ST446 Distributed Computing for Big Data

Week 1: Introduction

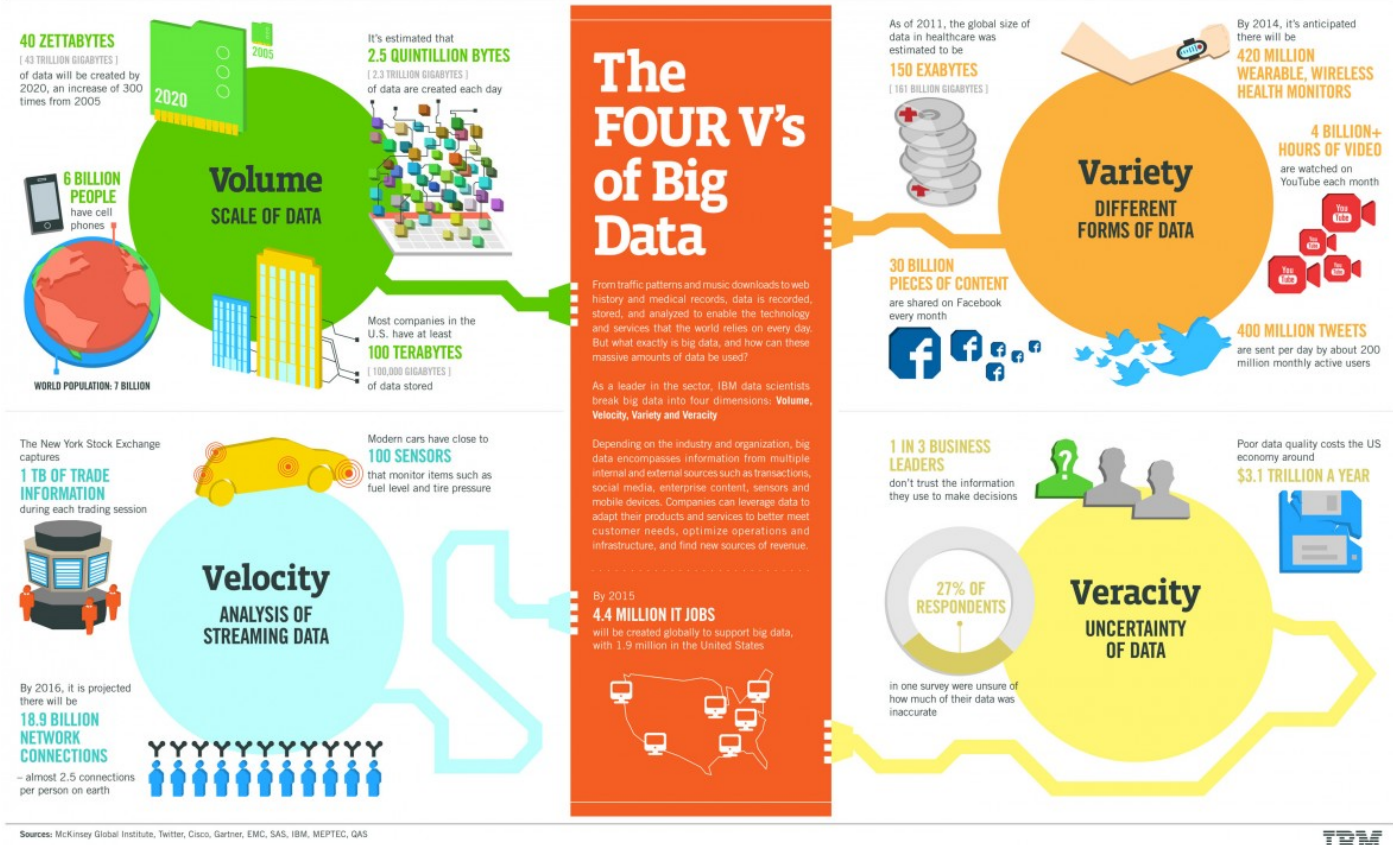
Milan Vojnovic, LT 2018

Outline

- What is big data?
- Data sources: unstructured, semi-structured, structured
- The world of SQL databases
- Modern big data analytics systems

Big Data

- The term in use since 1990s, no single universally accepted definition
- Our working definition:
  - A dataset whose *size* or *complexity* make traditional data processing application software inadequate
- Big data has at least one of the following properties:
  - **Volume**: large quantity of generated and stored data
  - **Variety**: data may be of different type
  - **Velocity**: data may be generated at different speeds
  - **Veracity**: the quality of captured data can vary greatly



Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

IBM

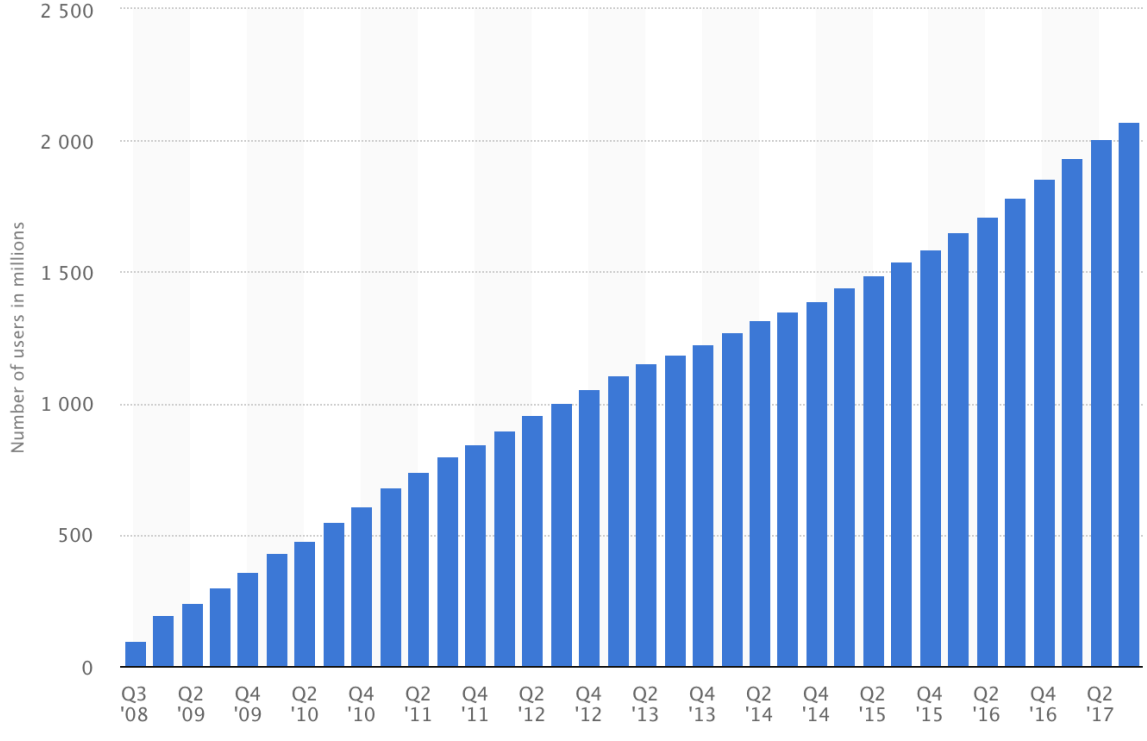
Data volume units

unit	abbreviation	total bytes	nearest decimal
kilobyte	**KB**	1024	10^3
megabyte	**MB**	1024^2	10^6
gigabyte	**GB**	1024^3	10^9
terabyte	**TB**	1024^4	10^{12}
petabyte	**PB**	1024^5	10^{15}
exabyte	**EB**	1024^6	10^{18}
zettabyte	**ZB**	1024^7	10^{21}
yottabyte	**YB**	1024^8	10^{24}

Data volume examples

- **Human population**: **7.6B** people
- **Mobile phones**: **4.77B** devices
- **Wikipedia**: **43,783,651** total wiki pages, **5,553,854** English articles, as of Dec 2017; **10TB** uncompressed, **100GB** compressed (7-zip) size of the dump of all pages with complete edit history in XML format as of Jun 2015
- **WWW**: **>= 4.56B** web pages
- **Facebook**: **2.07B** monthly active users, **1.37B** daily active users, **300M** photo uploads per day, **338** friends per user in 2017
- **Linkedin**: **500M** users
- **Twitter**: **328M** monthly active users
- **Google knowledge graph**: more than **70M** facts
- **Yago knowledge base**: more than **10M** entities and **120M** facts
- **Criteo Terabyte click logs**: **50GB** raw data per day, **195M** ad impressions per day, **26** features, **24** days
- **ImageNet dataset**: **14,197,122** images, **1.2TB** compressed size

Facebook: number of monthly active users



Source (https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/)

- Growing user base: need systems that can scale

Data source types

- **Structured**: any data that has a *schema*, i.e. a known set of fields for each record
  - Relational Database Management Systems (RDBMS) such as Oracle database, IBM DB2, Microsoft SQL Server
- **Semi-structured**: a form of structured data that does not conform with the formal structure of data models associated with *relational databases* or other forms of *data tables*, but contains *tags* or other markers to separate semantic elements and enforce hierarchies of records and fields within the data
  - XML (Extensible Markup Language) an other markup languages, JSON (JavaScript Object Notation), ...
- **Unstructured**: a data that either does not have a pre-defined data model or is not organized in a pre-defined manner
  - Textual data, speech, video, ...

Note: semi-structured data discussed in Buneman's PODS paper (https://homepages.inf.ed.ac.uk/opb/papers/PODS1997a.pdf)

Structured data sources

Field	Type	Null	Key
Host	char(60)	NO	PRI
User	char(32)	NO	PRI
Select_priv	enum('N','Y')	NO	
Insert_priv	enum('N','Y')	NO	
Update_priv	enum('N','Y')	NO	
Delete_priv	enum('N','Y')	NO	
Create_priv	enum('N','Y')	NO	
Drop_priv	enum('N','Y')	NO	
x509_issuer	blob	NO	
max_connections	int(11) unsigned	NO	
plugin	char(64)	NO	
authentication_string	text	YES	
password_last_changed	timestamp	YES	
password_lifetime	smallint(5) unsigned	YES	
account_locked	enum('N','Y')	NO	

Note: thinned down table from mysql database (so that it can fit into a slide)

Semi-structured data examples

- Delimited-separated file formats (e.g. csv and tab files)
- JSON files
- XML files
- ...

