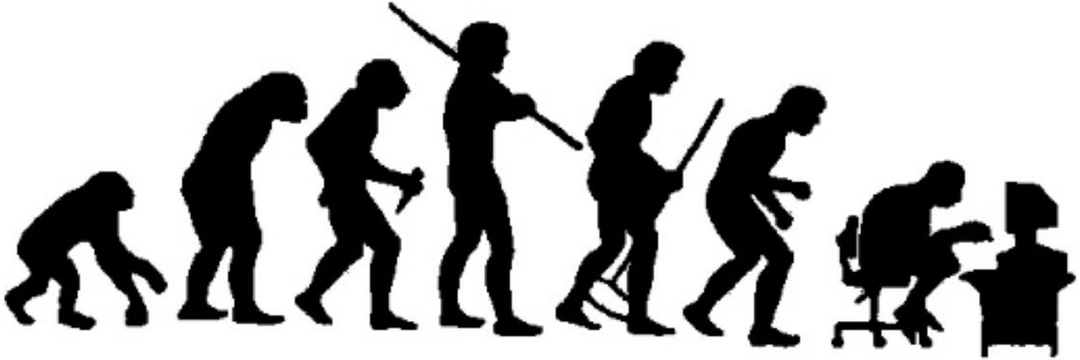
Cloud Computing Day #1

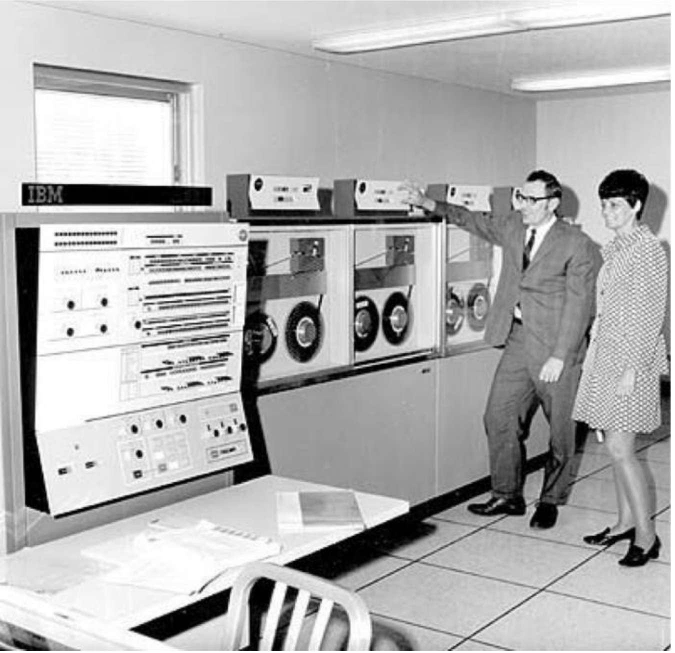
Disclaimer

Outline

- Evolution of Computing Paradigms
- Overview of existing hosting platforms
- Introduction to Cloud Computing
- Workload patterns for the cloud
- “Big Data”
- IT as a Service
- Technology behind Cloud Computing

Computing Paradigm Evolution





+

Highly Centralised
Very Powerful

–

Programmer
Productivity

1960s

Mainframes ruled the earth.

sw

Low level & Sequential
(Assembly, COBOL,
ForTran)

dx

None

+

Programmer
Productivity

–

Less Powerful
More Administration

1970s

The rise of the **Minis**

sw

Structured
(Pascal, C)

dx

None

+

Programmer Productivity Surge in Demand

–

Even Lesser Power Increased Administration

1980s

Proliferation of the **PCs**

sw

Object Oriented Programming (Smalltalk, C++)

dx

Network (Socket) Programming

+

More Centralized
More Powerful

–

Administration Explosion

1990s

Network the Servers

sw

Component Oriented Programming (COM)

dx

Homogeneous Client-Server, 3-Tier, RPC Based (DCOM, CORBA, RMI)

+

Centralized Applications
Lower Administration
Rise of Mobile platform

−

Lack of Customization
Lack of Multi-Tenancy

2000s

The **Internet** comes of age

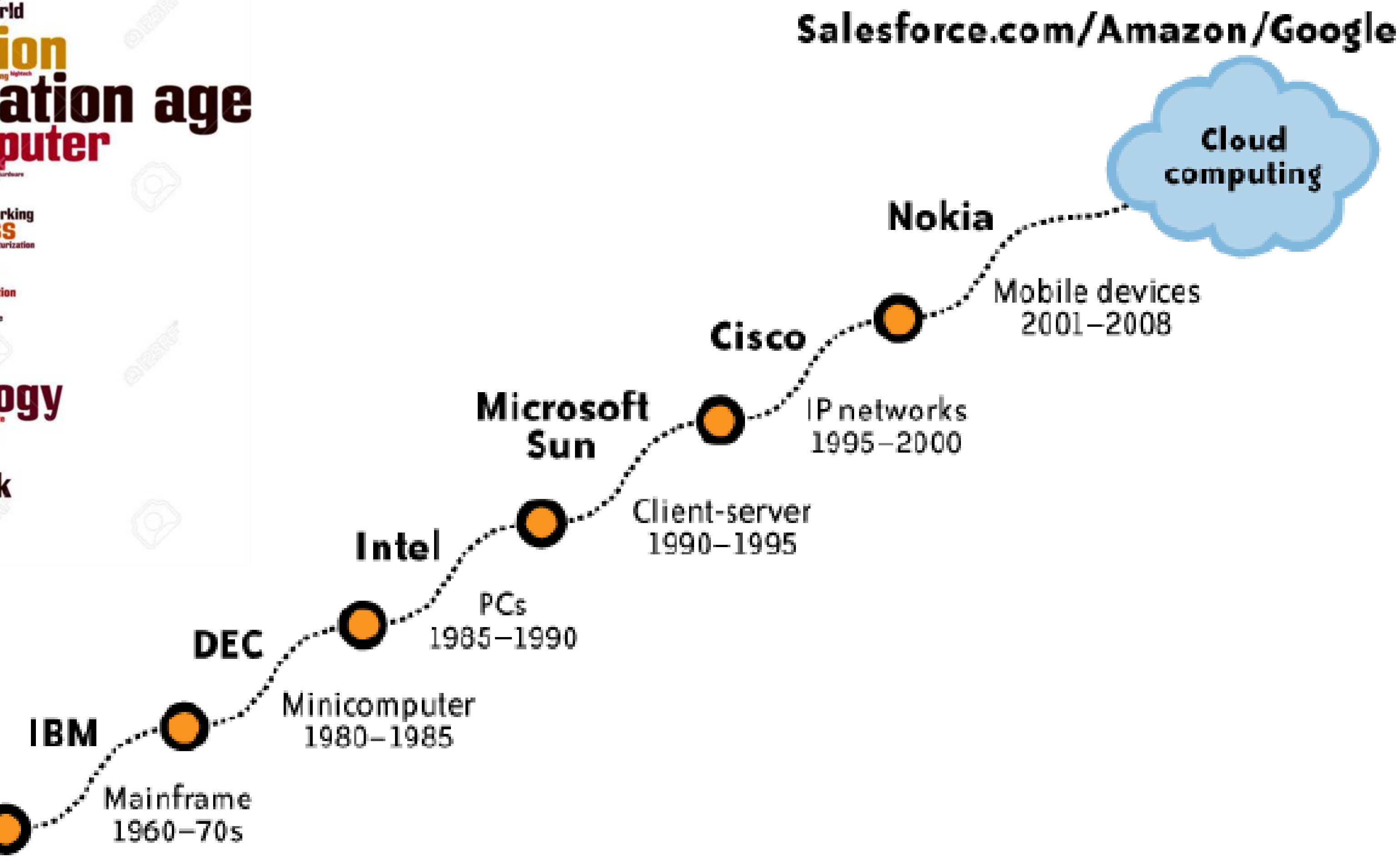
sw

Component Oriented
Programming
(.NET)

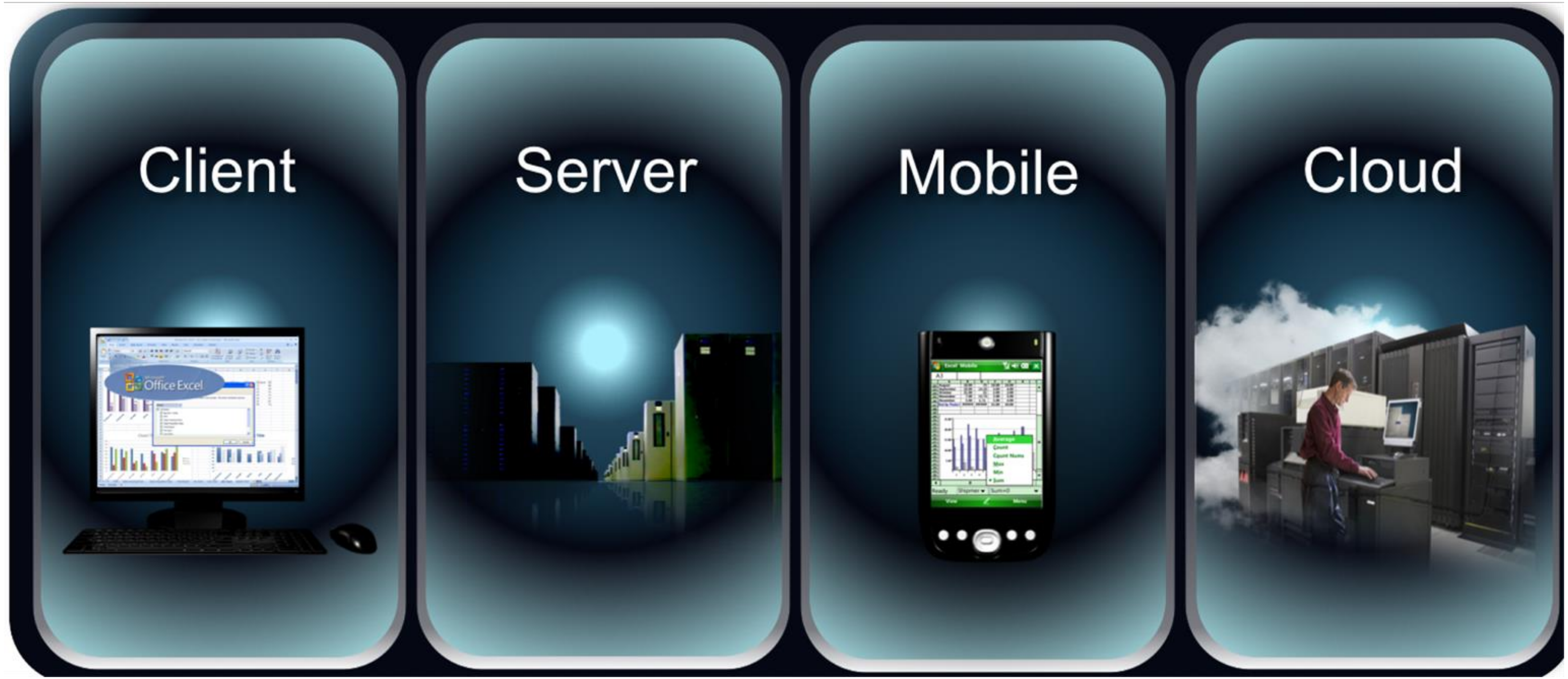
dx

Heterogeneous 3/n-Tier,
Message Passing based
Web Computing
(Web Services, SaaS)


Internet
world
information
information age
computer
communication
digital
global
business
age
tech
web
communications
technology
software
knowledge
network
data
binary
computing


Salesforce.com/Amazon/Google
Cloud computing
Nokia
Mobile devices
2001–2008
Cisco
IP networks
1995–2000
Microsoft
Sun
Client-server
1990–1995
Intel
PCs
1985–1990
DEC
Minicomputer
1980–1985
IBM
Mainframe
1960–70s

Platform Evolution



Platform Evolution

- Today there are a number of options you have when building applications.
- There are three types of platforms established in the industry:
  - Client - Rich client applications
  - Server - Many of us are developing server-based applications today, ranging from web applications to web services.
  - Mobile - Virtually all of you have a mobile device.

Shift to Cloud Computing

- Today there is a shift emerging that we believe will eventually effect virtually every type of organization from small start-ups to large enterprises.
- The shift represents the emergence for a fourth type of platform – the cloud
- This industry shift is the use of cloud computing and cloud services.
- Organizations extend compute, storage, and other workloads to the cloud – where these workloads will be operated and managed by a software vendor.

