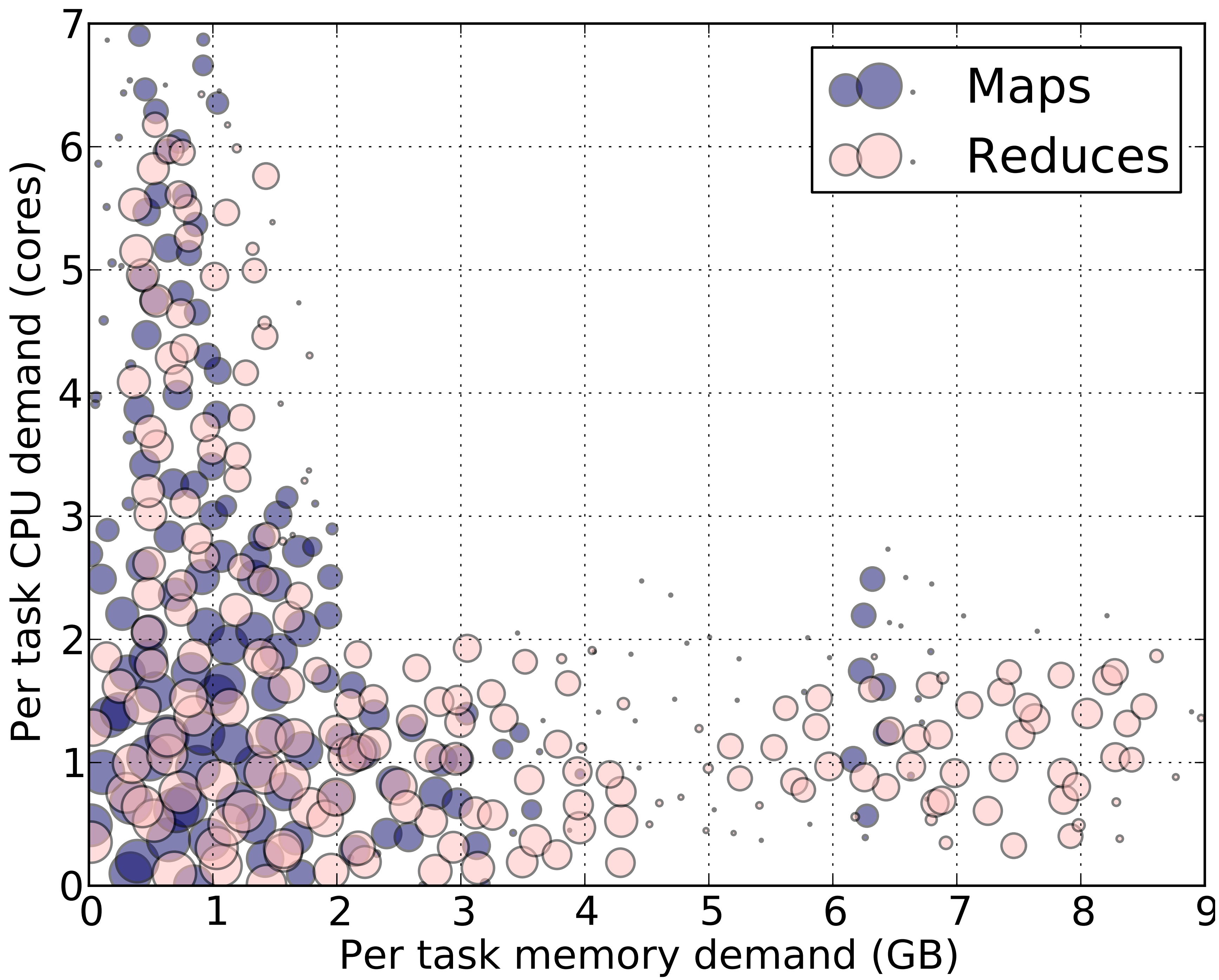
Server heterogeneity

- Servers span multiple generations representing different points in the configuration space

Number of machines	Platform	CPUs	Memory
6732	B	0.50	0.50
3863	B	0.50	0.25
1001	B	0.50	0.75
795	C	1.00	1.00
126	A	0.25	0.25
52	B	0.50	0.12
5	B	0.50	0.03
5	B	0.50	0.97
3	C	1.00	0.50
1	B	0.50	0.06

Source: C. Reiss, "Heterogeneity and dynamicity of Clouds at scale: Google trace analysis," ACM SoCC'12.

