COMP9313: Big Data Management

Lecturer: Xin Cao

Course web site: http://www.cse.unsw.edu.au/~cs9313/

Chapter 1: Course Information and Introduction to Big Data Management

Part 1: Course Information

Course Info

- Lectures: 6:00 – 9:00 pm (Tuesday)
- Location:
  - Old Main Building 230 (K-K15-230)
  - Webstream
- Labs: Weeks 2-13
- Consultation (Weeks 1-12): Questions *regarding lectures, course materials, assignements, exam, etc.*
  - Time: 3:00 – 4:00 pm (Tuesday)
  - Place: 201D, K-17
- TA:
  - Xuefeng Chen, *xuefeng.chen@student.unsw.edu.au*
- Tutors: Xuefeng Chen, Wei Li, You Peng, Yu Hao
- Discussion and QA: WebCMS3

Lecturer in Charge

- Lecturer: Xin Cao
  - Office: 201D K17 (outside the lift turn left)
  - Email: *xin.cao@unsw.edu.au*
  - Ext: 55932
- Research interests
  - Database
  - Data Mining
  - Big Data Technologies
  - My homepage: http://www.cse.unsw.edu.au/~z3515164/
  - My publications list at google scholar:
    https://scholar.google.com.au/citations?user=kJIkUagAAAAJ&hl=en

Course Aims

- This course aims to introduce you to the concepts behind Big Data, the core technologies used in managing large-scale data sets, and a range of technologies for developing solutions to large-scale data analytics problems.

- This course is intended for students who want to understand modern large-scale data analytics systems. It covers a wide range of topics and technologies, and will prepare students to be able to build such systems as well as use them efficiently and effectively to address challenges in big data management.

- *Not possible to cover every aspect of big data management.*

Lectures

- Lectures focusing on the frontier technologies on big data management and the typical applications
- Try to run in more interactive mode and provide more examples
- A few lectures may run in more practical manner (e.g., like a lab /demo) to cover the applied aspects
- Lecture length varies slightly depending on the progress (of that lecture) λ
- Note: attendance to every lecture is assumed

Resources

- Text Books
  - Hadoop: The Definitive Guide. Tom White. 4th Edition - O'Reilly Media
  - Mining of Massive Datasets. Jure Leskovec, Anand Rajaraman, Jeff Ullman. 2nd edition - Cambridge University Press
  - Data-Intensive Text Processing with MapReduce. Jimmy Lin and Chris Dyer. University of Maryland, College Park.
  - Learning Spark . Matei Zaharia, Holden Karau, Andy Konwinski, Patrick Wendell. O'Reilly Media
- Reference Books and other readings
  - Apache MapReduce Tutorial
  - Apache Spark Quick Start
  - Many other online tutorials … …
- Big Data is a relatively new topic (so no fixed syllabus)

Prerequisite

- Official prerequisite of this course is COMP9024 (Data Structures and Algorithms) and COMP9311 (Database Systems).
- Before commencing this course, you should:
  - have experiences and good knowledge of algorithm design (equivalent to COMP9024)
  - have a solid background in database systems (equivalent to COMP9311)
  - **have solid programming skills in Java**
  - **be familiar with working on a Unix-style operating systems**
  - have basic knowledge of linear algebra (e.g., vector spaces, matrix multiplication), probability theory and statistics , and graph theory
- No previous experience necessary in
  - MapReduce/Spark
  - Parallel and distributed programming

Please do not enrol if you

- Don’t have COMP9024/9311 knowledge
- Cannot produce correct Java program on your own
- Never worked on Unix-style operating systems
- Have poor time management
- Are too busy to attend lectures/labs

- *Otherwise, you are likely to perform badly in this subject*

Learning outcomes

- After completing this course, you are expected to:
  - elaborate the important characteristics of Big Data
  - develop an appropriate storage structure for a Big Data repository
  - utilize the map/reduce paradigm and the to manipulate Big Data
  - utilize the Spark platform to manipulate Big Data
  - develop efficient solutions for analytical problems involving Big Data

Assessment

Number	Name	Full Mark
1*	Coding Project 1	10
2**	Coding Project 2	25
3**	Coding Project 3	25
4**	Coding Project 4	40
5	Final Exam	100

Later Submission Penalties:
* : zero marks
** : 10% reduction of your marks for the 1st day, 30% reduction/day for the following days

The final mark is calculated by the harmonic mean:
Final Mark= 2 * (proj1 + proj2 + proj3 + proj4) * FinalExam / (proj1 + proj2 + proj3 + proj4 + FinalExam)

You also need to achieve at least 40 marks in the final exam to pass the course.

Coding Projects

- Projects:
  - 1 warm-up programming project on Hadoop MapReduce
  - 1 harder project on Hadoop MapReduce
  - 1 project on Spark
  - 1 project on AWS (MapReduce/Spark)

- Both results and source codes will be checked.
  - If not able to run your codes due to some bugs, you will not lose all marks.

CSE Computing Environment

- Use Linux/command line (virtual machine image will be provided)
  - Projects marked on Linux servers
  - You need to be able to upload, run, and test your program under Linux
- Assignment submission
  - Use Give to submit (either command line or web page)
  - Classrun. Check your submission, marks, etc. Read https://wiki.cse.unsw.edu.au/give/Classrun

Final exam

- Final written exam (100 pts)
- If you are ill on the day of the exam, do not attend the exam – I will not accept any medical special consideration claims from people who already attempted the exam
- **You need to achieve at least 40 marks in the final exam**
- No supplementary exam will be given

You May Fail Because …

- *Plagiarism*
- Code failed to compile due to some mistakes
- Late submission
  - 1 sec late = 1 day late
  - submit wrong files
- Program did not follow the spec

- I am unlikely to accept the following excuses:
  - "Too busy"
  - "It took longer than I thought it would take"
  - "It was harder than I initially thought"
  - … …

Tentative Course Schedule

Week	Topic	Assignment
1	Course info and introduction to big data	
2	Hadoop MapReduce 1	
3	Hadoop MapReduce 2	Proj1
4	Hadoop MapReduce 3	
5	Graph data processing	Proj2
6	Spark 1	
7	Spark 2	Proj3
8	Data stream mining	
9	Finding Similar Items	Proj4
10	Recommender Systems	
11	NoSQL and High Level MapReduce Tools	
12	Revision and exam preparation	

Labs

- 5 labs on MapReduce
- 3 labs on Spark
- 1 lab on high level MapReduce tools
- 1 lab on AWS
- 1 lab on big data machine learning platform [tentative]

