Cloud Computing

Gestão de Infraestruturas de Computação

deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

João Paulo Barraca

Current Trends

- Massive amounts of data
  - Petabyte is common for many business
- Thousands to millions of cores
  - Consolidated data centers
  - Shift from clock rate battle to multicore, to many core…
- Cheap, COTS hardware
- Failures are common, but not common to users
- Virtualization based systems
- Making accessible (Easy to use)
  - More people requiring large scale data processing

Current Trends

- Computing Clouds
  - Cloud Infrastructure Services
  - Cloud infrastructure Software
- Distributed File Systems (Distributed *)
- Data intensive parallel application frameworks
  - MapReduce
  - High level languages
- Science in the clouds
  - High Performance Computing (HPC)

Information Services Infrastructure

Some numbers (USA)

- 38 million physical servers
  - +700% growth in next 15 years
- $140b unused capacity
- 30%-50% server cost is related to power
- Average costs for a datacenter
  - $5K-$15K / sq meter
  - $2.5K to $20K / server
  - $80K to $700K / rack
- 20-30 : 1 – Server / Administrator ratio
  - … but can reach >1000:1
  - 1 server can have >200 VMs

Information Services Infrastructure

- Datacenters are not green!
  - 1 server = ~150W at average load
  - 1 rack, 32-42 servers = up to ~6.3KW (<4.8KW typical)
  - 1 DC, 50K servers = 7.5MW (for servers only!)

The result is HEAT, which must be removed out of the premises

- Power Usage Effectiveness
  - PUE = Total Energy / IT Energy
  - Currently: 1.2-3
  - 30% to 100% more in other devices (cooling, network, etc…)
  - >15% is simply lost

Some Power Estimates[1]

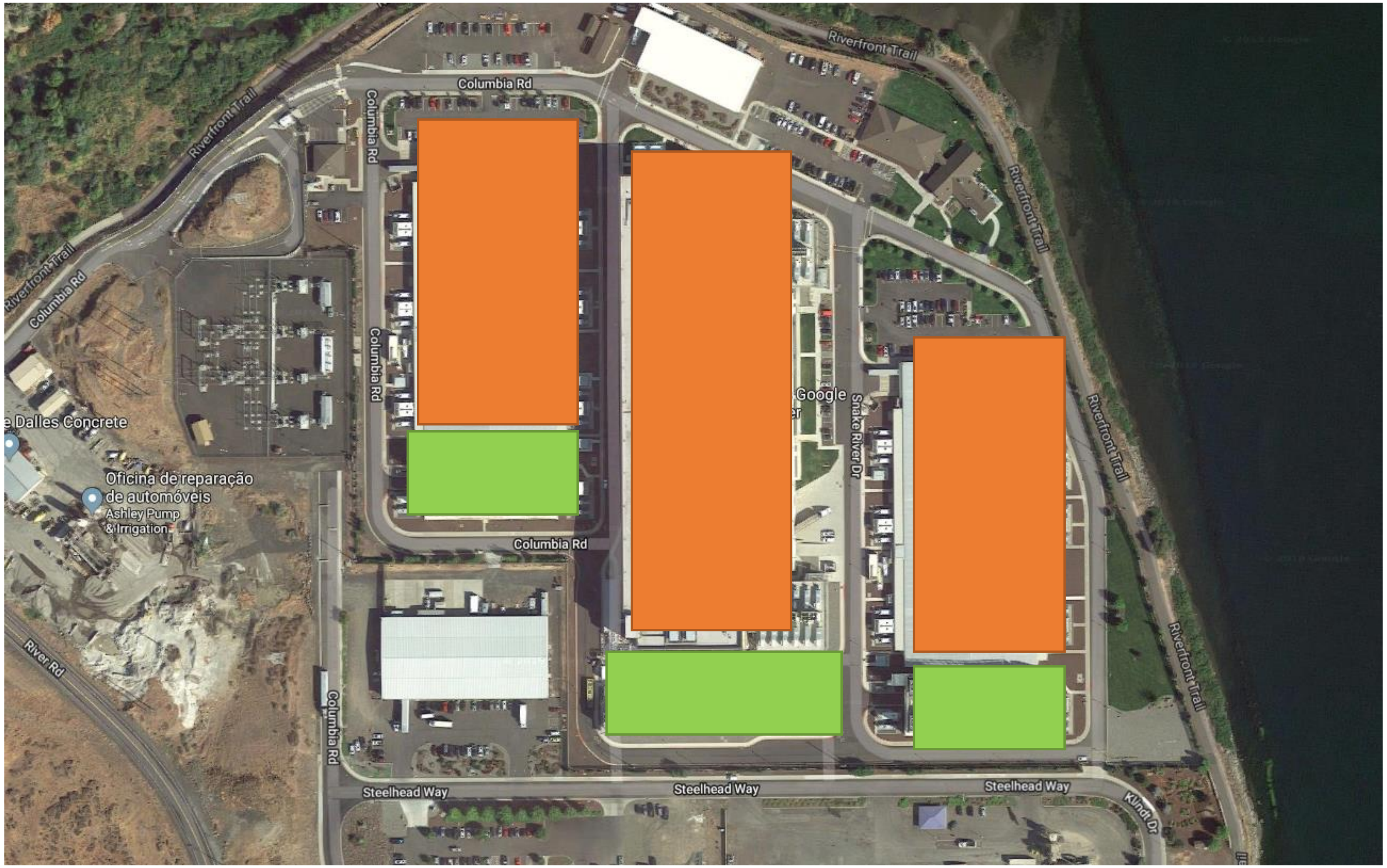
- Google: >1M servers, >400MW power
- Facebook: >240MW
- Amazon: >160MW
- Microsoft: >1M servers, >160MW
- Equinix: >740MW (in >175DCs)
- Total estimated : >400TW/h = 0.03% world power

