

Cloud Computing

Riccardo Torlone
Università Roma Tre



Credits: A. Haeberlen, Z. Ives (University of Pennsylvania)

Computing at scale

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

Scaling up



PC



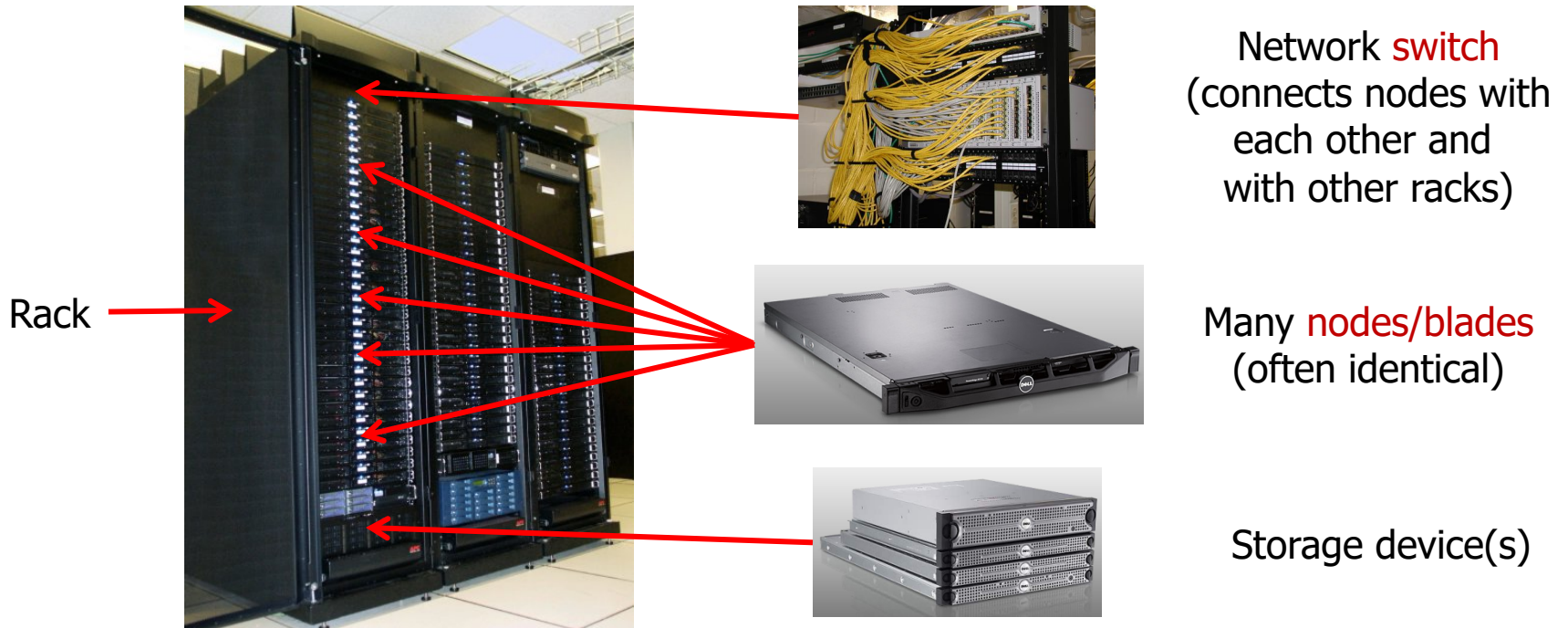
Server



Cluster

- What if one computer is not enough?
 - Buy a bigger (server-class) computer
- What if the biggest computer is not enough?
 - Buy many computers

Clusters



- Characteristics of a cluster:
 - Many similar machines, close interconnection (same room?)
 - Often special, standardized hardware (racks, blades)
 - It can be owned and used by a single organization

Power and cooling

- Clusters need lots of power
 - Example: 140 Watts per server
 - Rack with 32 servers: 4.5kW (needs special power supply!)
 - Most of this power is converted into heat
- Large clusters need massive cooling
 - 4.5kW is about 3 space heaters
 - And that's just one rack!



Further scaling up



PC



Server



Cluster



Data center

- What if your cluster is too big (hot, power hungry) to fit into your office building?
 - Build a separate building for the cluster
 - Building can have lots of cooling and power
 - Result: Data center

What does a data center look like?

Data centers
(size of a
football field)

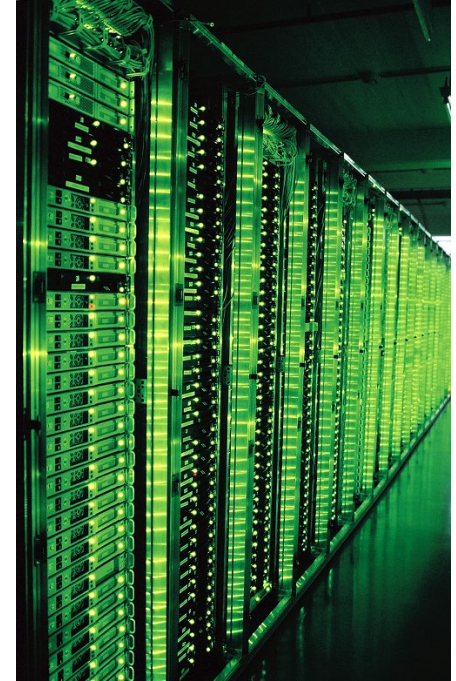
Cooling
plant



Google data center in The Dalles, Oregon

- A warehouse-sized computer
 - A single data center can easily contain 10,000 racks with 100 cores in each rack (1,000,000 cores total)

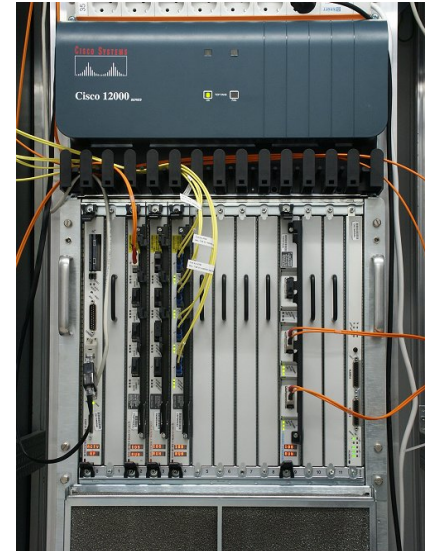
What's in a data center?



Source: 1&1

- Hundreds or thousands of racks

What's in a data center?



Source: 1&1

- Massive networking

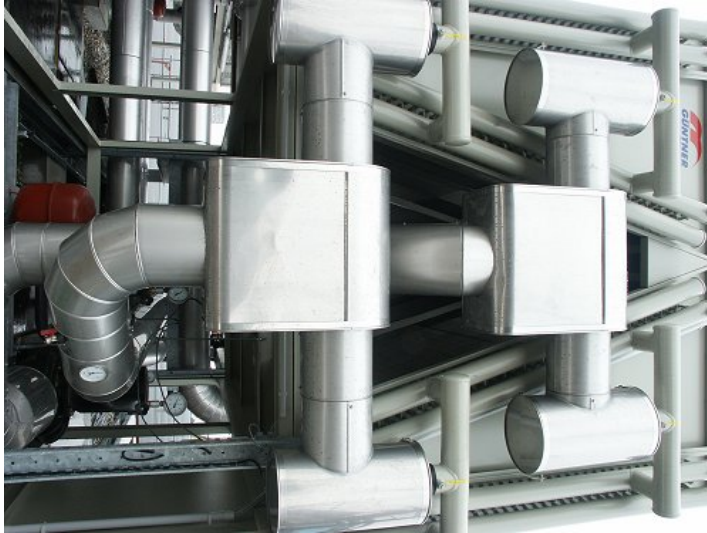
What's in a data center?



Source: 1&1

- Emergency power supplies

What's in a data center?



Source: 1&1

- Massive cooling

Energy matters!

Company	Servers	Electricity	Cost
eBay	16K	$\sim 0.6 \times 10^5$ MWh	$\sim \$3.7\text{M/yr}$
Akamai	40K	$\sim 1.7 \times 10^5$ MWh	$\sim \$10\text{M/yr}$
Rackspace	50K	$\sim 2 \times 10^5$ MWh	$\sim \$12\text{M/yr}$
Microsoft	>200K	$> 6 \times 10^5$ MWh	$> \$36\text{M/yr}$
Google	>500K	$> 6.3 \times 10^5$ MWh	$> \$38\text{M/yr}$
USA	10.9M	610×10^5 MWh	$\$4.5\text{B/yr}$

Source: Qureshi et al., SIGCOMM 2009

- Data centers consume a lot of energy
 - Makes sense to build them near sources of cheap electricity
 - Example: Price per KWh is 10.1ct in Idaho (near hydroelectric power), 18.5ct in California (long distance transmission), 30.8ct in Hawaii (must ship fuel)
 - Most of this is converted into heat → Cooling is a big issue!

Scaling up



PC



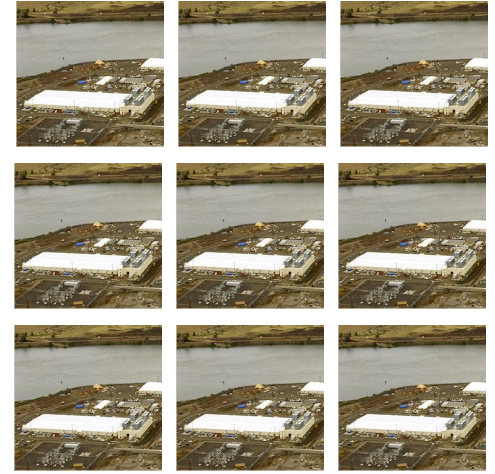
Server



Cluster



Data center



Network of data centers

- What if even a data center is not big enough?
 - Build additional data centers
 - Where? How many?

Global distribution

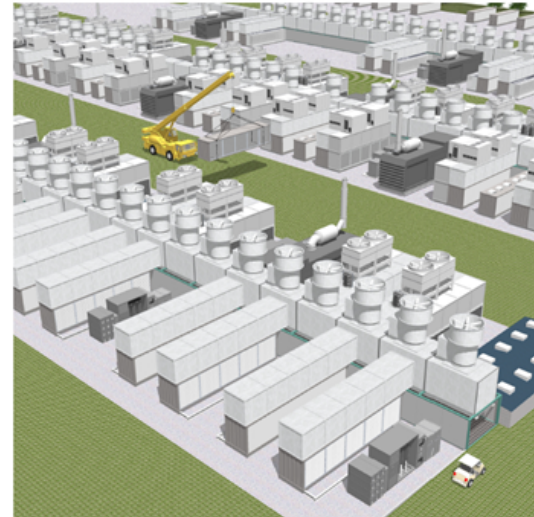


- Data centers are often globally distributed
 - Example above: Google data center locations (inferred)
- Why?
 - Need to be close to users (physics!)
 - Cheaper resources
 - Protection against failures

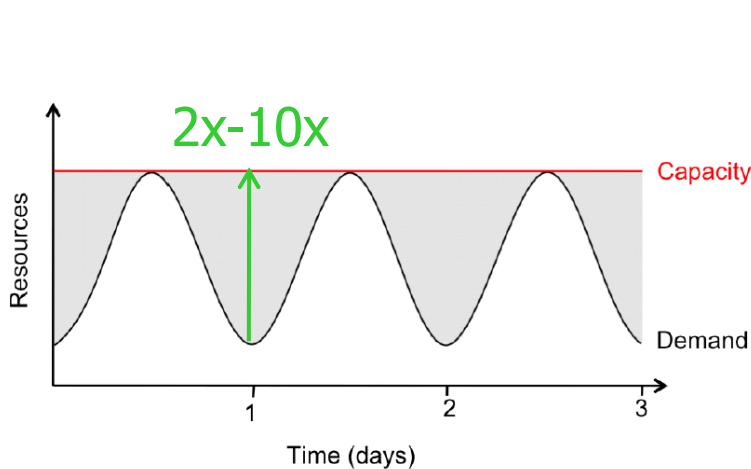
Trend: Modular data center



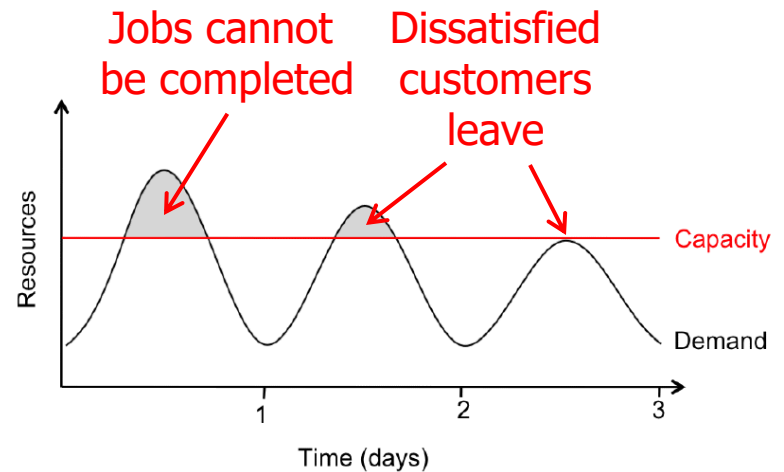
- Need more capacity? Just deploy another container!



