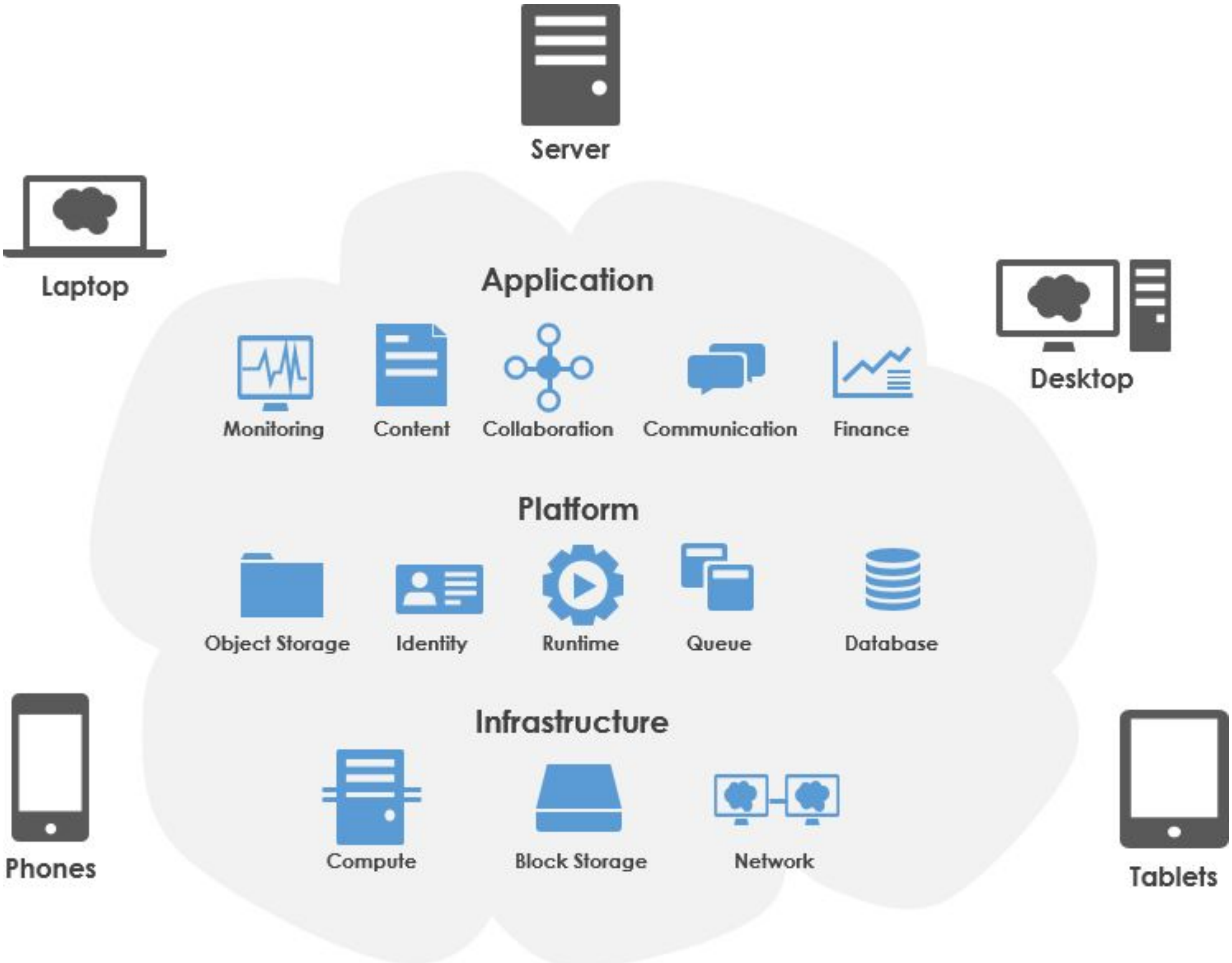
INTRODUCTION TO CLOUD
(Virtualization Core Concept)

What is Cloud Computing?

Informal: computing with large data centers.

Formal:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable **computing** resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.



Cloud Computing

Reference Example : Benefits for Cloud Computing

What is Cloud Computing?

~~Informal: computing with large datacenters~~

Our focus: computing as a utility (Pay as you Go)