(1) Ali Ghiasi, Overview of Largest Data Centers, IEEE, 2014

The Dalles

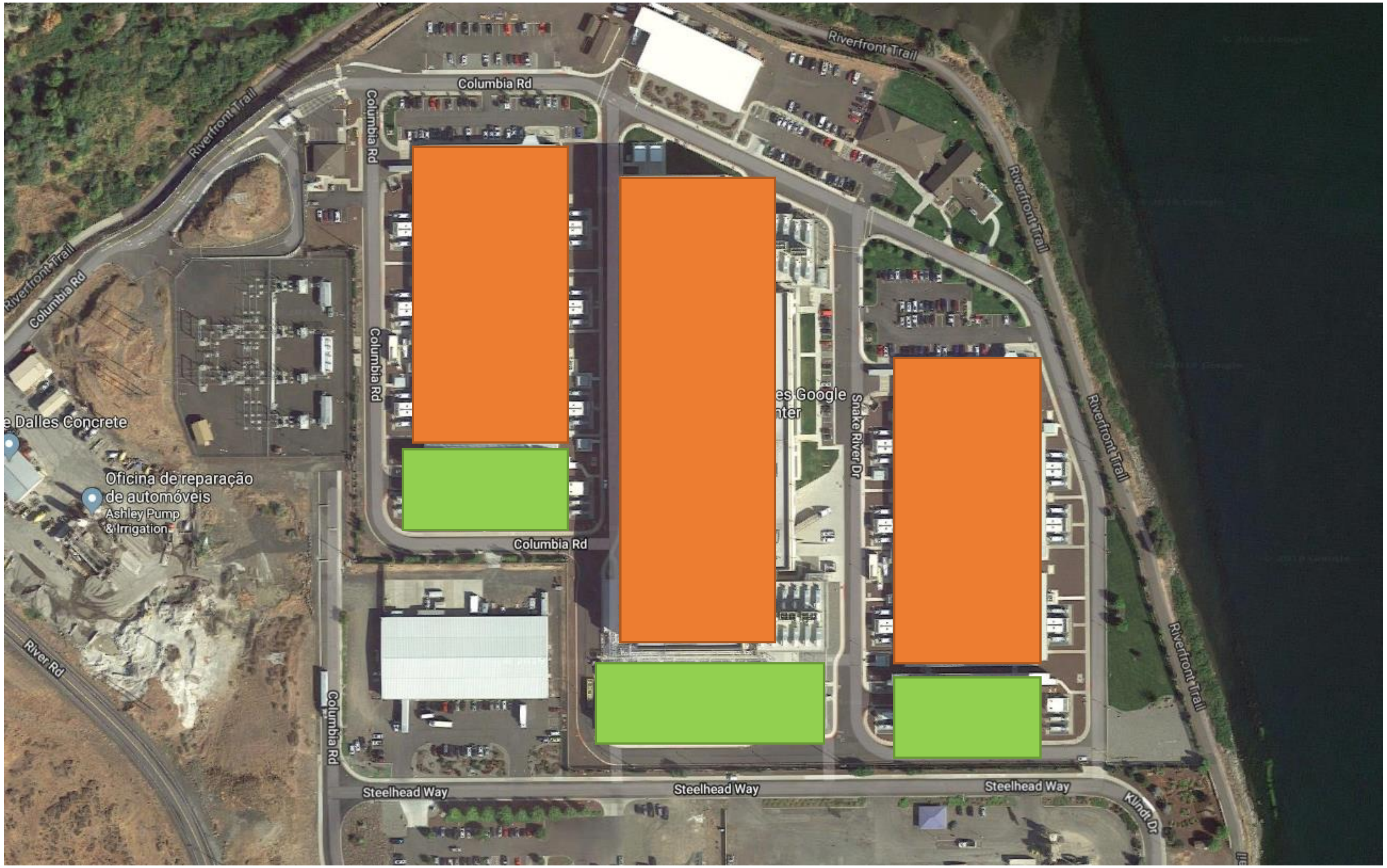

Riverfront Trail
Columbia Rd
Columbia Rd
Riverfront Trail
Riverfront Trail
Columbia Rd
Columbia Rd
e Dalles Concrete
Oficina de reparação
de automóveis
Ashley Pump
& Irrigation
The Dalles Google
Data Center
Snake River Dr
Riverfront Trail
Riverfront Trail
Riverfront Trail
Columbia Rd
River Rd
Columbia Rd
Steelhead Way
Steelhead Way
Steelhead Way
Klindt Dr

The Dalles



Columbia Rd
Riverfront Trail
Google
Snake River Dr
e Dalles Concrete
Oficina de reparação de automóveis
Ashley Pump & Irrigation
River Rd
Steelhead Way
Klindt Dr

The Dalles


Riverfront Trail
Columbia Rd
Columbia Rd
Riverfront Trail
Riverfront Trail
Columbia Rd
Columbia Rd
e Dalles Concrete
Oficina de reparação
de automóveis
Ashley Pump
& Irrigation
Columbia Rd
River Rd
Columbia Rd
Riverfront Trail
Snake River Dr
Riverfront Trail
Riverfront Trail
Steelhead Way
Steelhead Way
Steelhead Way
Klindt Dr

Scalability

- Vertical Scaling: Add more power to a server
  - More RAM, more storage, more CPUs
- Horizontal Scaling: Add more servers
  - Homogeneous or not
  - Usually not homogeneous as servers are replaced in chunks
- Datacenters are designed to scale horizontally
  - Adding more sections, with more servers

Scalability

- Systems are designed to scale locally and globally
  - Increase reliability
  - Increase performance
  - Reduce Cost

- Local Scaling: Distribute resource usage in same DC
- Global Scaling: Distribute resource usage across world

