London Business School

AM05 Data Management

05. Database Connections and Big Data

Dr. David Tilson

Agenda

Database connections and semi-structured data

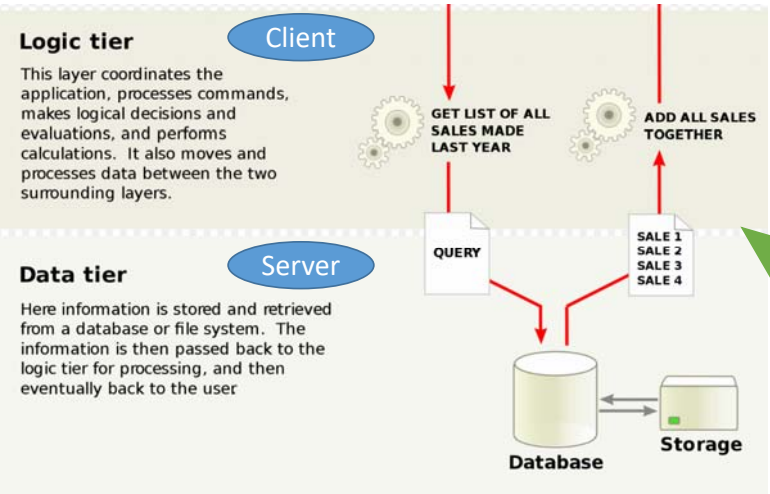
Business analytics and the advent of big data

Technical challenges of big data

Intro to Hadoop and other **Big Data** technologies

Wrapping up the course

Database connection allows **client software** to talk to database **server software** (on same machine or over a network)

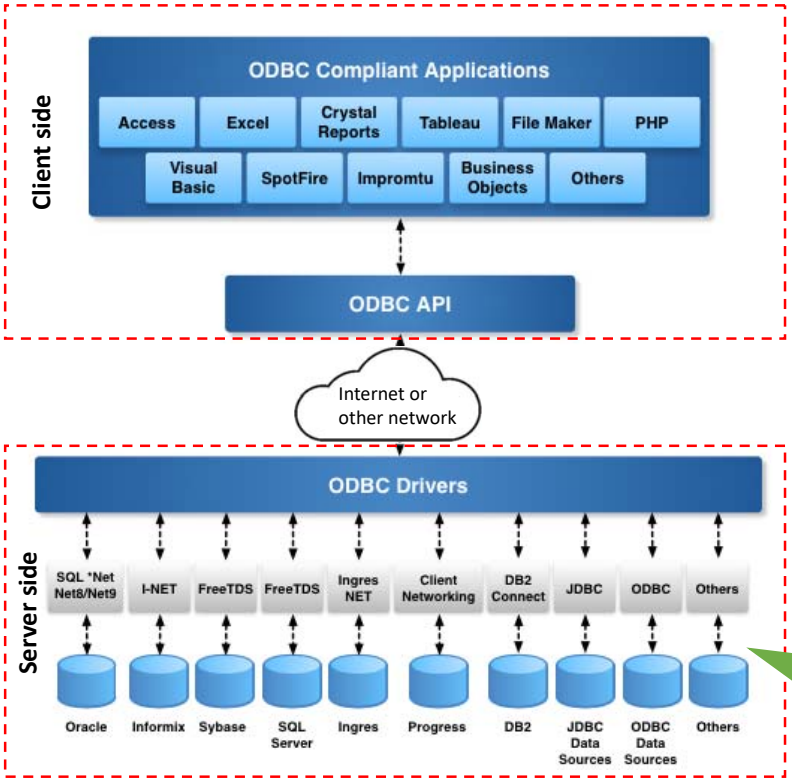


A connection is required to send commands and receive answers, usually in the form of a result set.

Commands can only be performed against a database with an "open and available" connection to it

There are many way of doing this (see chapter 5 of DSC). Some more efficient / secure than others

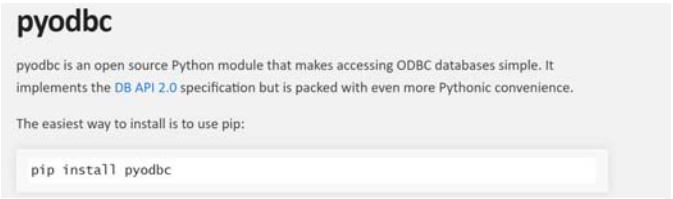
Open Database Connectivity (ODBC) is a standard application programming interface (API) for accessing DBMSs



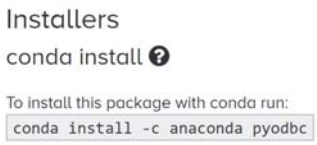
- **ODBC** aims to be independent of database and operating systems
- ODBC-based applications can be ported to other platforms with few changes to the data access code

Servers could be
- On same machine as client
- On different machine in data center
- In a different corporate data center
- On a cloud provider data service

In practice the ODBC API is available as a library on the client side (as are other types of database connectors)



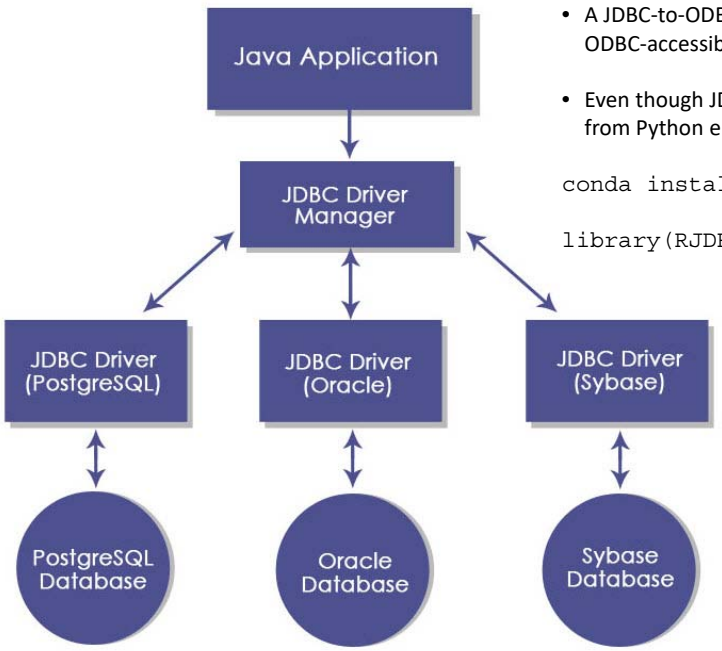
http://mkleehammer.github.io/pyodbc/



https://anaconda.org/anaconda/pyodbc

https://db.rstudio.com/odbc/

Java Database Connectivity (JDBC) is another common API which defines how clients can access RDBMs

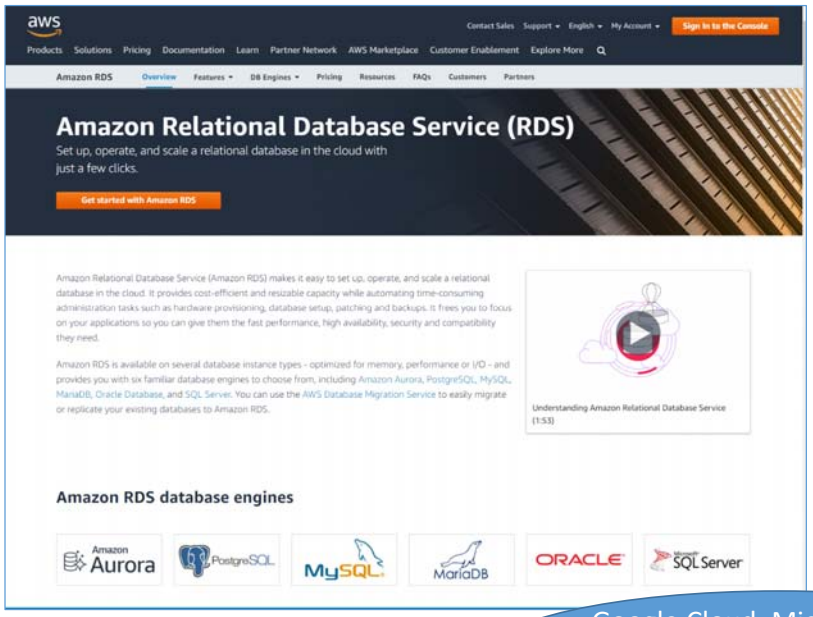


- A JDBC-to-ODBC bridge also enables connections to any ODBC-accessible data source
- Even though JDBC is a Java technology it can be used from Python e.g. to install from anaconda

```
conda install -c conda-forge jaydebeapi
```