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
 - But: Few users deliberately provision for less than the peak
 - Result: Server utilization in existing data centers $\sim 5\%-20\%!!$
 - Dilemma: Waste resources or lose customers!

Problem #2: Expensive

- Need to invest many \$\$\$ in hardware
 - Even a small cluster can easily cost \$100,000
 - Microsoft recently invested \$499 million in a single data center
- 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, ...



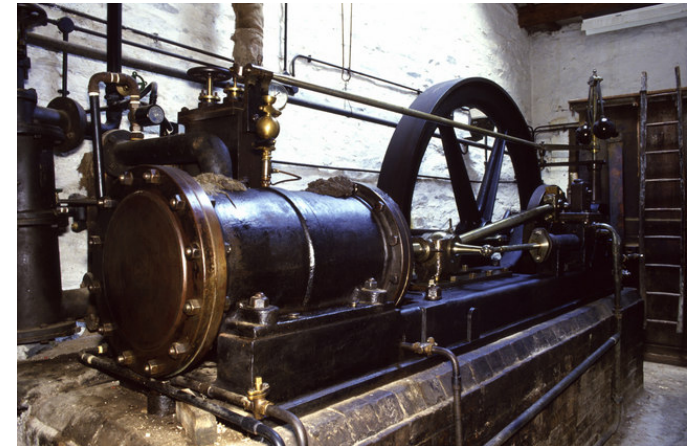
Problem #3: Difficult to scale

- Scaling up is difficult
 - Need to order new machines, install them, integrate with existing cluster - can take weeks
 - 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 → Energy is consumed even when no work is being done
 - Many fixed costs, such as construction

Solution: the power plant analogy



Waterwheel at the Neuhausen ob Eck Open-Air Museum



Steam engine at Stott Park Bobbin Mill

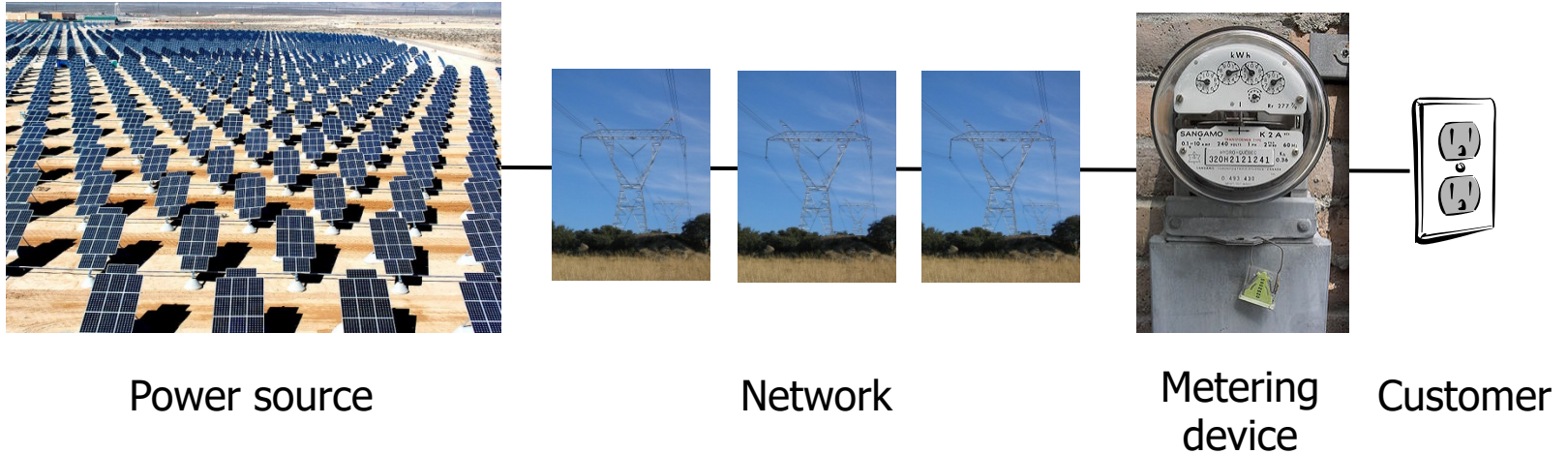
- It used to be that everyone had their own power source
 - Challenges are similar to the cluster: Needs large up-front investment, expertise to operate, difficult to scale up/down...

Scaling the power plant



- Then people started to build large, centralized power plants with very large capacity...

Metered usage model



- Power plants are connected to customers by a network
- Usage is metered, and everyone (basically) pays only for what they actually use

Why is this a good thing?

Electricity

- Economies of scale
 - Cheaper to run one big power plant than many small ones
- Statistical multiplexing
 - High utilization!
- No up-front commitment
 - No investment in generator; pay-as-you-go model
- Scalability
 - Thousands of kilowatts available on demand; add more within seconds



Computing

Cheaper to run one big data center than many small ones

High utilization!

No investment in data center; pay-as-you-go model

Thousands of computers available on demand; add more within seconds

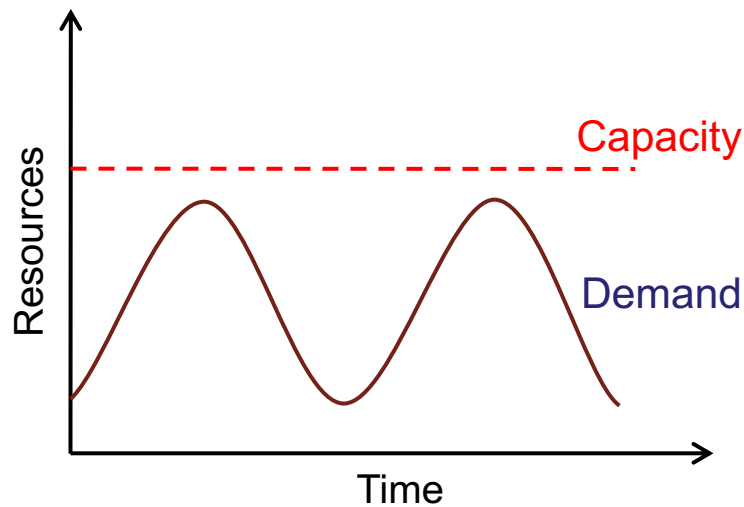
So what is Cloud Computing, really?

- According to NIST:

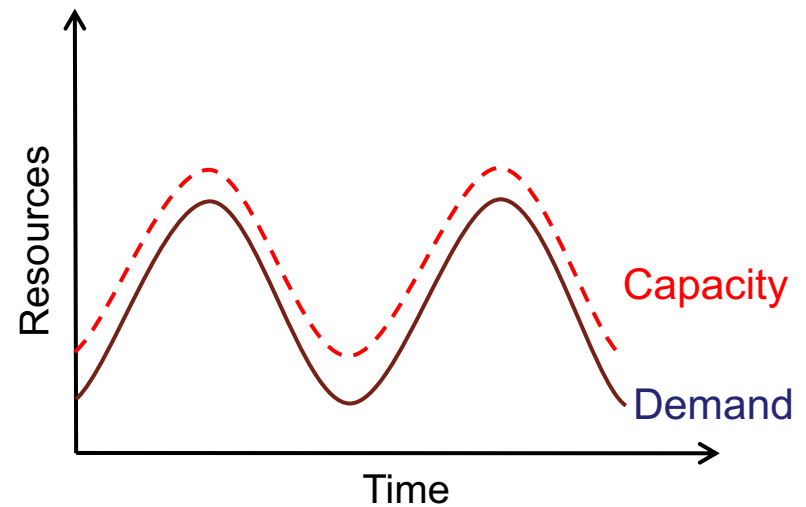
Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

- Essential characteristics:
 - On-demand self service
 - Broad network access
 - Resource pooling
 - Rapid elasticity
 - Measured service