Workload heterogeneity



Source: A. Ghodsi et al., "Dominant resource fairness: fair allocation of multiple resource types," USENIX/ACM NSDI'11.

Challenges due to heterogeneity

- Hard to provide predictable and consistent services
- Hard to monitor the system, identify the performance bottleneck, or reason about the stragglers
- Hard to achieve **fair sharing** among users

Despite all these challenges, we still want to achieve…

Objectives

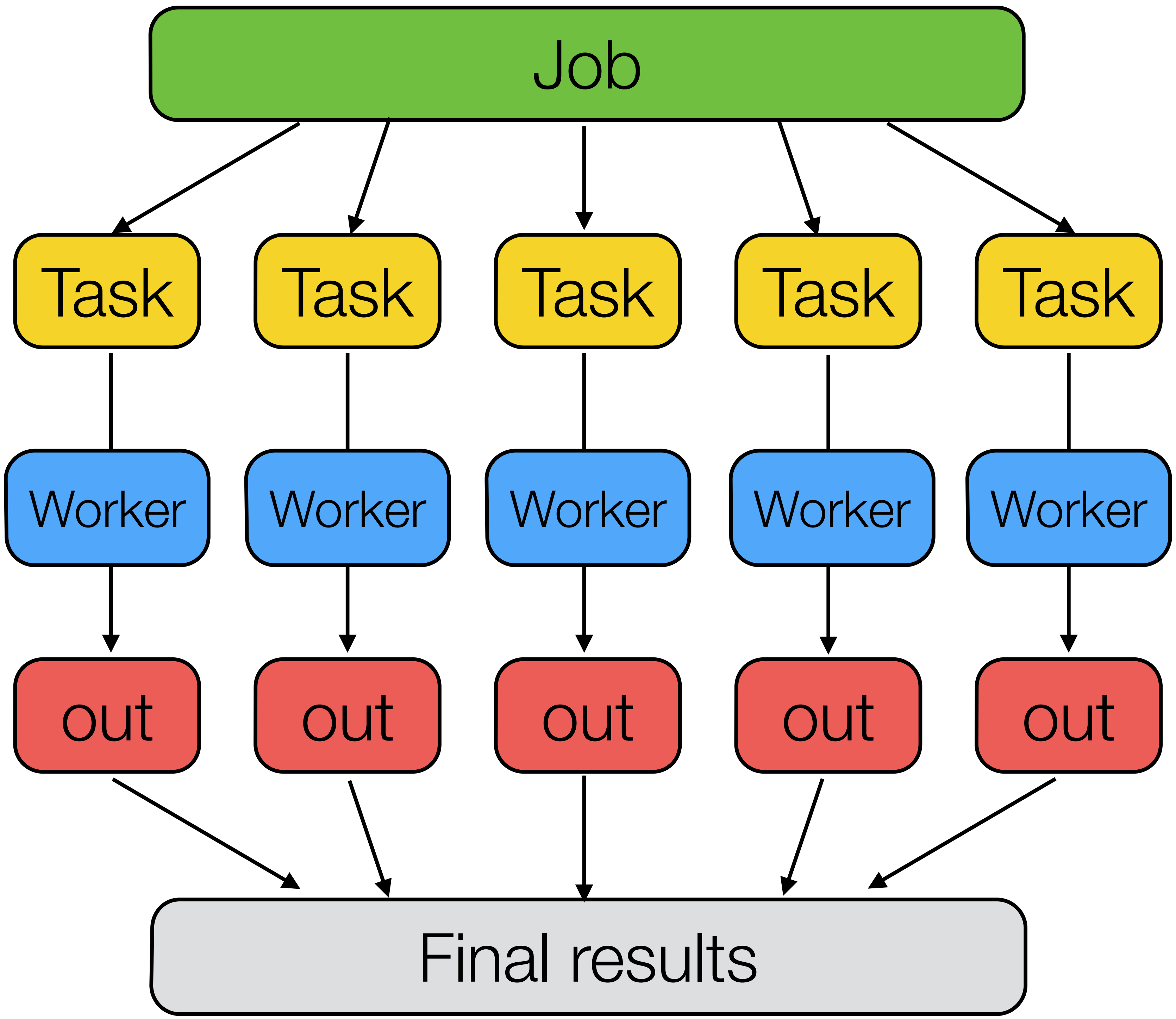
- Network with high bisection bandwidth
- Able to run everything at scale
- Fault tolerance
- Predictable services
- High utilization

With the minimum human intervention!