Platform Continuum

On-Premises Servers

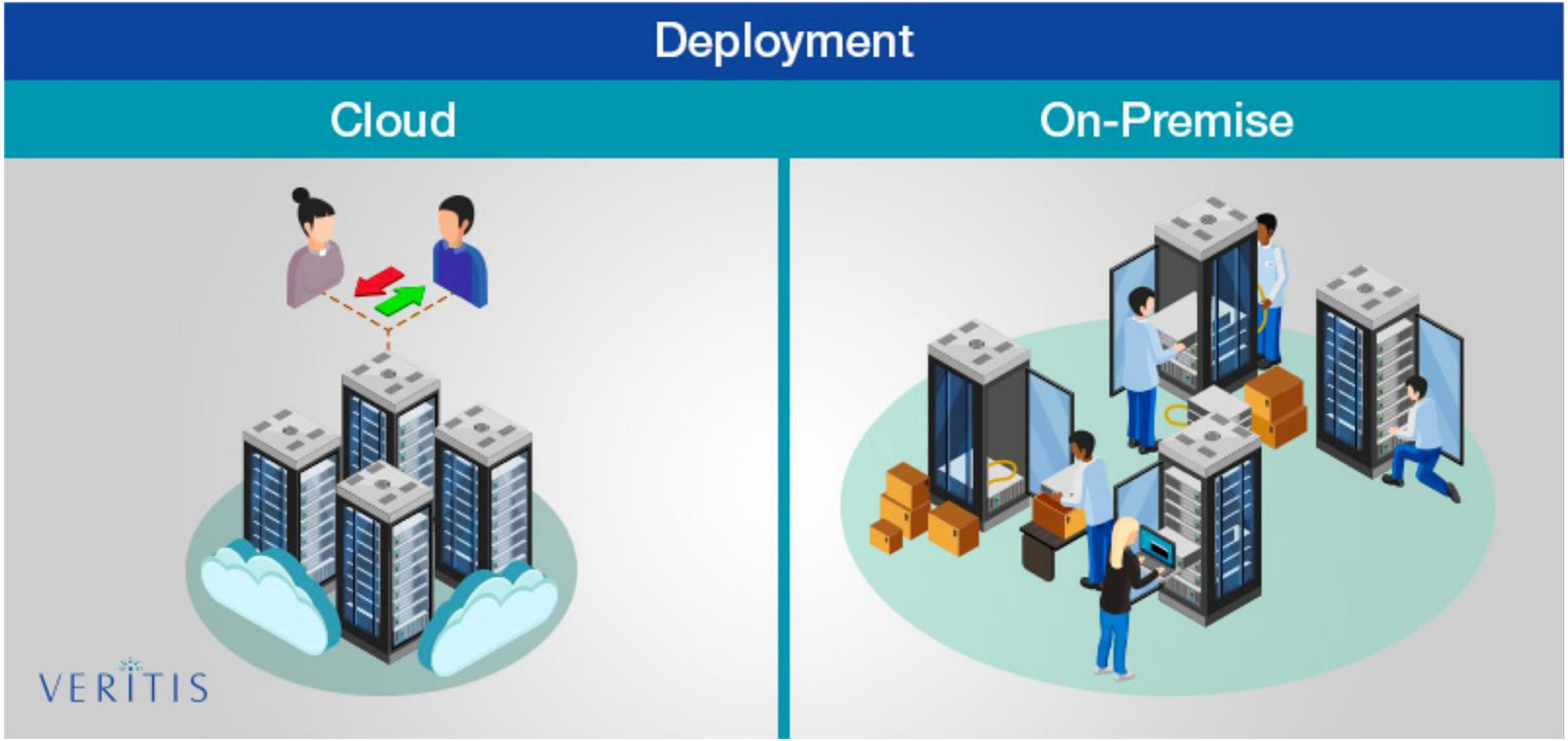
- Bring your own machines, connectivity, software, etc.
- Complete control
- Complete responsibility
- Static capabilities
- Upfront capital costs for the infrastructure

Hosted Servers

- Renting machines, connectivity, software
- Less control
- Fewer responsibilities
- Lower capital costs
- More flexible
- Pay for fixed capacity, even if idle

Cloud Platform

- Shared, multi-tenant infrastructure
- Virtualized & dynamic
- Scalable & available
- Abstracted from the infrastructure
- Higher-level services
- Pay as you go



On-Premise vs Cloud Computing Model

+ –

2010s

Welcome to the **Cloud**

Digital Transformation

sw

Service Oriented Architecture (.NET, WCF)

dx

Web Computing, Cloud Computing (Web Services – WS*, SOAP, REST)

Cloud Computing

- Store Data
- Run Applications
  - Combined with:
    - Utility model
    - Elastic Nature

What Is Cloud Computing?

Cloud Computing: App and Infrastructure over Internet

Compute as a Service:

Applications over the Internet

Utility Computing:

"Pay-as-You-Go" Datacenter Hardware and Software

What Is Cloud Computing?

Three New Aspects to Cloud Computing

The Illusion of Infinite Computing Resources Available on Demand

The Elimination of an Upfront Commitment by Cloud Users

The Ability to Pay for Use of Computing Resources
on a Short-Term Basis as Needed

What is cloud computing?

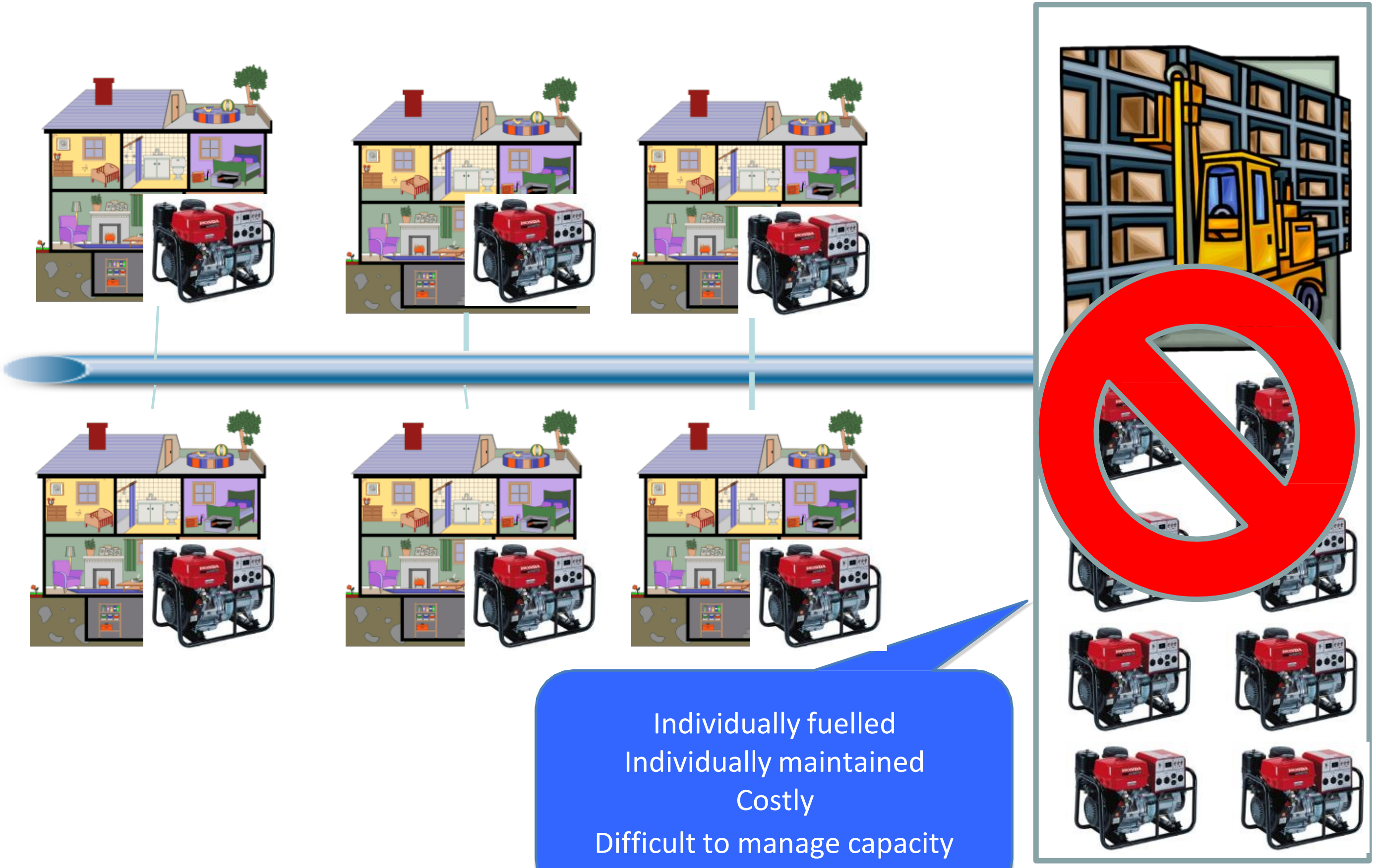
- “Cloud” is actually a metaphor for the Internet.
- User do not have or need knowledge, control, ownership in the computer infrastructure.
- Users simply rent or access the software, paying only what they use

What is cloud computing?

Cloud computing is using the Internet to **access someone else's software** running on **someone else's hardware** in **someone else's data centre** while **paying only for what you use**.

- I pay ONLY for what I use
- ONLY when I use it
- With the ability to SCALE capacity up and down on-demand

Advantage of Multi-tenant cloud based architecture?

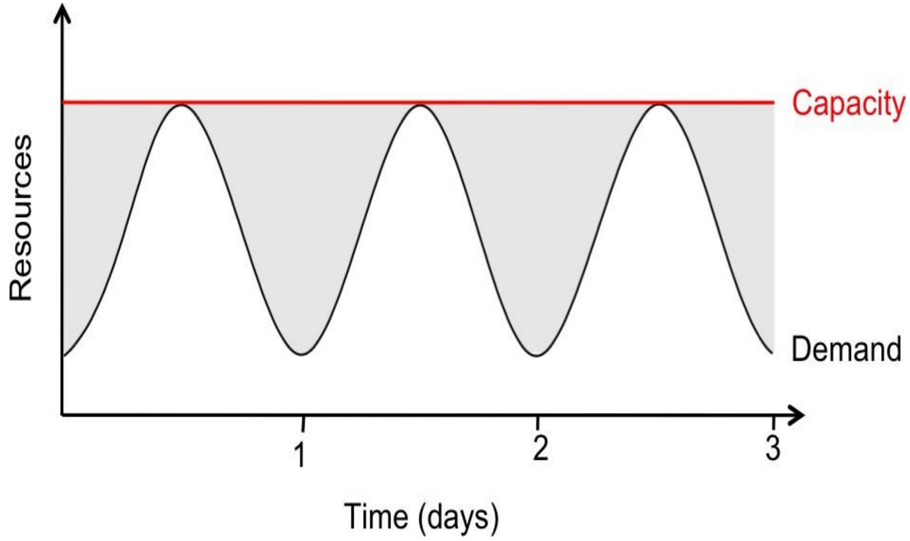


Individually fuelled
Individually maintained
Costly
Difficult to manage capacity

Elasticity – Provisioning for Peak

Real World Server Utilization Is 5% to 20%

- Many Services Peak Exceeds Average by a Factor of 2 to 10
- Most Provision for Peak
- Painful to Under-Provision (Lost Customers)



<u>Provisioning for Peak</u>
Without Elasticity,
We Waste Resources
(Shaded Areas)
During Non-Peak Times

Assignment

- You are starting a new startup with food delivery mobile app which has following requirements:
- Initially just a web server to host the web application
    - Which Computing Model should be preferred and why?
- In one year your customers have grown from 100 to 1,00,000 so the web traffic has increased
    - What will you do now?

Assignment

- In one year your customers have grown from 100 to 1,00,000 so the web traffic has increased
  - What will you do now?

If you shift to on premise server:

- You hire specialized employees for maintaining and managing the infrastructure
- Your employees keep leaving and you keep recruiting more
- You need more physical space
- You get into lot of problems and loose focus on business
- As a start up you don't have so much money also
- What of your business further grows and now you have customers from **1,00,000 to 1,00,000,000**

Shift to Cloud Computing

Unique characteristics about a cloud platform.