Using a data center in the cloud



Static data center

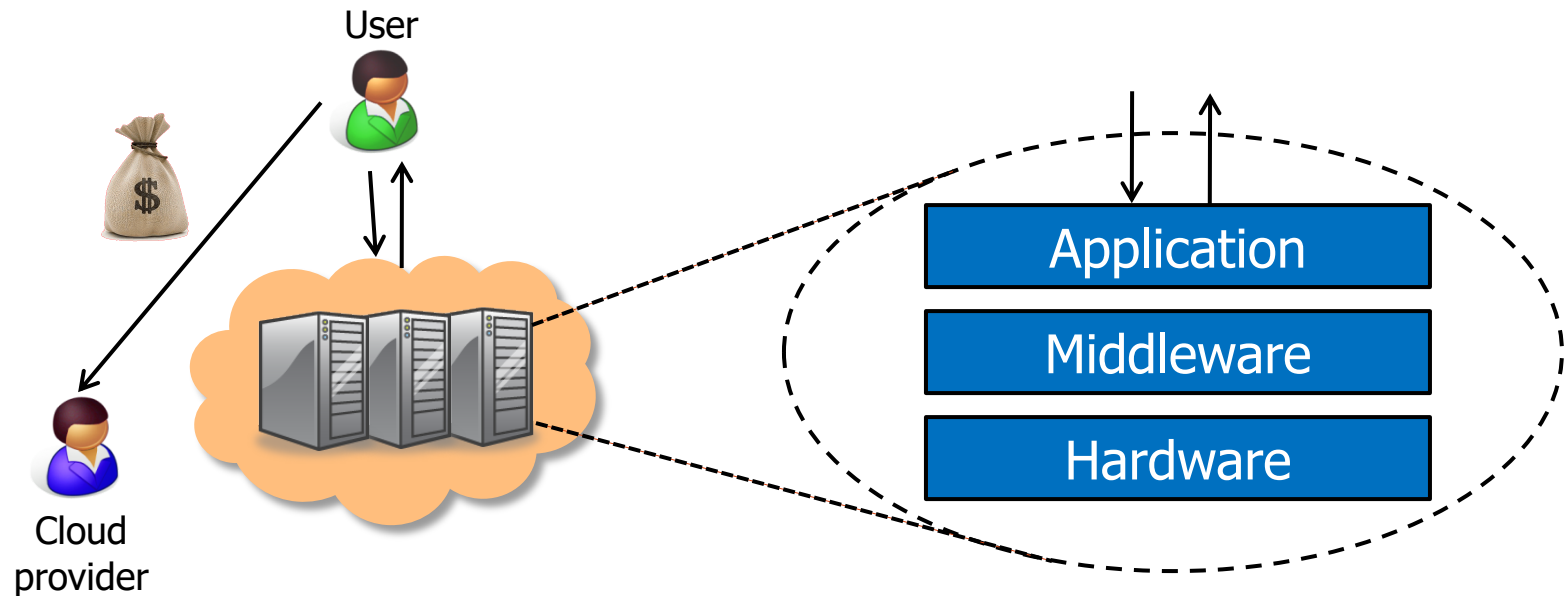


Data center in the cloud

Everything as a Service

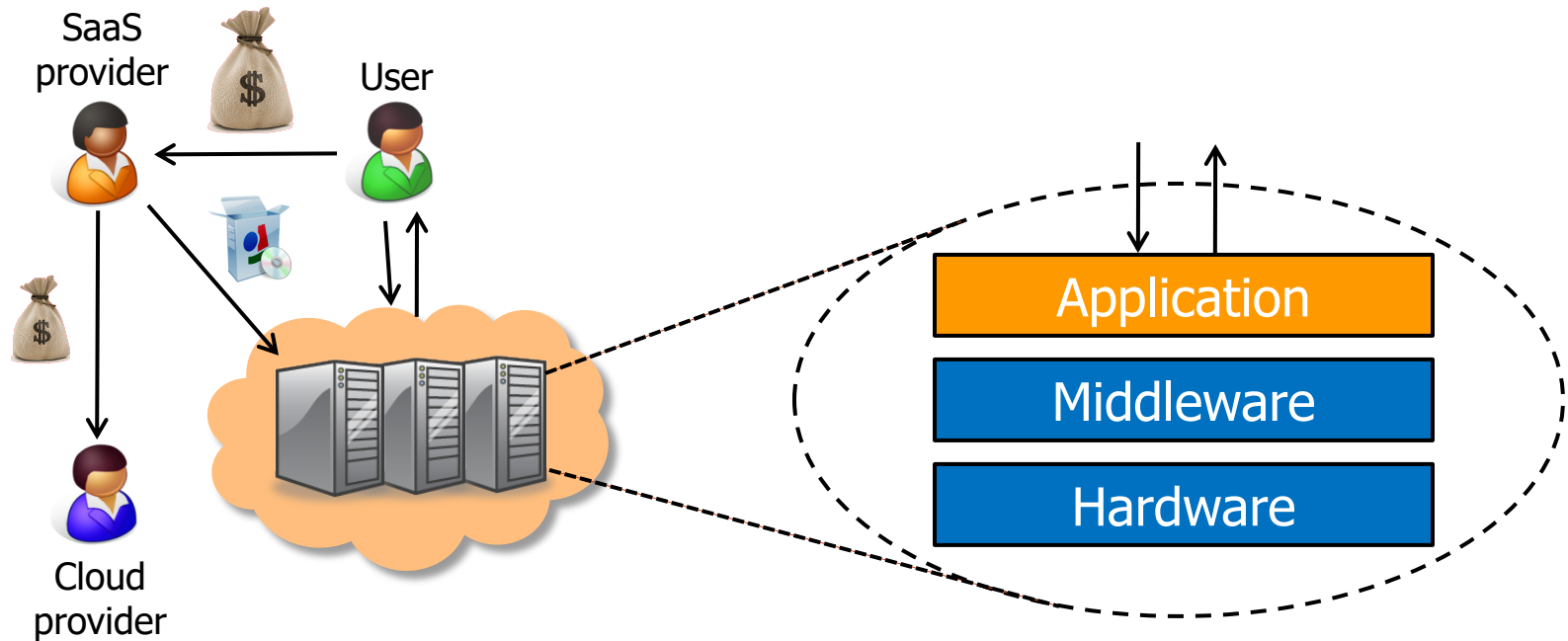
- What kind of service does the cloud provide?
 - Does it offer an entire application, or just resources?
 - If resources, what kind / level of abstraction?
- Three types commonly distinguished:
 - Software as a service (SaaS)
 - Analogy: Restaurant. Prepares&serves entire meal, does the dishes, ...
 - Platform as a service (PaaS)
 - Analogy: Take-out food. Prepares meal, but does not serve it.
 - Infrastructure as a service (IaaS)
 - Analogy: Grocery store. Provides raw ingredients.
 - Other XaaS types have been defined, but are less common
 - X=Data, Desktop, Backend, Communication, Network, Monitoring, ...

Software as a Service (SaaS)



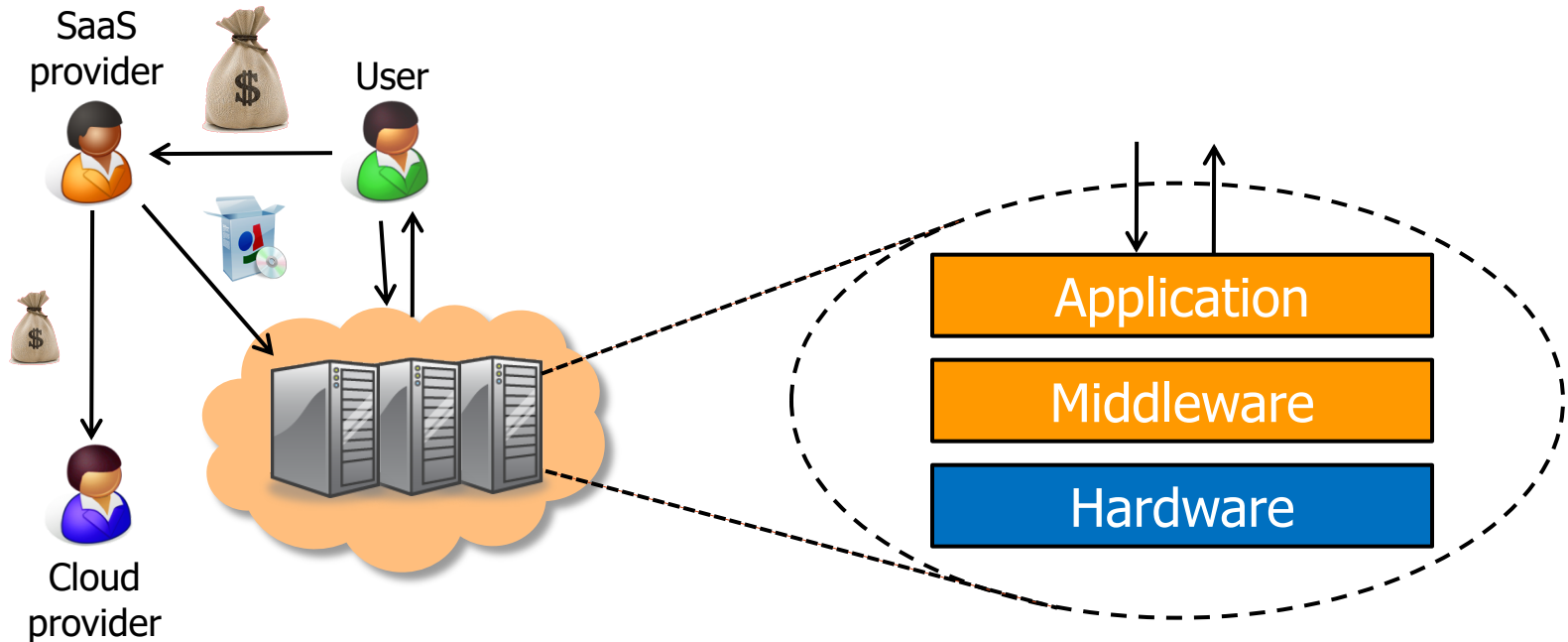
- Cloud provides an entire application
 - Word processor, spreadsheet, CRM software, calendar...
 - Customer pays cloud provider
 - Example: Google Apps (Docs, Gmail, Drive, Calendar), Salesforce.com

Platform as a Service (PaaS)



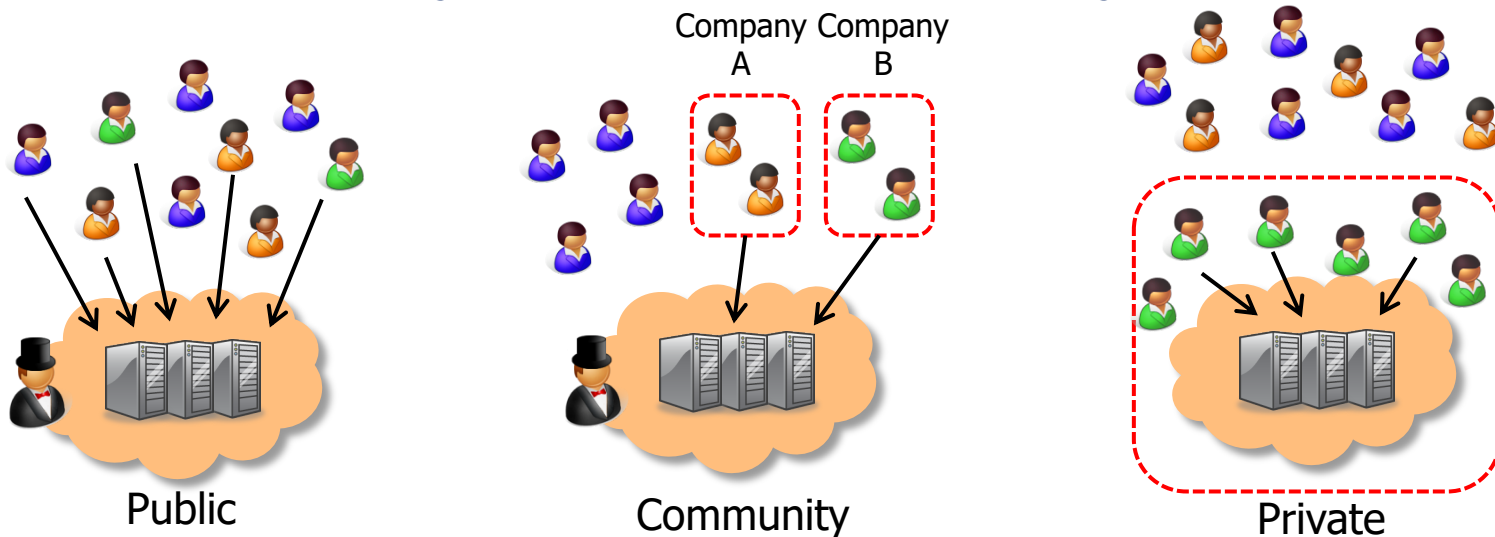
- Cloud provides middleware/infrastructure
 - Customer pays SaaS provider for the service; SaaS provider pays the cloud for the infrastructure
 - Example: Windows Azure, Google App Engine

Infrastructure as a Service (IaaS)



- Cloud provides raw computing resources
 - Virtual machine, blade server, hard disk, ...
 - Customer pays SaaS provider for the service; SaaS provider pays the cloud for the resources
 - Examples: Amazon Web Services, Rackspace Cloud, GoGrid

Private/hybrid/community clouds



- Who can become a customer of the cloud?
 - 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.