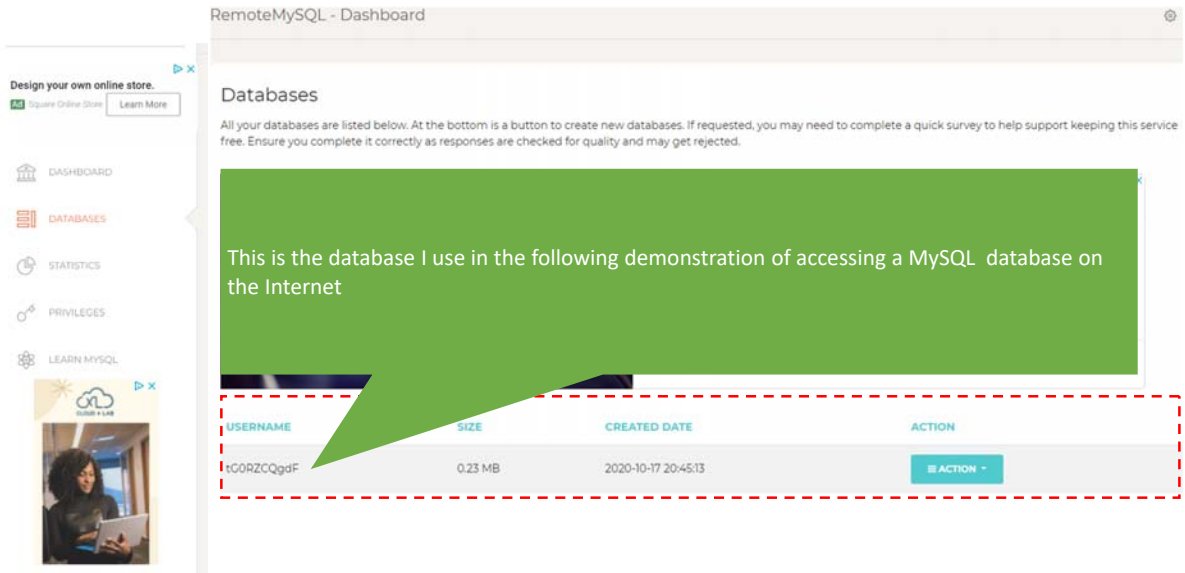
```
library(RJDBC)
```
in R (see rforge.net/RJDBC)

The major cloud infrastructure providers allow you to set up RDMSs in minutes (e.g. on Amazon Web Services)



Google Cloud, Microsoft Azure and many others have similar offerings

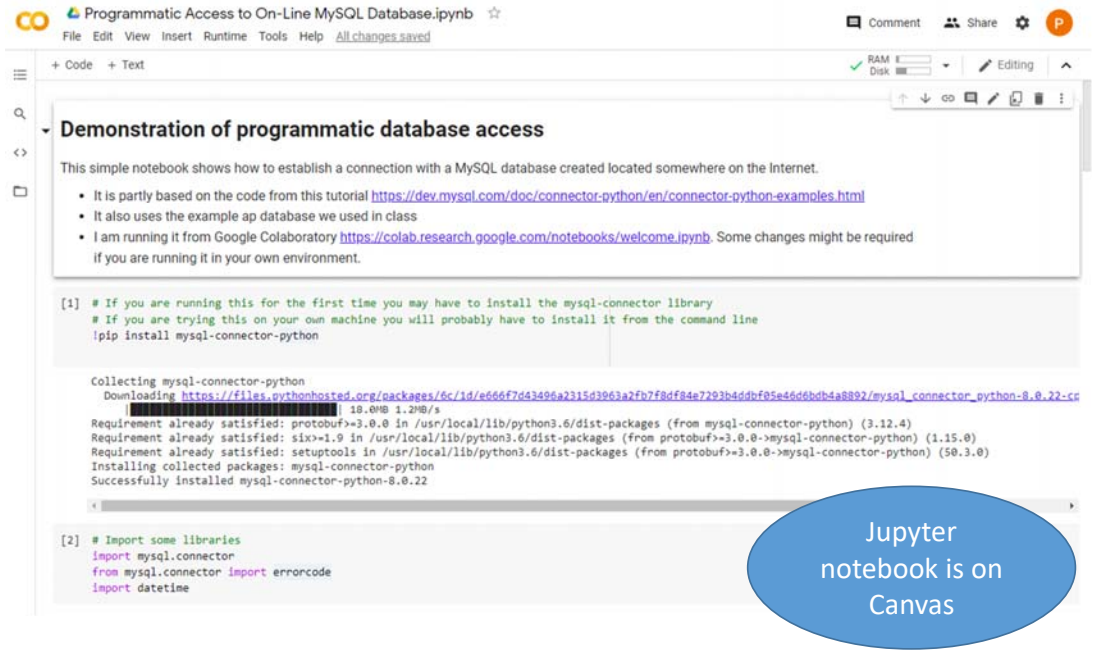
You can even set up free MySQL databases on-line to play with or test software (but you will get lots of adverts)



https://remotemysql.com

Obviously, don't use it for anything important

Let's look at example of connecting to an MySQL database over the Internet using python



JSON and XML are common ways of communicating semi-structured data across networks

JavaScript Object Notation (JSO) Example

```
{"products": [
        {"number": 1, "name": "Zoom X", "Price": 10.00},
        {"number": 2, "name": "Wheel Z", "Price": 7.50},
        {"number": 3, "name": "Spring 10", "Price": 12.75}
]}
```

eXtensible Markup Language (XML) Example

```
<products>
        <product>
                <number>1</number> <name>Zoom X</name> <price>10.00</price>
        </product>
        <product>
                <number>2</number> <name>Wheel Z</name> <price>7.50</price>
        </product>
        <product>
                <number>3</number> <name>Spring 10</name> <price>12.75</price>
        </product>
</products>
```

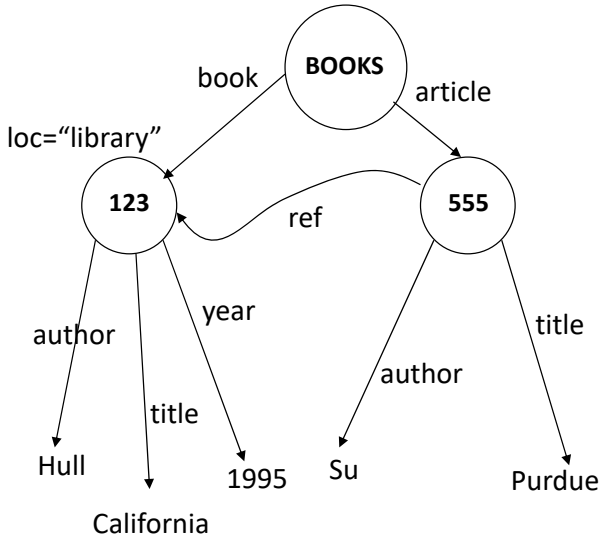
Both **JSON** and **XML** provide a **semi-structured** means of representing data hierarchically

Semi-structured data does not obey the formal structure of data models associated with RDBMSs but contains tags to separate semantic elements and enforce hierarchies of records and fields within (aka a self-describing structure)

XML documents define sets of rules for encoding documents in human and machine-readable format

Example of XML document elements

```
<BOOKS>
<book id="123" loc="library">
  <author>Hull</author>
  <title>California</title>
  <year> 1995 </year>
</book>
<article id="555" ref="123">
  <author>Su</author>
  <title> Purdue</title>
</article>
</BOOKS>
```

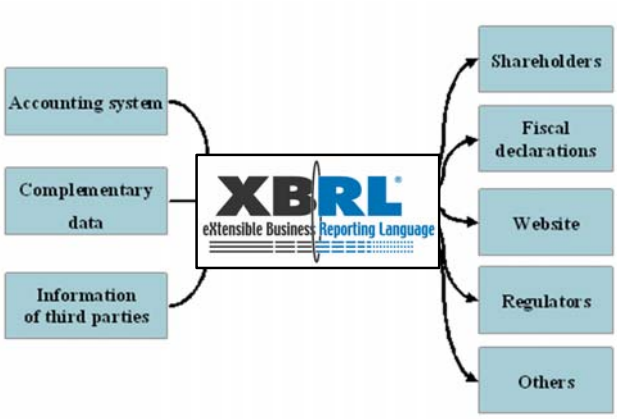


Hundreds of XML-based document formats have been defined e.g. RSS, SOAP for web services, and SOA. Even modern Microsoft Office file formats are XML-based