Delimiter-separated file formats

- File format for storing tabular data in plain text
- Each line of the file is a data record
- Each record consists of one or more fields, separated by a delimiter
- Common formats:
  - **csv**: delimiter is a comma (`,`)
  - **tsv**: delimiter is a tab (`\t`)
- The csv file format is not standardized

For more information on csv, see here (https://en.wikipedia.org/wiki/Comma-separated_values)

tsv example: Yago semantic knowledge base

From `yagoFacts.tsv`:

```
<id_rB6isMnplh_H?S_LT?Qo1Fzc!>  <Jesús_Rivera_Sánchez>  <isLeaderOf>    <Pueblo_o
f_Naranjito>
<id_BOK!FvTDPu_H?S_1NVSTKkFbS>  <Elizabeth_II>  <isLeaderOf>    <Royal_Numismatic
_Society>
<id_Uy5EwU3nX1_H?S_otuJrkvKs1>  <Richard_Stallman>      <isLeaderOf>    <Free_Sof
tware_Foundation>
<id_A?rIHtKpyX_H?S_Ap3TBzfE6b>  <Keith_Peterson>        <isLeaderOf>    <Cambridg
e_Bay>
<id_vzhzgmCR5Y_H?S_9xW07qYiaH>  <William_H._Seward_Jr.> <isLeaderOf>    <9th_New_
York_Heavy_Artillery_Regiment>
<id_GqUh9jFAN?_H?S_A39fu5FWu4>  <Andranik>      <isLeaderOf>    <Armenian_fedayi>

<id_s60Psk1DHb_H?S_OACCn8W8Kv>  <Ramasamy_Palanisamy>   <isLeaderOf>    <Democrat
ic_Action_Party_(Malaysia)>
<id_pII60Mnz8o_H?S_8mvRWxKXDG>  <Matt_Bevin>    <isLeaderOf>    <Kentucky_Air_Nat
ional_Guard>
...
```

JSON

- An open-standard file format that uses human-readable text to transmit data objects
- Objects consist of attribute-value pairs and array data types
- Commonly used data format for asynchronous browser-server communication
- A language-indepedent data format
  - Derived from JavaScript, specified in early 2000s
  - Two competing specifications (RFC 8259 and ECMA-404)
  - Many programming languages include code to generate and parse JSON files
- Basic data types: number (signed decimal), string, boolean, array, object (an unordered collection of name (key)-value pairs), null

JSON example: Yelp reviews


```
{
"business_id":"AGN788ObhwXu7rb8MEejIA",
"name":"Sheik Shoes",
"neighborhood":"Westside",
"address":"4300 Meadows Ln",
"city":"Las Vegas",
"state":"NV",
"postal_code":"89107",
"latitude":36.172259,
"longitude":-115.1963237,
"stars":1.5,
"review_count":4,
"is_open":1,
"attributes":["BusinessAcceptsCreditCards: True","RestaurantsPriceRange2: 2","Whe
elchairAccessible: True"],
"categories":["Shoe Stores","Fashion","Shopping"],
"hours":null,
"type":"business"
}
```

XML

- A markup language that defines a set of rules for encoding documents in a format that is both human and machine readable
- Defined by an open standard (W3C's XML 1.0 Spec and several other related specifications)
- Key concepts:
  - **Markup and content**: text that defines the structure and content, respectively
  - **Tag**: a markup construct that begins with < and ends with > (three types: start, end, empty element `<line-break />`)
  - **Element**: a logical document component that either begins with a start tag and ends with a matching end tag, or consists only of an empty element tag
  - **Attribute**: an attribute is a markup construct consisting of a name-value pair that exists within a start tag or empty element tag
  - **XML declaration**: XML documents may begin with an XML declaration, e.g., `<?xml version="1.0" encoding="UTF-8"?>`

XML example

```
<person>
  <firstname>Milan</firstname>
  <lastname>Vojnovic</lastname>
  <address>
    <streetaddress>70 Girton Road</streetaddress>
    <city>Cambridge</city>
    <postcode>CB30LN</postcode>
  </address>
  <phonenumber>
    <type>home</type>
    <number>xxx</number>
  </phonenumber>
  <phonenumber>
    <type>mobile</type>
    <number>xxx</number>
  </phonenumber>
</person>
```

XML example (cont'd)

```
<person firstname="Milan" lastname="Vojnovic">
        <address streetaddress="70 Girton Road" city="Cambridge" postcode="CB30L
N"/>
        <phonenumber type="home" number="xxx"/>
        <phonenumber type="mobile" number="yyy"/>
</person>
```

Here `firstname="Milan"` is an attribute

dblp example

dblp (http://dblp.uni-trier.de/): computer science bibliography

```
<dblp>
<article mdate="2017-05-28" key="journals/acta/Saxena96">
<author>Sanjeev Saxena</author>
<title>Parallel Integer Sorting and Simulation Amongst CRCW Models.</title>
<pages>607-619</pages>
<year>1996</year>
<volume>33</volume>
<journal>Acta Inf.</journal>
<number>7</number>
<url>db/journals/acta/acta33.html#Saxena96</url>
<ee>https://doi.org/10.1007/BF03036466</ee>
</article>

    ...
</dblp>
```

Available from dblp dump (http://dblp.uni-trier.de/xml/)

Resource Description Framework (RDF)