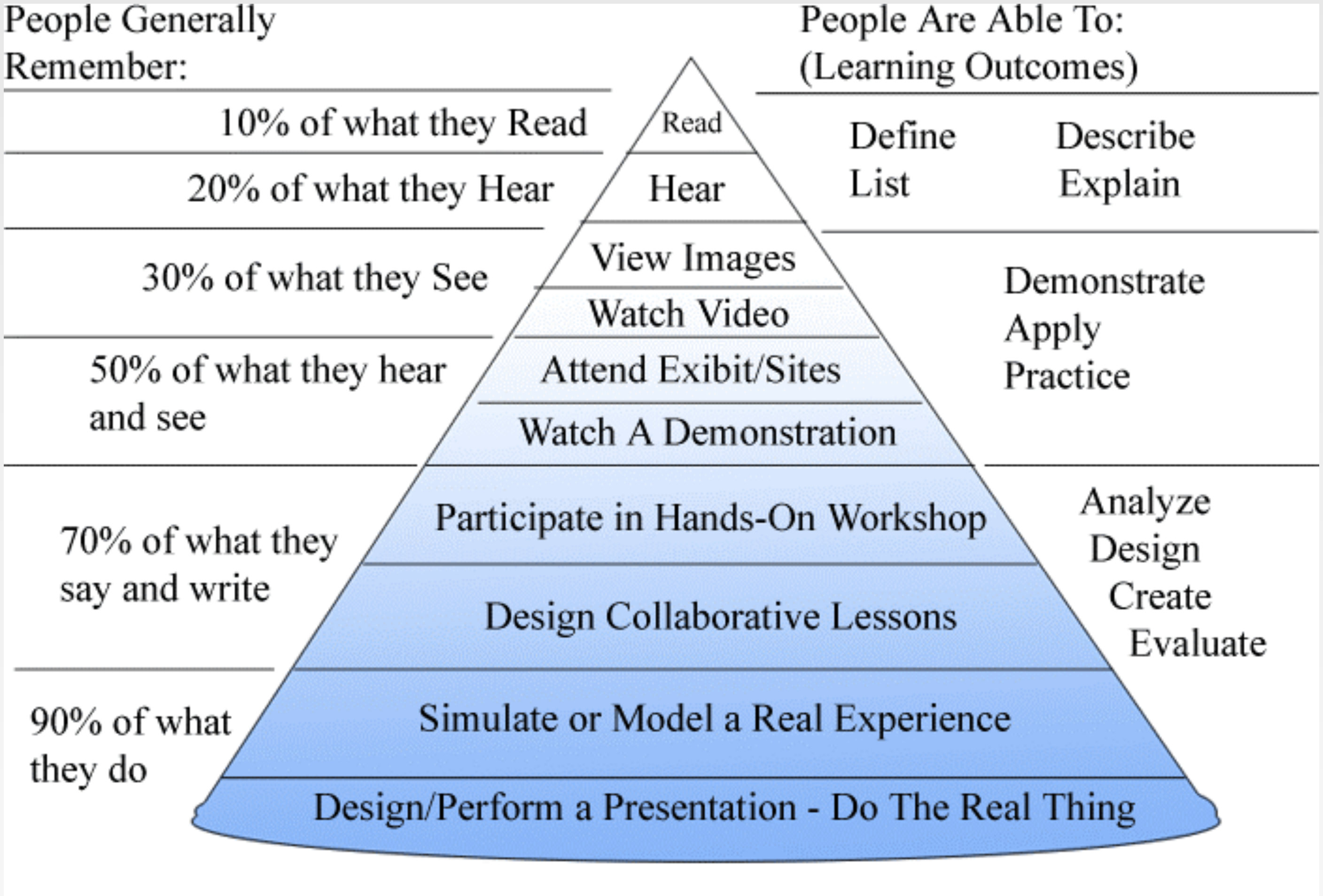
Virtual Machine

- Software: Virtualbox
  - Images:
    - Pure Xubuntu 14.04: http://www.cse.unsw.edu.au/~z3515164/Raw_Xubuntu.zip
    - Xubuntu 14.04 with pre-installed Hadoop and Eclipse plugin: http://mirror.cse.unsw.edu.au/pub/cs9313/Xubuntu.zip
      - Download the zip file and uncompress it, and rename the file "xubuntu-disk.vmdk" as "xubuntu-disk2.vmdk“
      - Open VirtualBox, File->Import Applicance
      - Browse the image folder, select the "*.ovf" file
      - The image will be imported to your computer, which may take 10 minutes
      - comp9313 is used as both username and password. The hadoop installation path is the same as in the virtual machine on lab computers.

Your Feedbacks Are Important

- Big data is a new topic, and thus the course is tentative
- The technologies keep evolving, and the course materials need to be updated correspondingly
- Please advise where I can improve after each lecturer, at the discussion and QA website
- myExperience system

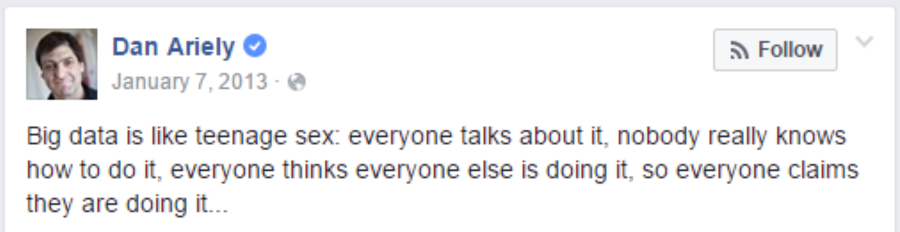
Why Attend the Lectures?



Part 2: Introduction to Big Data

What is Big Data?

- No standard definition! here is from Wikipedia:
  - Big data is a term for data sets that are so voluminous or complex that traditional data processing application software are inadequate to deal with them
  - Challenges include capture, storage, analysis, data curation, search, sharing, transfer, visualization, querying, updating and information privacy.
  - The term "big data" often refers simply to the use of *predictive analytics*, *user behaviour analytics*, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set
  - Analysis of data sets can find new correlations to "spot business trends, prevent diseases, combat crime and so on."



Instead of Talking about "Big Data"…

- Let's talk about a crowded application ecosystem:
  - Hadoop MapReduce
  - Spark
  - NoSQL (e.g., HBase, MongoDB, Neo4j)
  - Pregel
  - … …

- Let's talk about data science and data management:
  - Finding similar items
  - Graph data processing
  - Streaming data processing
  - Machine learning technologies
  - … …

Who is generating Big Data?