XBRL is an XML-based standard developed for the exchange of financial reports over networks

XBRL (**eXtensible Business Reporting Language**) is a framework for exchanging business information

- XML-based: uses XML syntax/technologies
- Supports business reporting like the definition and exchange of financial information (e.g. financial statements)



Individual fields or whole XML documents can be stored in RDBMSs or other data stores

JSON is used extensively on the Web (e.g. Twitter feeds)

Example tweet in JSON format


```
[{
  "created_at": "Thu Jun 22 21:00:00 +0000 2017",
  "id": 877994604561387500,
  "id_str": "877994604561387520",
  "text": "Creating a Grocery List Manager Using Angular, Part 1: Add &amp;amp; Display
  "truncated": false,
  "entities": {
    "hashtags": [{
      "text": "Angular",
      "indices": [103, 111]
    }],
    "symbols": [],
    "user_mentions": [],
    "urls": [{
      "url": "https://t.co/xFox78juL1",
      "expanded_url": "http://buff.ly/2sr60pf",
      "display_url": "buff.ly/2sr60pf",
      "indices": [79, 102]
    }]
  },
  "source": "<a href=\"http://bufferapp.com\" rel=\"nofollow\">Buffer</a>",
  "user": {
    "id": 772682964,
    "id_str": "772682964",
    "name": "SitePoint JavaScript",
    "screen_name": "SitePointJS",
    "location": "Melbourne, Australia",
    "description": "Keep up with JavaScript tutorials, tips, tricks and article
    "url": "http://t.co/cCH13gqeUK",
    "entities": {
      "url": {
        "urls": [{
          "url": "http://t.co/cCH13gqeUK",
          "expanded_url": "http://sitepoint.com/javascript",
          "display_url": "sitepoint.com/javascript",
          "indices": [0, 22]
        }]
      },
      "description": {
        "urls": []
      }
    },
    "protected": false,
    "followers_count": 2145,
    "friends_count": 18,
    "listed_count": 328,
    "created_at": "Wed Aug 22 02:06:33 +0000 2012",
    "favourites_count": 57,
    "utc_offset": 43200,
    "time_zone": "Wellington",
  },
```


Individual tweet may have different sets of tags

MySQL database has JSON datatype

Popular MongoDB NoSQL database natively stores JSON documents

See https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json

Agenda

Database connections and semi-structured data

Business analytics and the advent of big data

Technical challenges of big data

Intro to Hadoop and other **Big Data** technologies

Wrapping up the course

Definitions of **big data** are a bit squishy

Some definitions

1. "Big data exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process in within a tolerable elapsed time for its user population."
   Gartner's Merv Adrian in Teradata Magazine

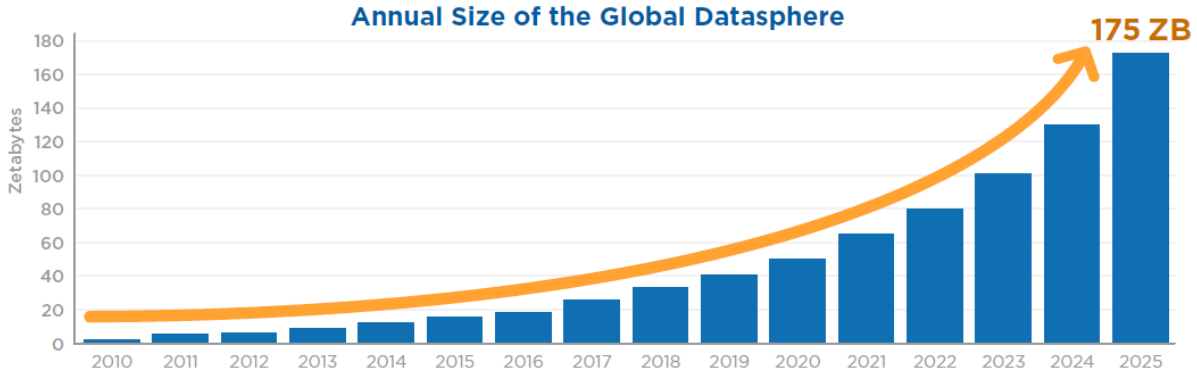
2. "Big data refers to data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze."
   McKinsey Global Institute, Big Data: The Next Frontier for Innovation, Competition and Productivity

Implies

- What qualifies as "big data" changes as technology advances
- Tomorrow's "big" will be bigger than today's

Big. . . and getting bigger

Figure 1 – Annual Size of the Global Datasphere



IDC DataAge 2025 whitepaper

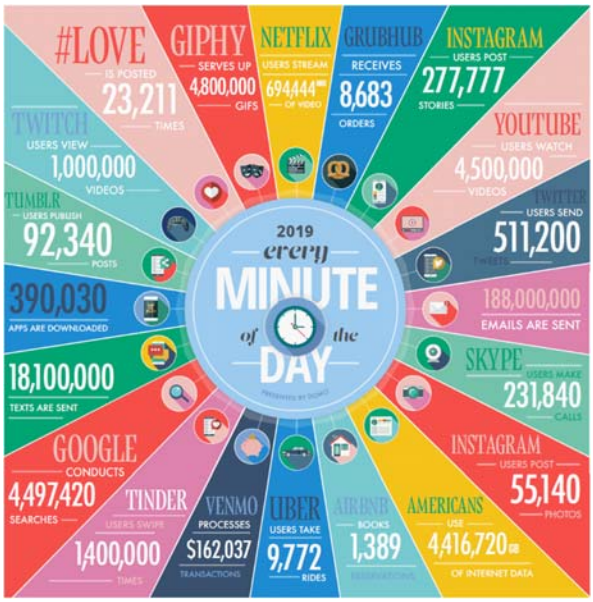
It's more than just lots of data . . . The 3Vs of "Big Data"

Three drivers of increased complexity

¶ **Volume** – a lot of data to work with

¶ **Velocity** – its coming at you fast,
- Not just periodic reports on structured data
- Continually arriving data needs to be processed for insight
- Moving to real-time analysis

¶ **Variety** – many different sources
- Structured: databases, sensor data
- Semi-structured: Weblogs, Social
- Unstructured: flat-file, images, video, audio



Data source: https://www.newgenapps.com/blog/big-data-statistics-predictions-on-the-future-of-big-data
Gartner identified the 3Vs – others added Value, Veracity, Vulnerability (security), Variability, Volatility, Visualization

Volume is a moving target . . . but petabytes are certainly big data

Quantities of bytes				
SI prefixes			Binary prefixes (IEC 60027-2)	
Name (Symbol)	Value in Popular Usage	Value in Standard SI	Name (Symbol)	Value
kilobyte (kB)	2^{10}	10^3	kibibyte (KiB)	2^{10}
megabyte (MB)	2^{20}	10^6	mebibyte (MiB)	2^{20}
gigabyte (GB)	2^{30}	10^9	gibibyte (GiB)	2^{30}
terabyte (TB)	2^{40}	10^{12}	tebibyte (TiB)	2^{40}
petabyte (PB)	2^{50}	10^{15}	pebibyte (PiB)	2^{50}
exabyte (EB)	2^{60}	10^{18}	exbibyte (EiB)	2^{60}
zettabyte (ZB)	2^{70}	10^{21}	zebibyte (ZiB)	2^{70}
yottabyte (YB)	2^{80}	10^{24}	yobibyte (YiB)	2^{80}