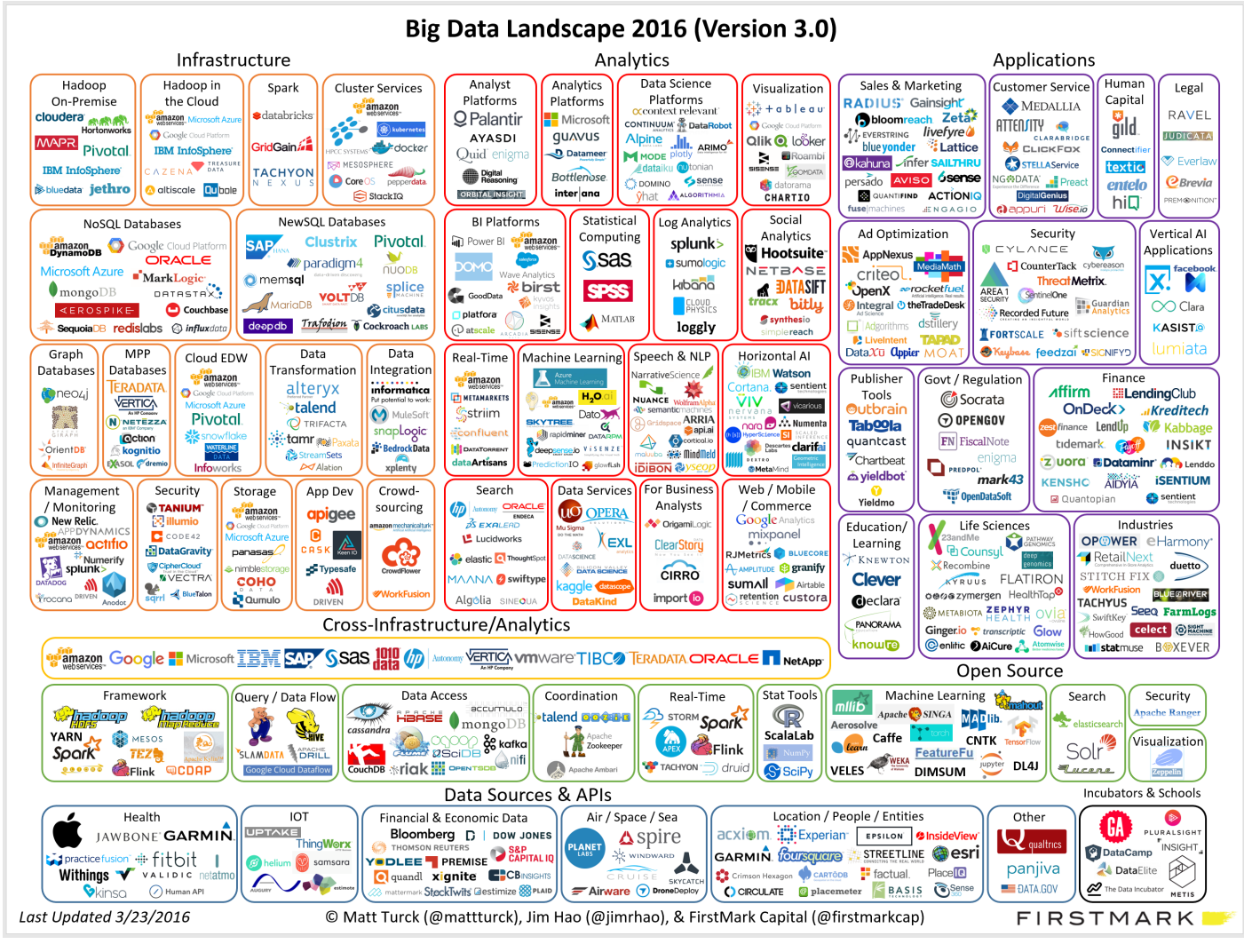
- A family of W3C specifications originally designed as a metadata data model expressed in XML
- Based on the idea of making *statements* about resources in expressions of the form *subject-predicate-object*, known as *triples*
- Query and inference languages: predominant is SPARQL (SQL like)