Social

User Tracking & Engagement

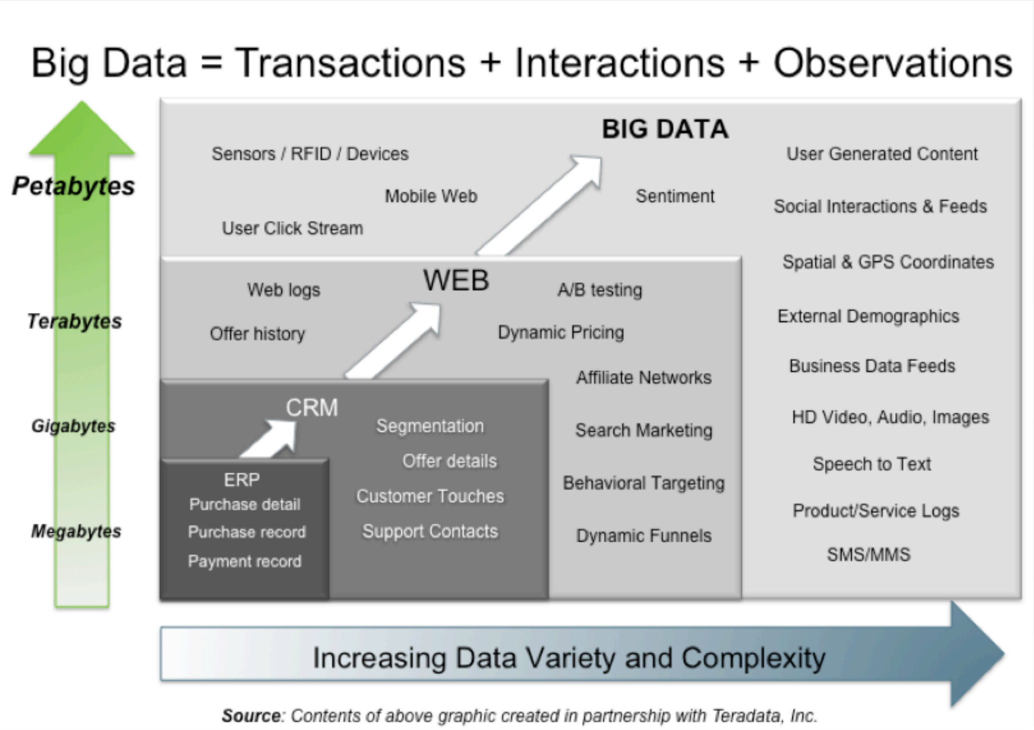
Homeland Security

eCommerce

Financial Services

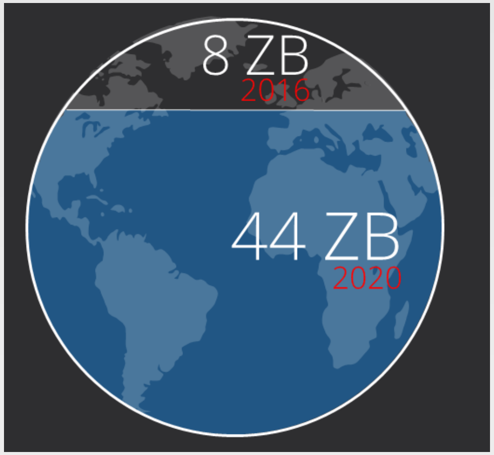
Real Time Search

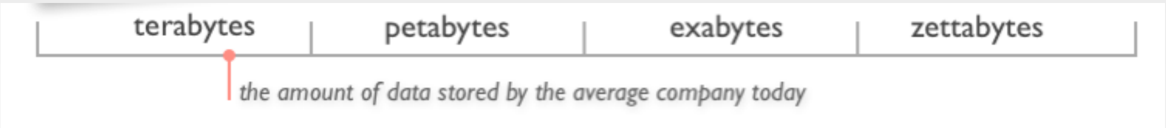
Big Data Characteristics: 3V



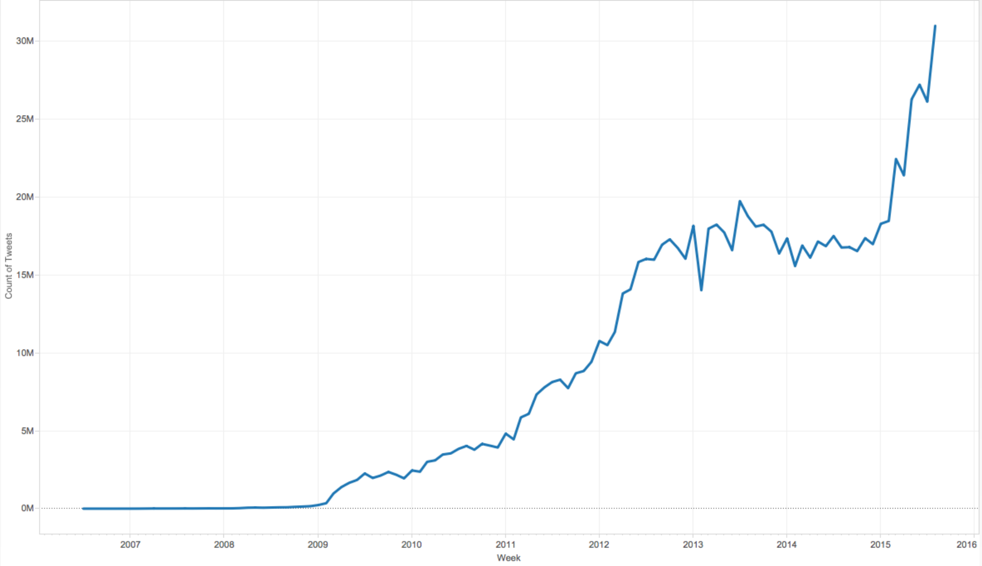
Volume (Scale)

- Data Volume
  - Growth 40% per year
  - From 8 zettabytes (2016) to 44zb (2020)
- Data volume is increasing exponentially





Number of Tweets



Recent Twitter Statistics

Total Number of Monthly Active Twitter Users:

330 million

Last updated: 1/1/18

Total Number of Tweets sent per Day:

500 million

Last updated: 1/24/17

Percentage of Twitter users on Mobile:

80%

Last updated: 1/24/17

Number of Twitter Daily Active Users:

100 million

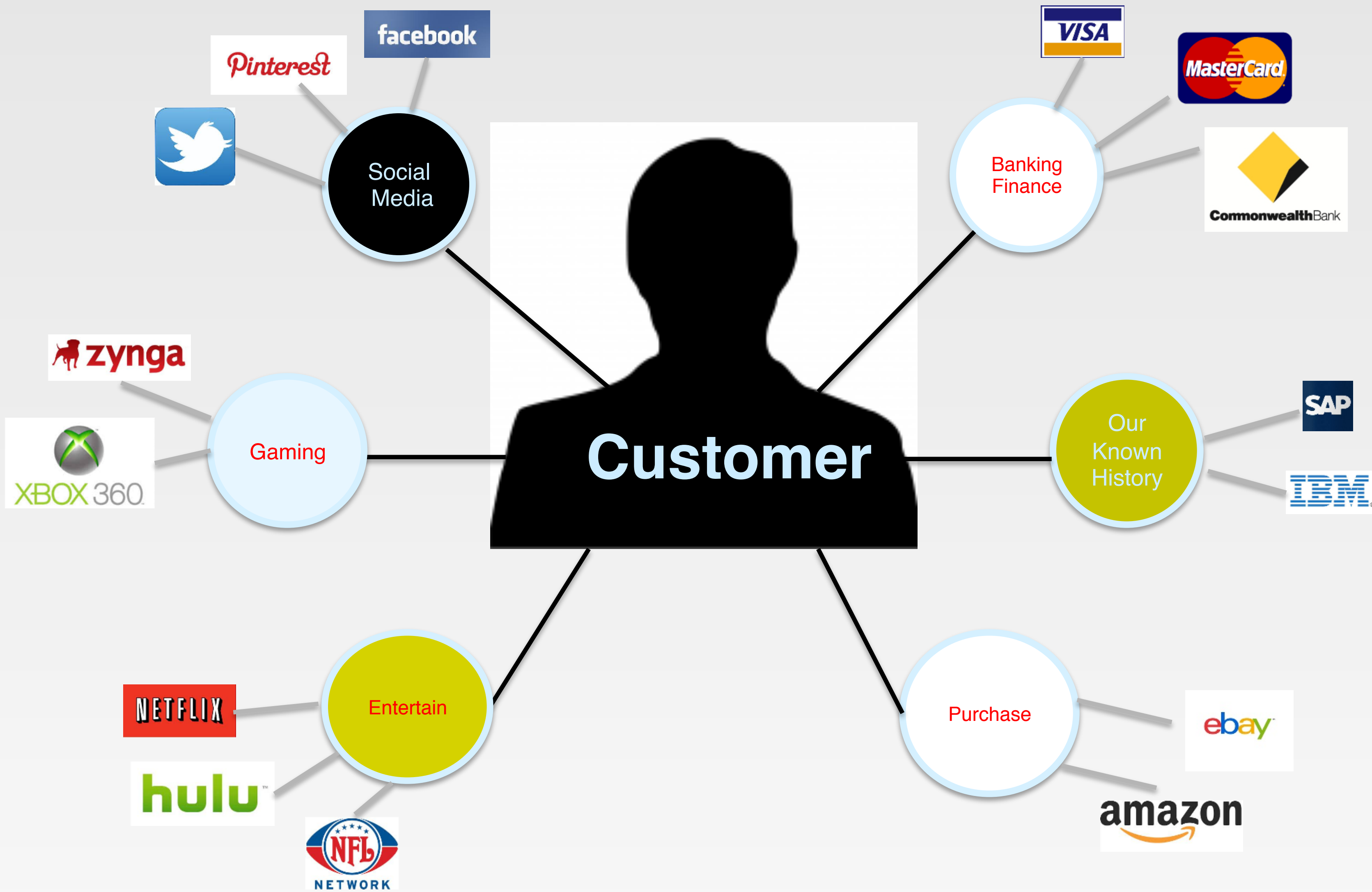
Last updated: 1/24/17

Variety (Complexity)

- Different Types:
  - Relational Data (Tables/Transaction/Legacy Data)
  - Text Data (Web)
  - Semi-structured Data (XML)
  - Graph Data
    - Social Network, Semantic Web (RDF), …
  - Streaming Data
    - You can only scan the data once
  - A single application can be generating/collecting many types of data
- Different Sources:
  - Movie reviews from IMDB and Rotten Tomatoes
  - Product reviews from different provider websites

To extract knowledge→ all these types of data need to linked together

A Single View to the Customer



A Global View of Linked Big Data

Diversified social network

Velocity (Speed)

- Data is being generated fast and need to be processed fast
- Online Data Analytics
- Late decisions ➔ missing opportunities
- Examples
  - **E-Promotions:** Based on your current location, your purchase history, what you like ➔ send promotions right now for store next to you
  - **Healthcare monitoring:** sensors monitoring your activities and body ➔ any abnormal measurements require immediate reaction
  - **Disaster management and response**

Velocity in Real-world

- Every second, on average, around **6,000** tweets are tweeted on Twitter (visualize them here), which corresponds to over **350,000** tweets sent per minute, **500 million**tweets per day and around **200 billion** tweets per year.

- The statistics for 1 second in many applications. http://www.internetlivestats.com/one-second/