Big data today

¶ Traditional databases can handle tens of terabytes

¶ PB definitely starting to be big data

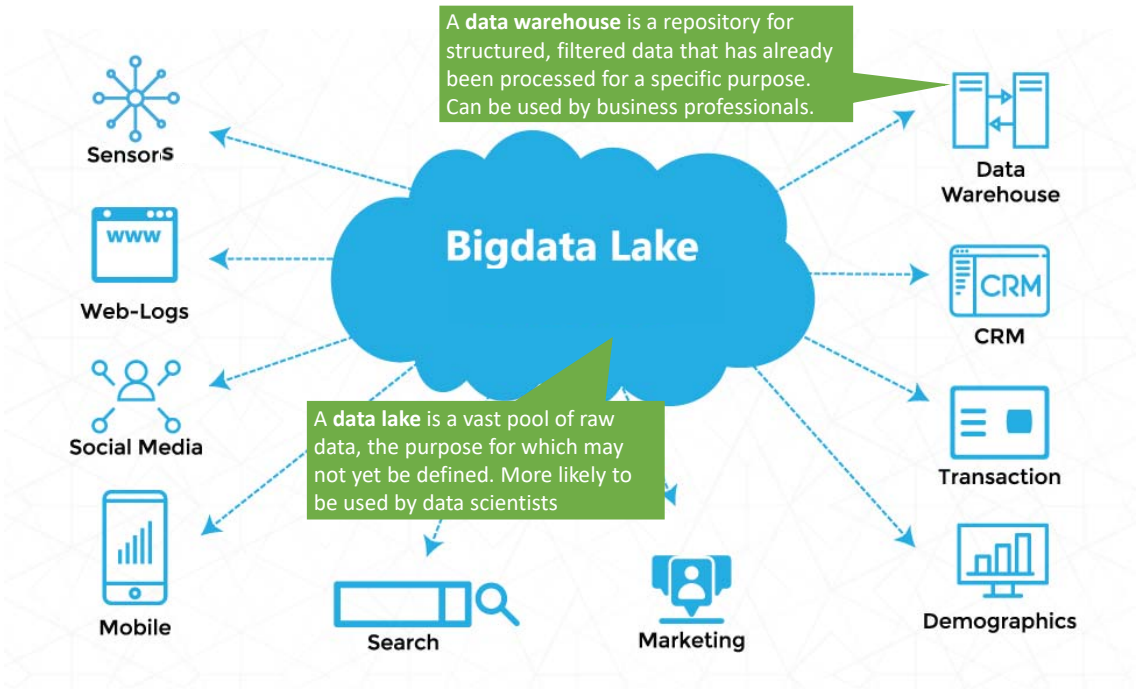
¶ Velocity – rate at which it is growing means you
- Might soon need big data technologies for large volume
- Need big data technologies to ingest the daily volume

¶ Variety makes things more complex
- Beyond rows and columns
- Messages (text), images, video

There are some features that tend to distinguish “big data” sources from traditional ones

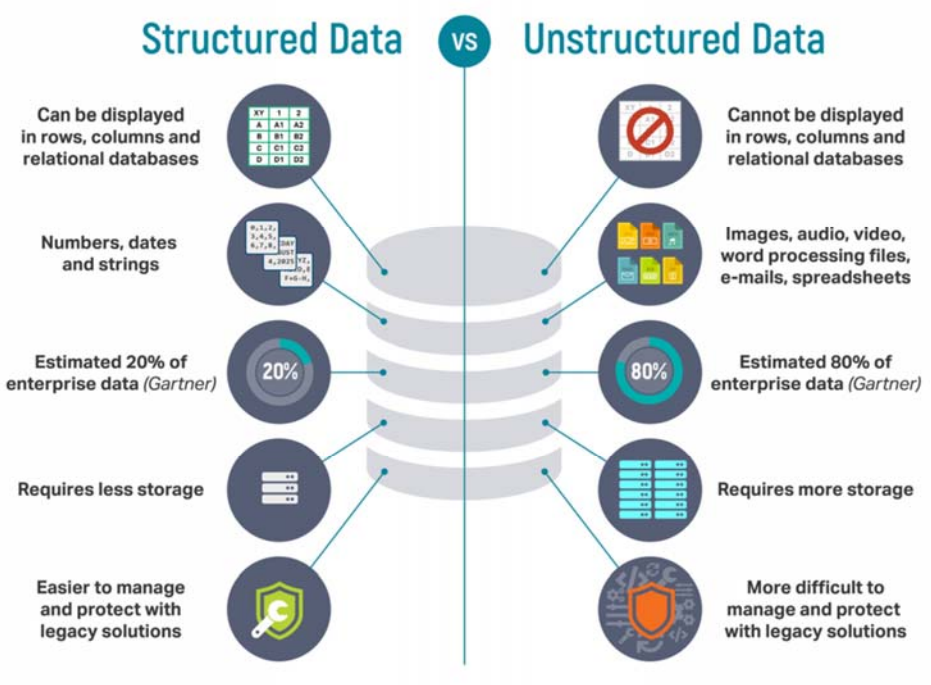
- ¶ Often generated automatically
  - Traditional data sources have people taking action: call detail records, bank/retail transactions, shipments, payments
  - Many “big data” sources don’t involve people: smart meters, engine sensors
- ¶ Novel data sources
  - Traditional data sources are often transactions, even if done on-line
  - New sources include detailed browsing behaviors
  - “More of the same” can be new: Smart meter readings every 15 minutes
- ¶ Not designed to be friendly (i.e. easy to process and analyze)
  - Traditional sources designed to be friendly (database transaction records)
  - Some not designed at all: text streams from social media
    - No standards of grammar, sentence ordering, or vocabulary
    - Get what you get
  - May have to wade through mess, junk filled data during analysis
- ¶ Much of it may be worthless
  - Traditional sources designed to be 100% relevant and make best use of limited storage and processing power
  - Big data tends to capture everything possible and figure out what matters later

In a business context Big Data often associated with predictive analytics and user\customer behavior analysis



Source: http://bigdatadimension.com/big-data-lake-analytics/

Much of the increase in volume is tied to increases in variety of data… structured, unstructured (text, logs, images, video, sensors)



https://www.igneous.io/blog/structured-data-vs-unstructured-data

Web log data is big(gest) source of "Big Data"

Web logs as rich new data source

- ¶ Detailed web behavior insight
  - Research behaviors
  - Decisions making
  - Purchase paths
- ¶ Factual info on customer
  - Preferences
  - Future intentions
  - Motivations
- ¶ Info across browsing sessions
- ¶ New data for segmentation (e.g. dreamers)

- Less than 2% of website visits result in purchases
- So, only tracking completed transactions means gaining no insight on >98% of visitors

```
#Fields: date time s-sitename s-computername s-ip cs-method cs-uri-stem cs-uri-query s-port cs
2010-03-24 07:00:01 ZZZZC941948879 RUFFLES 222.222.222.222 GET / - 80 - 220.181.7.113 HTTP/1.1
2010-03-24 07:00:23 ZZZZC941948879 RUFFLES 222.222.222.222 GET /2009/12/im_not_mean_im_just
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /terminal-blank.gif - 80
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /grep-options.gif - 80 -
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /terminal-cat.gif - 80 - 217
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /terminal-pwd-cd.gif - 80 - 217
2010-03-24 07:00:39 ZZZZC941948879 RUFFLES 222.222.222.222 GET /robots.txt - 80 - 95.55.207.95
2010-03-24 07:00:39 ZZZZC941948879 RUFFLES 222.222.222.222 GET /rss-short.xml - 80 - 173.45.2
```

But web logs are not "ready to analyze"

Unstructured text is massive source of potential insight

Natural Language Text Parsing

Meaning

¶ Lots of sources

- Email
- Text messages
- Tweets
- Social media posts
- Instant messaging / chats
- Audio transcriptions (e.g. support calls)

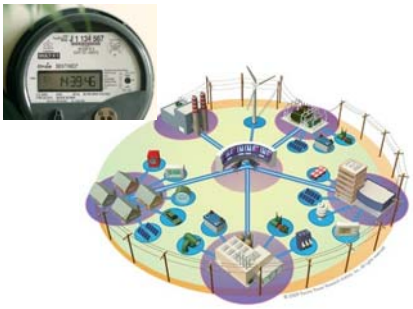
¶ Potential insights

- "Buzz" around product
- Customer sentiment about company / product / service
- Sources of complaints
- Fraud detection
- Legal discovery
- Targeted adverts

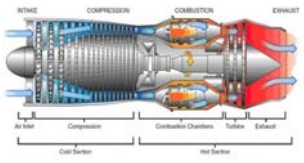
There are other novel sources of "Big Data"