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>
.
.
.
</rdf:RDF>
```

For more information see W3C semantic web standards (https://www.w3.org/RDF/), w3schools (https://www.w3schools.com/xml/xml_rdf.asp)



- Big data analytics technology space is big with different solutions available on the market, often with overlapping functionalities
- But how come the technology evolved as given now?

Relational database management systems

- The concept of relational data model dates back all the way to 1970s
  - Codd, E. F., A relational model of data for large shared data banks, 1970
- Support for transactions: online transaction processing (OTLP)
  - Transactions: a single logical operation on the data
  - Atomic operations ACID (next slide)
  - Imperative and declerative programming

ACID

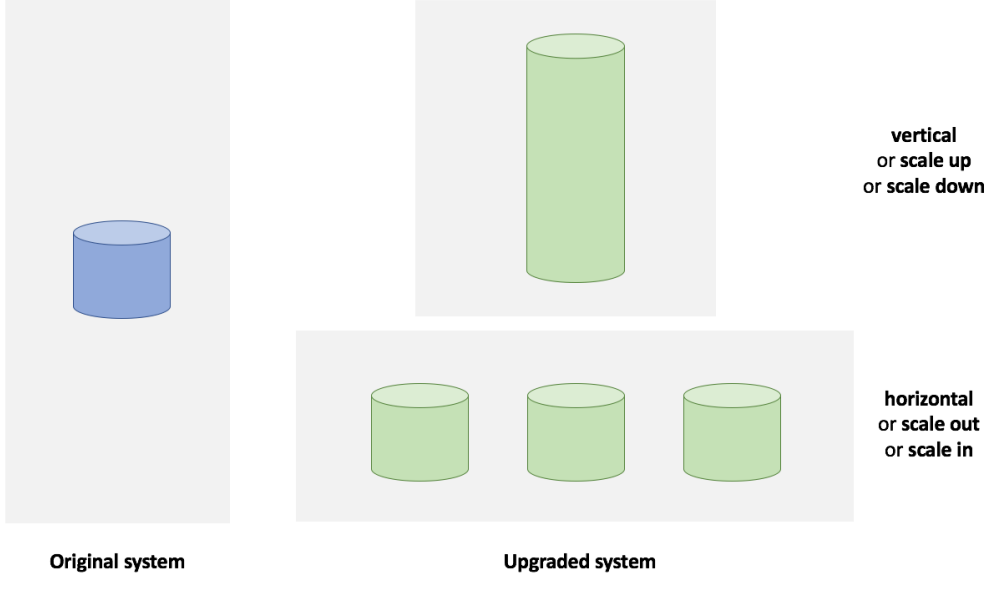
A set of four properties that guarantee that database transactions are processed reliably:

- **Atomic** means "all-or-nothing": when a statement is executed, every update within the transaction must succeed in order to be called succesful
- **Consistent** means that data moves from one correct state to another correct state, with no possibility that readers could view different values that don't make sense together
  - E.g. if a transaction attempts to delete a customer and her order history, it cannot leave order rows that reference the deleted customer's primary key; this is an inconsistent state that would cause errors if someone tried to read those order records
- **Isolated** means that transactions executing concurrently will not become entangled with each other
  - If two different transactions attempt to modify the same data at the same time, then one of them will have to wait for the other to complete
- **Durable** once a transaction has succeeded, the changes will not be lost

Scalability issues: may have poor availability, may use horizontal scaling but this requires distributed transactions (difficult)

Scalability

A capability of a system to handle a growing workload, or its potential to be enlarged to accomodate that growth



- **Horizontal**: adding more nodes to a distributed computing system
  - Ex. increasing the number of compute nodes in a cluster
- **Vertical**: adding resources to a single node system
  - Ex. adding CPUs or memory in a single computer

Data centers

- Clusters of large number of commodity machines (hundreds of thousand)
- Run by companies such as Amazon, Google, and Microsoft

Computer system bottlenecks

- **CPU utilization**: occurs when the processor is busy so that it cannot respond to requests for time
- **Memory utilization**: system does not sufficient or fast enough RAM
  - It cuts the speed at which the RAM can serve information to the CPU, which slows overall operations
  - The CPU will start offloading storage to a significantly slower HDD or SSD to keep things running
- **Network utilization**: the communication between processes residing on differewnt devices connected by a network lack the necessary bandwidth
- **Software limitations**: sometimes bottleneck-related performance originate from the software itself
- **Disk usage**: often the slowest component inside a computer

A modern high-end laptop

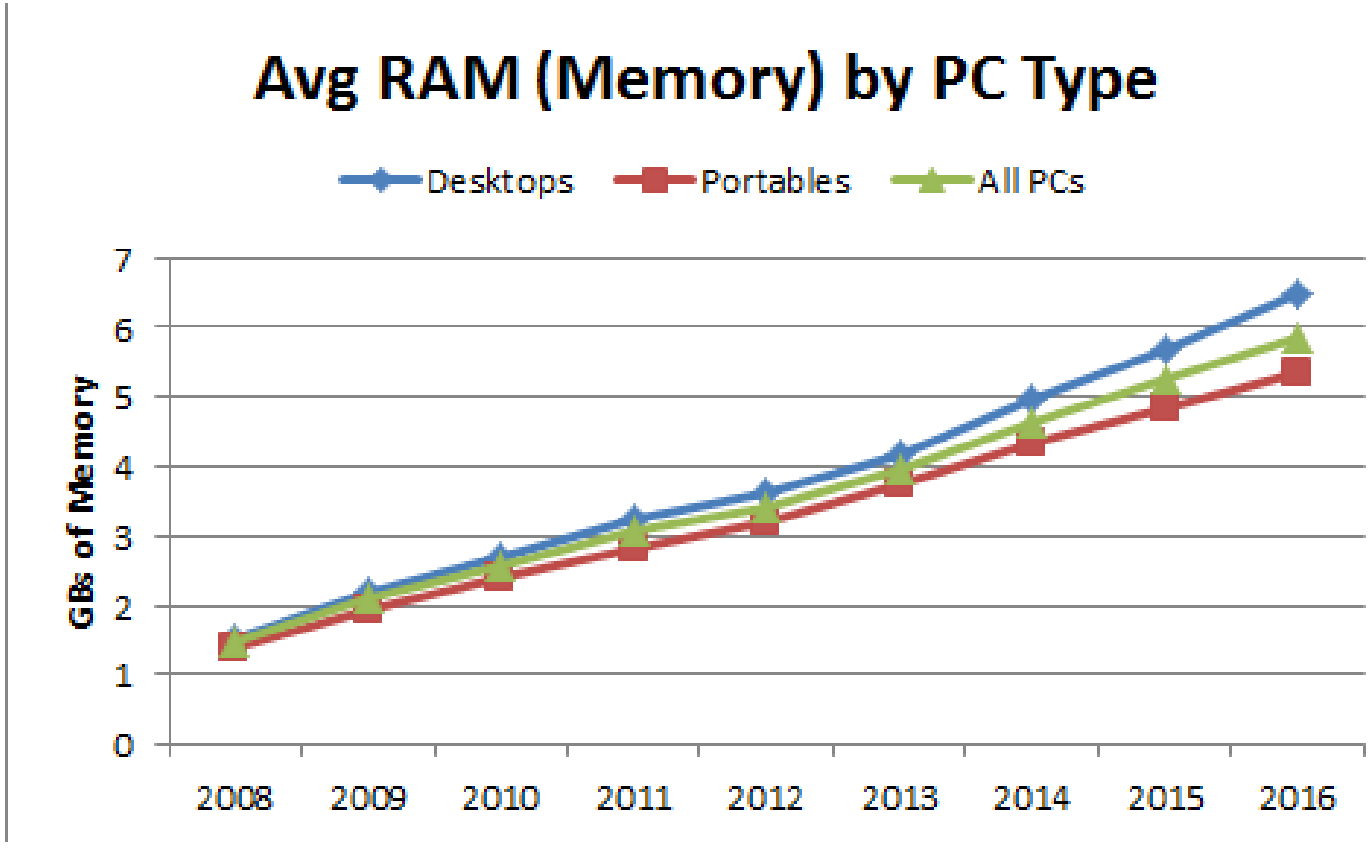
- CPU: Intel i7 (https://ark.intel.com/products/82932/Intel-Core-i7-5820K-Processor-15M-Cache-up-to-3_60-GHz) **3.30 GHz**, **6 cores**, **12 threads**
- RAM: **16 GB**
- HDD: **1TB**

Hard disk prize per GB

Source backblaze (https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/)

Note: SSD vs HDD (http://www.storagereview.com/ssd_vs_hdd)

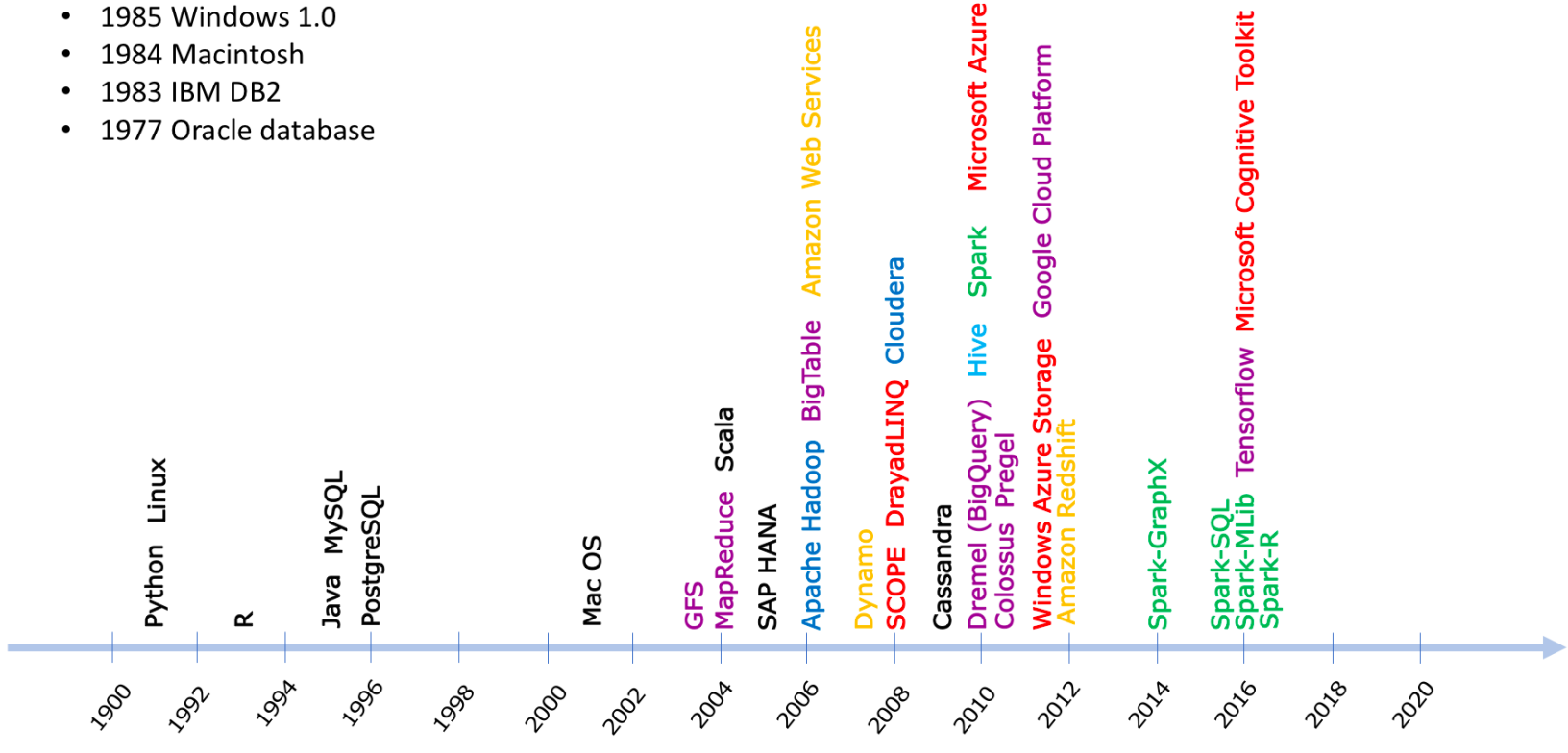
- Failure rate: mean inter-failure time of 2M hours (SSD), 1.5M hours (HDD)
- File copy/write speed: **>200 MB/s** up to 500 MB/s (SSD), **50-120MB/s** (HDD)