Extended Big Data Characteristics: 6V

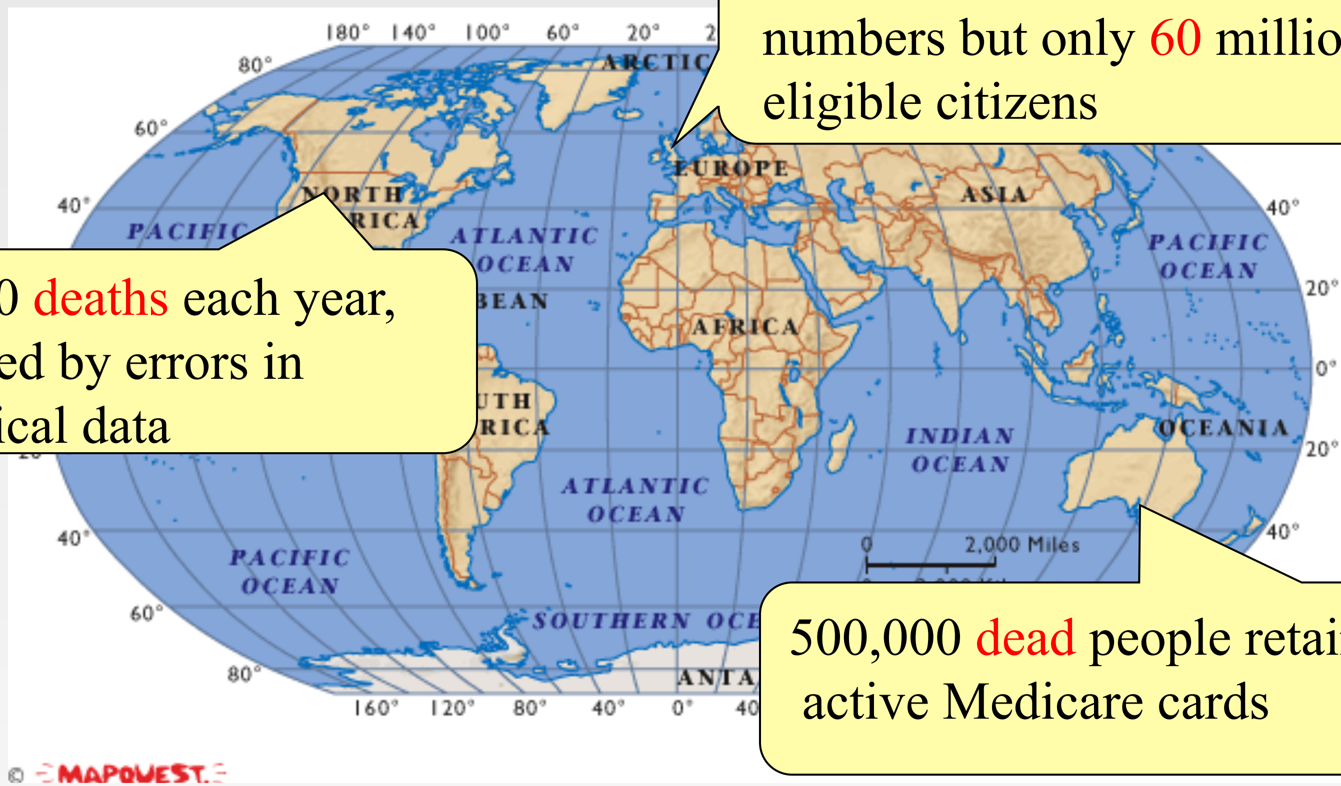
- Volume: In a big data environment, the amounts of data collected and processed are much larger than those stored in typical relational databases.
- Variety: Big data consists of a rich variety of data types.
- Velocity: Big data arrives to the organization at high speeds and from multiple sources simultaneously.

- Veracity: Data quality issues are particularly challenging in a big data context.
- Visibility/Visualization: After big data being processed, we need a way of presenting the data in a manner that's readable and accessible.
- Value: Ultimately, big data is meaningless if it does not provide value toward some meaningful goal.

Veracity (Quality & Trust)

- *Data = quantity + quality*
- When we talk about big data, we typically mean its quantity:
  - What capacity of a system provides to cope with the sheer size of the data?
  - Is a query feasible on big data within our available resources?
  - How can we make our queries tractable on big data?
  - . . .
- Can we trust the answers to our queries?
  - Dirty data routinely lead to misleading financial reports, strategic business planning decision ⇒ loss of revenue, credibility and customers, disastrous consequences
- *The study of data quality is as important as data quantity*

Data in real-life is often dirty



81 million National Insurance numbers but only 60 million eligible citizens

98000 deaths each year, caused by errors in medical data

500,000 dead people retain active Medicare cards

Visibility/Visualization

- Visibility: the state of being able to see or be seen is implied.
  - Big Data – visibility = Black Hole?

- Visualization: Making all that vast amount of data comprehensible in a manner that is easy to understand and read.

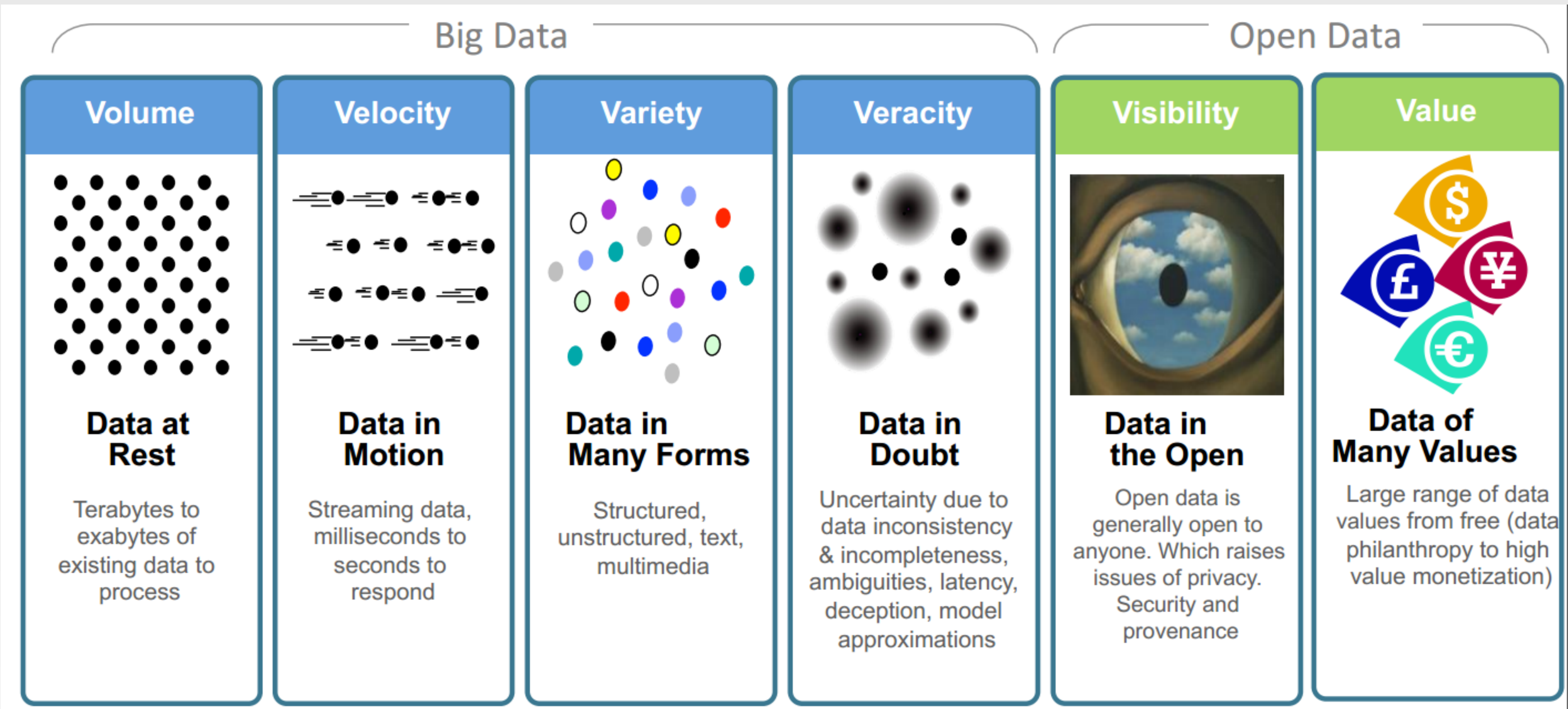
A visualization of Divvy bike rides across Chicago

- Big data visualization tools:

Value

- Big data is meaningless if it does not provide value toward some meaningful goal

Big Data: 6V in Summary



Transforming Energy and Utilities through Big Data & Analytics. By Anders Quitzau@IBM

Other V’s

- Variability
  - Variability refers to data whose meaning is constantly changing. This is particularly the case when gathering data relies on language processing.
- Viscosity
  - This term is sometimes used to describe the latency or lag time in the data relative to the event being described. We found that this is just as easily understood as an element of Velocity.
- Volatility
  - Big data volatility refers to how long is data valid and how long should it be stored. You need to determine at what point is data no longer relevant to the current analysis.
- More V’s in the future …
  - How many v’s are there in big data? http://www.clc-ent.com/TBDE/Docs/vs.pdf

Tag Clouds of Big Data

Why Study Big Data Technologies?