- » Outsourced to a third party or internal organization

Types of Cloud Services

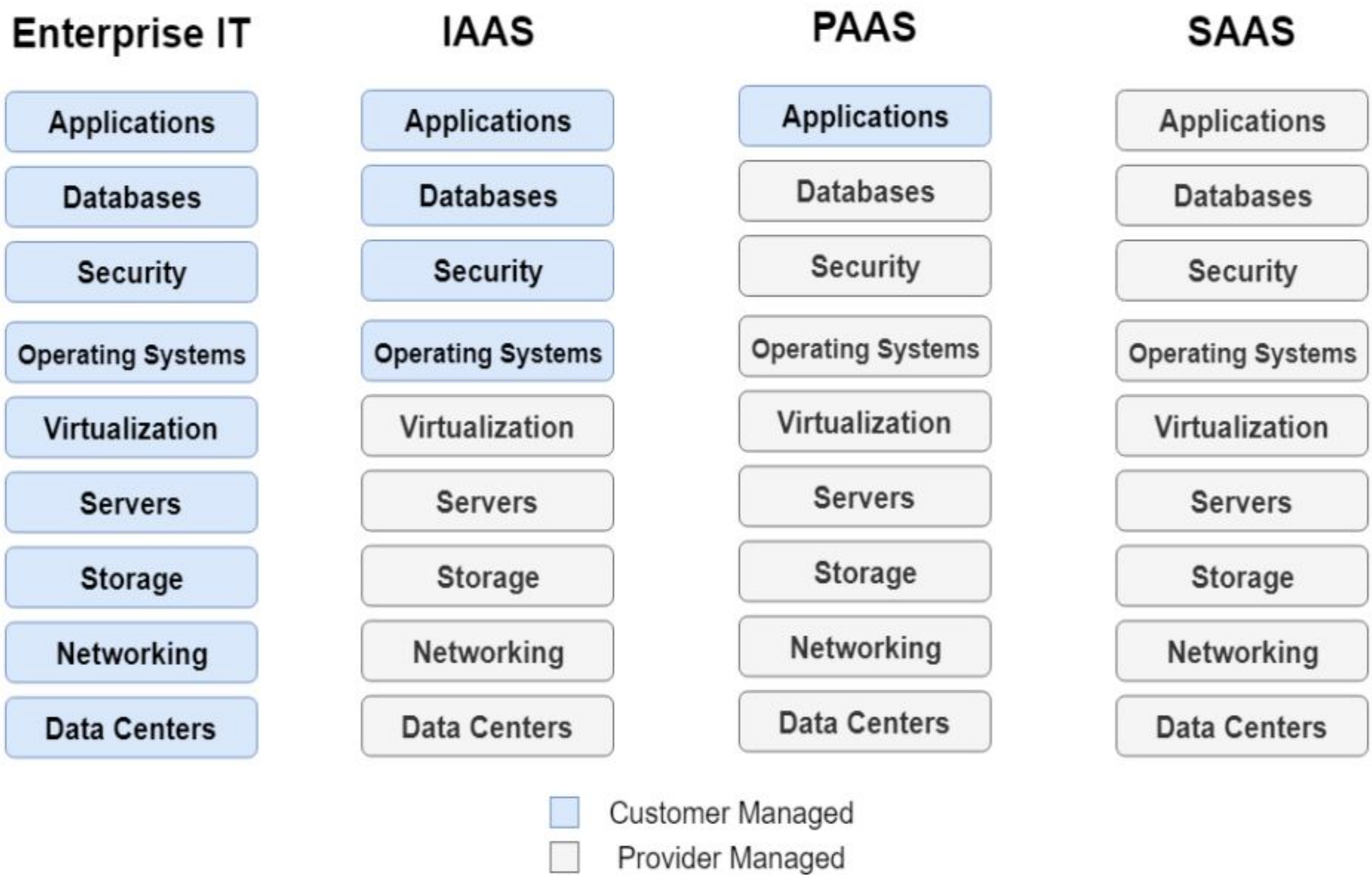
Infrastructure as a Service (IaaS): VMs, disks

Platform as a Service (PaaS): Web, MapReduce

Software as a Service (SaaS): Email, GitHub

Public vs private clouds: -

Shared across arbitrary orgs/customers
vs internal to one organization


Enterprise IT
IAAS
PAAS
SAAS
Applications
Databases
Security
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers
Applications
Databases
Security
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers
Applications
Databases
Security
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers
Applications
Databases
Security
Operating Systems
Virtualization
Servers
Storage
Networking
Data Centers
Customer Managed
Provider Managed

Example

AWS Lambda functions-as-a-service

- Runs functions in a Linux container on events
- Used for web apps, IoT apps, stream processing, highly parallel MapReduce and video encoding

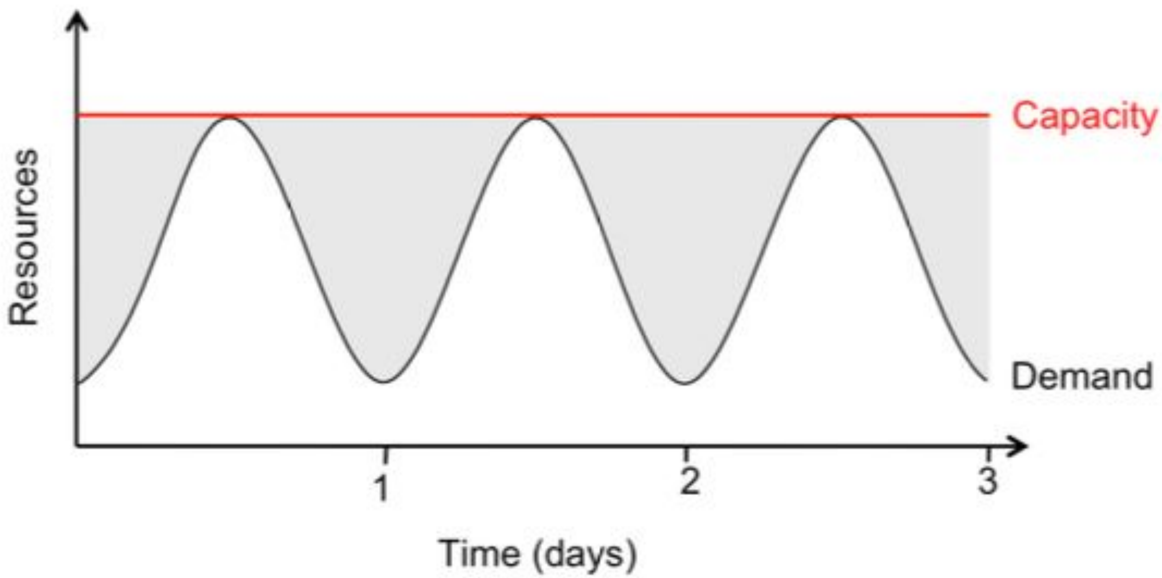
Cloud Economics: For Users

Pay-as-you-go (usage-based) pricing:

» Most services charge per minute, per byte, etc

» No minimum or up-front fee

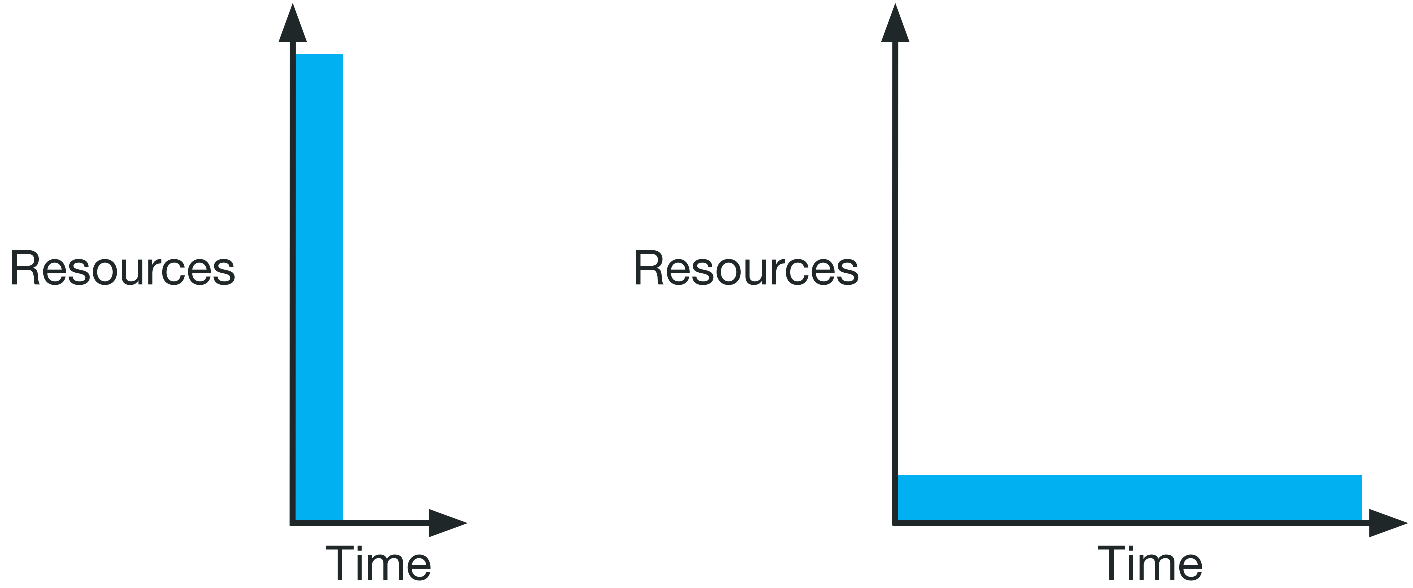
» Helpful when apps have *variable utilization*



Cloud Economics: For Users

Elasticity:

- Using 1000 servers for 1 hour costs the same as 1 server for 1000 hours
- Same price to get a result faster!

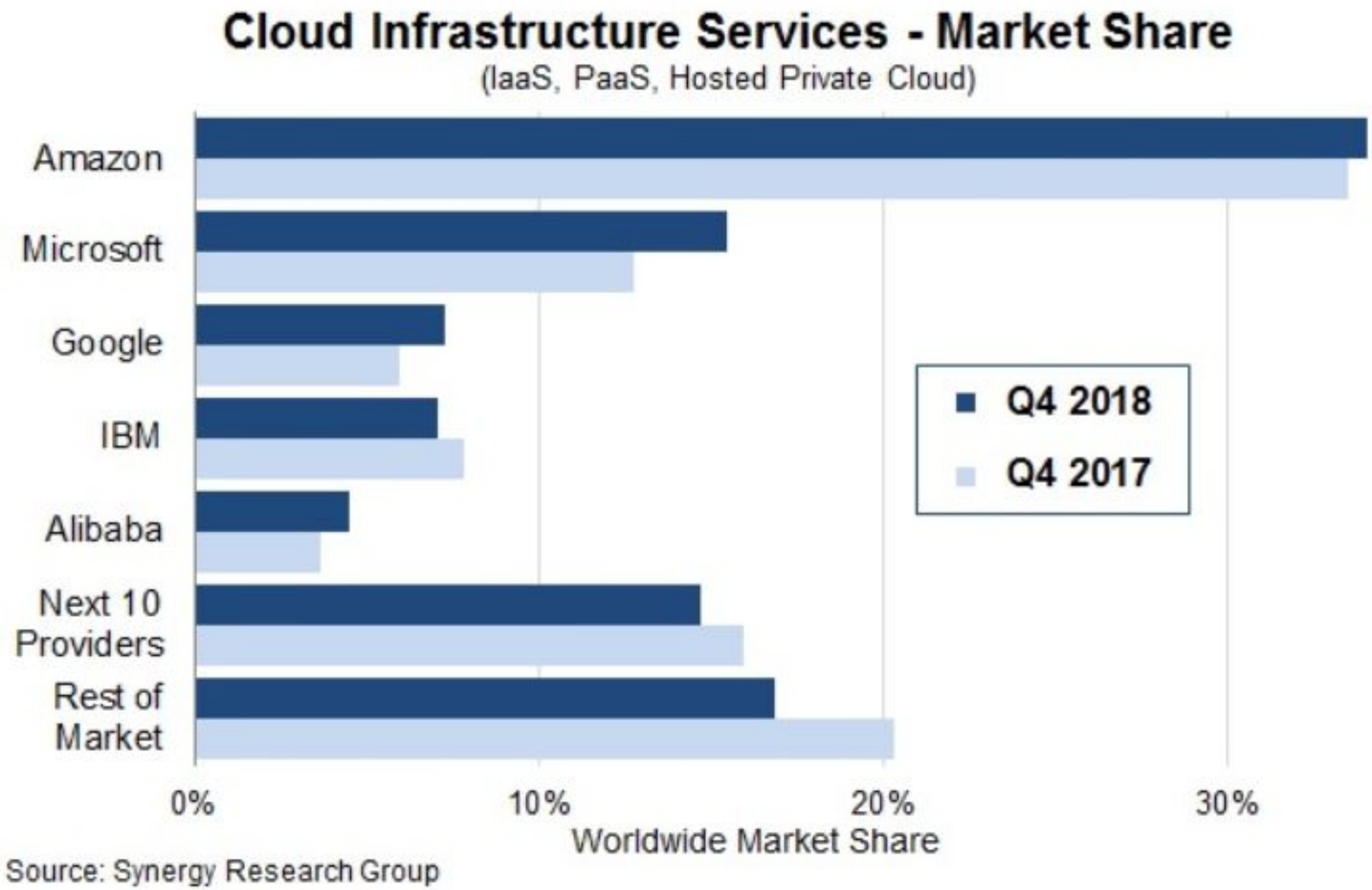


Cloud Economics: For Providers

Economies of scale:

- Purchasing, powering & managing machines at scale gives lower per-unit costs than customers’
- Tradeoff: fast growth vs efficiency
- Tradeoff: flexibility vs cost