- File opening speed: SSD up to 30% faster than HDD
- Cost: cca **$0.20/GB** (for 1TB drive) (SSD), **$0.03/GB** (for 4TB drive) (HDD)



Source: TechTalk (https://techtalk.pcpitstop.com/2016/10/05/average-pc-memory-ram-continues-climb/)

The rise of modern big data analytics systems

- 1989 Microsoft SQL Server
- 1985 Windows 1.0
- 1984 Macintosh
- 1983 IBM DB2
- 1977 Oracle database



Data processing types

- **Batch processing**: an execution of a series of jobs in a program on computing system without manual intervention (non-interactive)
  - Running a data analytics mapreduce job
- **Interactive computing**: software which accepts input from humans as it runs
  - Running ad-hoc queries on a dataset
- **Stream processing**: given a sequence of data (a stream), a series of operations is applied to each element in the stream
  - Referred to also as *dataflow programming*, *event stream processing*, *reactive programming*
- **Lambda architecture**: data-processing designed to handle massive quantities of data by taking advance of both batch and stream processing methods
  - Attempts to balance latency, throughput, and fault-tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data

Batch data processing

Example application scenarios:

- **Distributed grep**: `line of text if matches a user provided pattern`
- **Count URL access frequency**: `(url, total count)`
- **Reverse web-link graph**: `(target_url, list(src_url linking to target_url))`
- **Term-vector per host**: `(host name, list(term)`
- **Inverted index**: `(word, list(document id))`
- **Distributed sort**: `(key, value)` data by `key`

Led to development of new distributed file systems (starting with Google File System) and new computation models (starting with Mapreduce)

Note: 2016 Soft Benchmark (http://sortbenchmark.org/) record: **100 TB** in **134 seconds**, 512 nodes with 2 10-core CPUs, 512 GB RAM

Distributed file systems and mapreduce

- **Distributed file systems** (covered in week 2)
  - Needed new distributed file systems that scale to large volumes of data
  - Designed for fault tolerance, working on large clusters of commodity machines
  - Examples:
    - Google File System (now Colossus)
    - Apache HDFS
- **Mapreduce** (covered in week 3)
  - Needed a new computation simple
  - Required to scale, have simple API, and cover a set of computation tasks of interest

Apache Hadoop

An open-source software framework used for distributed storage and processing of dataset of big data using the mapreduce programming model

- For computer clusters built from commodity hardware
- All the modules designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework
    - Similar to the original design requirements of Internet protocols

The basic framework is composed of the following modules:

- **Hadoop Common**: libraries and utilities needed by other Hadoop modules
- **Hadoop Distributed File System (HDFS)**: a distributed file-system that stores data on commodity machines, providing high aggregate bandwidth across the cluster
- **Hadoop YARN**: a platform responsible for managing computing resources in clusters and using them for scheduling users' applications and
- **Hadoop MapReduce**: an implementation of the MapReduce programming model for large-scale data processing

Amazon shopping cart example

- **Shopping cart service**: maintain information about customer's shopping cart

Amazon facts:

- Amazon runs a world-wide e-commerce platform that serves tens of millions customers at peak times
- Strict operational requirements on the platform in terms of *performance*, *reliability* and *efficiency*
- Reliability is one of the most important requirements
  - Even the slightest outage has significant financial consequences and impacts customer trust