Location information

RFIDs

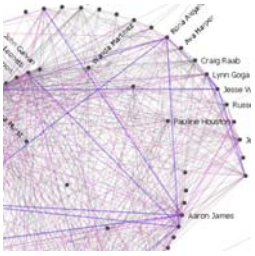


Smart grid / meters



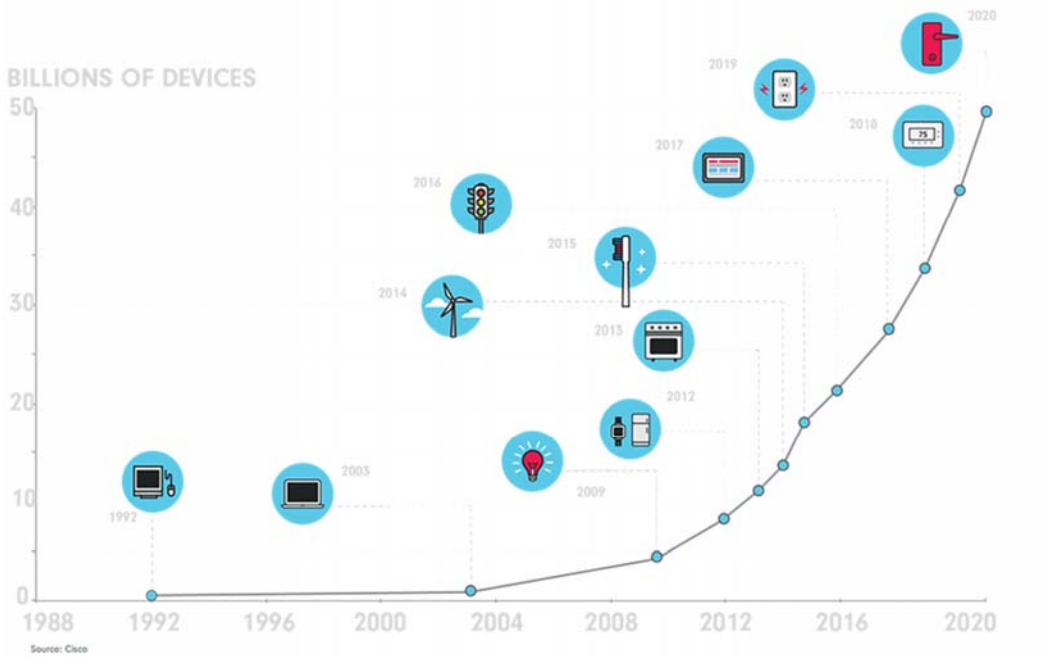
Sensor data

Sensor data



(Social) Network graphs

Data from Internet of Things (IoT) growing exponentially



Agenda

Database connections and semi-structured data

Business analytics and the advent of big data

Technical challenges of big data

Intro to Hadoop and other **Big Data** technologies

Wrapping up the course

Digital computers have several key components coordinated by software

Input/Output (I/O)

Network(s)

User Interface
(if used by directly by a person)

Processor(s)

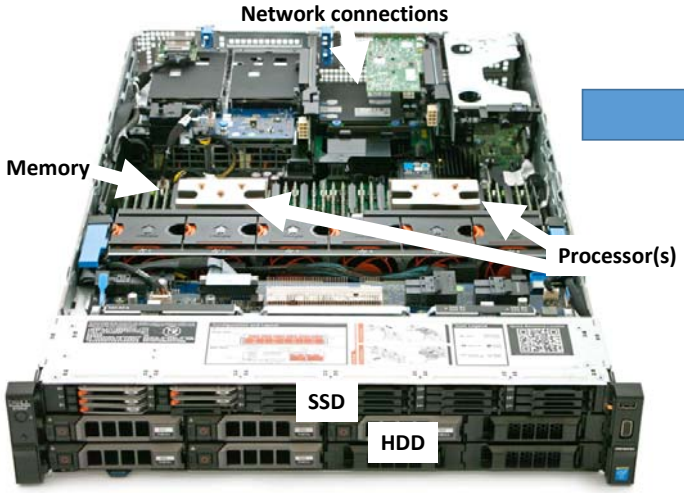
Memory

Persistent Storage

Hard Disk Drives (HDD)

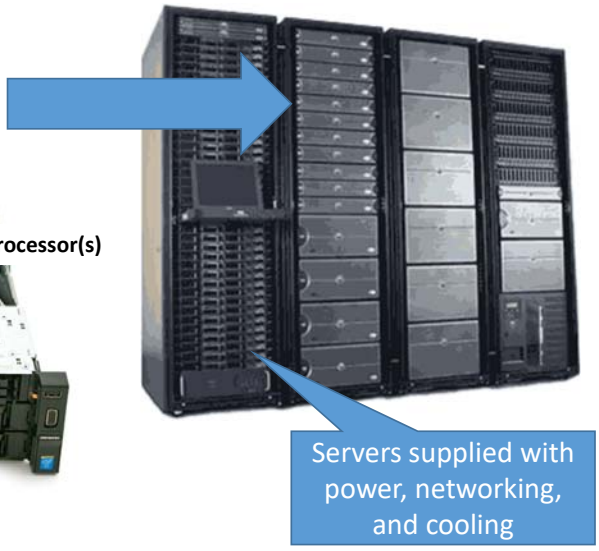
Solid State Drives (SDD)

Enterprise IT and Cloud services supported by servers installed in racks



Dell PowerEdge 13G R730xd

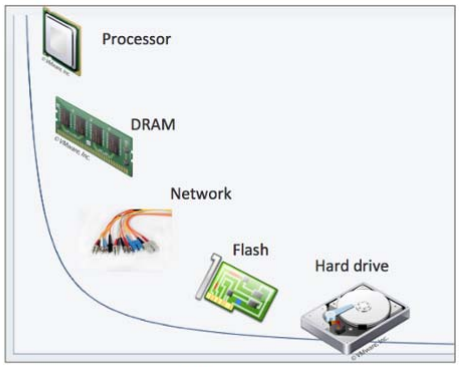
Typical Server – "commodity hardware"



Typical Rack in Data Center

When big data is spread across servers it becomes impractical to bring it all to one machine for processing

Components of computers operate at very different speeds



If Memory = **Minute**
Network = **Weeks***
Flash = **Months**
Disk = **Decades**

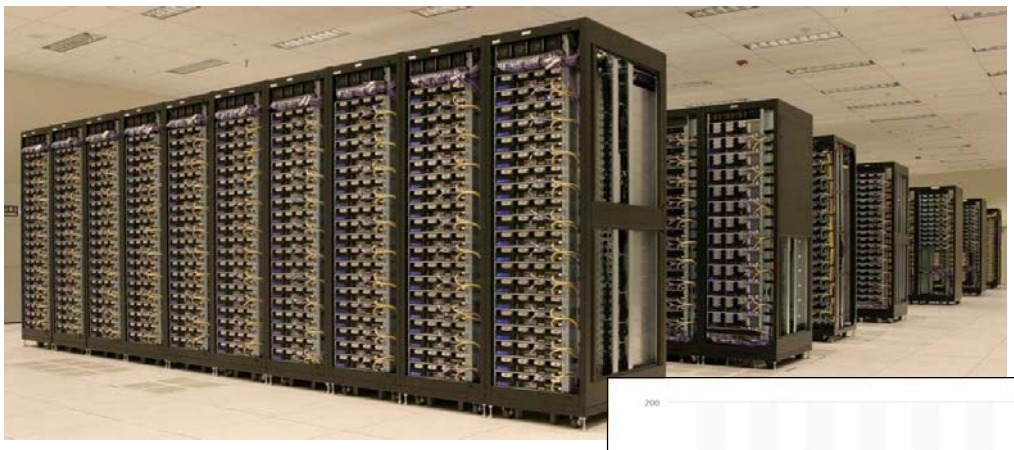
* Still want to avoid moving data over networks as it still has to transfer on/off disks

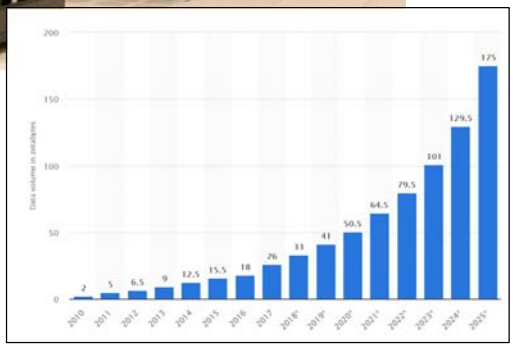
Getting data to processors becomes the bottleneck