Dimensioning

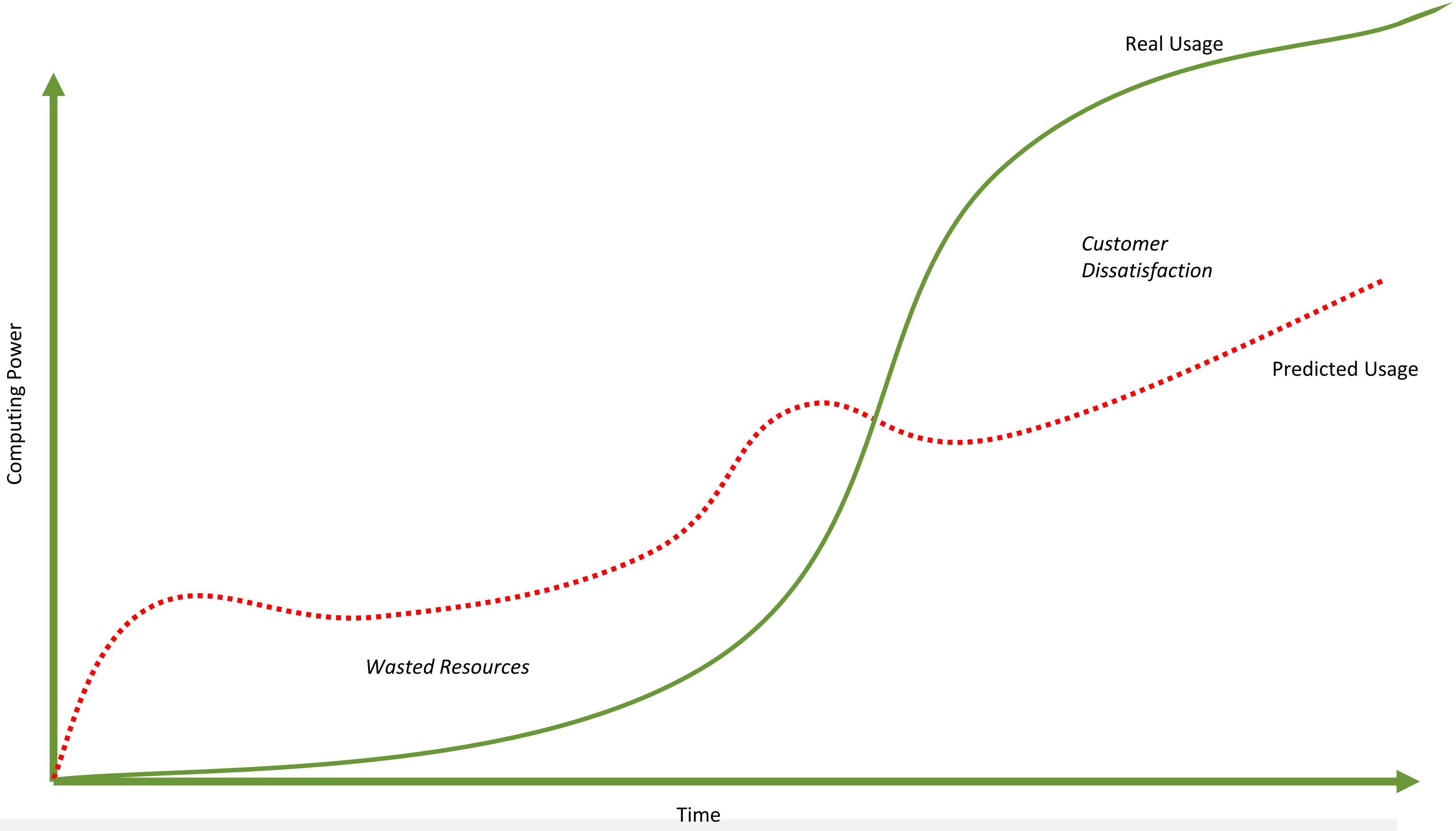
- Current landscape is too dynamic and unpredictable

- Provisioning for average user load will fail at peak time
  - Weekends, Holidays, Black Friday

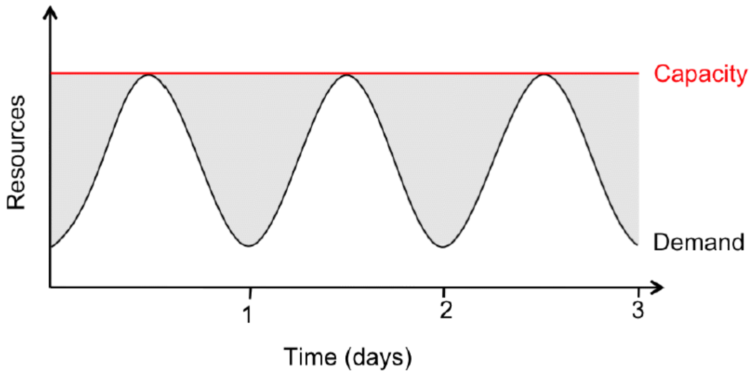
- Provisioning for peak time results in a huge waste
  - Peak should reach 80% capacity at most

- What about flash peaks?
  - Viral content, Promotions, Popular content on Twitter, Reddit, FB