Now what is the Cloud user’s problem?


How to handle big data?
google.com/datacenters

Basic idea: Divide and Conquer



The degree of parallelism depends on the problem scale

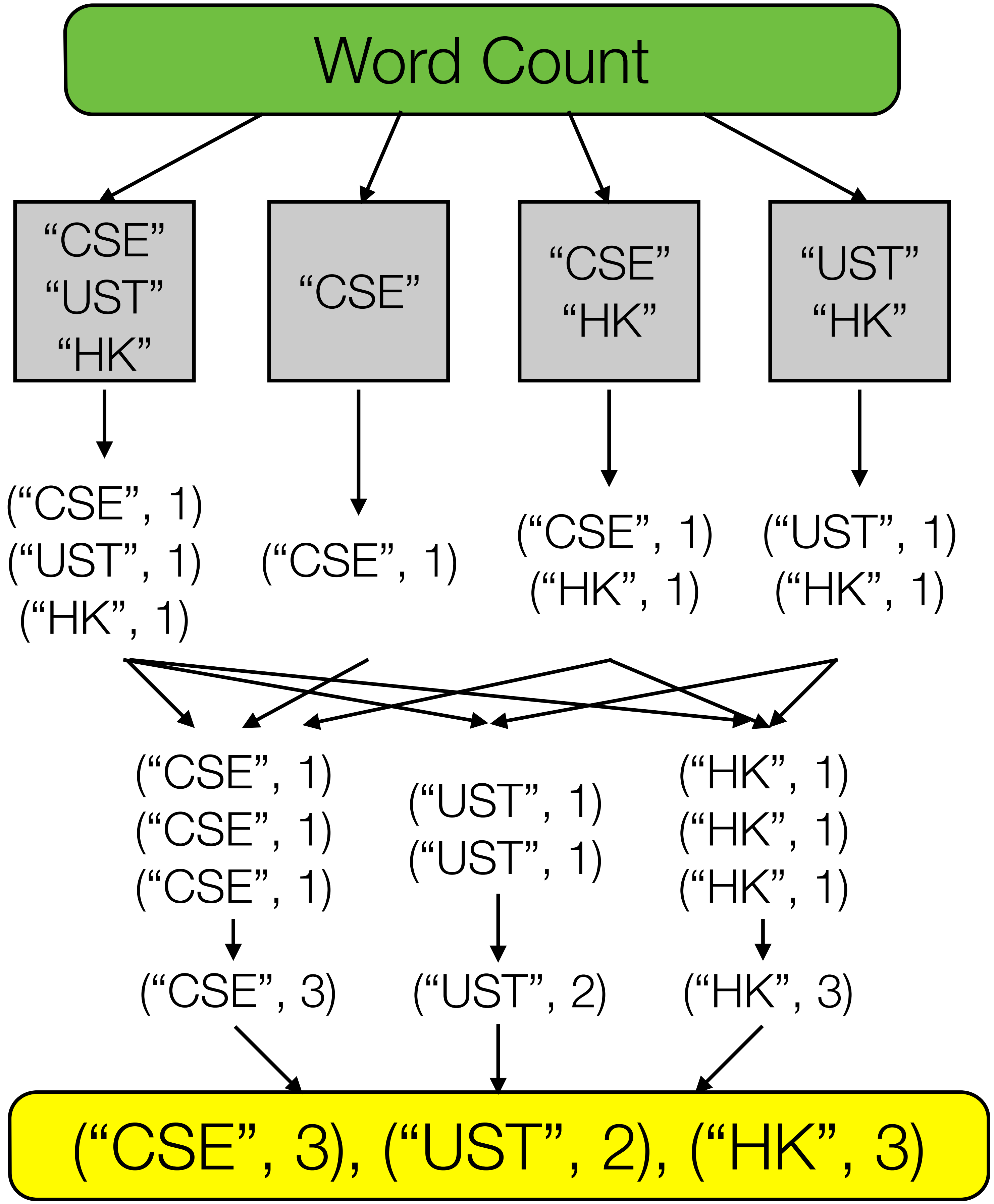
Implementation challenges

- How to schedule tasks onto the worker nodes?
- How to communicate with workers?
- How to collect/aggregate results?
- What if workers want to share intermediate results?
- What if workers become stragglers or die?
- How to monitor and reason about the problem?