Impossible for data to go to the code → So the code must go to the data

Big data does not fit on one computer

Yahoo's Hadoop Cluster





Distributed Systems have certain problems

- ¶Can use multiple machines for single task
- ¶Programming distributed systems much more complex
  - Synchronizing data exchanges
  - Managing a finite bandwidth
  - Controlling computation timing
- ¶Vulnerable to failure of one machine

Agenda

Database connections and semi-structured data

Business analytics and the advent of big data

Technical challenges of big data

Intro to Hadoop and other **Big Data** technologies

Wrapping up the course

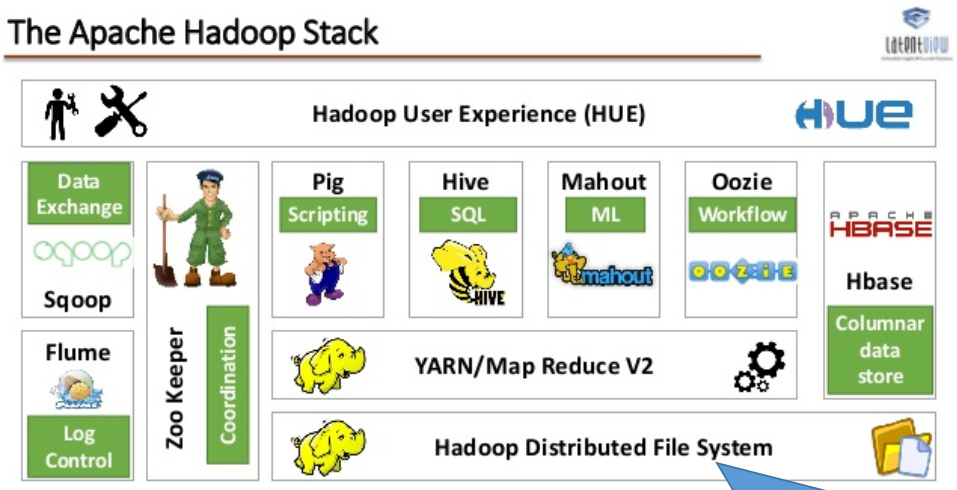
What is Apache Hadoop?

¶ Open source software framework designed for storage and processing of large-scale data on clusters of "commodity hardware"

¶ Used for

- Data-intensive text processing
- Assembly of large genomes
- Graph mining
- Machine learning and data mining
- Large scale social network analysis

Hadoop designed to alleviate the problems with distributed computing



HDFS file system hides the fact that data is stored across a cluster of machines from applications

¶ Must support partial failure

¶ Must be scalable

- Increasing resources should increase load capacity
- Increasing load on the system should result in graceful decline in performance for all jobs . . . rather than system failure

Ideas behind Apache Hadoop emerged from Google… one of the first companies to grapple with big data

Google

THE APACHE SOFTWARE FOUNDATION
Open Source

2003 — **The Google File System**

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*

→ hadoop HDFS

2004 — **MapReduce: Simplified Data Processing on Large Clusters**

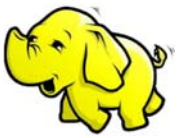
Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com
Google, Inc.

→ hadoop MapReduce

2006 — **Bigtable: A Distributed Storage System for Structured Data**

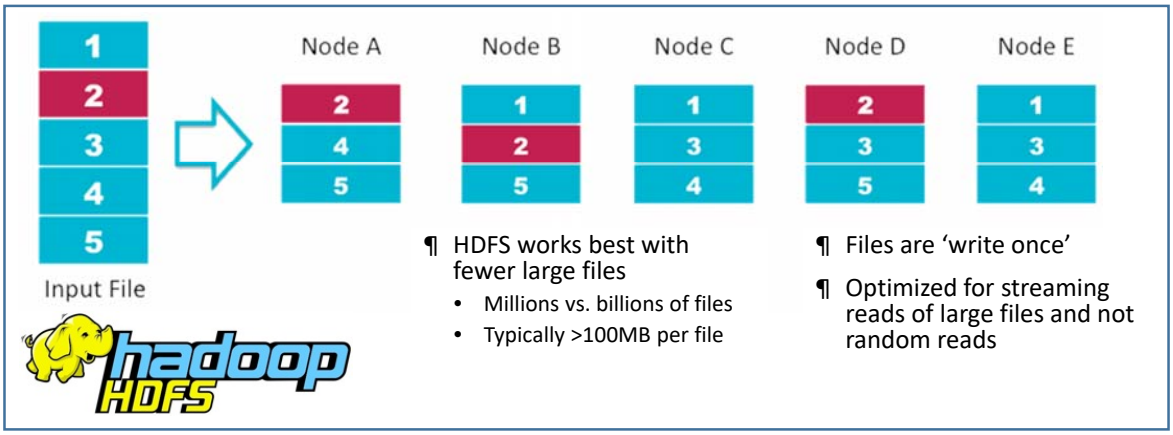
Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com
Google, Inc.

→ APACHE HBASE



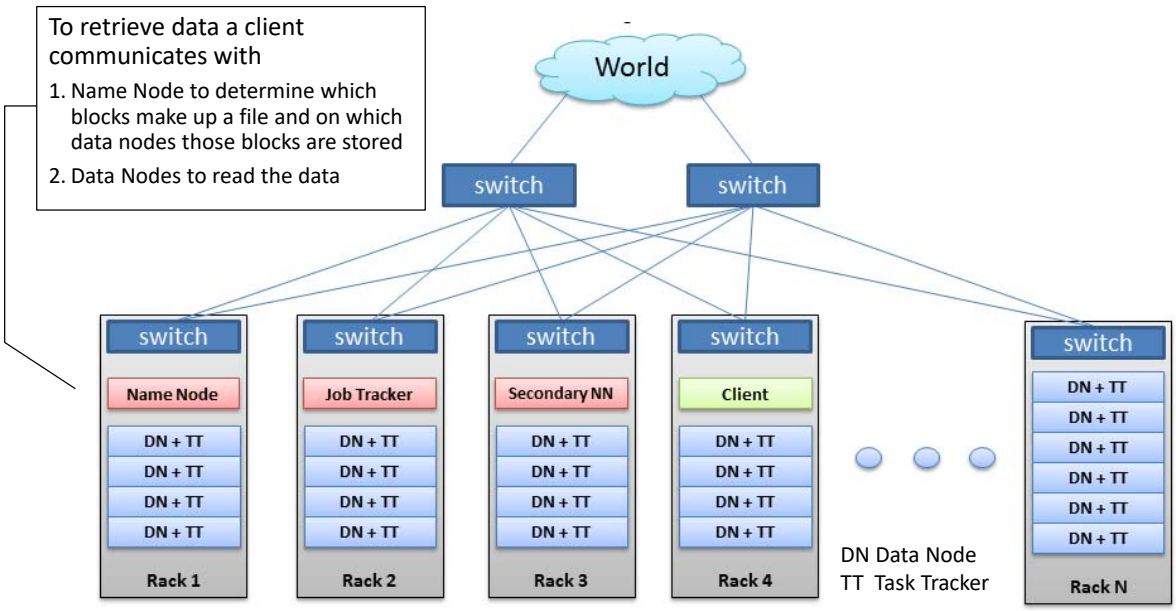
Hadoop Distributed File System (HDFS) provides redundant storage for massive amounts of data

- Files split into blocks (64 or 128 MB)
  - Split across many machines at load time
  - Replicated across multiple machines (for fault tolerance)
- *NameNode* keeps track of which blocks make up a file and where they are stored