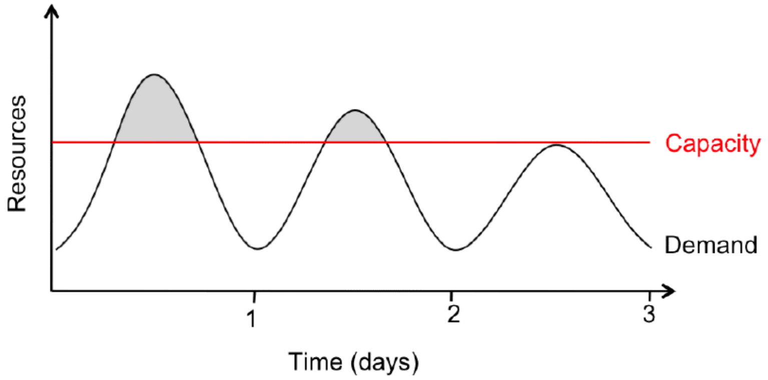
Dimensioning



Problem #1: Difficult to dimension



Provisioning for the peak load



Provisioning below the peak

- Problem: Load can vary considerably
  – Peak load can exceed average load by factor 2x-10x [Why?]
  – But: Few users deliberately provision for less than the peak
  – Result: Server utilization in existing data centers ~5%-20%!!
  – Dilemma: Waste resources or lose customers!

Problem #2: Expensive

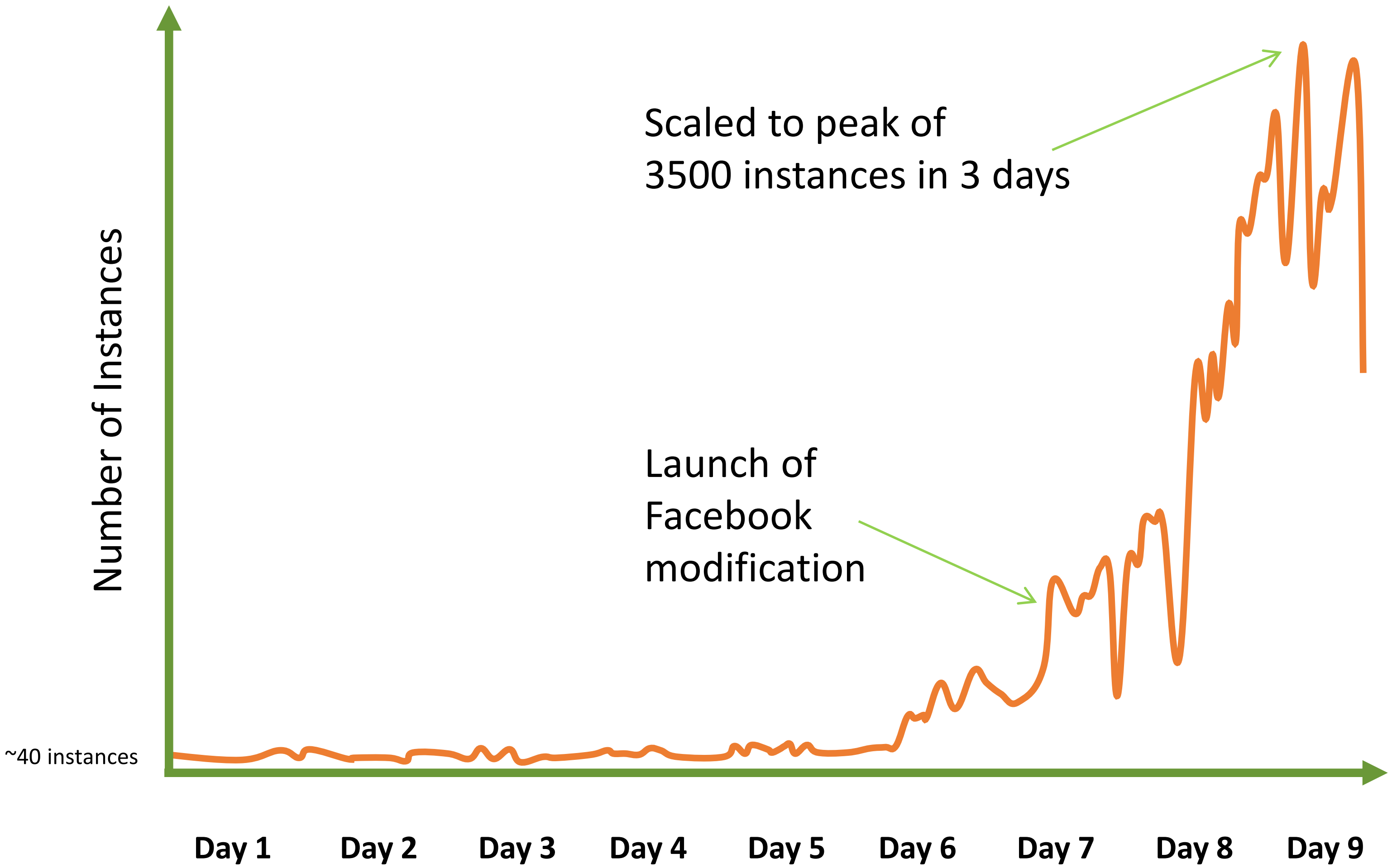
- Need to invest many $$$ in hardware
  - Even a small cluster can easily cost $100,000
  - Google The Dalles: 1.8B$
- Need expertise
  - Planning and setting up a large cluster is highly nontrivial
  - Cluster may require special software, etc.
- Need maintenance
  - Someone needs to replace faulty hardware, install software upgrades, maintain user accounts, ...