- The hottest topic in both research and industry
- Highly demanded in real world
- A promising future career
  - Research and development of big data systems:

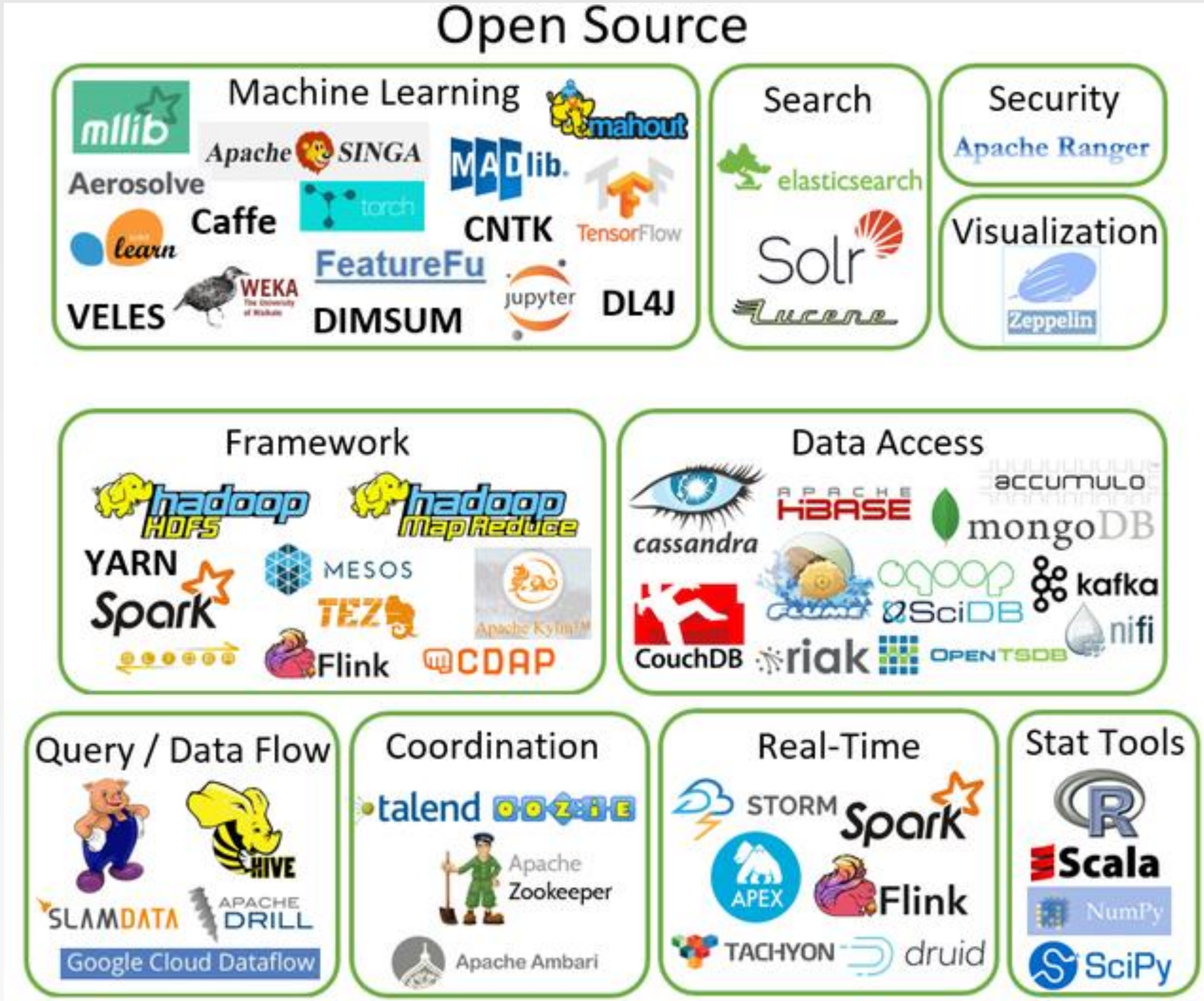
    distributed systems (eg, Hadoop), visualization tools, data warehouse, OLAP, data integration, data quality control, …
  - Big data applications:

    social marketing, healthcare, …
  - Data analysis: to get values out of big data

    discovering and applying patterns, predicative analysis, business intelligence, privacy and security, …

- Get enough credits

Big Data Open Source Tools



What will the course cover

- Topic 1. Big data management tools
  - Apache Hadoop
    - MapReduce
    - YARN/HDFS/HBase/Hive/Pig (briefly introduced)
    - Spark
    - AWS platform
    - Mahout [tentative]

- Topic 2. Big data typical applications
  - Finding similar items
  - Graph data processing
  - Data stream mining
  - Recommender Systems

Distributed processing is non-trivial

- How to assign tasks to different workers in an efficient way?
- What happens if tasks fail?
- How do workers exchange results?
- How to synchronize distributed tasks allocated to different workers?

Big data storage is challenging

- Data Volumes are massive
- Reliability of Storing PBs of data is challenging
- All kinds of failures: Disk/Hardware/Network Failures
- Probability of failures simply increase with the number of machines …

What is Hadoop

- Open-source data storage and processing platform
- Before the advent of Hadoop, storage and processing of big data was a big challenge
- Massively scalable, automatically parallelizable
  - Based on work from Google
    - Google: GFS + MapReduce + BigTable (Not open)
    - Hadoop: HDFS + Hadoop MapReduce + HBase（opensource)
- Named by Doug Cutting in 2006 (worked at Yahoo! at that time), after his son's toy elephant.

Hadoop offers

- Redundant, Fault-tolerant data storage
- Parallel computation framework
- Job coordination

Programmers

No longer need to worry about

Q: Where file is located?

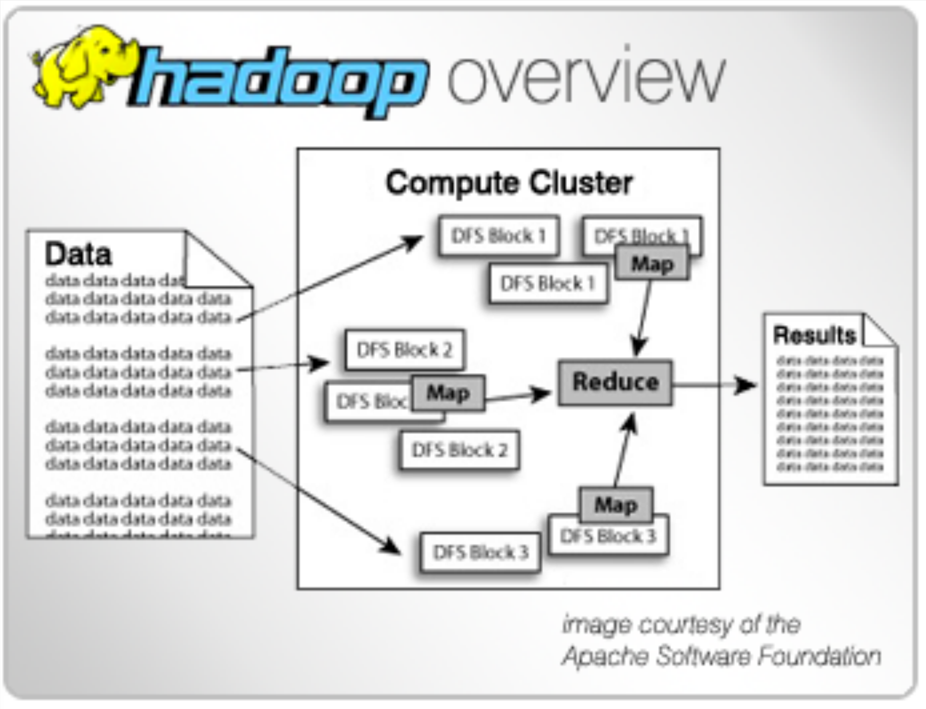
Q: How to handle failures & data lost?

Q: How to divide computation?

Q: How to program for scaling?

Why Use Hadoop?

- Cheaper
  - Scales to Petabytes or more easily
- Faster
  - Parallel data processing
- Better
  - Suited for particular types of big data problems



Companies Using Hadoop

facebook

The New York Times

ebay

hp

IBM

JPMorganChase

eHarmony

twitter

intel

NETFLIX

rackspace HOSTING

amazon.com

VISA

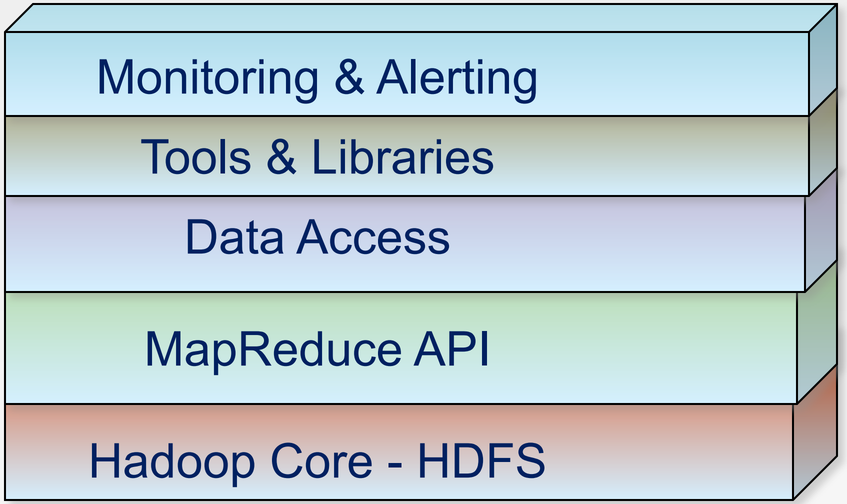
NING

SAMSUNG

YAHOO!