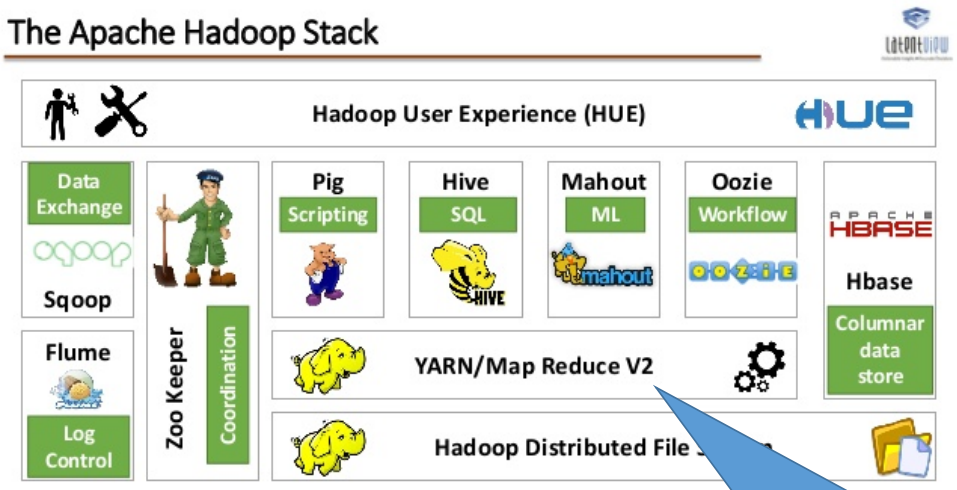
- HDFS works best with fewer large files
  - Millions vs. billions of files
  - Typically >100MB per file
- Files are 'write once'
- Optimized for streaming reads of large files and not random reads

Hadoop clusters can scale to thousands of nodes



This approach to increasing processing power, memory, and storage is referred to as **scaling-out** … as opposed to scaling-up by buying or building a single more capable server

MapReduce is Hadoop's distributed computing platform



MapReduce Features

- Automatic parallelization and distribution of computing across nodes in the cluster
- Fault-Tolerance
- Clean abstraction for programmers

Developers \ Analysts can write Map Reduce jobs (e.g. in Java or Python) without worrying about details of the cluster

Core concepts in MapReduce distributed computing model

- Code goes to the data (parallel computing)
  - Each node can perform computations on data it stores
  - No need to move data – at least for initial (map) processing
  - Tries to minimize slow data transfers between nodes and racks
- Applications written in high-level programming language
  - No network programming needed
  - Handles timing issues
  - Fault tolerance built in
    - Failures detected and tasks reassigned to a different nodes
    - Restarting a task does not affect nodes working on other tasks
    - If failed node restarts, it is added back to cluster and assigned new tasks
- Need to be a (Java) programmer to use MapReduce directly… not really for data analysts or data scientists

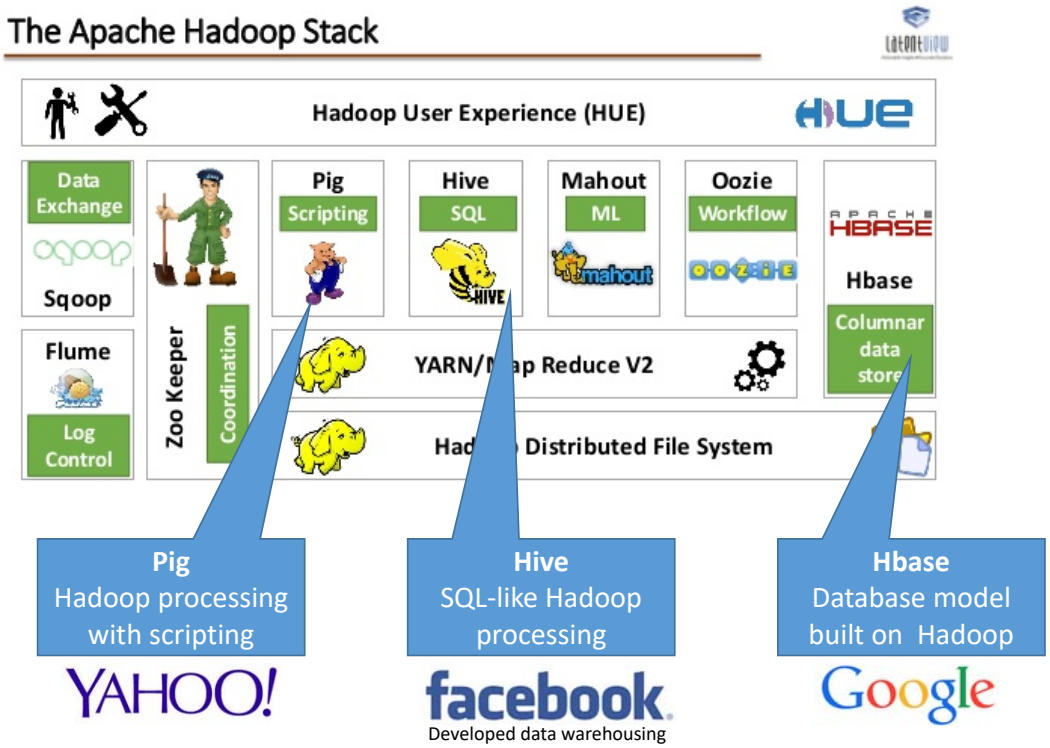
MapReduce is a relatively simple but flexible model

- Tasks divided into two phases
  - Map tasks done on portions of data where it is stored
  - Reduce tasks combine data from map tasks to produce final output
- Job Tracker allocates work to individual nodes
- Key functionality of SQL and other tasks can be implemented in MapReduce
  - Select \ filter
  - GroupBy
  - Joins
  - Linear Algebra (matrix math)

Map Reduce speeds up Big Data processing in two ways

1. Minimizing movement of data across networks
2. Using the power of many processors across the cluster (parallel computing)

Other tools in Hadoop ecosystem allow analysts to be more productive while harnessing MapReduce and HDFS



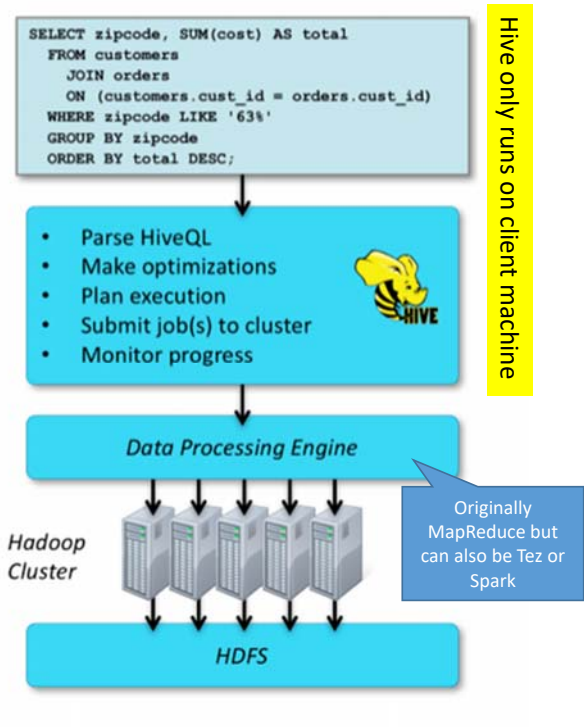
Many large companies rely on Hadoop

Apache Hive allows business analysts with SQL expertise gain insight from big data on Hadoop clusters

Hive runs on client machine

- ¶ Turns HiveQL queries into MapReduce jobs
- ¶ Submits those jobs to the cluster
  - As execution plan (not Java code)
- ¶ Shares many architectural similarities with Pig

Uses SQL-like language called HiveQL which is closely related to ANSI standard SQL-92



Hive brings 'big data' to a broader range of potential users

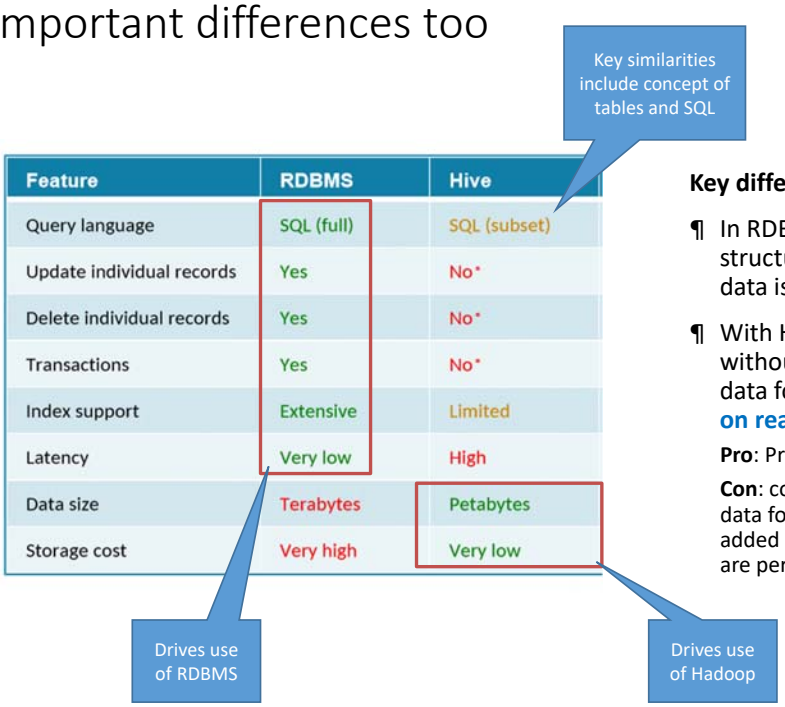
¶ More productive than writing MapReduce directly
- Five lines of HiveQL might be equivalent to 100 lines or more of Java