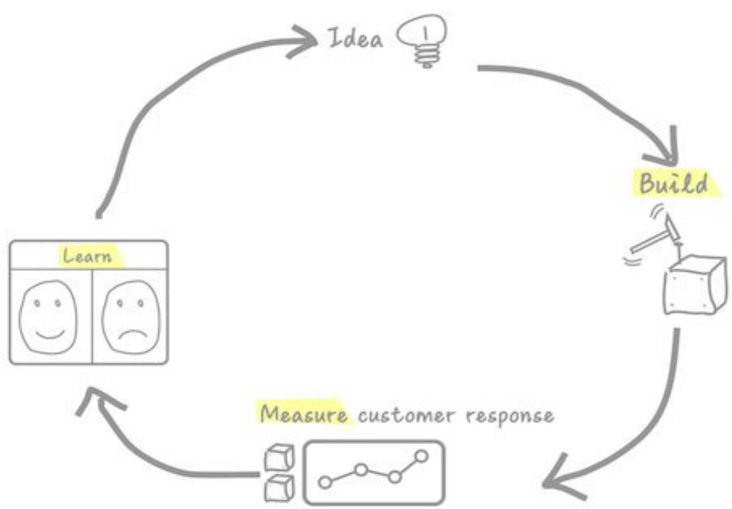
Cloud Market Research ..


Cloud Infrastructure Services - Market Share
(IaaS, PaaS, Hosted Private Cloud)
Amazon
Microsoft
Google
IBM
Alibaba
Next 10 Providers
Rest of Market
Q4 2018
Q4 2017
0%
10%
20%
30%
Worldwide Market Share
Source: Synergy Research Group

Cloud Economics: For Providers

Speed of iteration:

- Software as a service means fast time-to-market, updates, and detailed monitoring/feedback
- Compare to speed of iteration with ordinary software distribution



Questions

- Assume you are a cloud provider

  How do you avoid having many your customers spike at the time time?