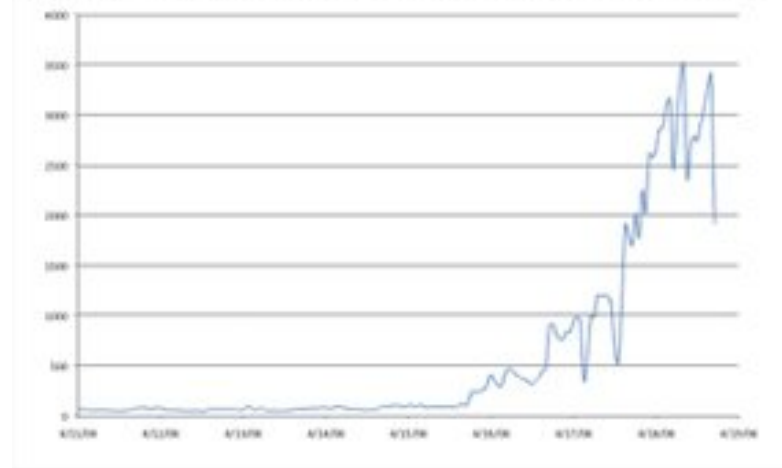
Examples of cloud applications

- Application hosting
- Backup and Storage
- Content delivery
- E-commerce
- High-performance computing
- Media hosting
- On-demand workforce
- Search engines
- Web hosting

Case study: ANIMOTO®

- Animoto: Lets users create videos from their own photos/music
 - Auto-edits photos and aligns them with the music, so it "looks good"
- Built using Amazon EC2+S3+SQS
- Released as Facebook app in mid-April 2008
 - More than 750,000 people signed up within 3 days
 - EC2 usage went from 50 machines to 3,500 (x70 scalability!)

Animoto: This Week's EC2 Instance Usage

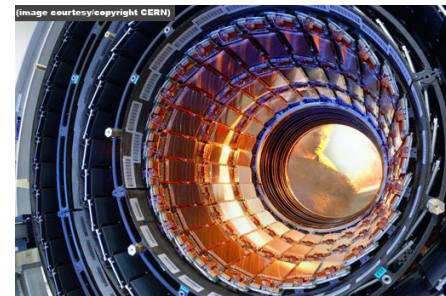


Case study: *The Washington Post*

- March 19, 2008: Hillary Clinton's official White House schedule released to the public
 - 17,481 pages of non-searchable, low-quality PDF
 - Very interesting to journalists, but would have required hundreds of man-hours to evaluate
 - Peter Harkins, Senior Engineer at The Washington Post: Can we make that data available more quickly, ideally within the same news cycle?
 - Tested various Optical Character Recognition (OCR) programs; estimated required speed
 - Launched 200 EC2 instances; project was completed within nine hours using 1,407 hours of VM time (\$144.62)
 - Results available on the web only 26 hours after the release

Other examples

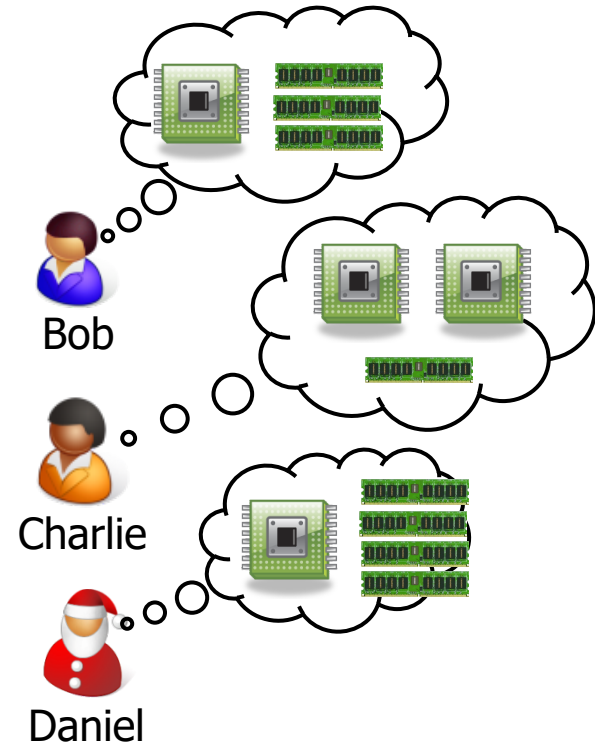
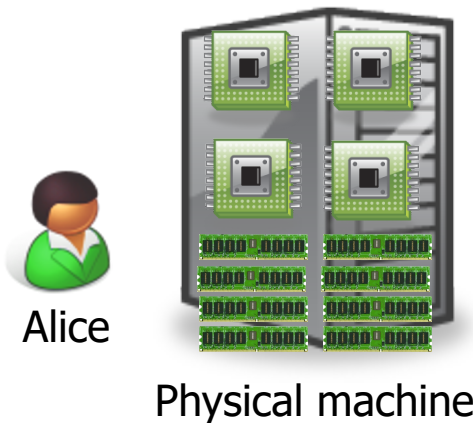
- DreamWorks used the Cerelink cloud to render animation movies
 - Cloud was already used to render parts of *Shrek Forever After* and *How to Train your Dragon*
- CERN has launched a "science cloud" to process experimental data
- Virgin Atlantic hosted their travel portal on Amazon AWS. Recently, they moved to Tata Consultancy Services (TCS).



Is the cloud good for everything?

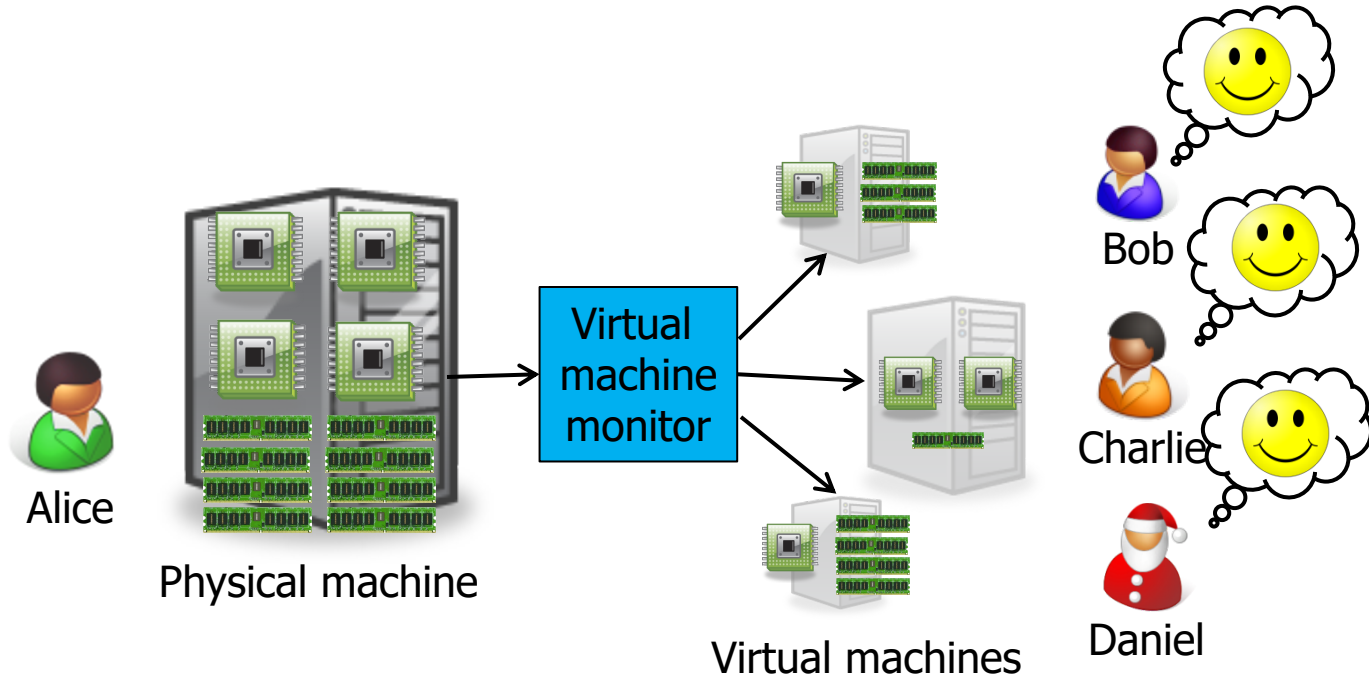
- No.
- Sometimes it is problematic, e.g., because of privacy requirements
- Example: Processing medical records
 - PPACA (Patient Protection and Affordable Care Act) privacy and security rule
 - GDPR (General Data Protection Regulation) on the protection of natural persons with regard to the processing of personal data and on the free movement of such data
- Example: Processing financial information
 - Sarbanes-Oxley act
- Would you put your medical data on the cloud?

What is virtualization?



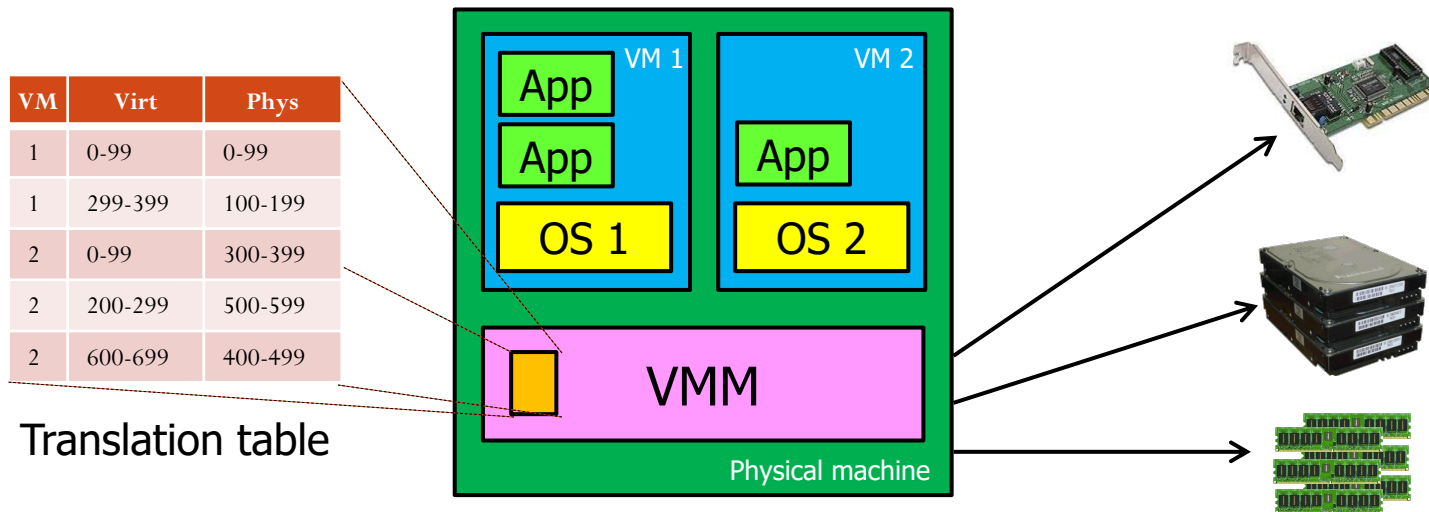
- Suppose Alice has a machine with 4 CPUs and 8 GB of memory, and three customers:
 - Bob wants a machine with 1 CPU and 3GB of memory
 - Charlie wants 2 CPUs and 1GB of memory
 - Daniel wants 1 CPU and 4GB of memory
- What should Alice do?

What is virtualization?