A system that handles all the challenges of parallelism, allowing users to focus on the high-level logic, not low-level implementation details

Typical operations

- Iterate over a large number of records across servers
- Extract some intermediate results from each
- Shuffle and sort intermediate results
- Collect and aggregate
- Generate final output


Word Count
“CSE” “UST” “HK”
“CSE”
“CSE” “HK”
“UST” “HK”
(“CSE”, 1) (“UST”, 1) (“HK”, 1)
(“CSE”, 1)
(“CSE”, 1) (“HK”, 1)
(“UST”, 1) (“HK”, 1)
(“CSE”, 1) (“CSE”, 1) (“CSE”, 1)
(“UST”, 1) (“UST”, 1)
(“HK”, 1) (“HK”, 1) (“HK”, 1)
(“CSE”, 3)
(“UST”, 2)
(“HK”, 3)
(“CSE”, 3), (“UST”, 2), (“HK”, 3)

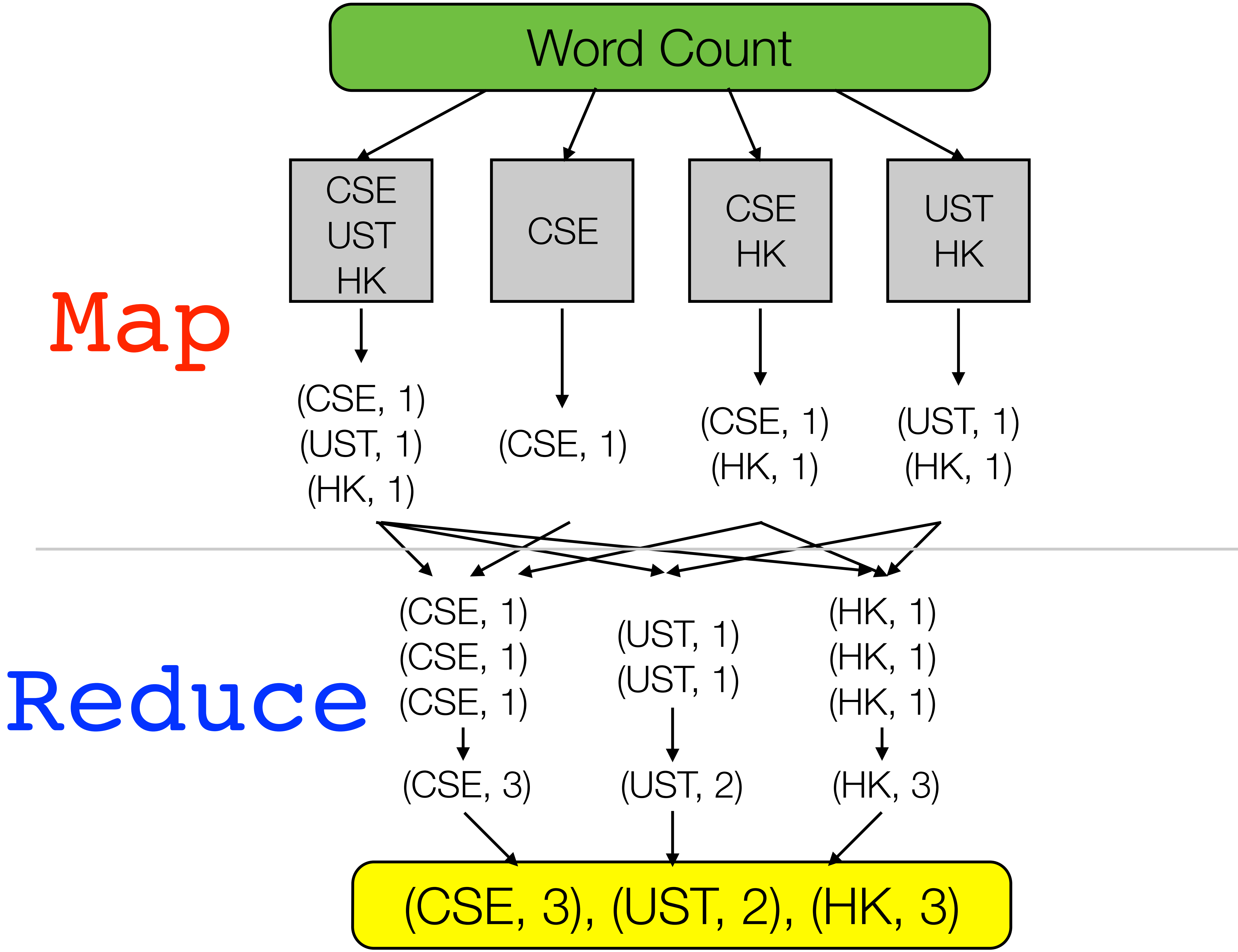
Abstract, abstract, abstract!