Problems #3: Difficult to Scale

- Scaling up is difficult
  - Need to order new machines, install them, integrate with existing cluster - can take months!
  - Large scaling factors may require major redesign, e.g., new storage system, new interconnect, new building (!)

- Scaling down is difficult
  - What to do with superfluous hardware?
  - Server idle power is about 60% of peak $\rightarrow$ Energy is consumed even when no work is being done
  - Many fixed costs, such as construction

Use Case: Animoto



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

Case Studies: Medical Research

- Novartis Institutes for Biomedical Research
  - focused on the drug discovery phase of the ~10 year / $1 billion drug development process
- 2013: ran a project to screen 10 M compounds against a common cancer target
- Compute requirements >> internal capacity / $
- Project ran across 10,500 EC2 Spot instances (~87,000 cores) for $4,232 in 9 hours <u>(peanuts)</u>
- **Equiv. of 39 years** of computational chemistry

Problem #4: Availability is hard

- No single computer can handle today's workloads
  - The Growth of Ebay: https://bit.ly/2BG8FBB
- No single computer can provide high availability
  - Hard disk replacements, upgrades, hardware failure?
- Typical availability
  - 99.999% uptime=5.26 minutes downtime per year
  - 99.9999% uptime = 31.8 seconds downtime per year
- Availability is highly demanded
  - Google failed? What?

Summary

- Modern applications require huge amounts of processing and data
  - Measured in petabytes, millions of users, billions of objects
  - Need special hardware, algorithms, tools to work at this scale

- Clusters and data centers can provide the resources we need
  - Main difference: Scale (room-sized vs. building-sized)
  - Special hardware; power and cooling are big concerns

- Clusters and data centers are not perfect
  - Difficult to dimension; expensive; difficult to scale

Cloud Computing

- Web and Internet based on **on demand** computational services
- Infrastructure complexity **transparent** to end user
- **Horizontal scaling** with no additional delay
  – Increased throughput
- Public Clouds
  – Amazon Web Services, Windows Azure, Google AppEngine, …
- Private Cloud Infrastructure Software
  – Eucalyptus, Nimbus, OpenNebula, OpenStack, Kubernetes,

Cloud Computing

- Running a DataCenter is expensive.
  - Costs too much to built (CapEx)
  - Costs too much to run (OpEx)

“Need milk? Don’t buy the cow… buy the milk”

- Rent what you need instead of buying and running everything!
- Cloud Computing advantages:
  - Pay per use
  - Instant Scalability
  - Security
  - Reliability
  - APIs

Cloud Computing

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. "