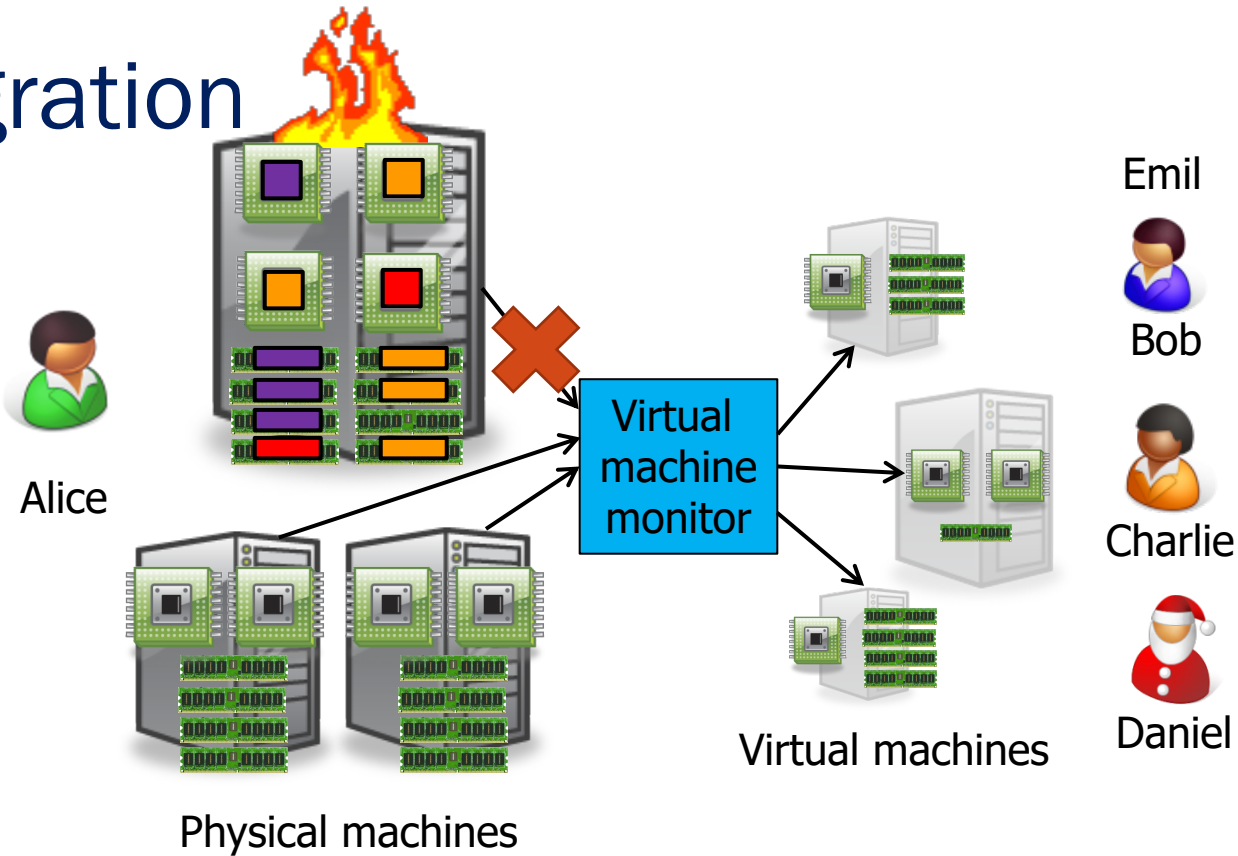
- Alice can sell each customer a **virtual machine** (VM) with the requested resources
 - From each customer's perspective, it appears as if they had a physical machine all by themselves (isolation)

How does it work?



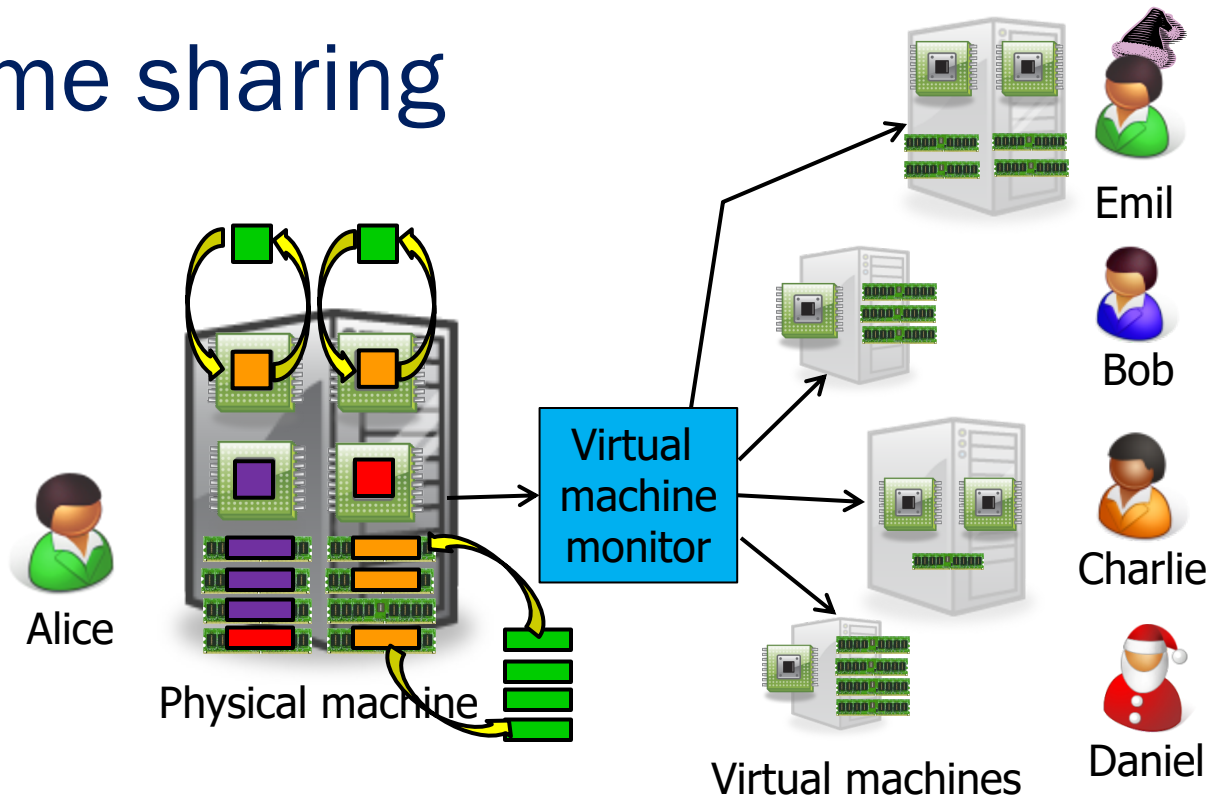
- Resources (CPU, memory, ...) are virtualized
 - VMM ("Hypervisor") has translation tables that map requests for virtual resources to physical resources
 - Example: VM 1 accesses memory cell #323; VMM maps this to physical memory cell #123.

Migration



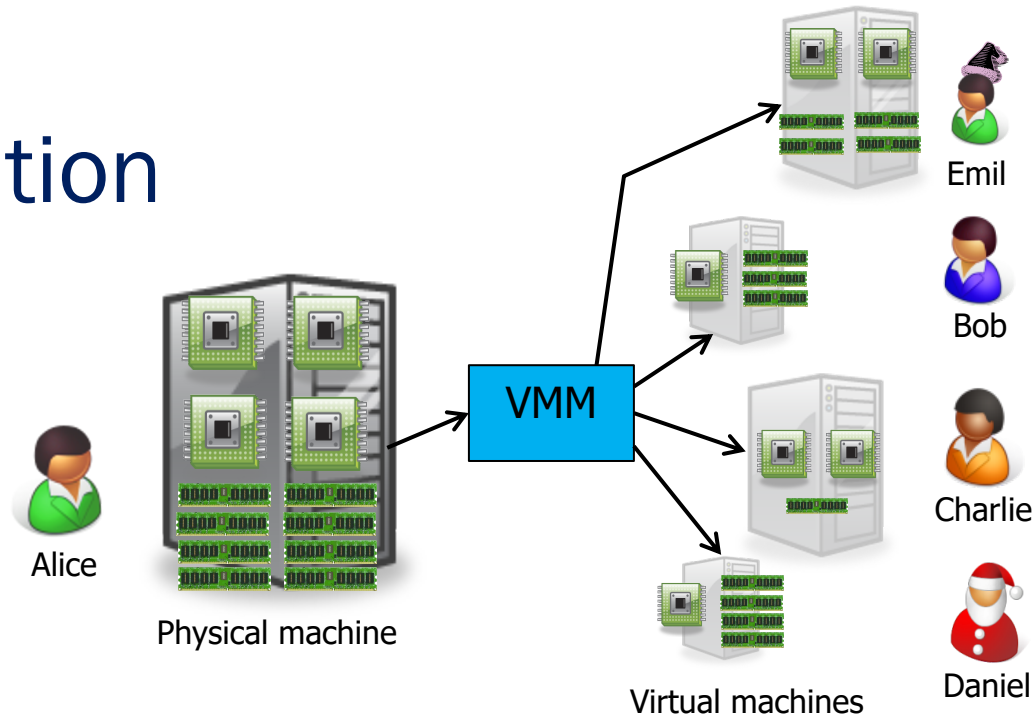
- What if the machine needs to be shut down?
 - e.g., for fault, maintenance, consolidation, ...
 - Alice can **migrate** the VMs to different physical machines without any customers noticing

Time sharing



- What if Alice gets another customer?
 - Multiple VMs can **time-share** the existing resources
 - Result: Alice has more virtual CPUs and virtual memory than physical resources (but not all can be active at the same time)

Isolation



- Good: Emil can't access Charlie's data
- Bad: What if the load suddenly increases?
 - Example: Emil's VM shares CPUs with Charlie's VM, and Charlie suddenly starts a large compute job
 - Emil's performance may decrease as a result
 - VMM can move Emil's software to a different CPU, or migrate it to a different machine

Recap: Virtualization in the cloud

- Gives cloud provider a lot of flexibility
 - Can produce VMs with different capabilities
 - Can migrate VMs if necessary (e.g., for maintenance)
 - Can increase load by overcommitting resources
- Provides security and isolation
 - Programs in one VM cannot influence programs in another
- Convenient for users
 - Complete control over the virtual 'hardware' (can install own operating system own applications, ...)
- But: Performance may be hard to predict
 - Load changes in other VMs on the same physical machine may affect the performance seen by the customer

10 obstacles and opportunities

1. Availability

- What happens to my business if there is an outage in the cloud?

2. Data lock-in

- How do I move my data from one cloud to another?

3. Data confidentiality and auditability

- How do I make sure that the cloud doesn't leak my confidential data?
- Can I comply with regulations like PPACA and GDPR?

Service	Duration	Date
S3	6-8 hrs	7/20/08
AppEngine	5 hrs	6/17/08
Gmail	1.5 hrs	8/11/08
Azure	22 hrs	3/13/09
EBS	>3 days	4/21/11
Verizon	~1 day	1/14/16
ECC	~10 hrs	6/4/16

Example of cloud outages

10 obstacles and opportunities

4. Data transfer bottlenecks

- How do I copy large amounts of data from/to the cloud?
- Example: 10 TB from UC Berkeley to Amazon in Seattle, WA
- Motivated Import/Export feature on AWS

Method	Time
Internet (20Mbps)	45 days
FedEx	1 day

Time to transfer 10TB [AF10]

5. Performance unpredictability

- Example: VMs sharing the same disk → I/O interference
- Example: HPC tasks that require coordinated scheduling

Primitive	Mean perf.	Std dev
Memory bandwidth	1.3GB/s	0.05GB/s (4%)
Disk bandwidth	55MB/s	9MB/s (16%)

Performance of 75 EC2 instances in benchmarks

10 obstacles and opportunities

6. Scalable storage

- Cloud model (short-term usage, no up-front cost, infinite capacity on demand) does not fit persistent storage well

7. Bugs in large distributed systems

- Many errors cannot be reproduced in smaller configs

8. Scaling quickly

- Problem: Boot time; idle power
- Fine-grain accounting?