- Iterate over a large number of records across servers
- Extract some intermediate results from each record

Map

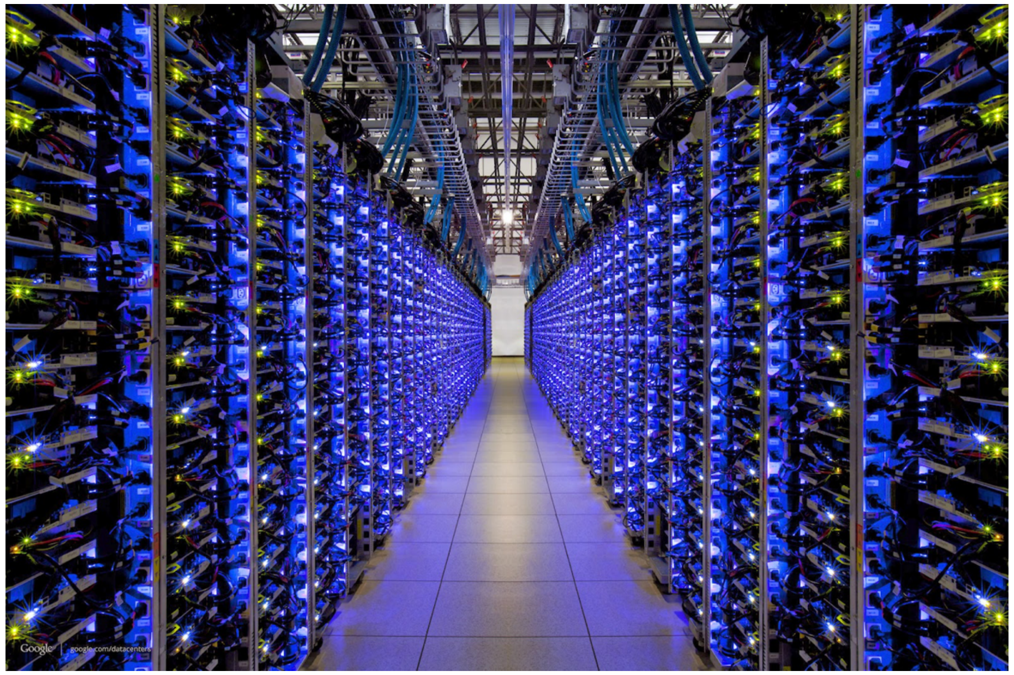
- Shuffle and sort intermediate results
- Collect and aggregate
- Generate final output