Common Cloud Applications

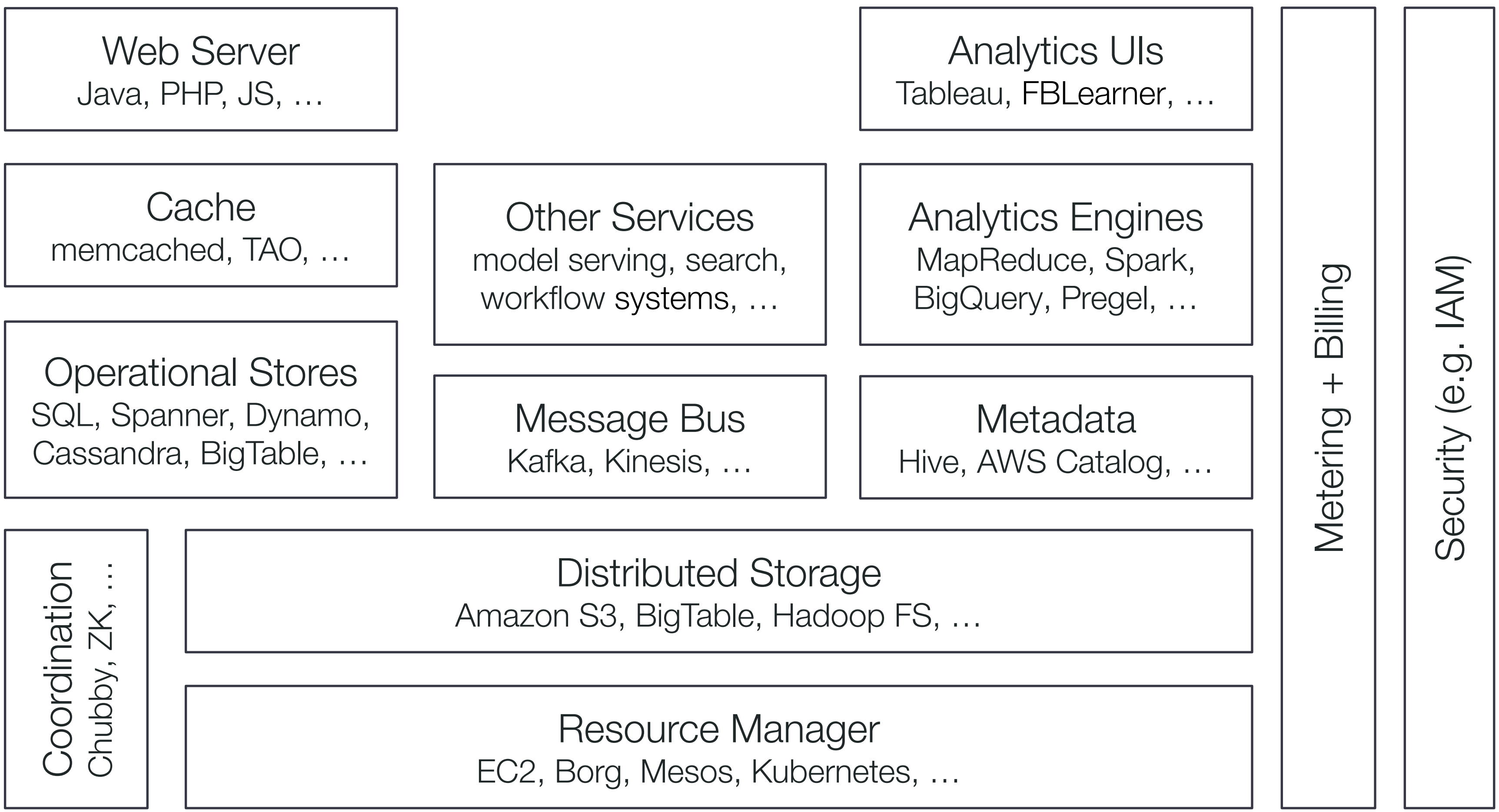
1. Web and mobile applications

2. Data analytics (MapReduce, SQL, ML, etc)

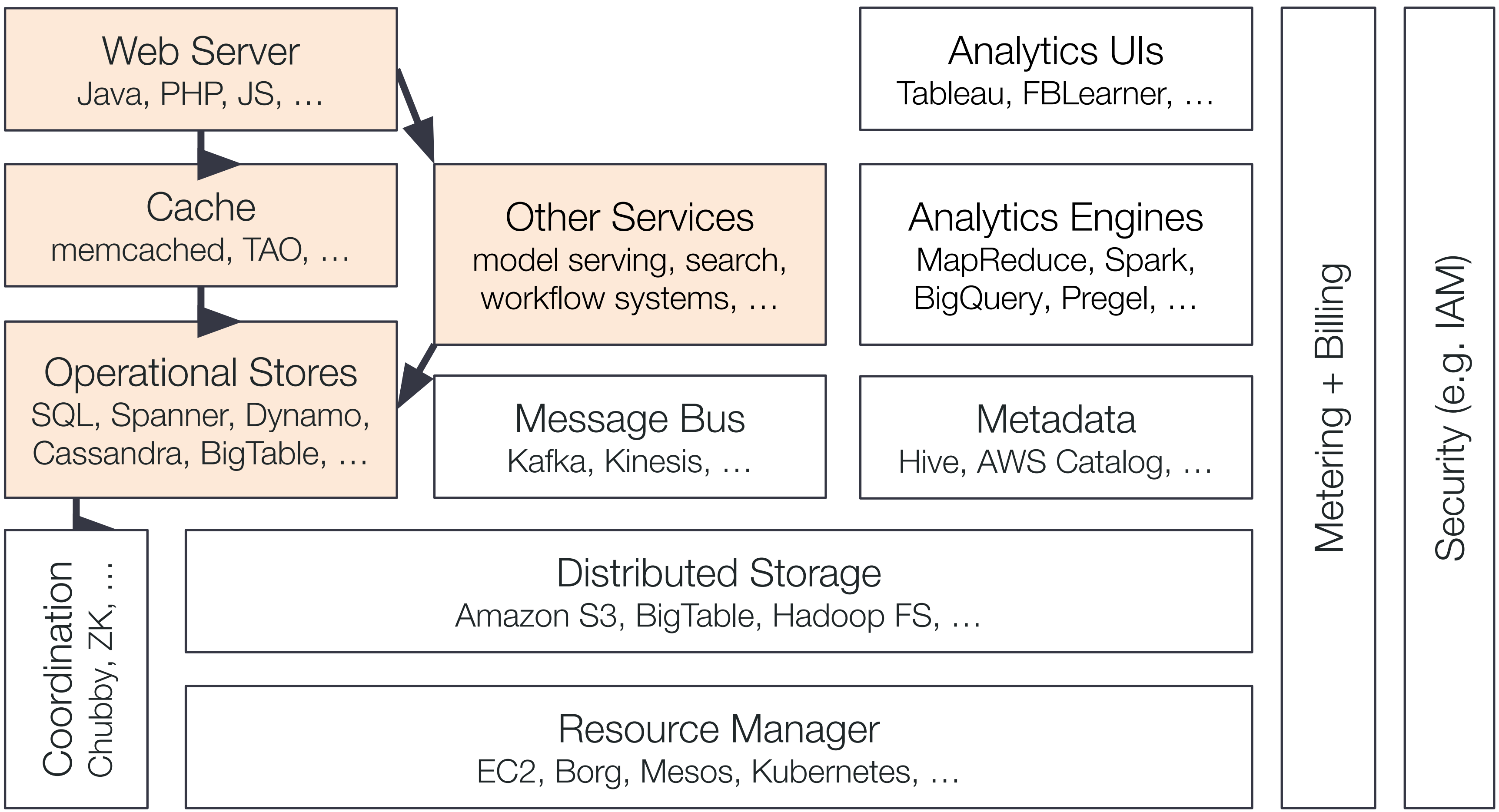
3. Stream processing

4. Batch computation (HPC, video, etc)

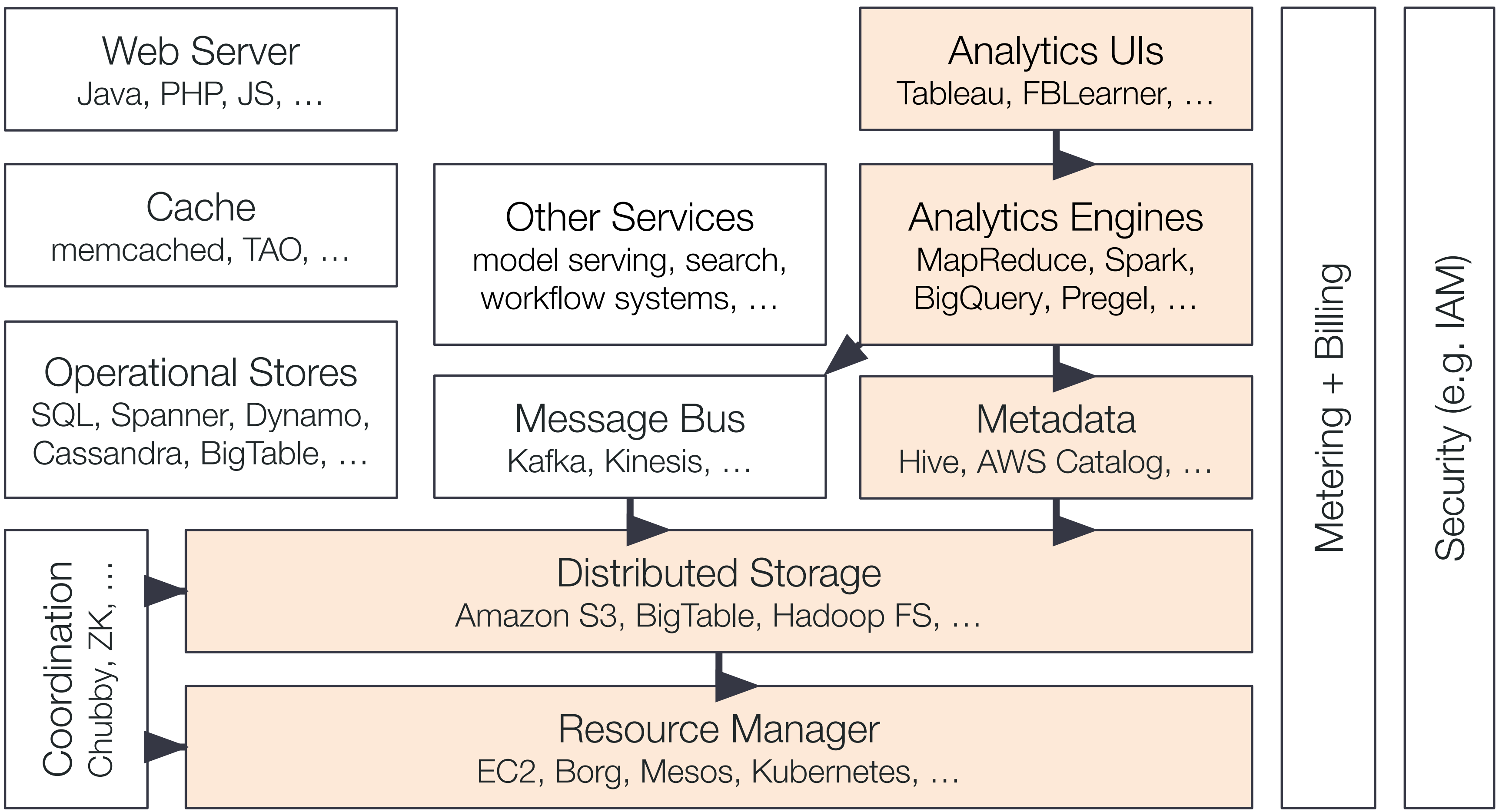
Cloud Software Stack



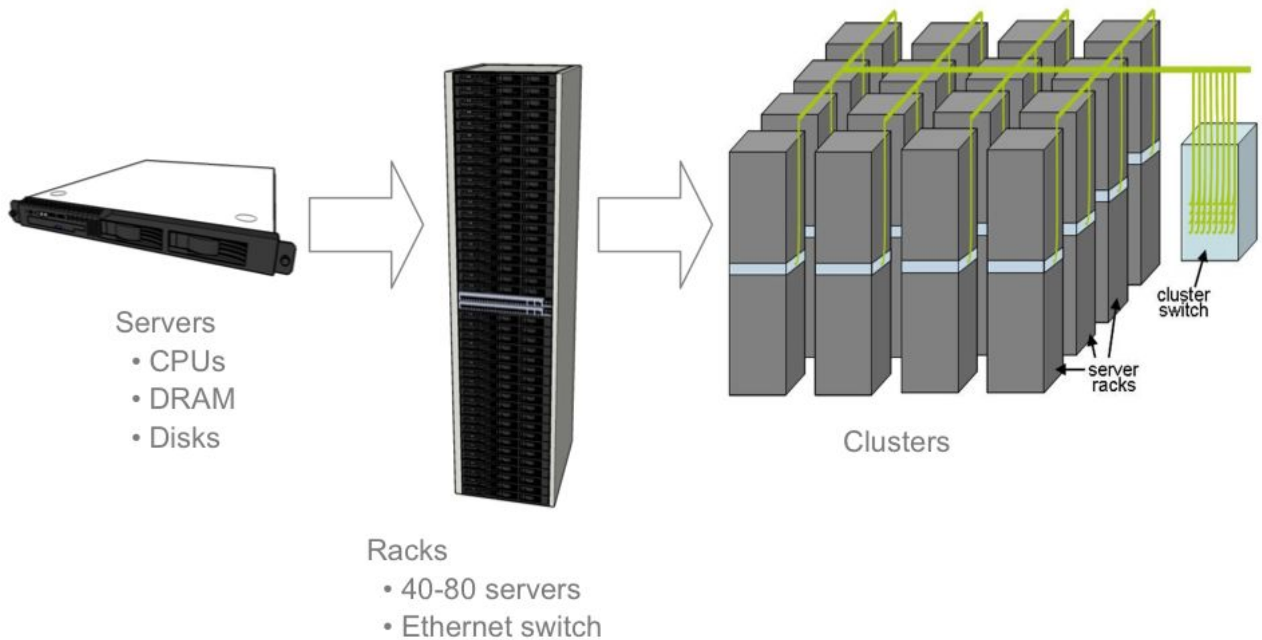
Example: Web Application



Example: Analytics Warehouse



Datacenter Hardware

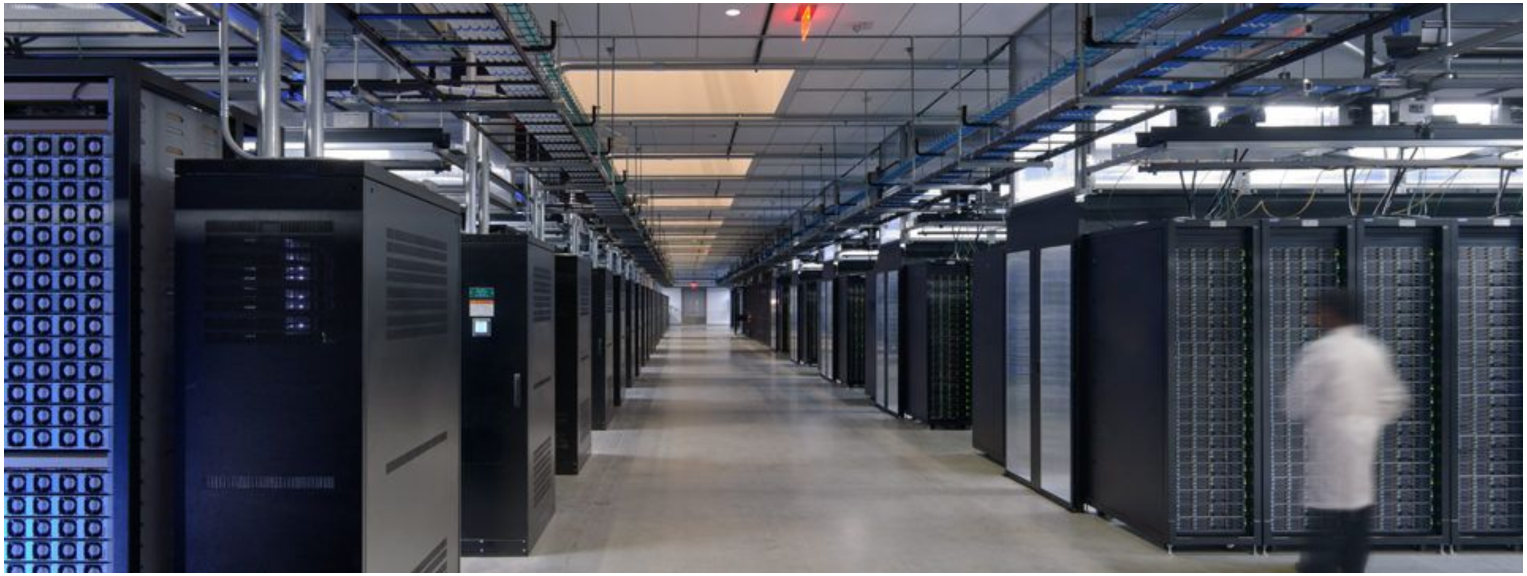


Rows of rack-mounted servers

Datacenter: 50 – 200K of servers, 10 – 100MW

Often organized as few and mostly independent clusters

Datacenter Example



Datacenter HW: Compute

The basics

Multi-core CPU servers

1 & 2 sockets

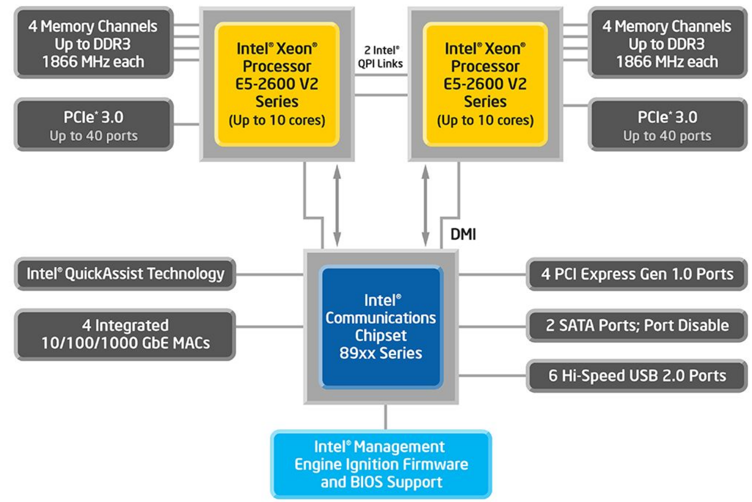