Hadoop is a set of Apache Frameworks and more…

- Data storage (HDFS)
  - Runs on commodity hardware (usually Linux)
  - Horizontally scalable
- Processing (MapReduce)
  - Parallelized (scalable) processing
  - Fault Tolerant
- Other Tools / Frameworks
  - Data Access
    - HBase, Hive, Pig, Mahout
  - Tools
    - Hue, Sqoop
  - Monitoring
    - Greenplum, Cloudera



What are the core parts of a Hadoop distribution?

HDFS Storage

- Redundant (3 copies)
- For large files – large blocks
- 64 or 128 MB / block
- Can scale to 1000s of nodes

MapReduce API

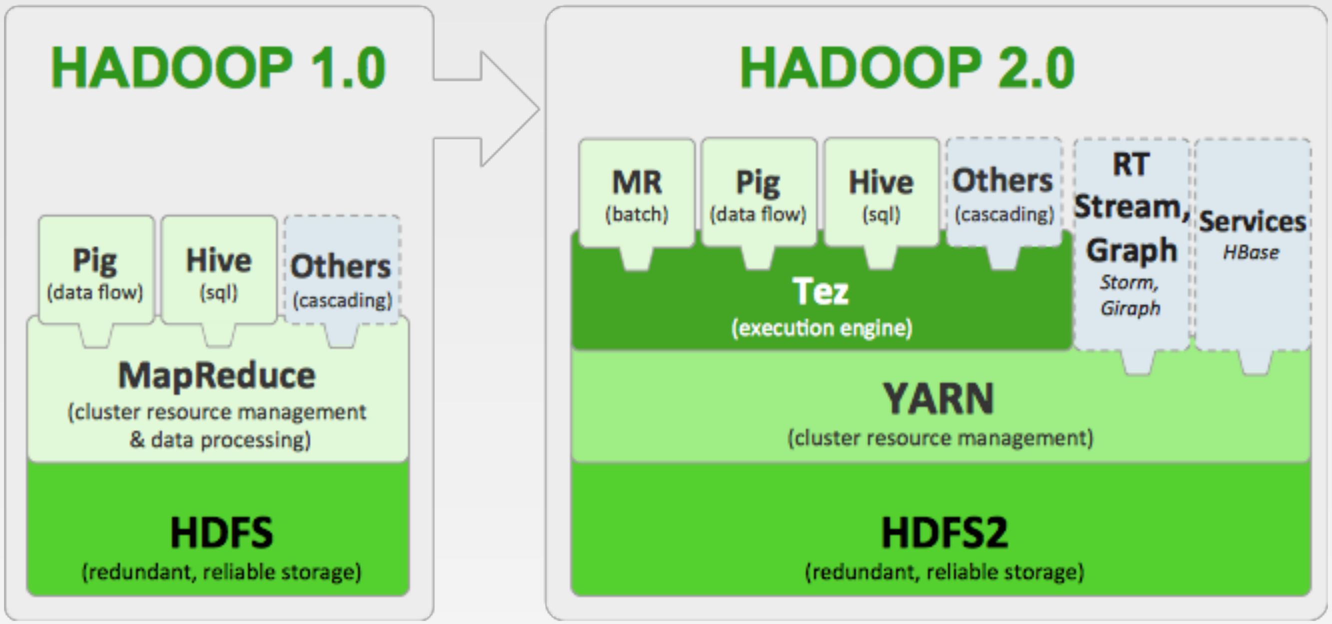
- Batch (Job) processing
- Distributed and Localized to clusters (Map)
- Auto-Parallelizable for huge amounts of data
- Fault-tolerant (auto retries)
- Adds high availability and more

Other Libraries

- Pig
- Hive
- HBase
- Others

Hadoop 2.0

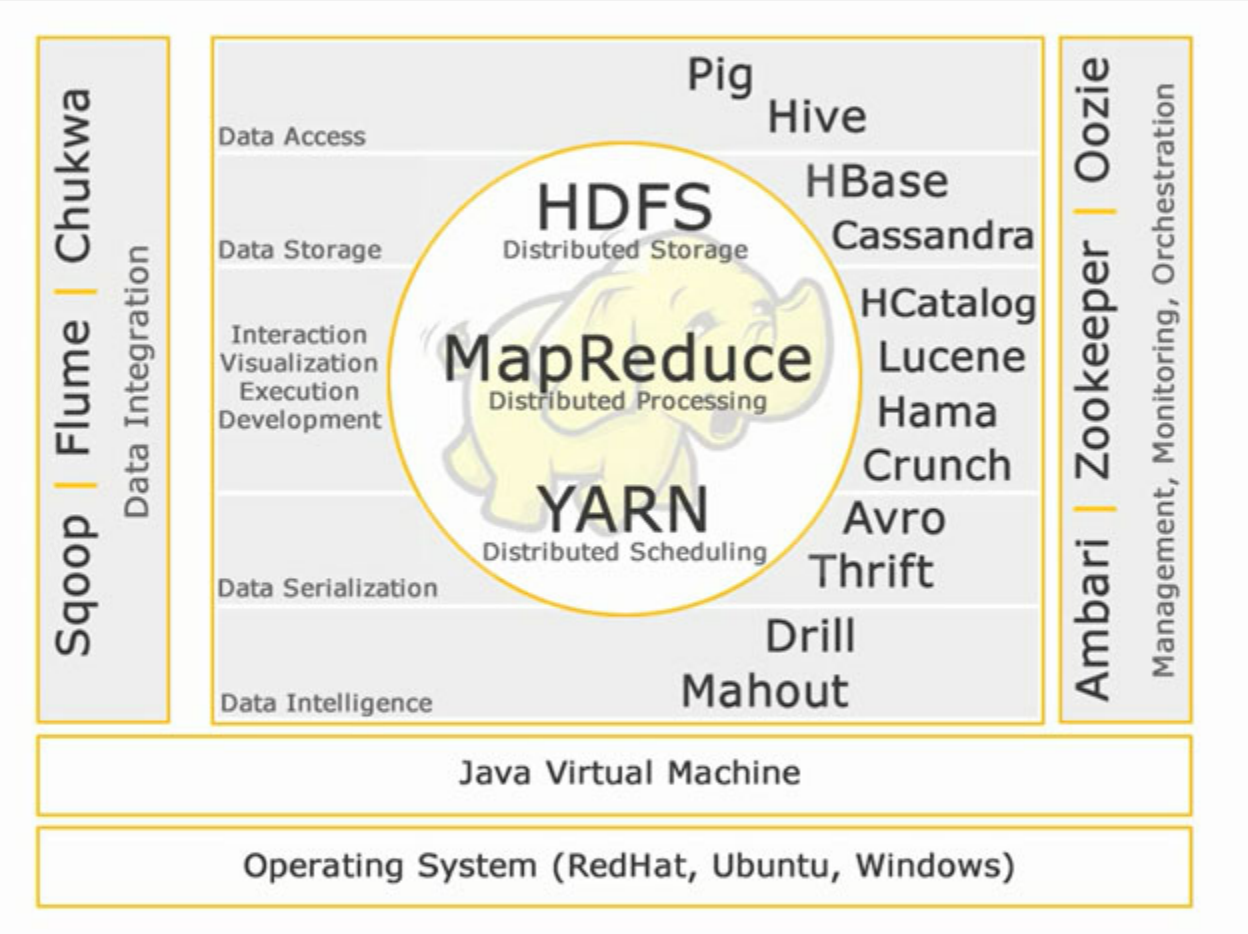
- Single Use System
  - Batch apps
- Multi-Purpose Platform
  - Batch, Interactive, Online, Streaming



Hadoop YARN *(Yet Another Resource Negotiator)*: a resource -management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications

Hadoop Ecosystem

A combination of technologies which have proficient advantage in solving business problems.



http://www.edupristine.com/blog/hadoop-ecosystem-and-components

Common Hadoop Distributions

- Open Source
  - Apache
- Commercial
  - Cloudera
  - Hortonworks
  - MapR
  - AWS MapReduce
  - Microsoft Azure HDInsight (Beta)

Setting up Hadoop Development

Hadoop Binaries

Local install
- Linux
- Windows

Cloudera's Demo VM
- Need Virtualization software, i.e. VMware, etc…

Cloud
- AWS
- Microsoft (Beta)
- Others

Data Storage

Local
- File System
- HDFS Pseudo-distributed (single-node)