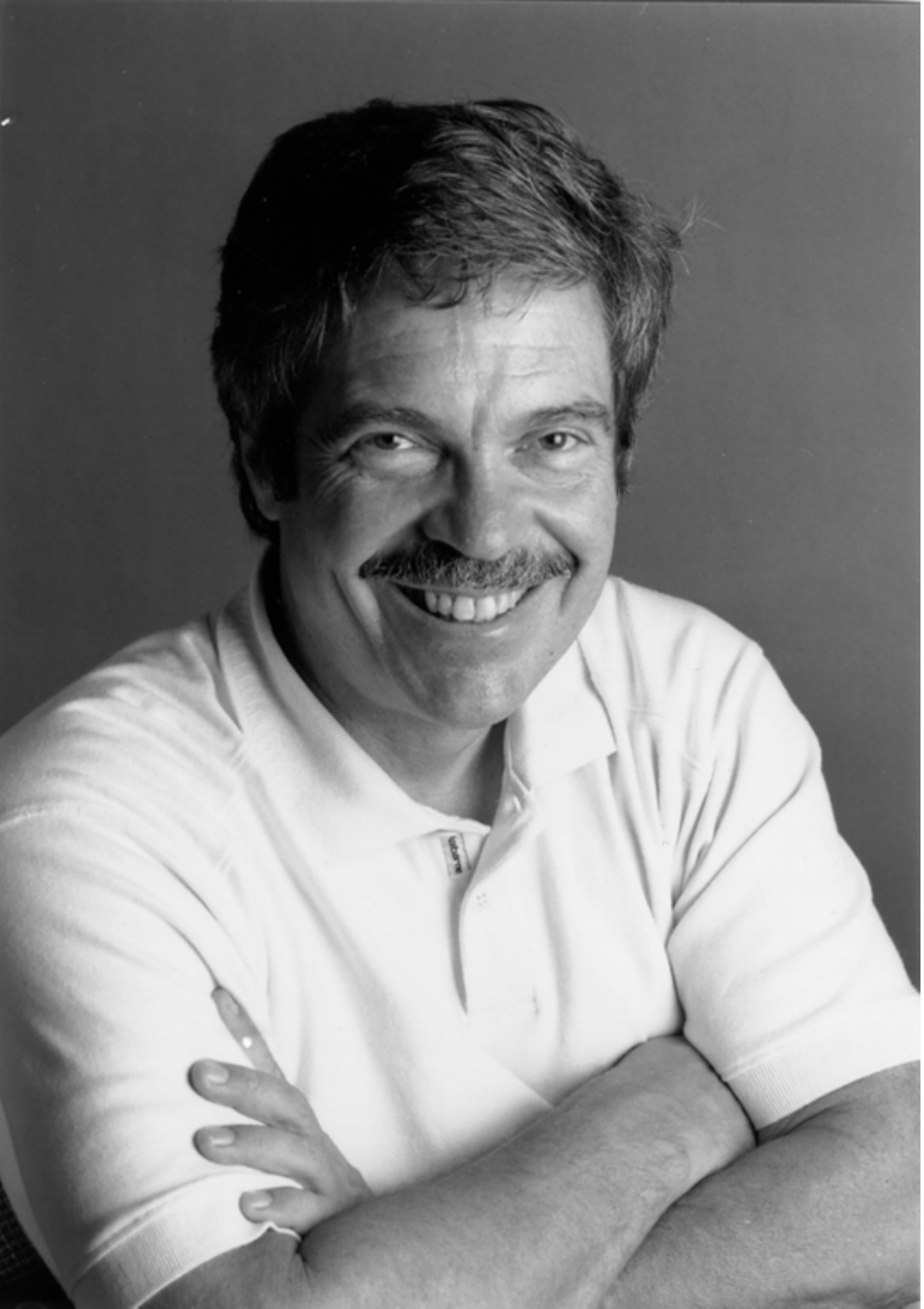
Reduce


Word Count
CSE
UST
HK
CSE
CSE
HK
UST
HK
Map
(CSE, 1)
(UST, 1)
(HK, 1)
(CSE, 1)
(CSE, 1)
(HK, 1)
(UST, 1)
(HK, 1)
Reduce
(CSE, 1)
(CSE, 1)
(CSE, 1)
(UST, 1)
(UST, 1)
(HK, 1)
(HK, 1)
(HK, 1)
(CSE, 3)
(UST, 2)
(HK, 3)
(CSE, 3), (UST, 2), (HK, 3)

MapReduce: programming on a 1000-node cluster is no more difficult than programming on a laptop

vs.





"Simple things should be simple, complex things should be possible."

— Alan Kay

Papers to be presented

Friday, Sep. 11

- MapReduce: Saethish
- Spark: Shengkai

Monday, Sep. 14

- SparkStreaming: Yaofeng
- Tez: Daizuo