- First, is a hosted software platform using shared infrastructure – where another organization is providing the infrastructure.
- Designed to be virtualized and dynamic – to achieve economic benefits and provide flexibility.
- A cloud platform must be much more than just hosting code.
- We believe that a cloud platform must deliver higher level services that enable the development and deployment of applications.
- Finally, a key characteristic is that a cloud platform must provide a pay as you go pricing model. The pricing model must be as flexible as the platform.

Cloud Computing: who should use it?

- Cloud computing definitely makes sense if your own security is weak, missing features, or below average.
- Ultimately, if
  - the cloud provider’s security people are “better” than yours (and leveraged at least as efficiently),
  - the web-services interfaces don’t introduce too many new vulnerabilities, and
  - the cloud provider aims at least as high as you do, at security goals.
- Then cloud computing has better security.

From [2] John McDermott, ACSAC 09

Benefits

- Agility
- Cost
- Device & Location Independence
- Multi-Tenancy
- Reliability
- Scalability
- Security
- Sustainability

Science / Research

What is the Big Data Problem

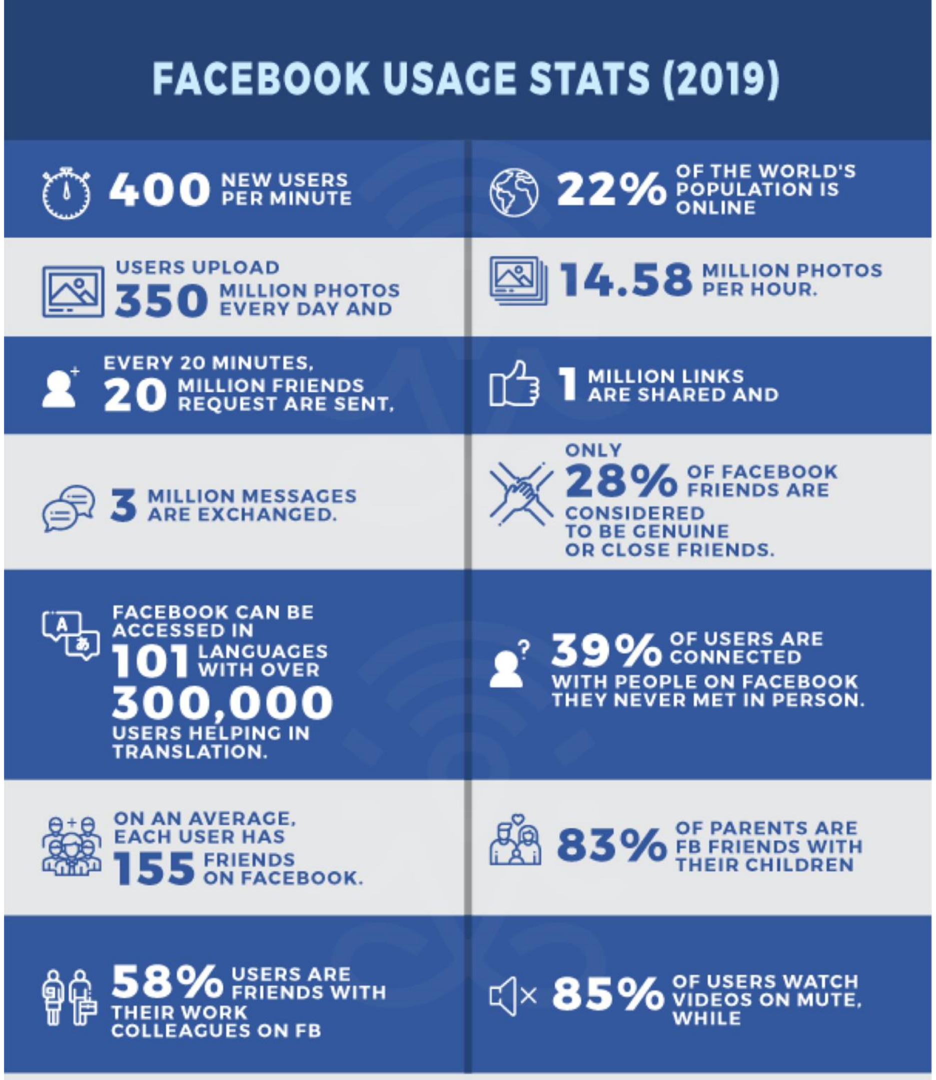
User Numbers(2018) **Google Statistics(2018)**

User numbers (2018)

- Airbnb: 150 million users
- Facebook: 2.271 billion users
- Flickr: 90 million users
- Google+: 111 million users
- Instagram: 1bn users
- LinkedIn: 562 million users
- Snapchat: 186 million daily users
- Twitter: 326 million users
- Wechat: 1.12 billion users
- Weibo: 600 million users
- WhatsApp: 900 million users
- Youtube: 1.5 billion users

- Google processes **100 billion searches a month**
- That's an average of **40,000 search queries every second**
- **91.47%** of all internet searches are carried out by Google
- Those searches are carried out by **1.17 billion unique users**
- Every day, **15%** of that day's queries have never been asked before
- Google has answered **450 billion** unique queries since 2003
- **60%** of Google's searches come from mobile devices
- By 2014, Google had indexed over **130,000,000,000,000 (130 trillion) web pages**
- To carry out all these searches, Google's data centre uses **0.01% of worldwide electricity**, although it hopes to cut its energy use by 15% using AI

Internet-Scale Application



facebook

* Core Datapoints Review & Updated October 30, **2019**, based upon **Facebook's** official investor relations information.

facebook

Worldwide, there are over 2.45 billion monthly active users (MAU) as of October 30, **2019**. This is an 8 percent increase in **Facebook** MAUs year over year.

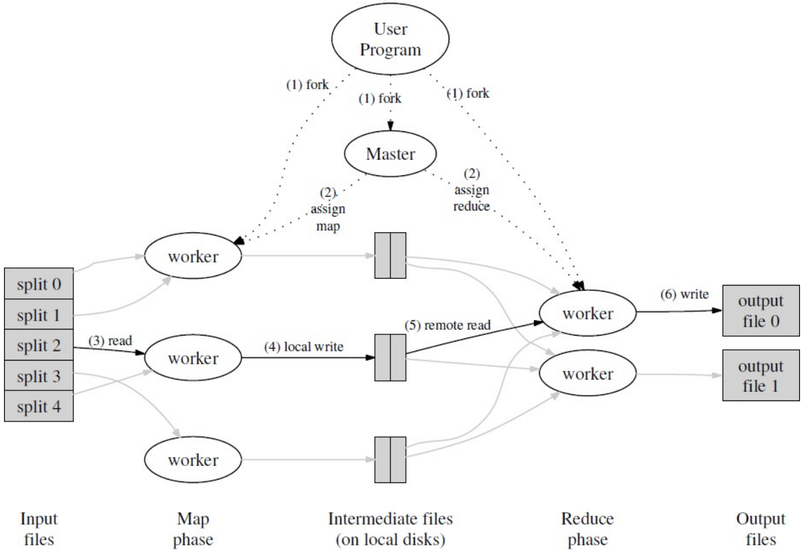
- its system processes 2.5 billion pieces of content and 500+ terabytes of **data** each **day**.
- It's pulling in 2.7 billion Like actions and 300 million photos **per day**, and it scans roughly 105 terabytes of **data** each half hour.

* Core Datapoints Review & Updated October 30, **2019**, based upon **Facebook's** official investor relations information.

Internet-Scale Application

- 2007 stats:
  - +20 petabytes of data processed / day by +100K MapReduce jobs
  - 1 petabyte sort took ~6 hours on ~4K servers replicated onto ~48K disks
  - +200 GFS clusters, each at 1-5K nodes, handling +5 petabytes of storage

- ~40 GB/sec aggregate read/write throughput across the cluster
- +500 servers for each search query < 500ms
- Scaling the process:
  - **MapReduce**: parallel processing framework
  - **BigTable**: structured hash database
  - **Google File System**: massively scalable distributed storage



Cloud Models

- **Delivery Models**
    - SaaS
    - PaaS
    - IaaS
- **Deployment Models**
    - Private cloud
    - Community cloud
    - Public cloud
    - Hybrid cloud

From [1] NIST

Cloud Models

- **Delivery Models**
  - SaaS
  - PaaS
  - IaaS

From [1] NIST

Cloud Computing – Delivery Model

PaaS

Platform as a Service

IaaS

Infrastructure as a Service

SaaS

Software as a Service

Security as a service

Cloud Computing – Delivery Model

SOFTWARE AS A SERVICE (SaaS)

- International Business machines (IBM)
- Salesforce.com (CRM)
- EFRO Tech.com (HRIS)
- Google (GOOG)
- NetSuit (N)
- Cordys
- Taleo (TLEO)
- Concur Technologies (CNQR)

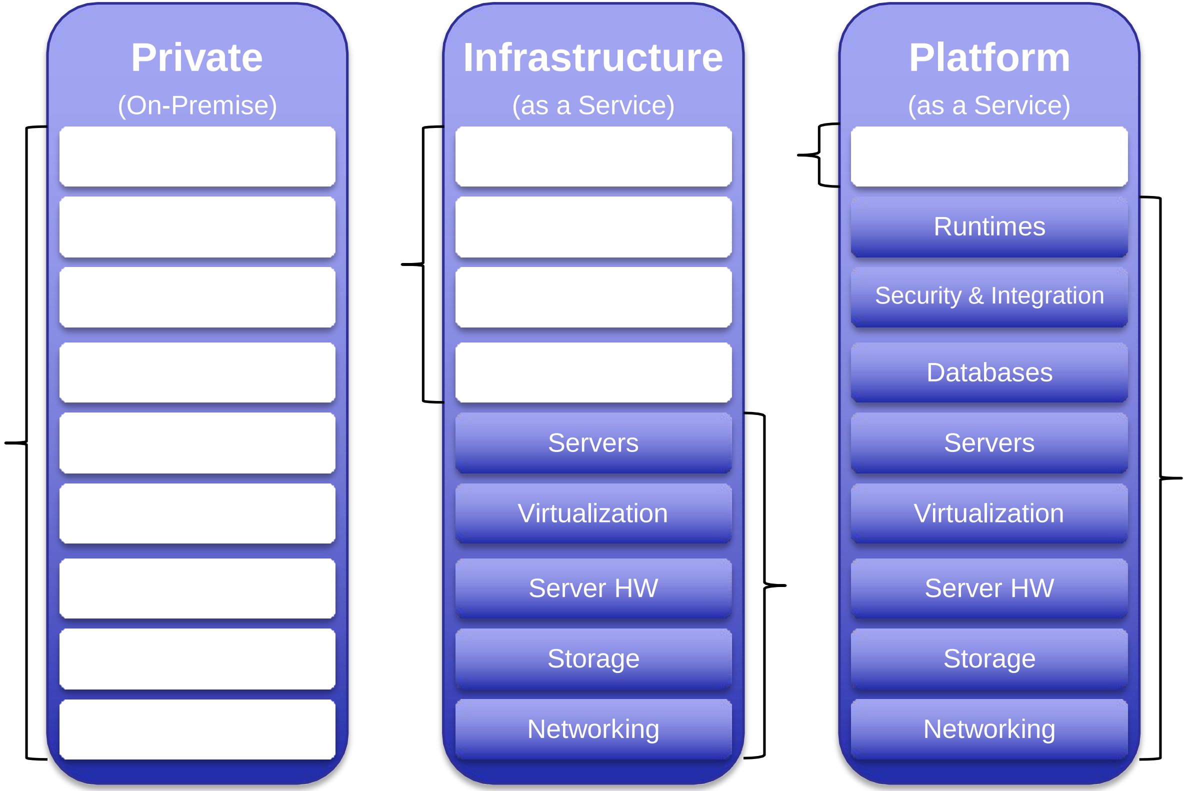
INFRASTRUCTURE AS A SERVICE (IaaS)