10 obstacles and opportunities

9. Reputation fate sharing

- One customer's bad behavior can affect the reputation of others using the same cloud
- Example: Spam blacklisting, FBI raid after criminal activity

10. Software licensing

- What if licenses are for specific computers?
 - Example: Microsoft Windows
- How to scale number of licenses up/down?
 - Need pay-as-you-go model as well

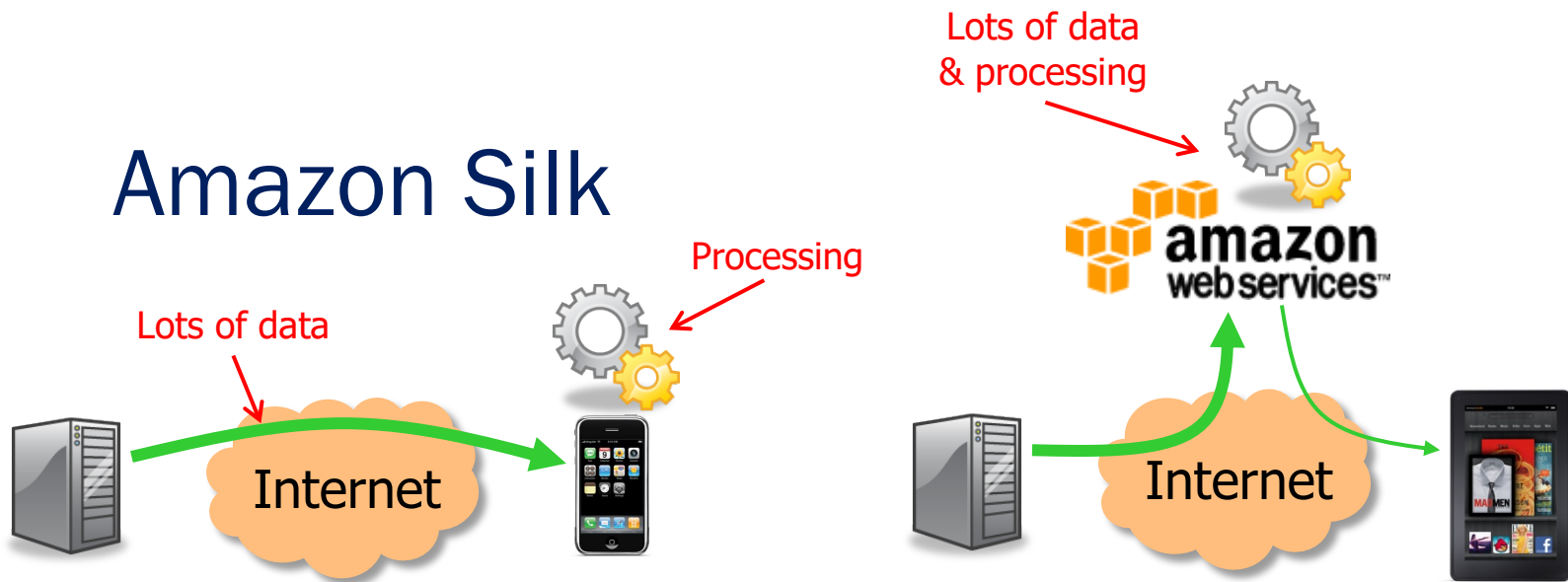
Public vs. Private Clouds

- As discussed previously, “cloud” is a broad term but comprises:
 - Very large data centers with thousands of commodity machines
 - Multiple, geographically distributed sites
 - Common management infrastructure
 - Common programming infrastructure that automatically allocates requests and/or jobs to available machines
- Difference between public and private clouds?
 - Public clouds sub-contract out to multiple clients; private clouds are controlled by one organization

The major public Cloud providers

- Amazon is the big player
 - Multiple services: infrastructure as a service, platform as a service (including Hadoop), storage as a service
- But there are many others:
 - Microsoft Azure — in many ways has similar services to Amazon, with an emphasis on .Net programming model
 - Google App Engine + GWT + services — offers servlet-level programming interface, Hadoop, ...
 - Also software as a service: gmail, docs, etc.
 - IBM, HP, Yahoo — seem to focus mostly on enterprise (often private) cloud apps (not small business-level)
 - Rackspace, Terremark — mostly infrastructure as a service

Amazon Silk



- Idea: Use the cloud to make browsers faster
 - Page rendering is split between the user's device & the cloud
 - Cloud performs 'heavy lifting' (rendering, script execution, ...)
 - Device just has to show the resulting page, so it doesn't need much bandwidth or processing power
 - Used on Kindle Fire
- Many opportunities for optimizations
 - Smart caching, on-the-fly optimizations
 - Learn about traffic patterns and pre-fetch pages

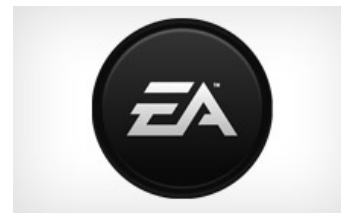
Software as a Service

- We'll look at three successful SaaS services hosted on companies' private clouds:
 - Salesforce.com
 - similar: NetSuite; Quicken's Web apps; TurboTax Web; etc.
 - GMail
 - similar: Hotmail, Yahoo Mail
 - Google Docs
 - similar: Office Web



Salesforce.com

- Founded in 1999: first proponents of the term ‘cloud’, with support from Larry Ellison (Oracle)
- First CRM offered as a SaaS (Software as a service)
- Software being provided: **Customer Relationship Management**
 - Tools for sales people to find customers: gives a view of customers’ status, in-flight orders, order history, leads, approvals, etc.
 - more than 90,000 customers



Outsourcing your e-mail: Gmail



- (and, to a lesser extent, Yahoo Mail, Hotmail)
- Basic architecture:
 - Distributed, replicated message store in **BigTable** – a key-value store like Amazon SimpleDB
 - “Multihomed” model – if one site crashes, user gets forwarded to another
 - Weak consistency model for some operations – “message read”
 - Stronger consistency for others – “send message”
- We all know Gmail: what is it that makes it special?
- What is the business model?

Outsourcing your documents: Google Docs

- The idea:
 - instead of buying software, worrying about security and administration...
 - simply put your docs on the Web and let Google do the rest!
- Today: much remains to be proven
 - Features? [right now, quite limited vs. MS Office]
 - Security? [cf. hackers' attack on Google]
- But some benefits
 - Sharing and collaboration are much easier



Users of Platform as a Service

- Facebook provides some PaaS capabilities to application developers
 - Web services – remote APIs – that allow access to social network properties, data, “Like” button, etc.
 - Many third-parties run their apps on Amazon EC2, and interface to Facebook via its APIs – PaaS + IaaS
- Facebook itself makes heavy use of PaaS services for their own private cloud
 - Key problems: how to analyze logs, make suggestions, determine which ads to place
- There is no common standard (yet)
 - App Engine is PaaS and supports Java/JVM or Python
 - Azure is PaaS and supports .NET/CLR
 - AWS is PaaS/IaaS and supports IA-64 virtual machines

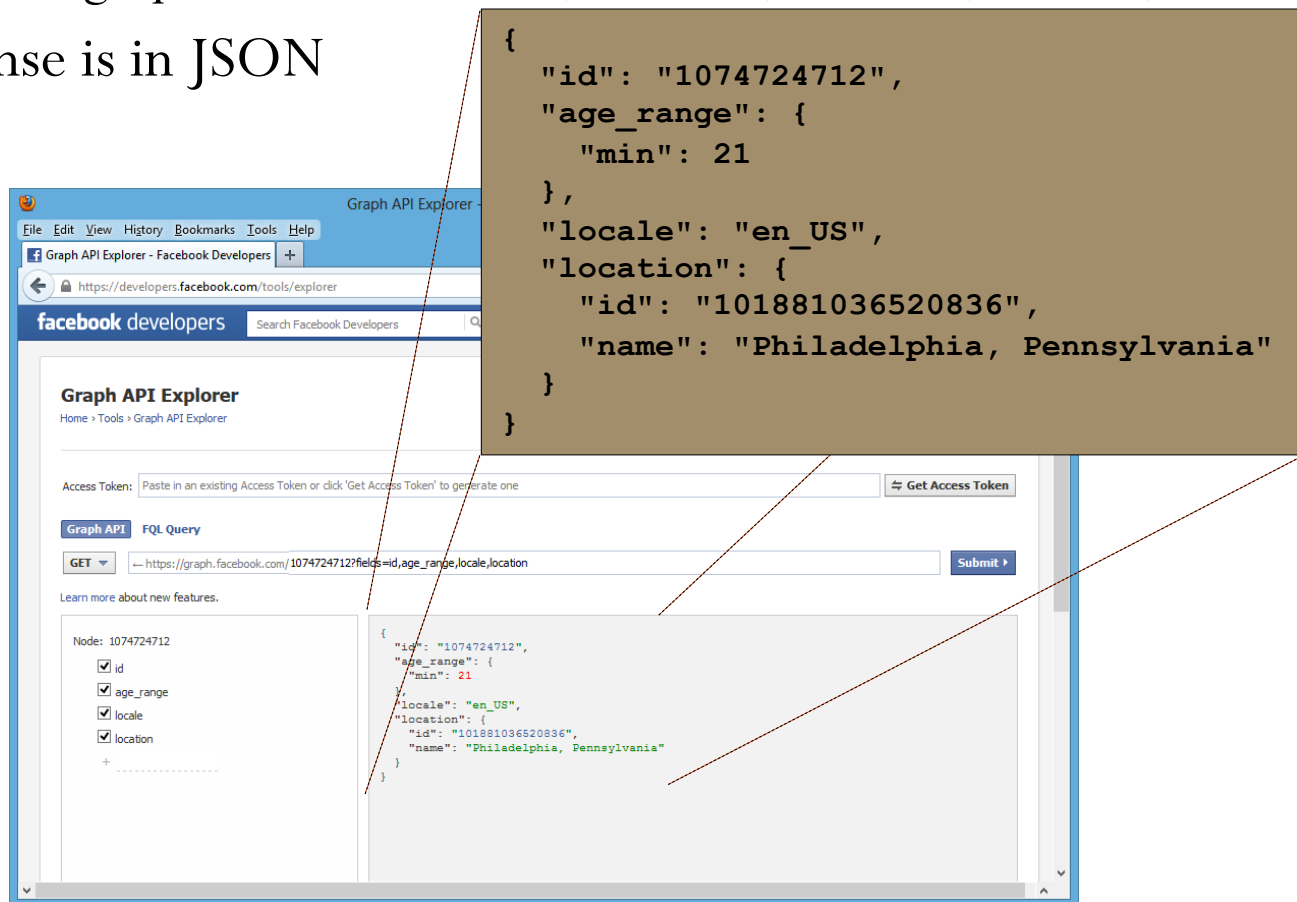
Facebook API: Overview



- What you can do:
 - Read data from profiles and pages
 - Navigate the graph (e.g., via friends lists)
 - Issue queries (for posts, people, pages, ...)
 - Add or modify data (e.g., create new posts)
 - Get real-time updates, issue batch requests, ...
- How you can access it:
 - Graph API
 - FQL
 - REST API

Facebook API: The Graph API (1/2)

- Requests are mapped directly to HTTP:
 - `https://graph.facebook.com/(identifier)?fields=(fieldList)`
- Response is in JSON