Everything As a Service

SaaS

- Salesforce, Google Apps, MS Office 365

PaaS

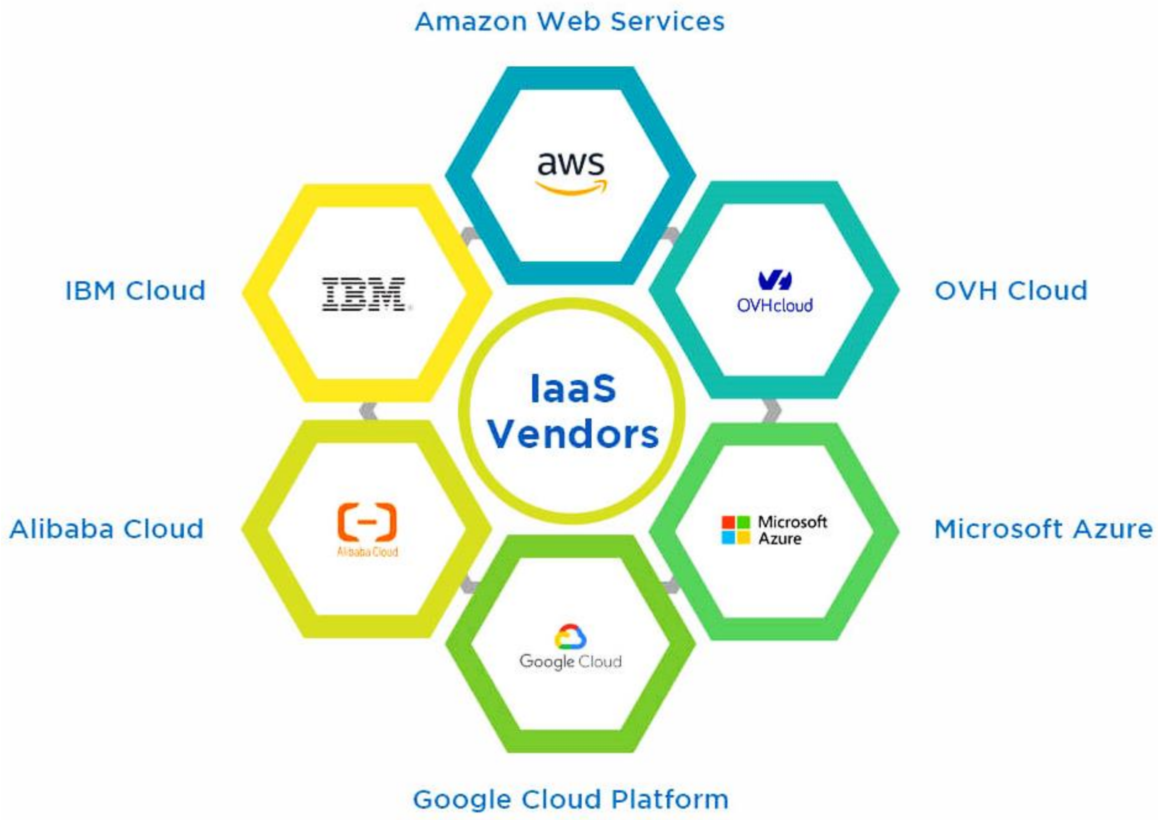
- MS Azure, Google App Engine, Heroku

IaaS

- Amazon, Google Cloud Platform, IBM Bluemix

IaaS: Infrastructure As A Service

- Grids of virtualized servers, storage & networks
  - E.g. Amazon (EC2, S3, EBS), IBM Bluemix, Google Cloud Platform
- Access to infrastructure stack:
  - Full OS access
  - Firewalls
  - Routers
  - Load balancing
- Advantages
  - Pay per use
  - Instant Scalability
  - Security
  - Reliability
  - APIs



PaaS: Platform as a Service

- **The abstraction of applications from traditional limits of hardware**
  - allowing developers to focus on application development
  - and not worry about operating systems, infrastructure scaling, load balancing and so on.
  - Examples include Google App Engine (Java, Python), MS Azure (.net), Heroku (RoR)
- **Platform delivery model**
  - Platforms are built upon Infrastructure, which is expensive
  - Estimating demand is not a science!
  - Platform management is not fun!
- **Advantages**
  - Pay per use
  - Instant Scalability
  - No sysadmin tasks
  - Better Security