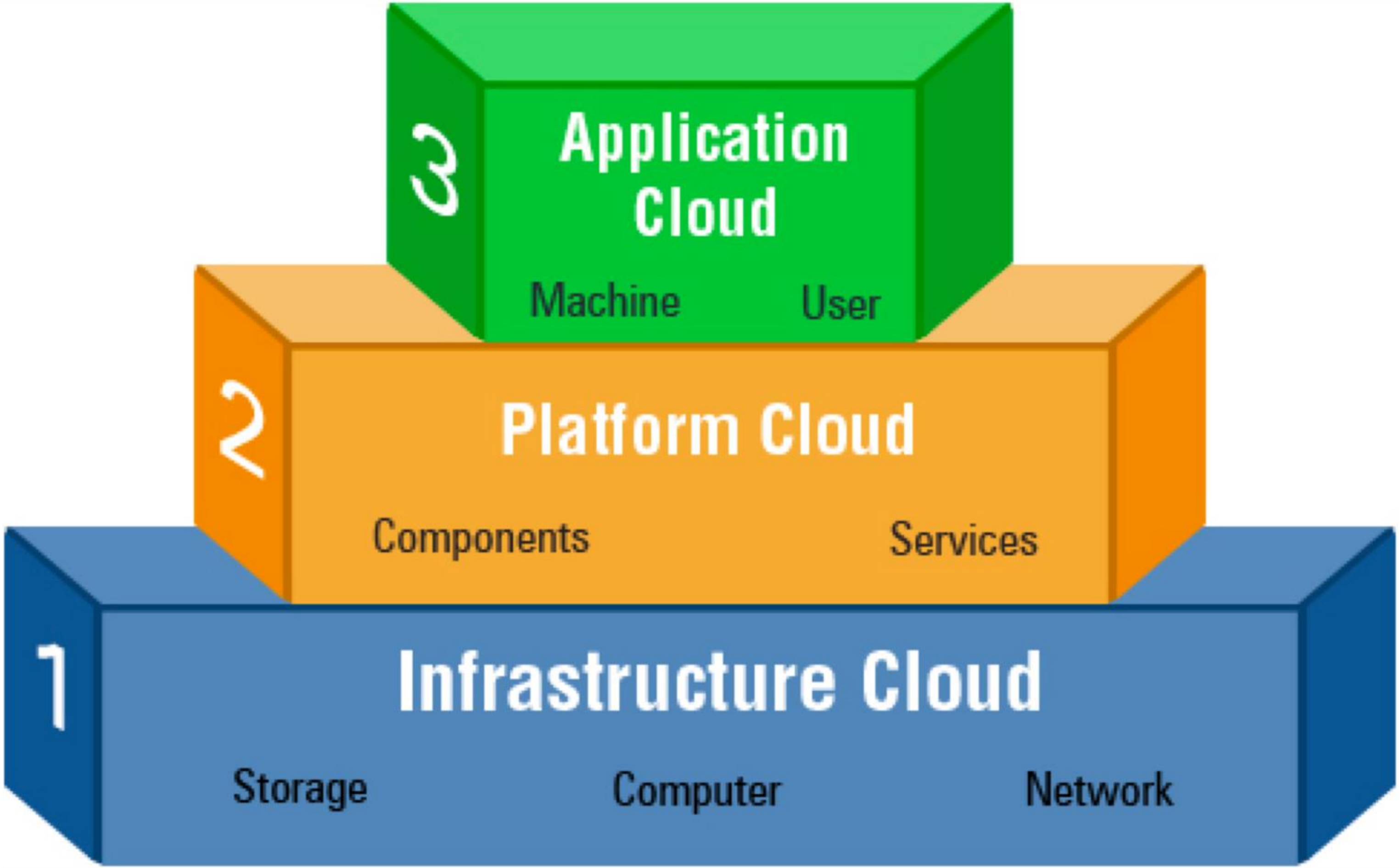
- Google (GOOG)- managed hosting, development environment
- International Business machines (IBM)- managed environment
- SAVVIS (SVVS)- managed hosting and cloud computing
- Terremark Worldwide (TRMK)- managed hosting
- Amazon.com (AMZN)- cloud storage
- Rockspace Hosting (RAX)- managed hosting and cloud computing

PLATFORM AS A SERVICE (PaaS)

- International Business machines (IBM)- IBM cloud and IBM lotus
- Google (GOOG)- Apps engines
- Amazon.com (AMZN)- EC2
- Microsoft (MSFT)- Windows Azure
- SAVVIS (SVVS)- Symphony VPDC
- Terremark Worldwide (TRMK)- The enterprise cloud
- Salesforce.com (CRM)- Force.com

IT as a Service




3
Application
Cloud
Machine
User
2
Platform Cloud
Components
Services
1
Infrastructure Cloud
Storage
Computer
Network

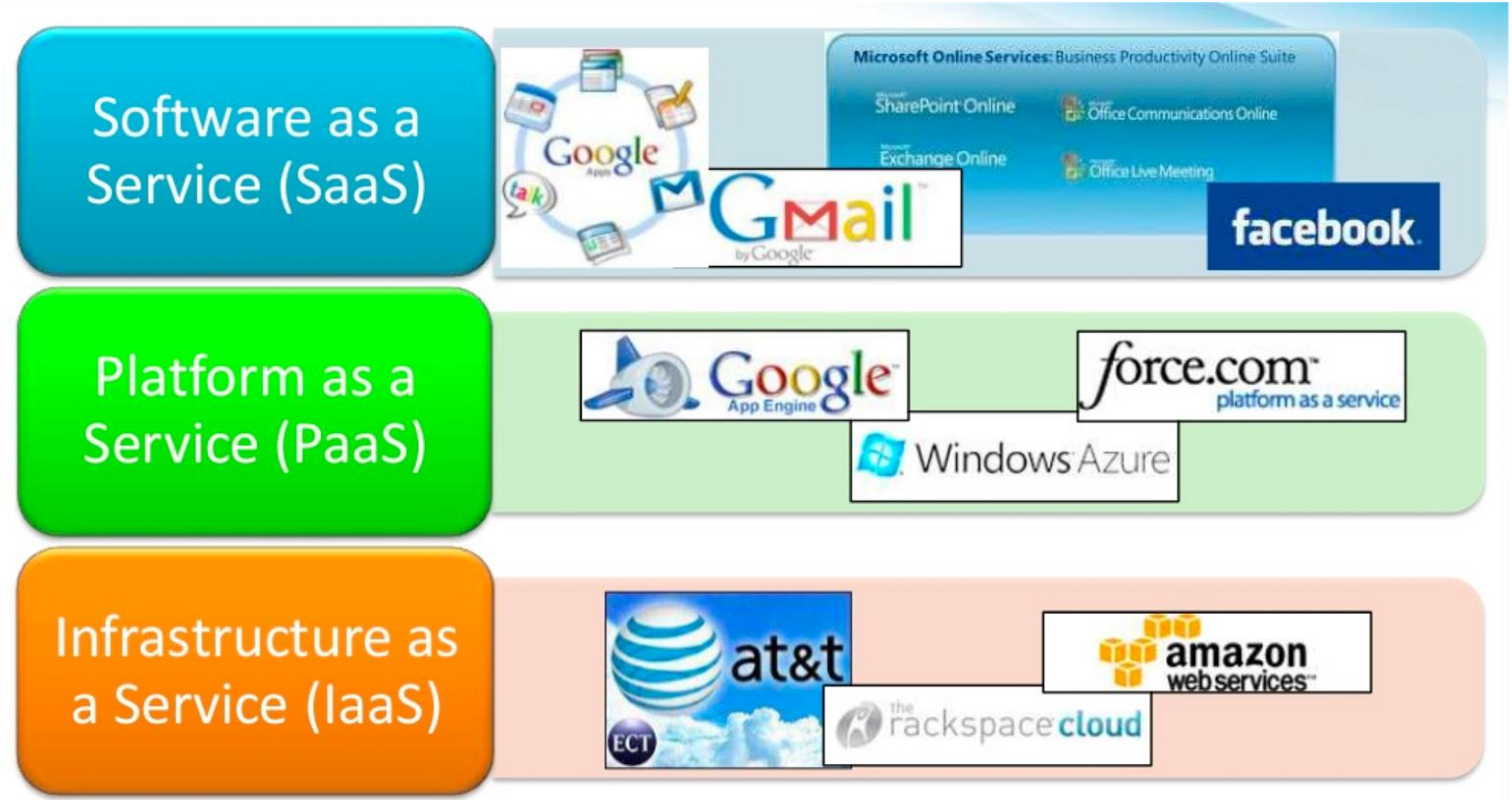
Dropbox

Google Drive

YouTube

compology

amazon drive


Software as a Service (SaaS)
Google Apps
Microsoft Online Services: Business Productivity Online Suite
SharePoint Online
Office Communications Online
Exchange Online
Office Live Meeting
Gmail by Google
facebook
Platform as a Service (PaaS)
Google App Engine
force.com platform as a service
Windows Azure
Infrastructure as a Service (IaaS)
at&t
ECT
amazon web services
the rackspace cloud

Cloud Models

- **Deployment Models**
  - Private cloud
  - Community cloud
  - Public cloud
  - Hybrid cloud

From [1] NIST

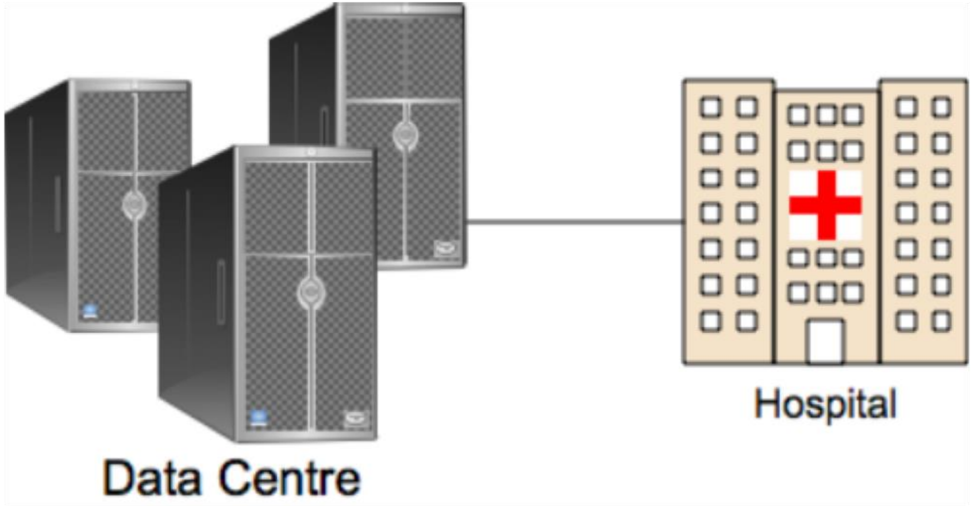
Cloud Deployment Models

- **Private cloud**
  - Cloud infrastructure is operated solely for an organization.
  - Managed by the organization or a third party and may exist on premise or off premise.
- **Community cloud**
  - The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns.
  - May be managed by the organizations or a third party and may exist on premise or off premise.