- To support continuous growth, the platform needs to be highly scalable

- Back in 2004, Amazon was running Oracle's enterprise edition with clustering and replication

- Limits were reached to sustain the availability, scalability and performance
- Led to the development of Amazon Dynamo distributed key-value store
  - Designed by trading strict consistency guarantees for availability and scalability

Vogels, W., A Decade of Dynamo (http://www.allthingsdistributed.com/2017/10/a-decade-of-dynamo.html), Blog, All Things Distributed

NoSQL databases

The term NoSQL refers to *non-relational* databases, began gaining popularity around 2009

Categories:

- **Key-value stores**
  - The data items are keys that have a set of attributes; all data relevant to a key is stored with the key
  - Data is frequently duplicated
  - Ex: Amazon's Dynamo and popular caching technologies used as key-value stores such as MemcacheD
- **Column stores**
  - Referred also as wide-column stores
  - Ex: Google's Bigtable, served as inspiration for other systems such as Apache Cassandra
- **Document stores**
  - The basic unit of storage is the complete document
  - Typical document formats: JSON, XML, YAML
  - Ex: MongoDB and Apache CouchDB

NoSQL databases (cont'd)

- **Graph databases**
  - Data represented as a graph - a network of nodes and edges that connect the nodes
  - Both nodes and edges can have attributes
  - Use cases: social networking and semantic web applications
  - Ex: FlockDB, Neo4J, Polyglot, Amazon's Neptun
- **Object databases**
  - The basic unit is an object, not relations and columns
  - Ex: db4o, InterSystems Cache
- **XML databases**
  - Special form of document databases, optimized specifically for working with XML
  - Ex: Tamino and eXist

Notes:

- See http://nosql-database.org (http://nosql-database.org) for a comprehensive list of NoSQL databases
- See db-engines.com (https://db-engines.com/en/ranking) for a database popularity ranking

Structured queries

- Execute queries using a SQL like language
- Need to allow for loading data into a structured data model
- Need to allow for fast evaluation of queries
- Want to use a SQL like API, not some imperative programming (e.g. Mapreduce)

Note: covered in week 04 (we will look into Hive and Spark SQL)

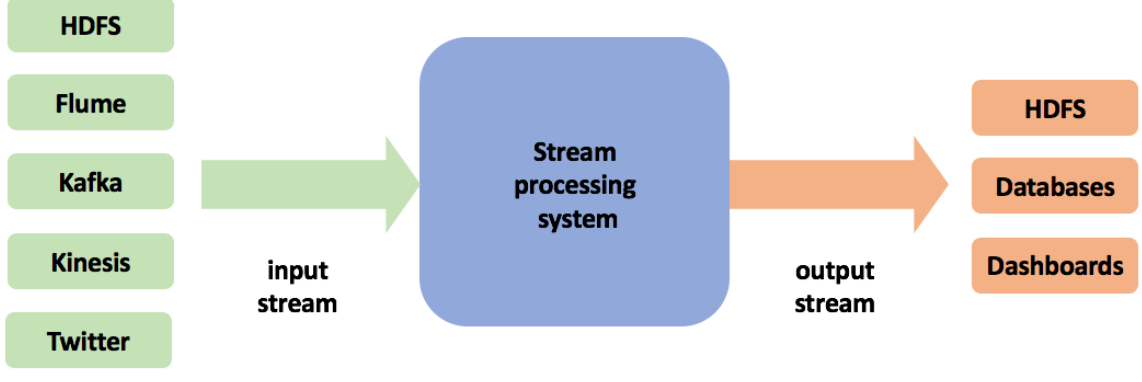
Graph queries

- Input data naturally fits into a graph data model
- Graph data model: *entities* related with *relations*
- Both entities and relations can have *attributes*
- Ex. Yago: `<Richard_Stallman> <isLeaderOf> <Free_Software_Foundation>`
  - Entities: `<Richard_Stallman>` and `<Free_Software_Foundation>`
  - Relatonship: `<isLeaderOf>` attribute specifies the type of relationship 'is leader of'
- Types of qraph data processing:
  - Aggregation: ex. node degrees
  - Centrality measures: ex. PageRank (requires iterative computation)
  - Motif queries: ex. give me all people who are leaders of Free_Software_Foundation (requires graph traversal)

Note: graph processing covered in week 5

Stream data processing

Ex. applications: fraud detection in bank transactions, anomaly detection in sensor data, select tweets about Donald Trump



Requirements: scaling to support data velocity, low latency, efficienty recovery from failures, integration with batch and interactive processing

Machine learning task: ad click prediction

- Criteo terabyte click logs dataset (http://labs.criteo.com/2013/12/download-terabyte-click-logs/)
- day_0 data: **196M** impressions, **26** features, **50GB** raw data size

Row sample: `1 5 110 16 1 0 14 7 1 306 62770d79 e21f5d58 afea442f 945c7fcf 38b02748 6fcd6dcb 3580aa21 28808903 46dedfa6 2e027dc1 0c7c4231 95981d1f 00c5ffb7 be4ee537 8a0b74cc 4cdc3efa d20856aa b8170bba 9512c20b c38e2f28 14f65a5d 25b1b089 d7c1fc0b 7caf609c 30436bfc ed10571d`