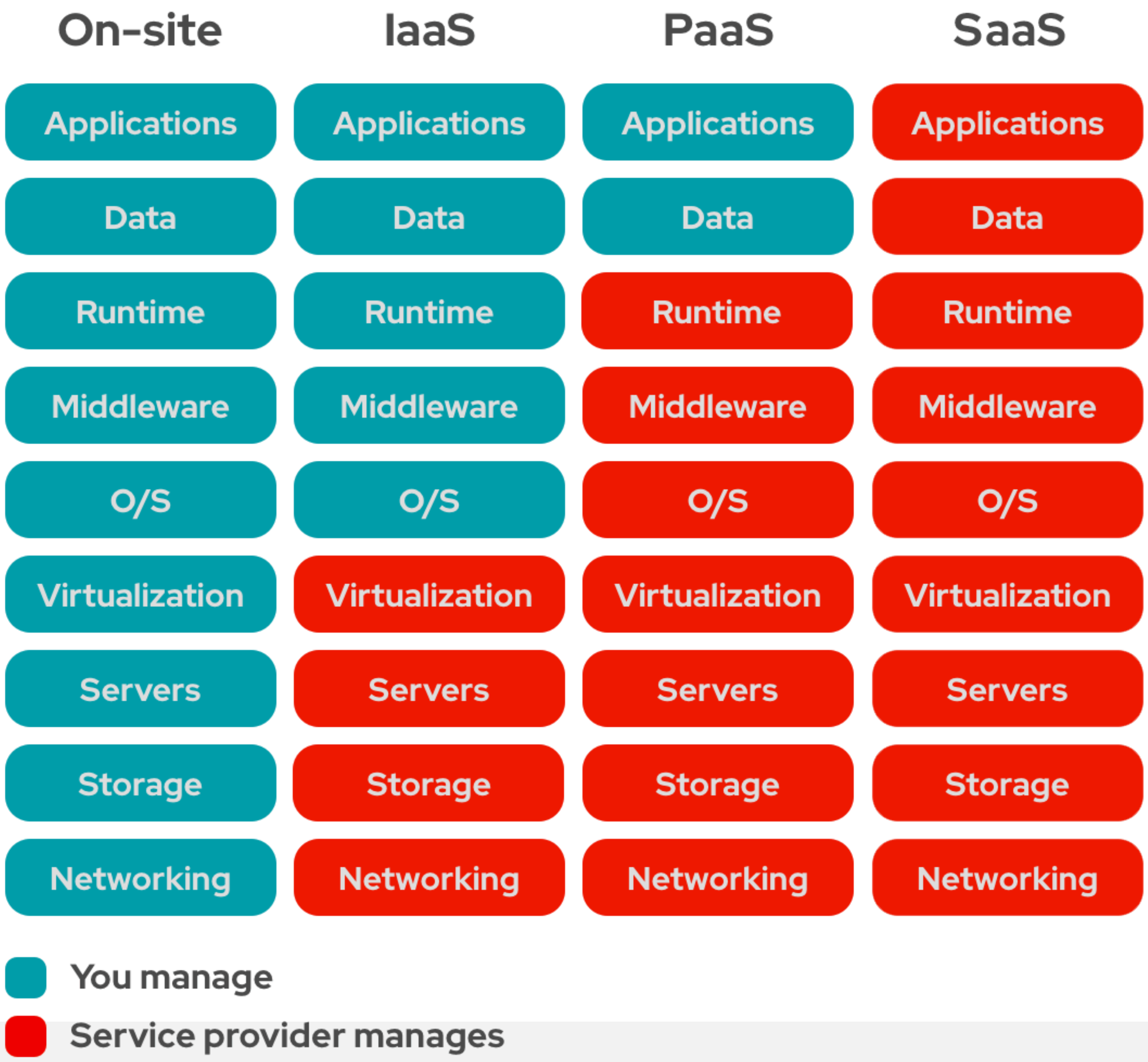
SaaS: Software as a Service

- **Applications with a Web-based interface accessed via Web Services and Web**
  - E.g. Google Apps, SalesForce.com and social network applications such as Facebook
- **Software delivery model**
  - Increasingly popular with SMEs
  - No hardware or software to manage
  - Service delivered through a browser
- **Advantages**
  - No Installation Required
  - Not platform specific
  - Automatic Upgrades
  - Access your data anywhere

Other

- Cloud as a Service
- Network as a Service
- Storage as a Service
- AI as a Service
- Energy Storage as a Service
- Security as a Service
- ...https://en.wikipedia.org/wiki/As_a_service

https://www.redhat.com/pt-br/topics/cloud-computing/iaas-vs-paas-vs-saas



Cloud Types

Cloud is presented with different flavors

- Public cloud: Commercial service; open to (almost) anyone
  - Example: Amazon AWS, Microsoft Azure, Google App Engine
- Community cloud: Shared by several similar organizations.
  - Example: Google's "Gov Cloud"
- Private cloud: Shared within a single organization.
  - Example: Internal datacenter of a large company.