Cloud Deployment Models

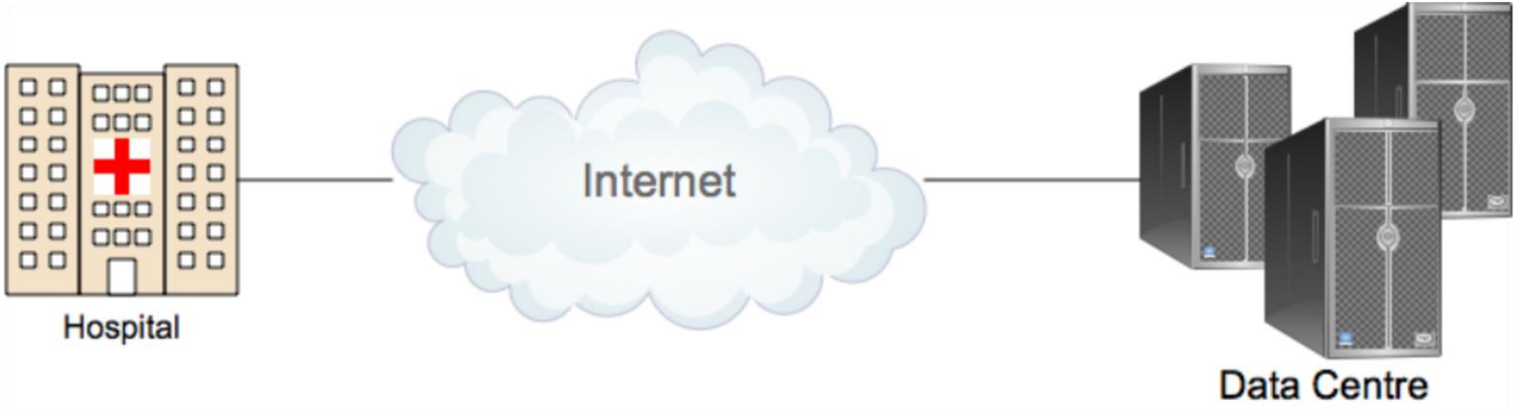
- **Public cloud**
  - The cloud infrastructure is made available to the general public or a large industry group
  - owned by an organization selling cloud services.
- **Hybrid cloud**
  - The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

Cloud Architecture



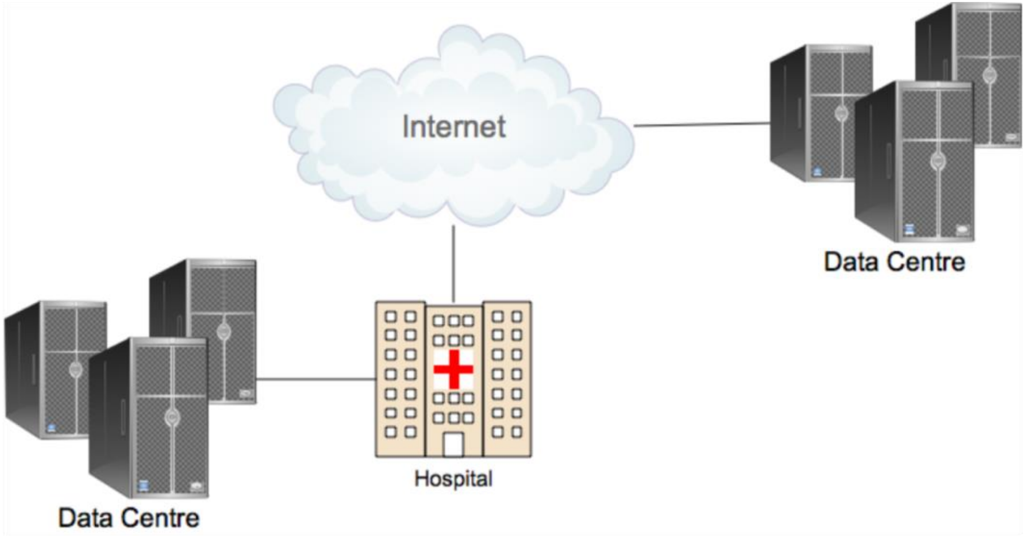
Private Cloud

Cloud Architecture



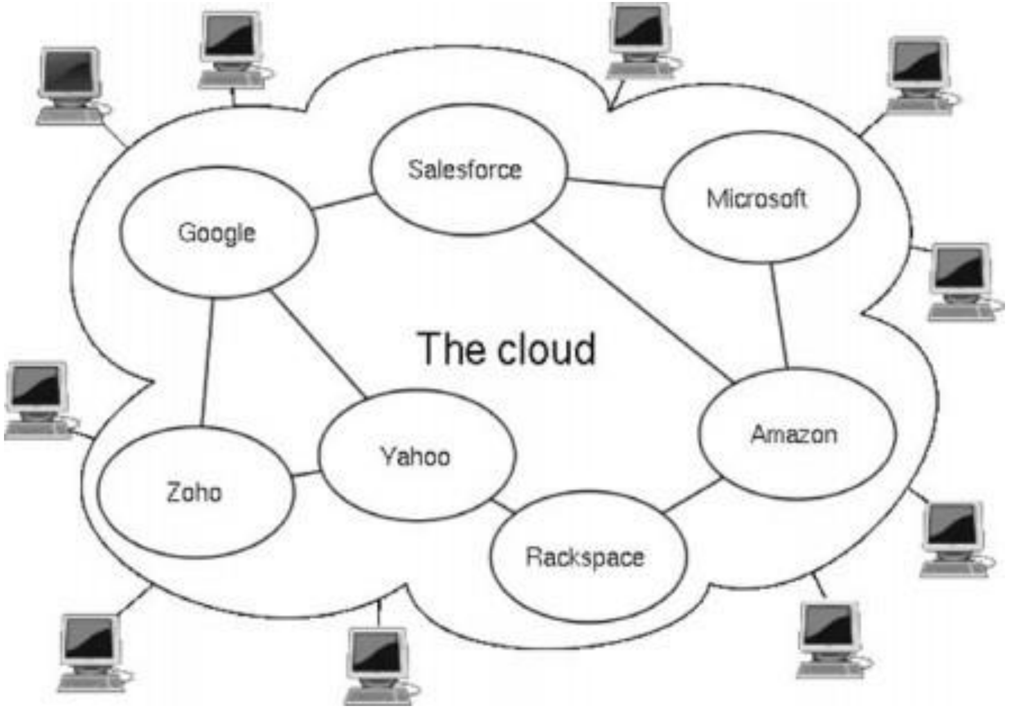
Public Cloud

Cloud Architecture



Hybrid Cloud

Cloud Computing Logical Diagram



But, How does it work?

The Cloud is typically a large data-center

and very different from a PC…

https://www.youtube.com/watch?v=DGDtujmOBKc

Task

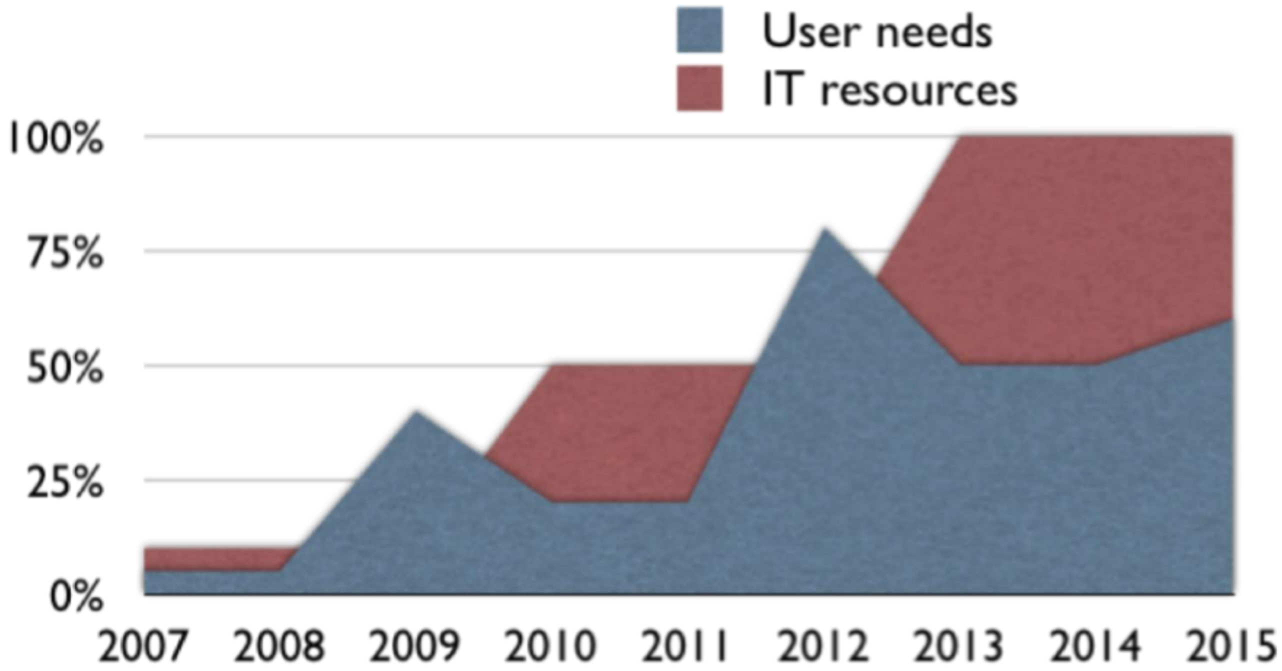
- We will do examples of Cloud Computing such as:
- Dropbox
- Setting up Google Classroom
- Google Docs/Sheet
- Google Drive

- Understand how they use different kind of Cloud Computing Delivery Models
- In later classes we will also show you how to use Amazon – Platform as a Service

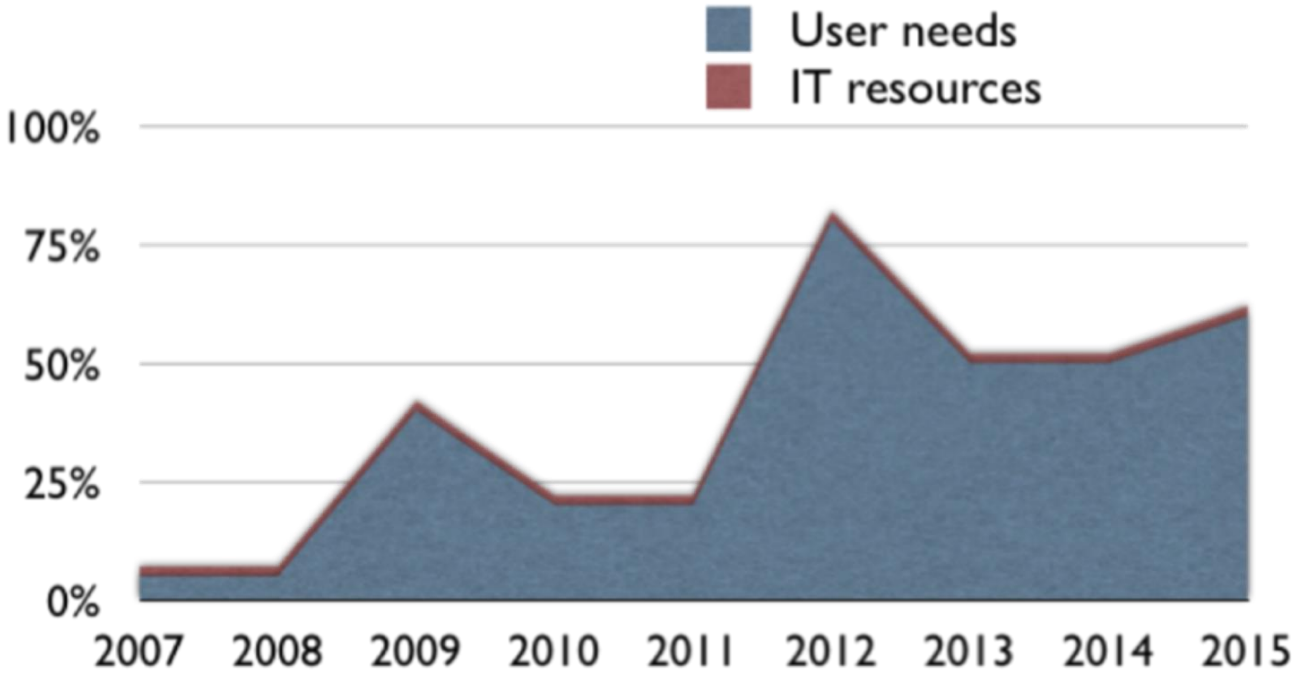
Why cloud computing

- Data centres are notoriously underutilised, often idle 85% of the time
  - Over provisioning
  - Insufficient capacity planning and sizing
  - Improper understanding of scalability requirements etc
- including thought leaders from Gartner, Forrester, and IDC—agree that this new model offers significant advantages for fast-paced startups, SMBs and enterprises alike.
- Cost effective solutions to key business demands
- Move workloads to improve efficiency

Why cloud computing - Traditional IT



Why cloud computing - Cloud



Who already used the cloud?

- Peter Harkins at The Washington Post: 200 EC2 instances (1,407 server hours), convert 17,481 pages of Hillary Clinton’s travel documents within 9 hours
- The New York Times used 100 Amazon EC2 instances + Hadoop application to recognise 4TB of raw TIFF image into 1.1 million PDFs in 24 hours ($240)
- The U.S. Defense Department is offering now cloud computing services.

Issues in Cloud Computing?