¶ Brings large-scale data analysis to a broader audience
- No software development (Java) experience required
- Leverages existing knowledge of SQL

¶ Offers interoperability with other systems
- Extensible through UDFs, JDBC/ODBC, and external scripts (could access via python programs for example)
- Many business intelligence (BI) tools support Hive

UDF stands for User Defined Functions

Hive shares many similarities with an RDBMS but there are important differences too



Feature	RDBMS	Hive
Query language	SQL (full)	SQL (subset)
Update individual records	Yes	No*
Delete individual records	Yes	No*
Transactions	Yes	No*
Index support	Extensive	Limited
Latency	Very low	High
Data size	Terabytes	Petabytes
Storage cost	Very high	Very low

Key differences include

- In RDBMS, you create tables with rigid structure that must be specified before data is added ("**schema on write**")
- With Hive you can store data in HDFS without knowing its format. Only specify data format when you read it ("**schema on read**")

  Pro: Provides far more flexibility

  Con: conflict between expected and actual data formats won't be detected as records added to table. Only discovered when queries are performed

* Hive now has limited support for UPDATE, DELETE, and transactions. Not industrial strength as yet. Hive has no triggers either.

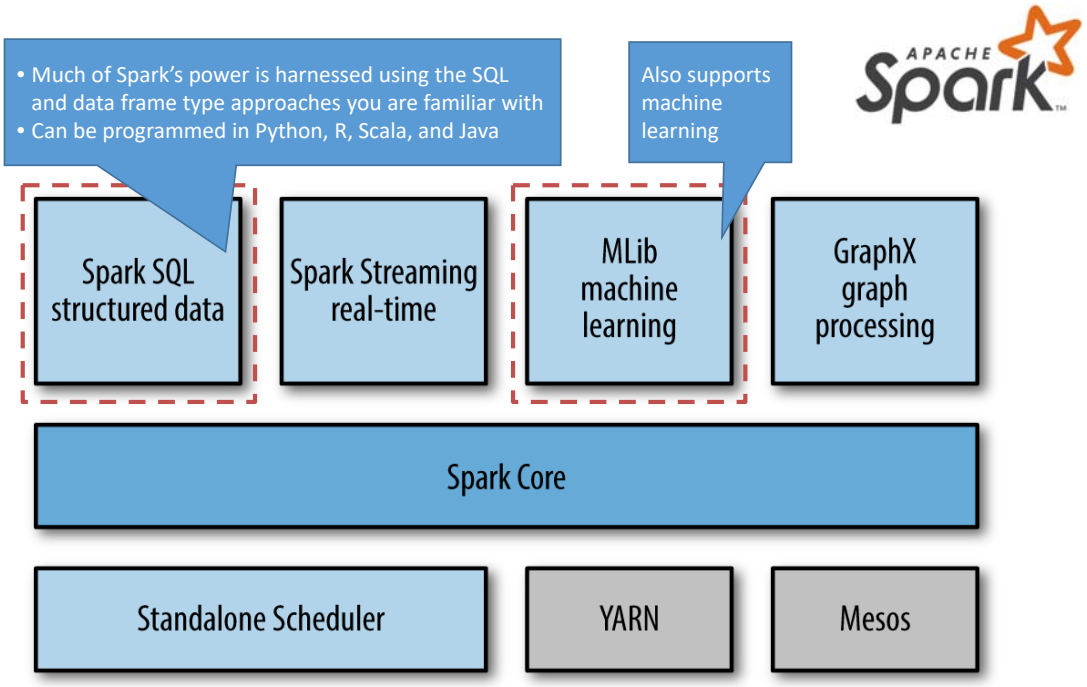
What is Spark?

amplab UC BERKELEY

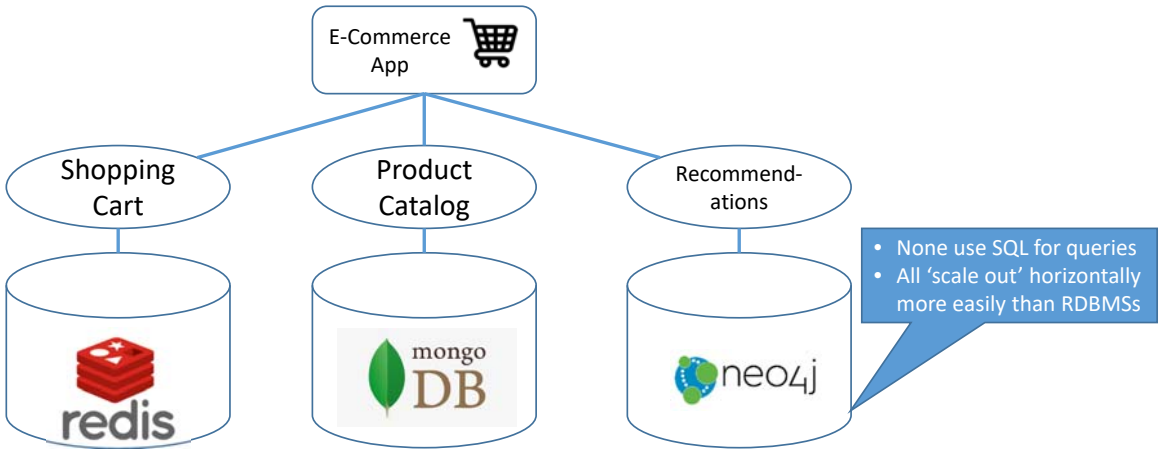
APACHE Spark™

- Apache Spark is a cluster computing platform on top of storage layer
  - a fast, general-purpose engine for large-scale data processing and analysis developed by AMPlab at UC Berkeley in 2009
  - Contributed to Apache Software Foundation in 2013
- Extends MapReduce with support for more components
  - Multi-pass analytics (e.g. machine learning, graph analytics)
  - Real-time streaming processing
  - Interactive ad-hoc analysis
- Runs in memory
  - Spark offers the ability to run computations in memory (can be 100x faster than MapReduce on some jobs

Key Spark component builds on the popularity of SQL



NoSQL (Not Only SQL) are non-relational databases that play an increasing role in data storage



Redis (Remote Dictionary Server) is an in-memory key-value database
- Stores python-like dictionary data structures
- Very fast in data caching applications

MongoDB is a leading document database – type of NoSQL database with a document-based data model
- Data stored in JSON like format
- Allows querying of semi-structured data
- Ideal in role to support a diverse product portfolio with complex querying and filtering across many product attributes – can populate web page with single query

Neo4j is a leading graph database – type of NoSQL database storing graph structures
- Keeping track of customer purchases is a common use case
- Helps in generating personalized product recommendations
- Can also be used to represent social network graphs

Examples of big data applications

- Recommender systems – Netflix, Amazon, Social Networks
- Netflix picked director and actors for "House of Cards" based on correlations in its data
- Wal-Mart uses text analysis and ML to produce better search results -> leading to billions in extra sales
- Morton Steak House used it to find a great PR opportunity
- Crime prediction
- Amex uses it to predict customer churn
- Insurance company used it to mine adjusters' reports to recover $12 million of fraudulent claims
- Finding relative with DNA sequencing
- Lots of other examples in health, manufacturing, government, water, energy . . . .

Agenda

Database connections and semi-structured data

Business analytics and the advent of big data

Technical challenges of big data

Intro to Hadoop and other **Big Data** technologies

Wrapping up the course

Final exam on Thursday

¶ Final exam 2hr 30min window from 9am 22/10 – 9am 23/10

- Open book and notes
- Covers all content of course (but less from Session 5)
- More emphasis on “hands on” topics than last year
- It will cover SQL and data modeling

¶ Any final questions??

¶ Exam

¶ MAM Programme

¶ Career