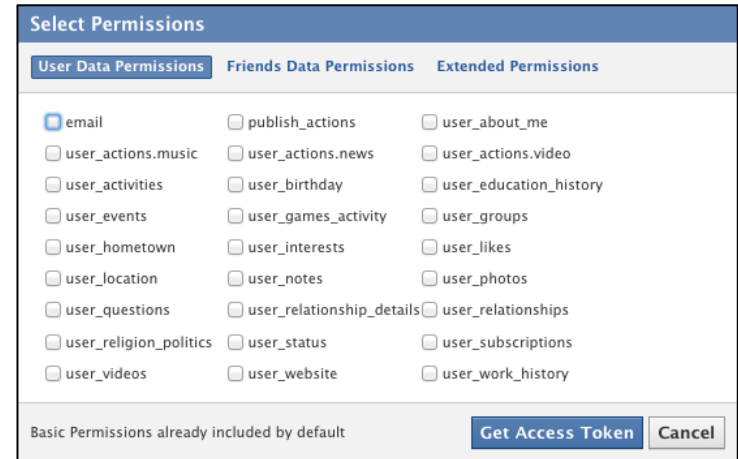


The screenshot displays the Facebook Graph API Explorer interface. The URL bar shows `https://developers.facebook.com/tools/explorer`. The main content area shows a GET request to `https://graph.facebook.com/1074724712?fields=id,age_range,locale,location`. The response is a JSON object, which is also shown in a separate callout box. The JSON response is as follows:

```
{
  "id": "1074724712",
  "age_range": {
    "min": 21
  },
  "locale": "en_US",
  "location": {
    "id": "101881036520836",
    "name": "Philadelphia, Pennsylvania"
  }
}
```

Facebook API: The Graph API (2/2)

- Uses HTTP methods:
 - GET for reading
 - POST for adding or modifying
 - DELETE for removing
- IDs can be numeric or names
 - /1074724712 or /andreas.haeberlen
 - Pages also have IDs
- Authorization is via ‘access tokens’
 - Opaque string; encodes specific permissions (access user location, but not interests, etc.)
 - Has an expiration date, so may need to be refreshed



The screenshot shows the 'Select Permissions' dialog box from the Facebook API. It has three tabs: 'User Data Permissions', 'Friends Data Permissions', and 'Extended Permissions'. The 'User Data Permissions' tab is selected. It contains a list of permissions with checkboxes. The 'email' permission is checked. Other permissions include 'user_actions.music', 'user_activities', 'user_events', 'user_hometown', 'user_location', 'user_questions', 'user_religion_politics', 'user_videos', 'publish_actions', 'user_actions.news', 'user_birthday', 'user_games_activity', 'user_interests', 'user_notes', 'user_relationship_details', 'user_status', 'user_website', 'user_about_me', 'user_actions.video', 'user_education_history', 'user_groups', 'user_likes', 'user_photos', 'user_relationships', 'user_subscriptions', and 'user_work_history'. At the bottom, there is a note 'Basic Permissions already included by default' and two buttons: 'Get Access Token' and 'Cancel'.

User Data Permissions	Friends Data Permissions	Extended Permissions
<input checked="" type="checkbox"/> email	<input type="checkbox"/> publish_actions	<input type="checkbox"/> user_about_me
<input type="checkbox"/> user_actions.music	<input type="checkbox"/> user_actions.news	<input type="checkbox"/> user_actions.video
<input type="checkbox"/> user_activities	<input type="checkbox"/> user_birthday	<input type="checkbox"/> user_education_history
<input type="checkbox"/> user_events	<input type="checkbox"/> user_games_activity	<input type="checkbox"/> user_groups
<input type="checkbox"/> user_hometown	<input type="checkbox"/> user_interests	<input type="checkbox"/> user_likes
<input type="checkbox"/> user_location	<input type="checkbox"/> user_notes	<input type="checkbox"/> user_photos
<input type="checkbox"/> user_questions	<input type="checkbox"/> user_relationship_details	<input type="checkbox"/> user_relationships
<input type="checkbox"/> user_religion_politics	<input type="checkbox"/> user_status	<input type="checkbox"/> user_subscriptions
<input type="checkbox"/> user_videos	<input type="checkbox"/> user_website	<input type="checkbox"/> user_work_history

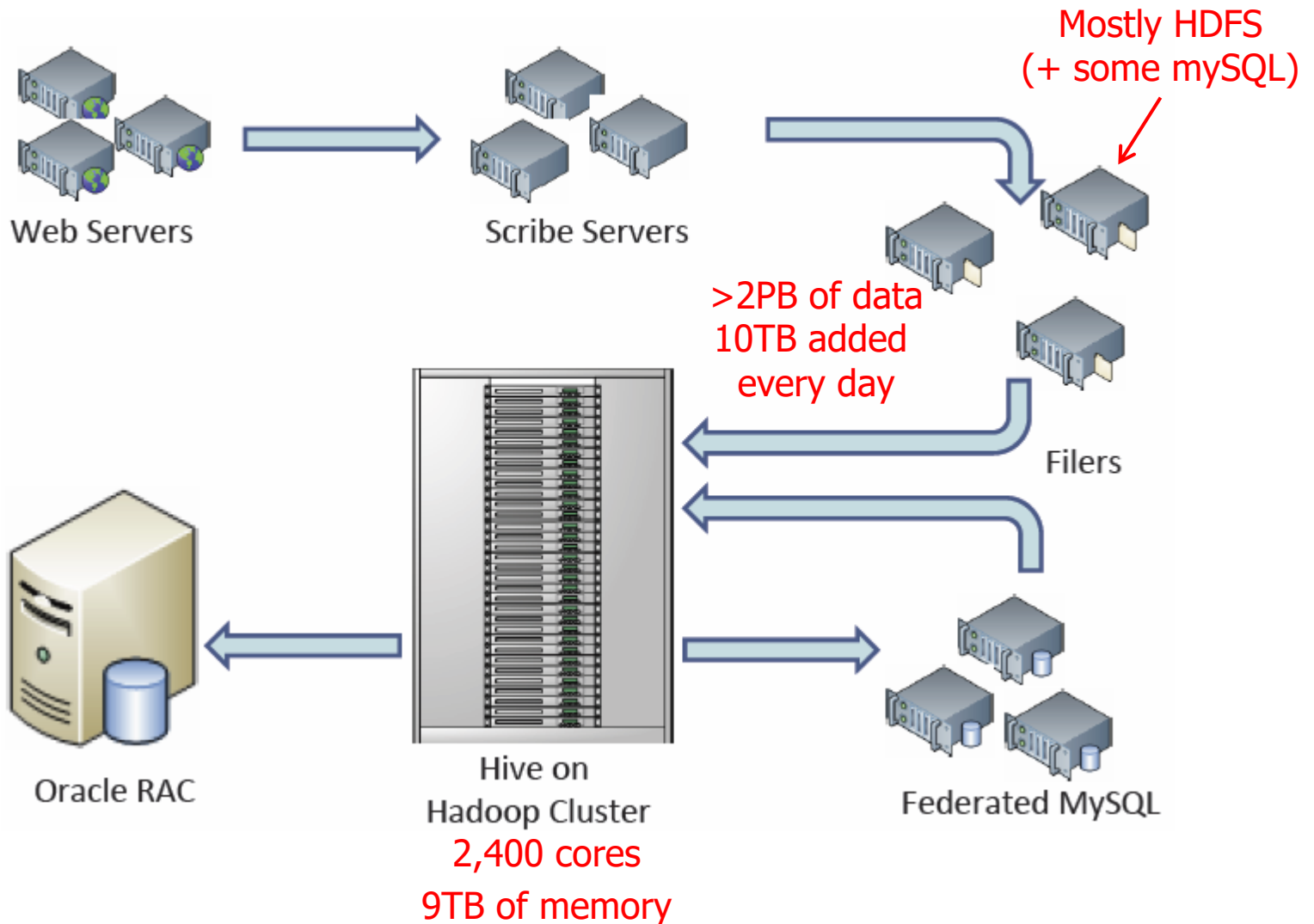
Basic Permissions already included by default

[Get Access Token](#) [Cancel](#)

Facebook Data Management

- Main tasks for “cloud” infrastructure:
 - Summarization (daily, hourly)
 - to help guide development on different components
 - to report on ad performance
 - recommendations
 - Ad hoc analysis:
 - Answer questions on historical data – to help with managerial decisions
 - Archival of logs
 - Spam detection
 - Ad optimization
 - ...
- Initially used Oracle DBMS for this
 - But eventually hit scalability, cost, performance bottlenecks
 - ... just like Salesforce does now

Data Warehousing at Facebook



PaaS at Facebook

- **Scribe** – open source logging, actually records the data that will be analyzed by Hadoop
- **Hadoop** as batch processing engine for data analysis
 - As of 2016: 1st largest Hadoop cluster in the world, 4000 cores, > 2PB data with > 10TB added every day
- **Hive** – SQL over Hadoop, used to write the data analysis queries
- **Federated MySQL, Oracle** – multi-machine DBMSs to store query results

IaaS example: Netflix



- Perhaps Amazon's highest-profile customer
 - Most of their traffic is served from AWS
 - In 2009, none of it was
- Why did Netflix take this step?
 - Needed to re-architect after a phase of growth
 - Ability to question everything
 - Focus on their core competence (content); leave the 'heavy lifting' (datacenter operation) to Amazon
 - Customer growth & device engagement hard to predict
 - With the cloud, they don't have to
 - Belief that cloud computing was the future
 - Gain experience with an increasingly important technology

How Netflix uses AWS

- Streaming movie retrieval and playback
 - Media files stored in S3
 - “Transcoding” to target devices (Smartphone, tablet, etc.) using EC2
- Web site modules
 - Movie lists and search — app hosted by Amazon Web Services
- Recommendations
 - Analysis of streaming sessions, business metrics — using Elastic MapReduce

Other users, and the future

- Startups, especially, are making great use of EC2, Rackspace, etc. for their hosting needs
 - compare to 20 years ago – dot-com boom – where you started by buying a cluster of SPARC machines
- Government, health care, science, many enterprises have great interest in cost savings of the cloud
 - But concerns remain – esp. with respect to security, privacy, availability
- ... And moreover: the last word has not been written on how to *program* the cloud

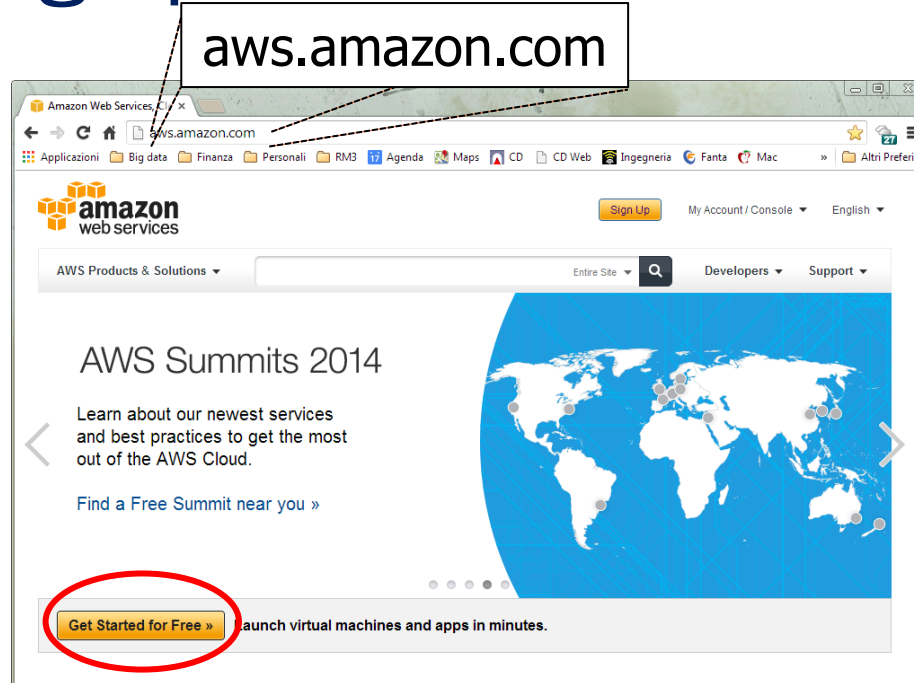
Amazon AWS



- Amazon Web Services (AWS) provides a number of different services, including:
 - Amazon Elastic Compute Cloud (EC2)
Virtual machines for running custom software
 - Amazon Simple Storage Service (S3)
Simple key-value store, accessible as a web service
 - Amazon RDS
Simple distributed database
 - Amazon Elastic MapReduce (EMR)
Scalable MapReduce computation
 - Amazon Mechanical Turk (MTurk)
A 'marketplace for work'
 - Amazon CloudFront
Content delivery network
 - ...