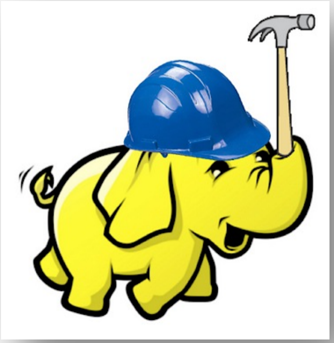
Cloud
- AWS
- Azure
- Others

MapReduce

- Local
- Cloud

Other Libraries & Tools

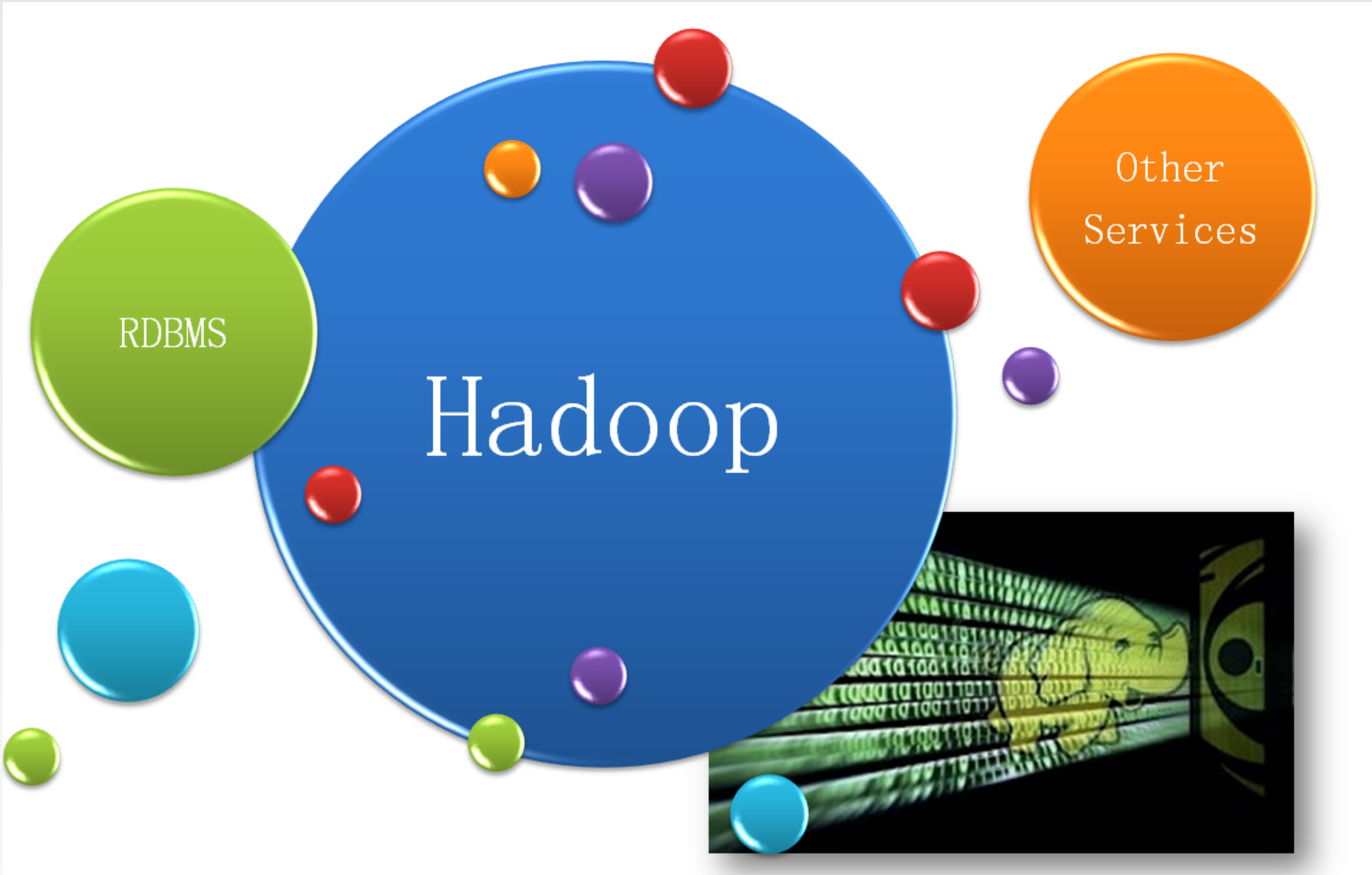
- Vendor Tools
- Libraries



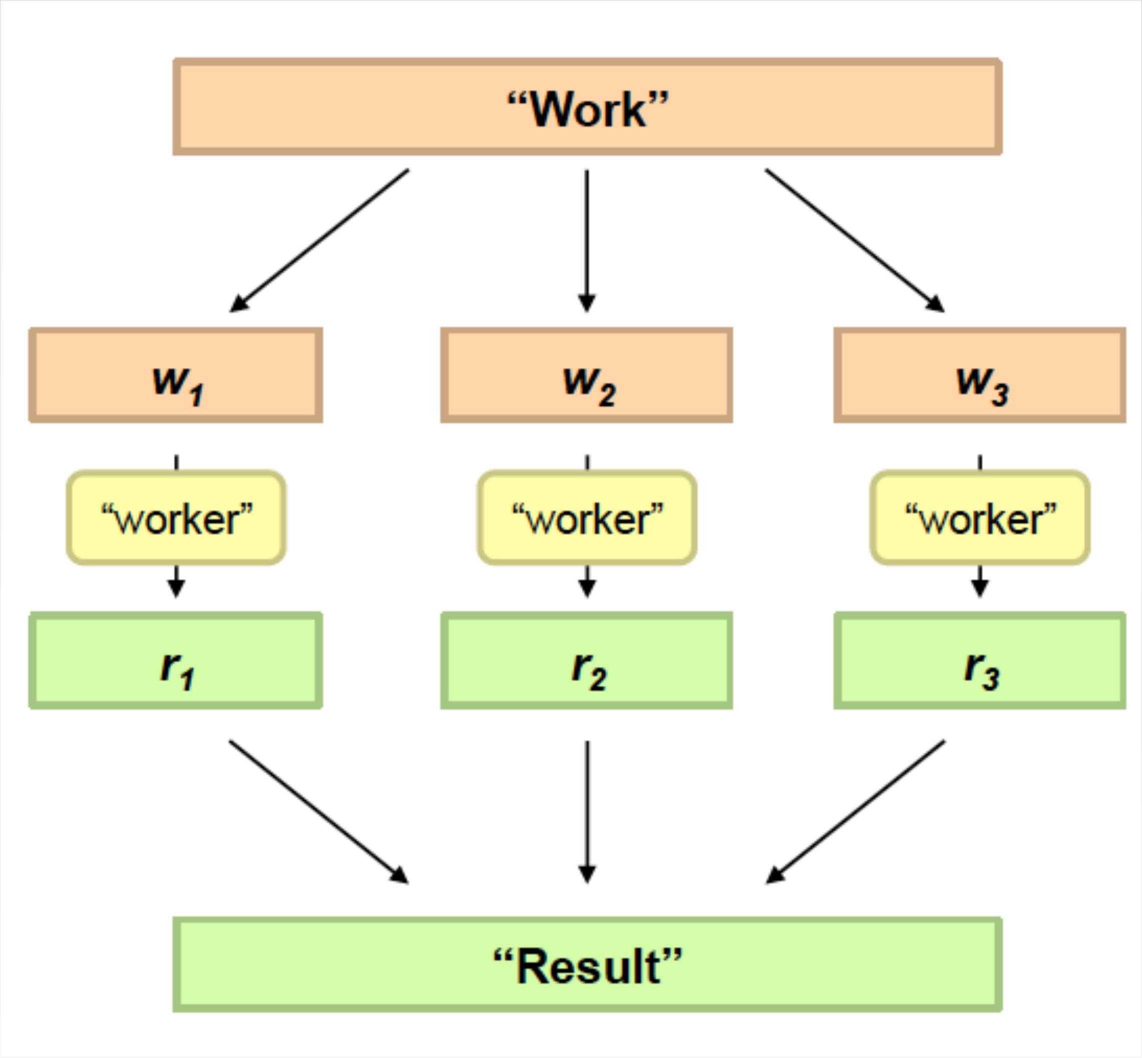
Comparing: RDBMS vs. Hadoop

Feature	RDBMS	Hadoop
Data Variety	Mainly for Structured data.	Used for Structured, Semi-Structured and Unstructured data
Data Storage	Average size data (GBS)	Use for large data set (Tbs and Pbs)
Querying	SQL Language	HQL (Hive Query Language)
Schema	Required on write (static schema)	Required on read (dynamic schema)
Speed	Reads are fast	Both reads and writes are fast
Cost	License	Free
Use Case	OLTP (Online transaction processing)	Analytics (Audio, video, logs etc), Data Discovery
Data Objects	Works on Relational Tables	Works on Key/Value Pair
Throughput	Low	High
Scalability	Vertical	Horizontal
Hardware Profile	High-End Servers	Commodity/Utility Hardware
Integrity	High (ACID)	Low