1. Web-scale problems
2. Large data centers
3. Different models of computing
4. Highly-interactive Web applications

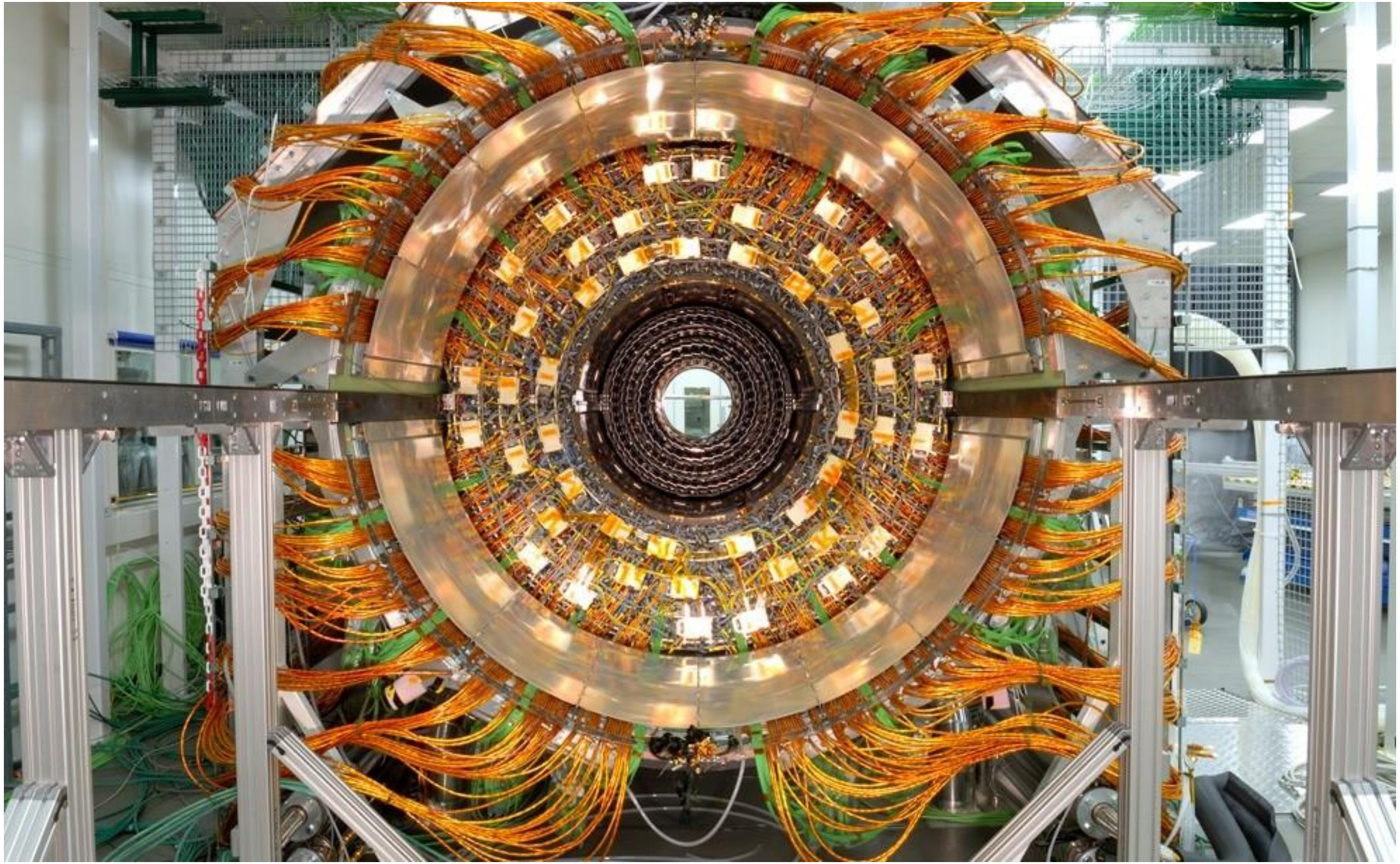
Web-Scale Problems

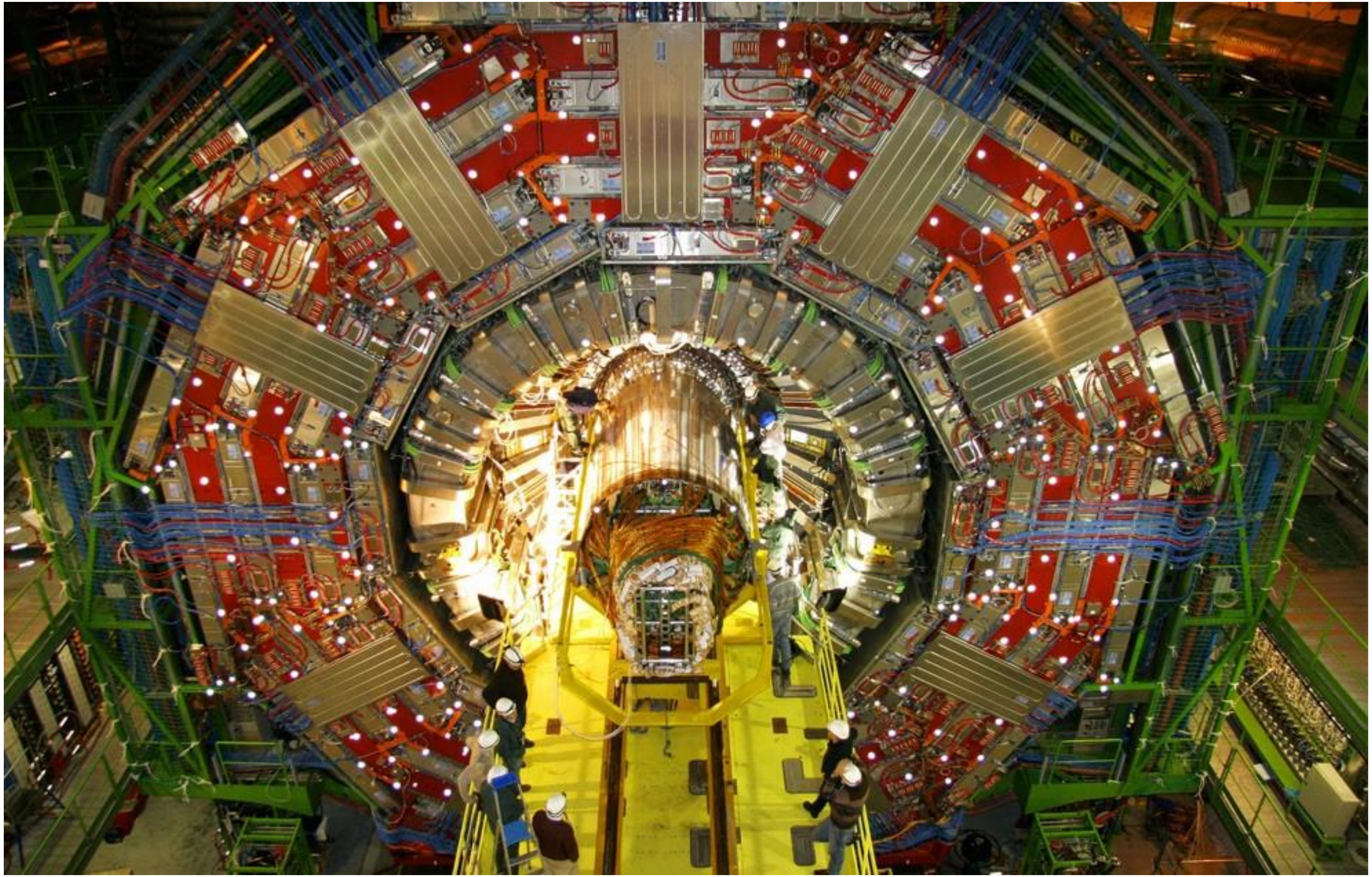
- Characteristics:
  - Definitely data-intensive
  - May also be processing intensive
- Examples:
  - Crawling, indexing, searching, mining the Web
  - “Post-genomics” life sciences research
  - Other scientific data (physics, astronomers, etc.)
  - Sensor networks
  - Web 2.0 applications
  - …

How much data?

- Wayback Machine has 2 PB + 20 TB/month (2006)
- Google processes 20 PB a day (2008)
- “all words ever spoken by human beings” ~ 5 EB
- NOAA has ~1 PB climate data (2007)
- CERN’s LHC will generate 15 PB a year (2008)

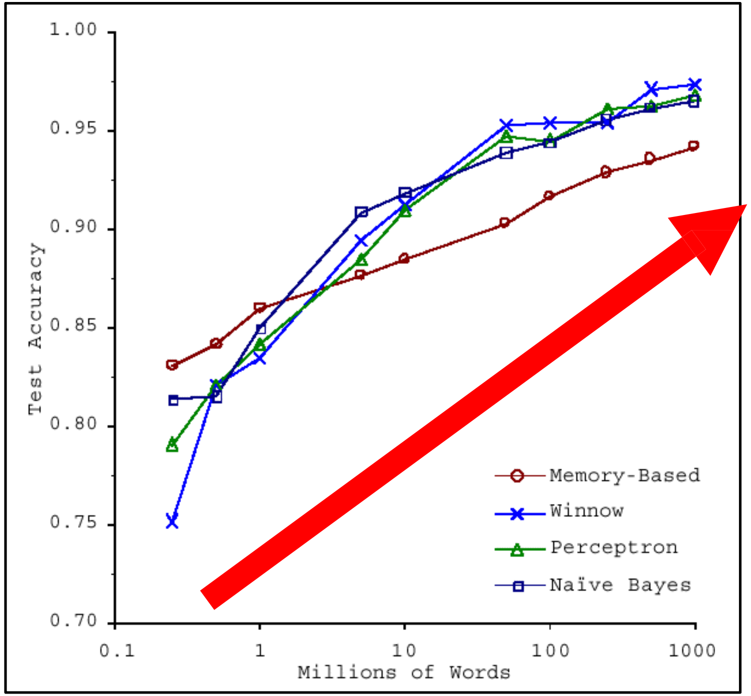
640K ought to be enough for anybody.

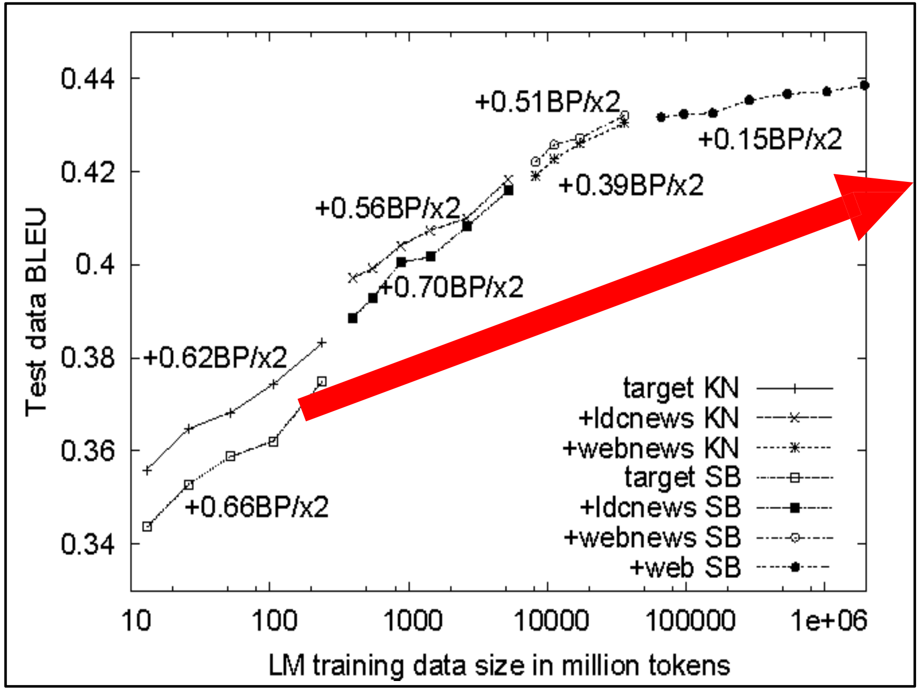




Maximilien Brice, © CERN

There's nothing like more data!