What's new

GPUs

FPGAs

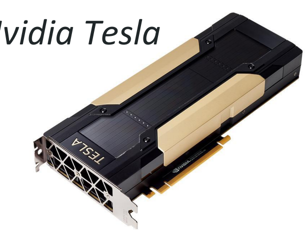
Custom accelerators (AI)

2-socket server

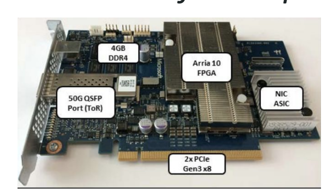


Google TPU²

Nvidia Tesla



Microsoft Catapult

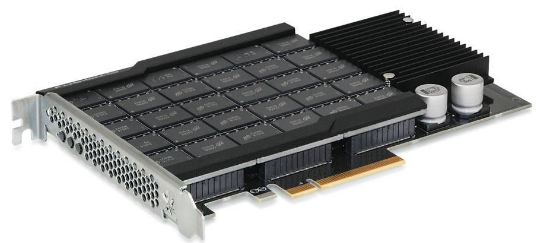


Datacenter HW: Storage

The basics

- Disk trays
- SSD & NVM Flash

NVMe Flash

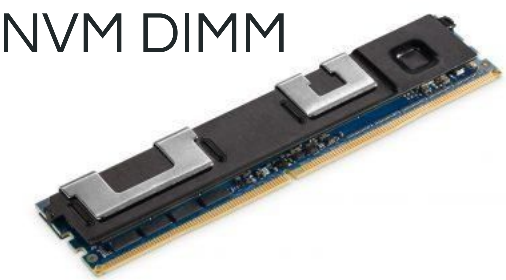


JBOD disk array

What's new

- Non-volatile memories
- New archival storage (e.g., glass)

NVM DIMM



Distributed with compute or NAS systems

- Remote storage access for many use cases (why?)

Datacenter HW: Networking

The basics

- 10, 25, and 40GbE NICs
- 40 to 100GbE switches
- Ciscos topologies

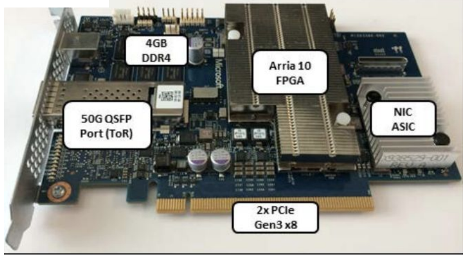
40GbE Switch

Smart NIC

What's new

- Software defined networking
- Smart NICs
- FPGAs

Microsoft Catapult



Useful Latency Numbers

Initial list from Jeff Dean, Google

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L3 cache reference	20 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K bytes with Snappy	3,000 ns
Send 2K bytes over 10Ge	2,000 ns
Read 1 MB sequentially from memory	100,000 ns
Read 4KB from NVMe Flash	50,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from disk	20,000,000 ns
Send packet CA → Europe → CA	150,000,000 ns

Useful Throughput Numbers

DDR4 channel bandwidth	20 GB/sec
PCIe gen3 x16 channel	15 GB/sec
NVMe Flash bandwidth	2GB/sec
GbE link bandwidth	10 – 100 Gbps
Disk bandwidth	6 Gbps
NVMe Flash 4KB IOPS	500K – 1M
Disk 4K IOPS	100 – 200