Used for the projects

Setting up an AWS account

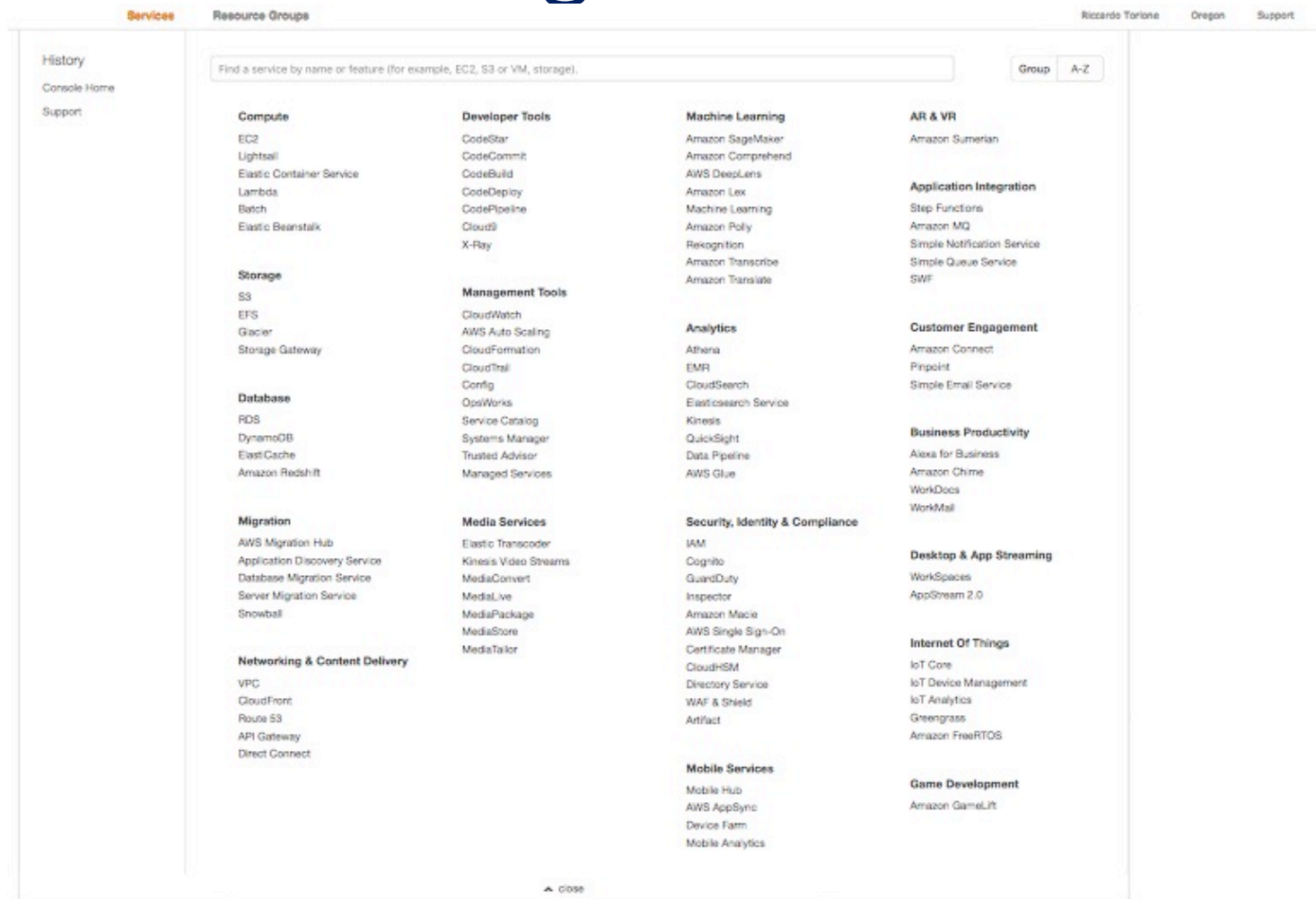


- Sign up for an account on aws.amazon.com
 - You need to choose a username and a password
 - These are for the management interface only
 - Your programs will use other credentials (RSA keypairs, access keys, ...) to interact with AWS

AWS Educate

- <https://aws.amazon.com/it/education/awseducate/>
- Register as student of Università Roma Tre
- 100\$ free for using AWS services

The AWS management console



- Used to control many AWS services:
 - For example, start/stop EC2 instances, create S3 buckets...

What is Amazon EC2?

Linux	RHEL	SLES	Windows	Windows con SQL Standard	Windows con SQL Web
Windows con SQL Enterprise					
Regione: Stati Uniti orientali (Ohio) 4					
	vCPU	ECU	Memoria (GiB)	Storage istanze (GiB)	Utilizzo di Linux/UNIX
Uso generale - Generazione attuale					
t2.nano	1	Variable	0.5	Solo EBS	\$0.0058 all'ora
t2.micro	1	Variable	1	Solo EBS	\$0.0116 all'ora
t2.small	1	Variable	2	Solo EBS	\$0.023 all'ora
t2.medium	2	Variable	4	Solo EBS	\$0.0464 all'ora
t2.large	2	Variable	8	Solo EBS	\$0.0928 all'ora
t2.xlarge	4	Variable	16	Solo EBS	\$0.1856 all'ora
t2.2xlarge	8	Variable	32	Solo EBS	\$0.3712 all'ora
m5.large	2	10	8	Solo EBS	\$0.096 all'ora
m5.xlarge	4	15	16	Solo EBS	\$0.192 all'ora
m5.2xlarge	8	31	32	Solo EBS	\$0.384 all'ora
m5.4xlarge	16	61	64	Solo EBS	\$0.768 all'ora
m5.12xlarge	48	173	192	Solo EBS	\$2.304 all'ora
m5.24xlarge	96	345	384	Solo EBS	\$4.608 all'ora
m4.large	2	6.5	8	Solo EBS	\$0.1 all'ora
m4.xlarge	4	13	16	Solo EBS	\$0.2 all'ora
m4.2xlarge	8	26	32	Solo EBS	\$0.4 all'ora
m4.4xlarge	16	53.5	64	Solo EBS	\$0.8 all'ora
m4.10xlarge	40	124.5	160	Solo EBS	\$2 all'ora
m4.16xlarge	64	188	256	Solo EBS	\$3.2 all'ora
Calcolo ottimizzato - Generazione attuale					
c5.large	2	8	4	Solo EBS	\$0.085 all'ora

- Infrastructure-as-a-Service (IaaS)
 - You can rent various types of virtual machines by the hour
 - In your VMs, you can run your own (Linux/Windows) programs
 - Examples: Web server, search engine, movie renderer, ...

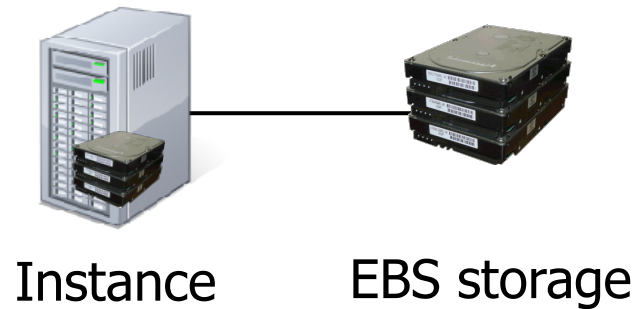
Amazon Machine Images

- When I launch an instance, what software will be installed on it?
 - Software is taken from an **Amazon Machine Image (AMI)**
 - Selected when you launch an instance
 - Essentially a file system that contains the operating system, applications, and potentially other data
 - Lives in S3
- How do I get an AMI?
 - Amazon provides several generic ones, e.g., Amazon Linux, Fedora Core, Windows Server, ...
 - You can make your own
 - You can even run your own custom kernel (with some restrictions)

Regions and Availability Zones

- Where exactly does my instance run?
 - No easy way to find out - Amazon does not say
- Instances can be assigned to **regions**
 - Currently 18 available: US East (Northern Virginia), US West (Northern California), US West (Oregon), EU (Ireland), Asia/Pacific (Singapore), Asia/Pacific (Sydney), Asia/Pacific (Tokyo), South America (Sao Paulo), ...
 - Important, e.g., for reducing latency to customers
- Instances can be assigned to **availability zones**
 - Purpose: Avoid correlated fault
 - Several availability zones within each region

What is Elastic Block Store (EBS)?



- Persistent storage
 - Unlike the local instance store, data stored in EBS is not lost when an instance fails or is terminated
- Should I use the instance store or EBS?
 - Typically, instance store is used for temporary data

Volumes

- EBS storage is allocated in **volumes**
 - A volume is a 'virtual disk' (size: 1GB - 1TB)
 - Basically, a raw block device
 - Can be attached to an instance (but only one at a time)
 - A single instance can access multiple volumes
- Placed in specific availability zones
 - Why is this useful?
 - Be sure to place it near instances (otherwise can't attach)
- Replicated across multiple servers
 - Data is not lost if a single server fails
 - Amazon: Annual failure rate is 0.1-0.5% for a 20GB volume

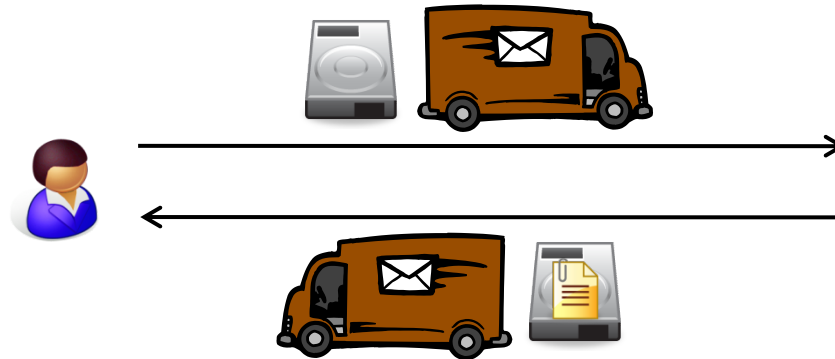
Pricing

- You pay for...
 - Storage space: \$0.10 per allocated GB per month
 - I/O requests: \$0.10 per million I/O requests
 - S3 operations (GET/PUT)
- Charge is only for actual storage used
 - Empty space does not count

AWS Import/Export

Method	Time
Internet (20Mbps)	45 days
FedEx	1 day

Time to transfer 10TB [AF10]



- Import/export large amounts of data to/from S3 buckets via physical storage device
 - Mail an actual hard disk to Amazon (power adapter, cables!)
 - Signature file for authentication