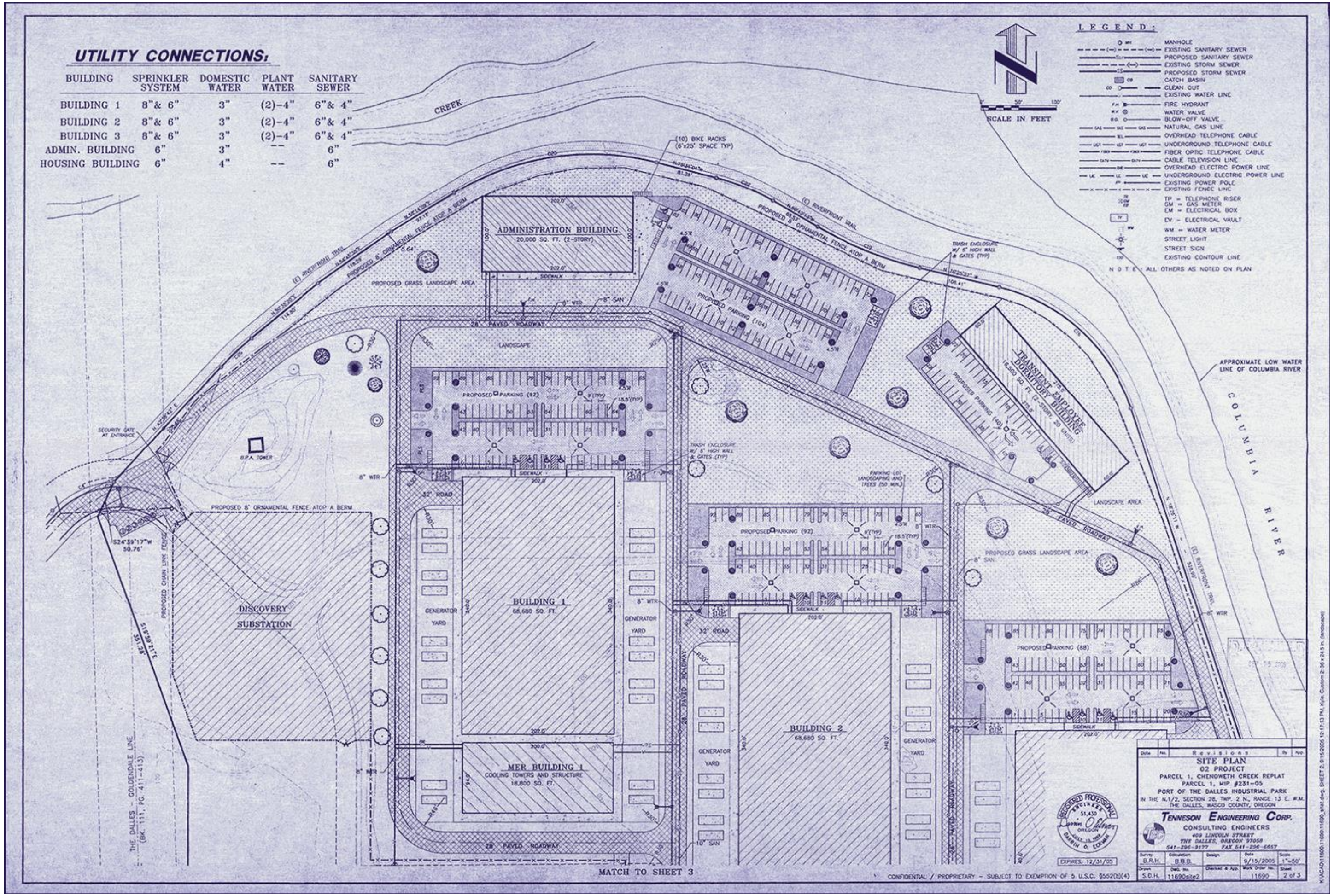


(Banko and Brill, ACL 2001)

(Brants et al., EMNLP 2007)

Large Data Centers

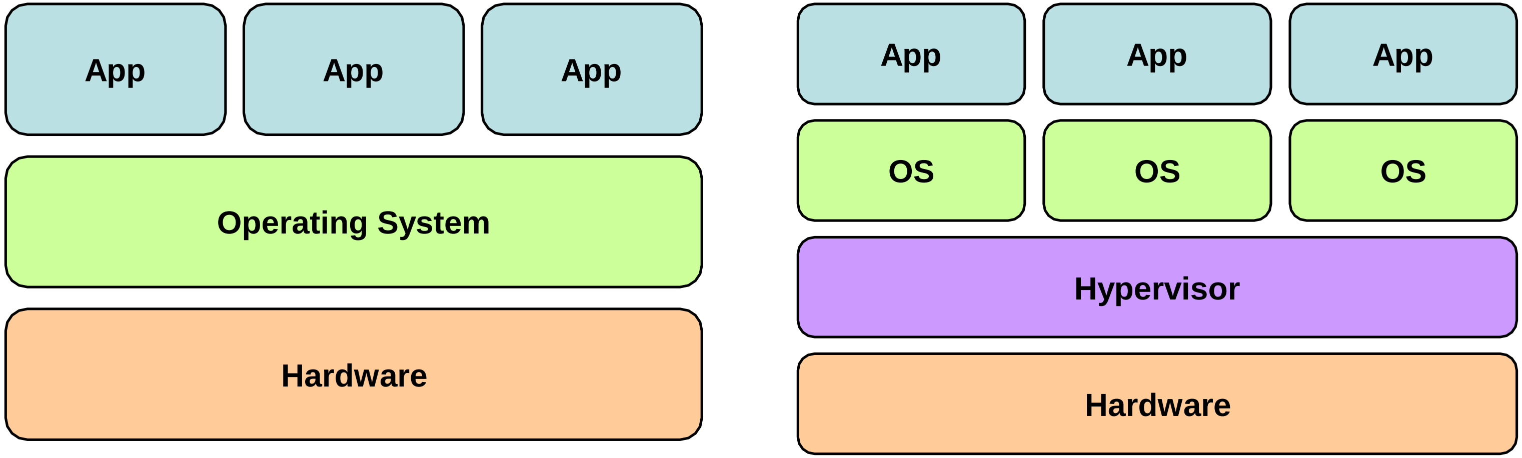
- Web-scale problems? Throw more machines at it!
- Clear trend: centralization of computing resources in large data centers
  - Necessary ingredients: fiber, juice, and space
  - What do Oregon, Iceland, and abandoned mines have in common?
- Important Issues:
  - Redundancy
  - Efficiency
  - Utilization
  - Management



Source: Harper's (Feb, 2008)

Maximilien Brice, © CERN

Key Technology: Virtualization



Traditional Stack

Virtualized Stack

Different Computing Models

"Why do it yourself if you can pay someone to do it for you?"

- Utility computing
  - Why buy machines when you can rent cycles?
  - Examples: Amazon's EC2, GoGrid, AppNexus
- Platform as a Service (PaaS)
  - Give me nice API and take care of the implementation
  - Example: Google App Engine
- Software as a Service (SaaS)
  - Just run it for me!
  - Example: Gmail

Web Applications

- A mistake on top of a hack built on sand held together by duct tape?
- What is the nature of software applications?
  - From the desktop to the browser
  - SaaS == Web-based applications
  - Examples: Google Maps, Facebook
- How do we deliver highly-interactive Web-based applications?
  - AJAX (asynchronous JavaScript and XML)
  - For better, or for worse…

What else does the course include?

- MapReduce: the “back-end” of cloud computing
    - Batch-oriented processing of large datasets
- Ajax: the “front-end” of cloud computing
    - Highly-interactive Web-based applications
- Computing “in the clouds”
    - Amazon’s EC2/S3 as an example of utility computing

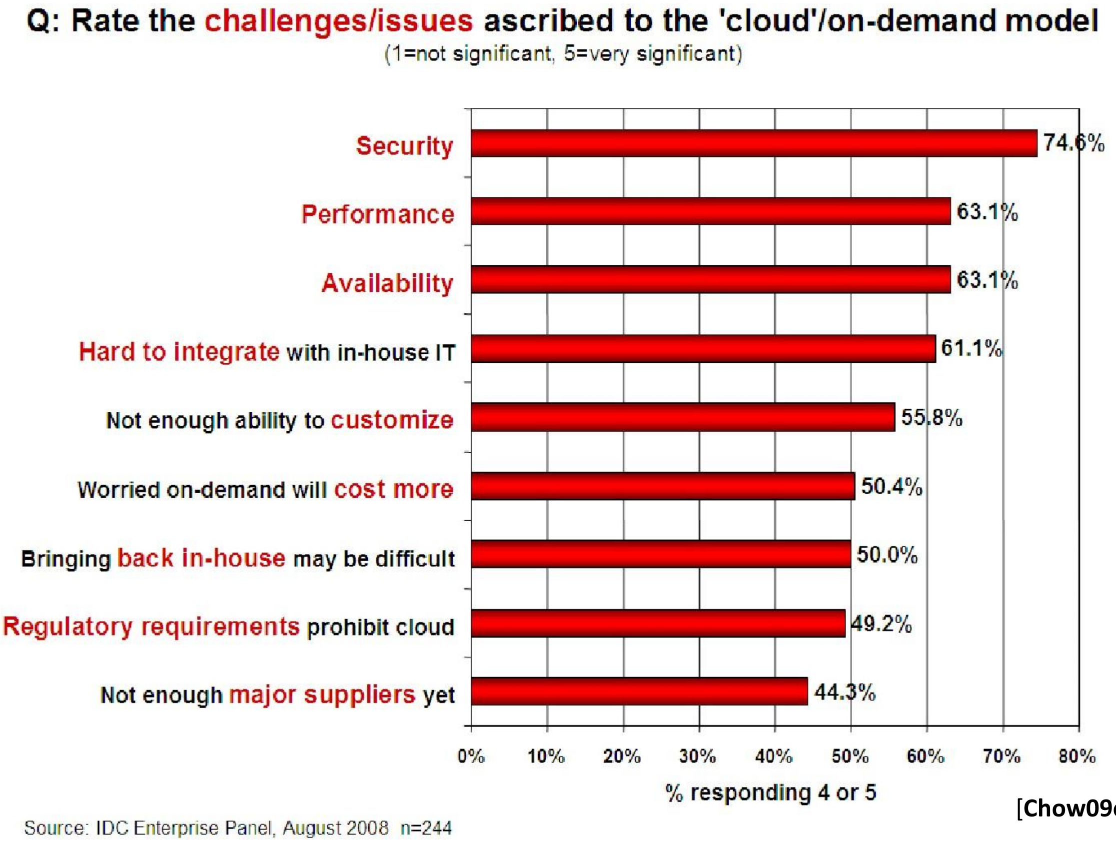
Amazon Web Services

- Elastic Compute Cloud (EC2)
  - Rent computing resources by the hour
  - Basic unit of accounting = instance-hour
  - Additional costs for bandwidth
- Simple Storage Service (S3)
  - Persistent storage
  - Charge by the GB/month
  - Additional costs for bandwidth
- You’ll be using EC2/S3 for course assignments!

If cloud computing is so great, why isn't everyone doing it?

- The cloud acts as a big black box, nothing inside the cloud is visible to the clients
- Clients have no idea or control over what happens inside a cloud
- Even if the cloud provider is honest, it can have malicious system admins who can tamper with the VMs and violate confidentiality and integrity
- Clouds are still subject to traditional data confidentiality, integrity, availability, and privacy issues, plus some additional attacks

Companies are still afraid to use clouds



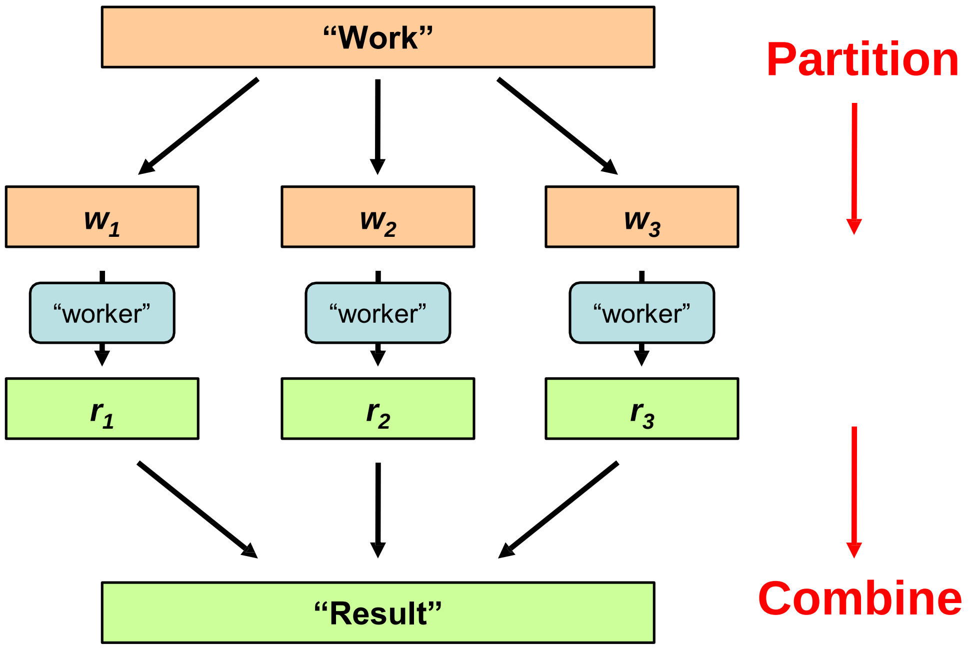
[Chow09ccsw]

Web-Scale Problems?