Performance Metrics

Throughput

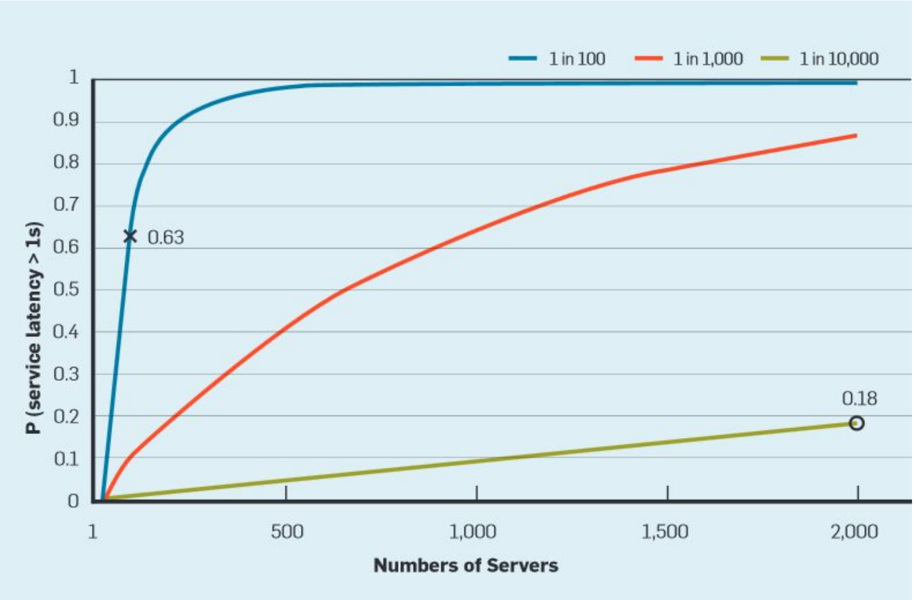
- Requests per second
- Concurrent users
- Gbytes/sec processed
- ...

Latency

- Execution time
- Per request latency

Tail Latency

[Dean & Barroso,'13]



The 95th or 99th percentile request latency

End-to-end with all tiers included

Larger scale → more prone to high tail latency

Total Cost of Ownership (TCO)

TCO = capital (CapEx) + operational (OpEx) expenses

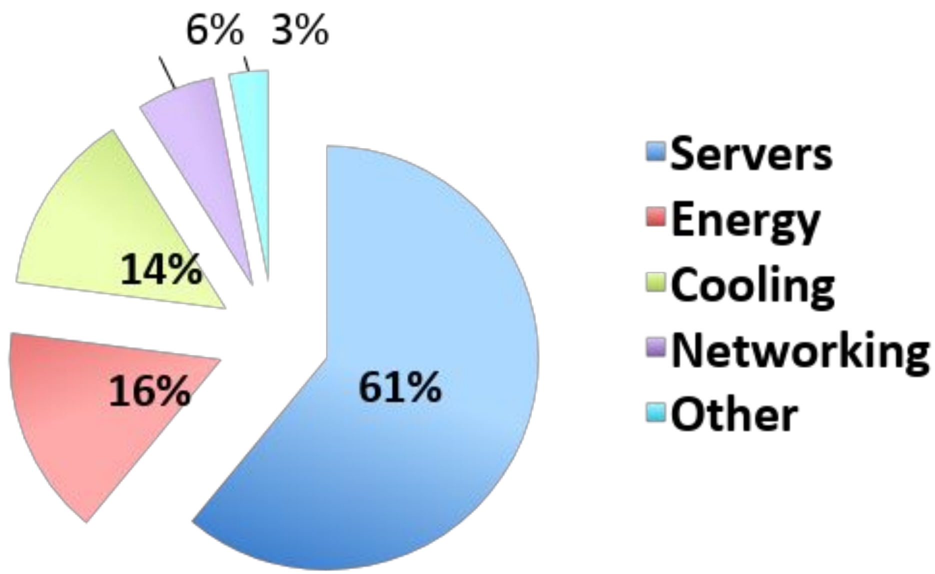
Operators perspective

- CapEx: building, generators, A/C, compute/storage/net HW
  - Including spares, amortized over 3 – 15 years
- OpEx: electricity (5-7c/KWh), repairs, people, WAN, insurance, …

Users perspective

- CapEx: cost of long term leases on HW and services
- OpeEx: pay per use cost on HW and services, people

Operator’s TCO Example



[Source: James Hamilton]

Hardware dominates TCO, make it cheap

Must utilize it as well as possible

Reliability

Failure in time (FIT)

- Failures per billion hours of operation = 10^9/MTTF

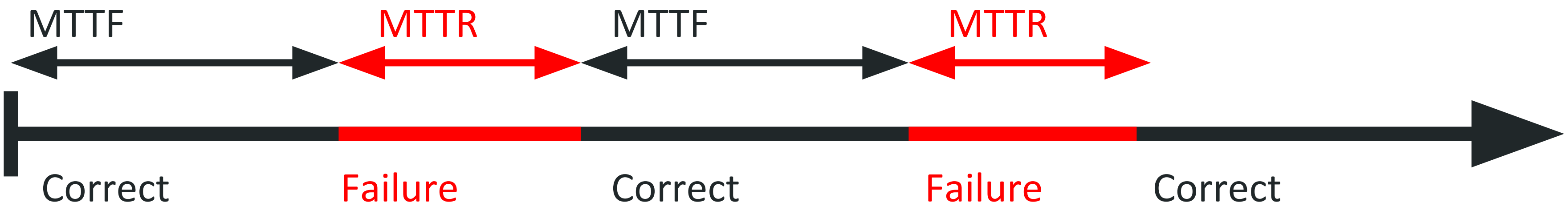
Mean time to failure (MTTF)

- Time to produce first incorrect output

Mean time to repair (MTTR)

- Time to detect and repair a failure

Availability



Steady state availability = MTTF / (MTTF + MTTR)

Yearly Datacenter Flakiness

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)

~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hrs to come back)

~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hrs)

~1 network rewiring (rolling ~5% of machines down over 2-day span)

~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)

~5 racks go wonky (40-80 machines see 50% packet loss)

~8 network maintenances (4 might cause ~30-minute random connectivity losses)

~12 router reloads (takes out DNS and external vIPs for a couple minutes)

~3 router failures (have to immediately pull traffic for an hour)

~dozens of minor 30-second blips for dns

~1000 individual machine failures (2-4% failure rate, machines crash at least twice)

~thousands of hard drive failures (1-5% of all disks will die)

Add to these SW bugs, config errors, human errors, …

Key Availability Techniques

Technique	Performance	Availability
Replication	✓	✓
Partitioning (sharding)	✓	✓
Load-balancing	✓	
Watchdog timers		✓
Integrity checks		✓
Canaries		✓
Eventual consistency	✓	✓

- Make apps do something reasonable when not all is right
  - Better to give users limited functionality than an error page
- Aggressive load balancing or request dropping
  - Better to satisfy 80% of the users rather than none