```
LSE021353:Criteo vojnovic$ wc -l day_0
 195841983 day_0

LSE021353:Criteo vojnovic$ grep "^1" day_0 | wc -l
 6286525
```

Click rate = **0.0321**

- **Machine learning task**: fitting a logistic regression model
  - It requires iterative optimization method to find parameter vector that minimizes a loss function

Machine learning task: collaborative filtering

- Netflix prize: was an open competition soliciting the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films
- Began on October 2nd, 2006, the winning team awarded **$1M** on September 21st, 2009
- Training data set: `<user, movie, date of grade, grade>`
  - **100M** ratings
  - **480,000** users
  - **18,000** movie titles
  - **2.1GB** raw data size
  - Data collected between Oct 1998 and Dec 2005
- **Machine learning task**: predict ratings of movies by users

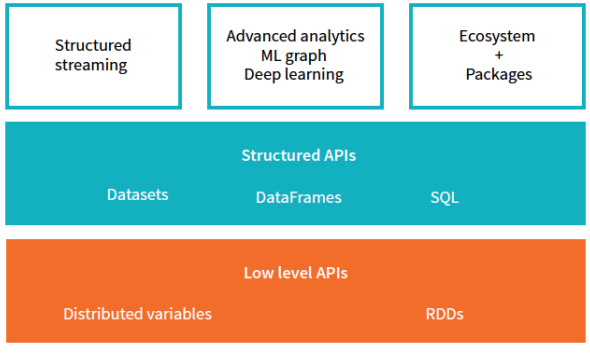
More about Netflix prize (https://en.wikipedia.org/wiki/Netflix_Prize)

Note: machine learning tasks covered in weeks 7 and 8

Apache Spark

- Introduced the novel concept of a resilient distributed data set (RDD) for fault-tolerant, distributed in-memory computing
- Integration of different types of data processing into a single system
- API interfaces in different programming languages (Scala, Java, Python, R)

Spark functionalities:



Note: we will cover various aspects of Apache Spark

A note on computer system conferences

The design of many big data analytics systems was first made public in a research paper, presented at a computer systems conference

Some computer systems conferences:

- OSDI (https://www.usenix.org/conferences/byname/179) USENIX Symposium on Operating Systems Design and Implementation
- SOSP (http://sosp.org/) ACM Symposium on Operating Systems Principles
- NSDI (https://www.usenix.org/conferences/byname/178) USENIX Symposium on Networked Systems Design and Implementation

some database system conferences:

- VLDB (http://www.vldb.org/conference.html) Very Large Databases
- SIGMOD (https://sigmod.org/) International Conference on Management of Data
- PODS (https://sigmod.org/pods/) Principles of Database Systems

Current trend: making a system open source in early stages (e.g. Tensorflow)

Lab preview

- Command line interface
  - Basic file system commands
  - Basic system administration commands
  - Network commands
- GCP
  - Install gcloud console
  - Create a cluster, work with a bucket
  - Open jupyter notebook run in a GCP cluster
- docker
  - Install docker
  - Run jupyter notebook within a cluster