- Don’t hold your breath:
  - Biocomputing
  - Nanocomputing
  - Quantum computing
  - …
- It all boils down to…
  - Divide-and-conquer
  - Throwing more hardware at the problem

Simple to understand… a lifetime to master…

Divide and Conquer



Different Workers

- Different threads in the same core
- Different cores in the same CPU
- Different CPUs in a multi-processor system
- Different machines in a distributed system

Choices, Choices, Choices

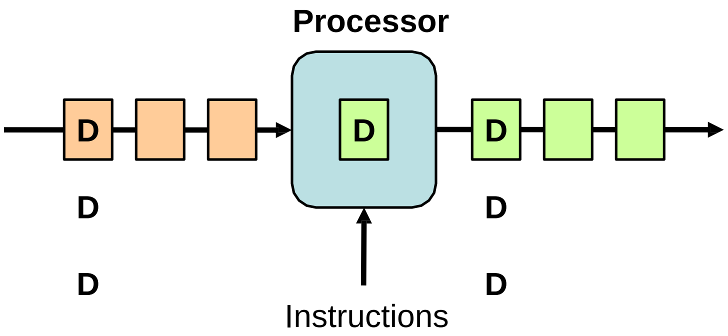
- Commodity vs. “exotic” hardware
- Number of machines vs. processor vs. cores
- Bandwidth of memory vs. disk vs. network
- Different programming models

Flynn’s Taxonomy

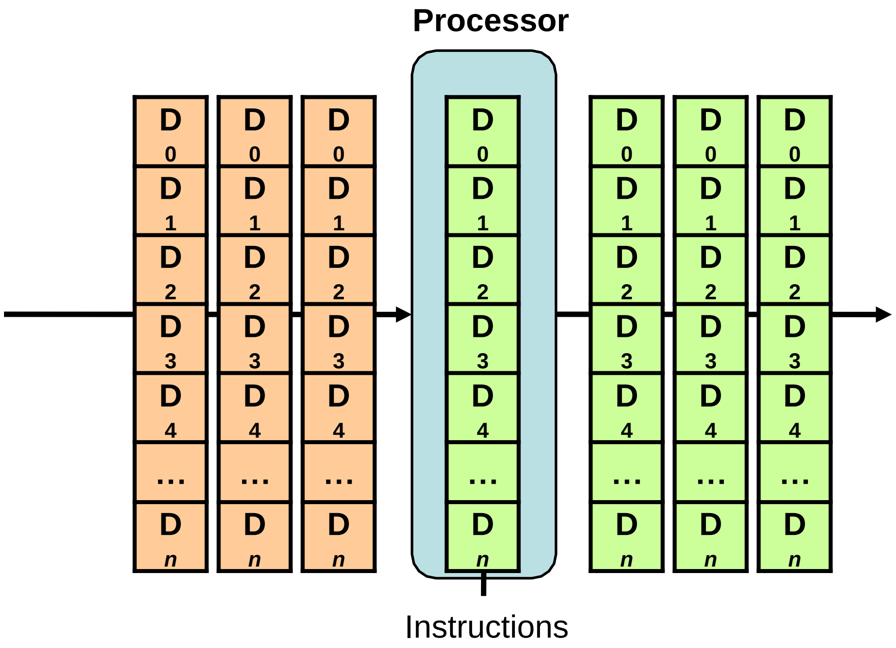
Instructions

Data	Single (SI)	Multiple (MI)
Single (SD)	**SISD** Single-threaded process	**MISD** Pipeline architecture
Multiple (MD)	**SIMD** Vector Processing	**MIMD** Multi-threaded Programming

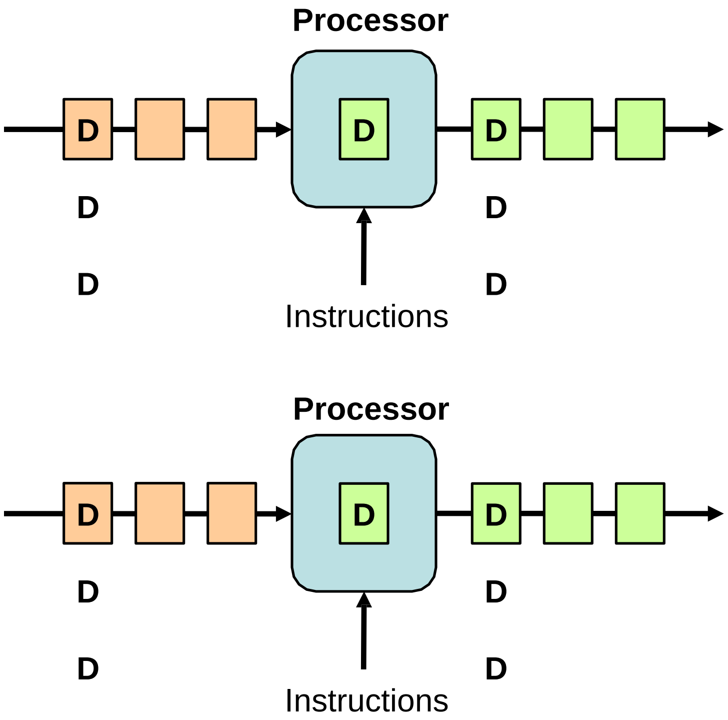
SISD



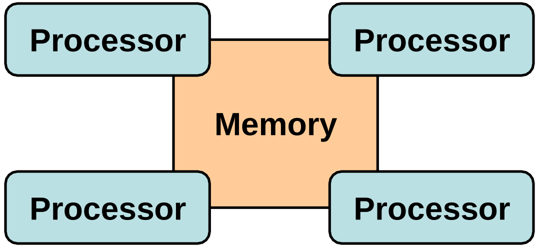
SIMD



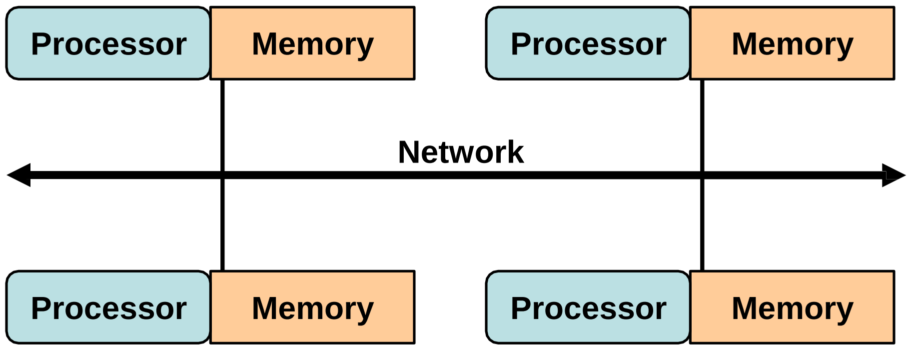
MIMD


Processor
D
D
D
D
D
D
D
D
Instructions
Processor
D
D
D
D
D
D
D
D
Instructions

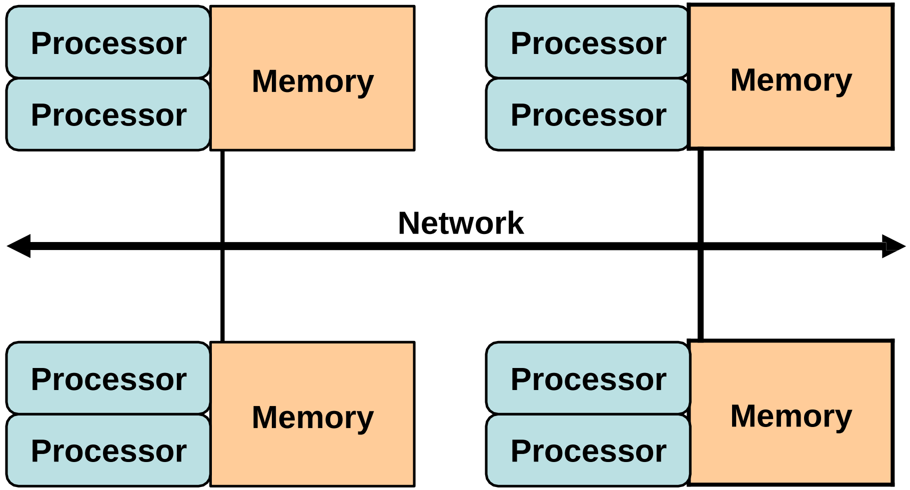
Memory Typology: Shared



Memory Typology: Distributed



Memory Typology: Hybrid



Parallelization Problems

- How do we assign work units to workers?
- What if we have more work units than workers?
- What if workers need to share partial results?
- How do we aggregate partial results?
- How do we know all the workers have finished?
- What if workers die?

What is the common theme of all of these problems?

General Theme?

- Parallelization problems arise from:
  - Communication between workers
  - Access to shared resources (e.g., data)
- Thus, we need a synchronization system!
- This is tricky:
  - Finding bugs is hard
  - Solving bugs is even harder

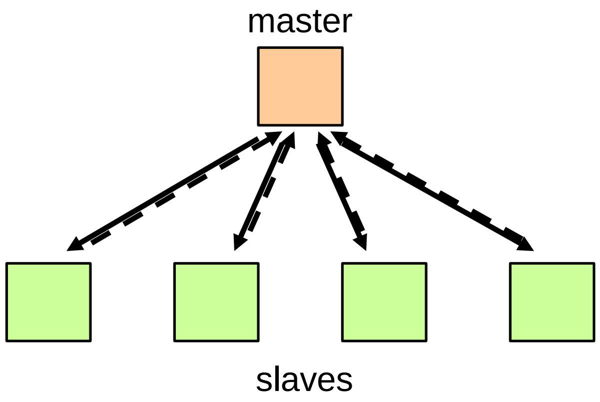
Managing Multiple Workers

- Difficult because
  - (Often) don’t know the order in which workers run
  - (Often) don’t know where the workers are running
  - (Often) don’t know when workers interrupt each other
- Thus, we need:
  - Semaphores (lock, unlock)
  - Conditional variables (wait, notify, broadcast)
  - Barriers
- Still, lots of problems:
  - Deadlock, livelock, race conditions, ...
- Moral of the story: be careful!
  - Even trickier if the workers are on different machines

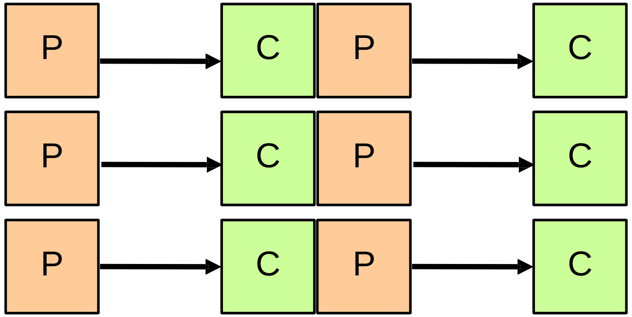
Patterns for Parallelism

- Parallel computing has been around for decades
- Here are some “design patterns” …

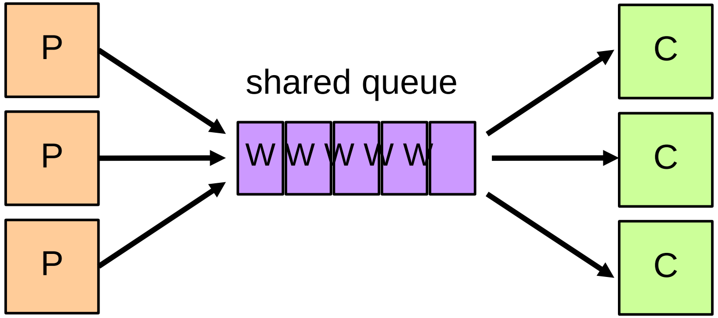
Master/Slaves



Producer/Consumer Flow



Work Queues



Rubber Meets Road

- From patterns to implementation:
  - pthreads, OpenMP for multi-threaded programming
  - MPI for clustering computing
  - …
- The reality:
  - Lots of one-off solutions, custom code
  - Write you own dedicated library, then program with it
  - Burden on the programmer to explicitly manage everything
- MapReduce to the rescue!
  - (for next class)

Thank you