The Changing Data Management Landscape



Philosophy to Scale for Big Data Processing



Divide Work

Combine Results

MapReduce

- Typical big data problem
  - Iterate over a large number of records
  - Extract something of interest from each
  - Shuffle and sort intermediate results
  - Aggregate intermediate results
  - Generate final output

Map

Reduce

Key idea: provide a functional abstraction for these two operations

- Programmers specify two functions:

  map $(k_1, v_1) \rightarrow [<k_2, v_2>]$

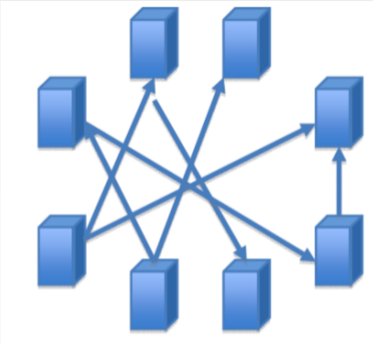
  reduce $(k_2, [v_2]) \rightarrow [<k_3, v_3>]$

  - All values with the same key are sent to the same reducer
- The execution framework handles everything else…

Understanding MapReduce

- **Map>>**
  - (K1, V1) →
    - Info in
    - Input Split
  - list (K2, V2)
    - Key / Value out (intermediate values)
    - One list per local node
    - Can implement local Reducer (or Combiner)

- **Shuffle/Sort>>**



- **Reduce**
  - (K2, list(V2)) →
    - Shuffle / Sort phase precedes Reduce phase
    - Combines Map output into a list
  - list (K3, V3)
    - Usually aggregates intermediate values

(input) <k1, v1> → **map** → <k2, v2> → **combine** → <k2, list(V2)> → **reduce** → <k3, v3> *(output)*

WordCount - Mapper

- Reads in input pair <k1,v1>
- Outputs a pair <k2, v2>
  - Let’s count number of each word in user queries (or Tweets/Blogs)
  - The input to the mapper will be <queryID, QueryText>:

    <Q1,“The teacher went to the store. The store was closed; the store opens in the morning. The store opens at 9am.” >

  - The output would be:

    <The, 1> <teacher, 1> <went, 1> <to, 1> <the, 1> <store,1> <the, 1> <store, 1> <was, 1> <closed, 1> <the, 1> <store,1> <opens, 1> <in, 1> <the, 1> <morning, 1> <the 1> <store, 1> <opens, 1> <at, 1> <9am, 1>

WordCount - Reducer

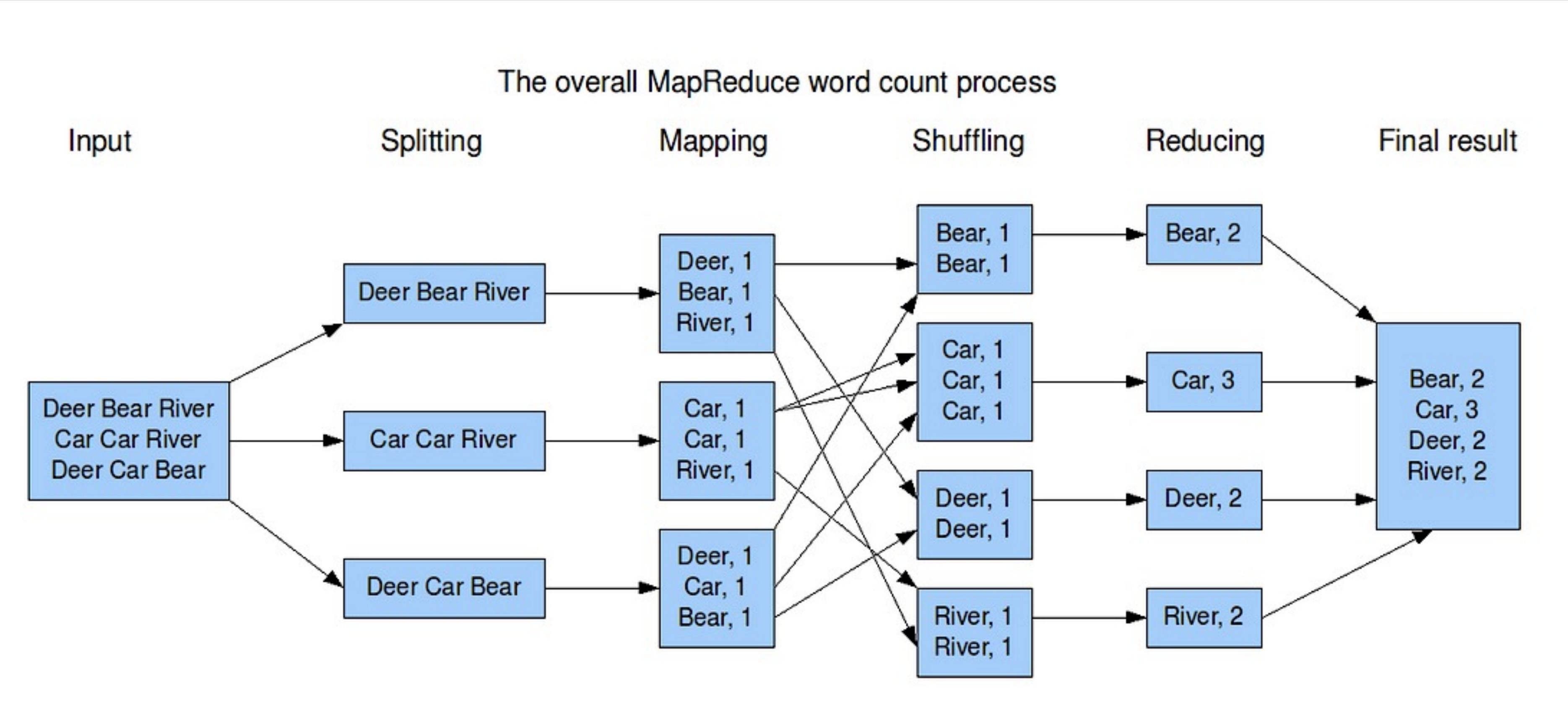
- Accepts the Mapper output (k2, v2), and aggregates values on the key to generate (k3, v3)
  - For our example, the reducer input would be:

    <The, 1> <teacher, 1> <went, 1> <to, 1> <the, 1> <**store**, 1>
    <the, 1> <store, 1> <was, 1> <closed, 1> <the, 1> <**store**, 1>
    <opens,1> <in, 1> <the, 1> <morning, 1> <the 1> <**store**, 1>
    <opens, 1> <at, 1> <9am, 1>

  - The output would be:

    <The, 6> <teacher, 1> <went, 1> <to, 1> **<store, 4>** <was, 1>
    <closed, 1> <opens, 2> <in, 1> <morning, 1> <at, 1> <9am, 1>

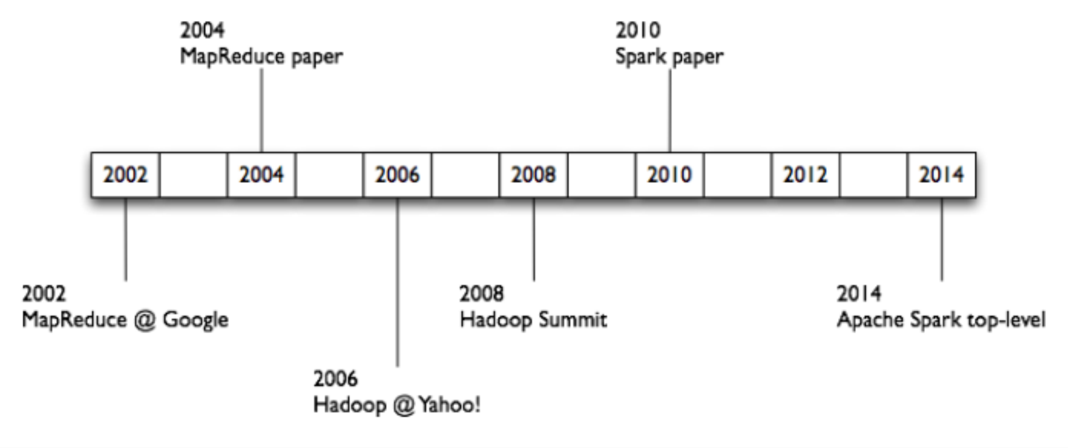
MapReduce Example - WordCount



- Hadoop MapReduce is an implementation of MapReduce
  - MapReduce is a computing paradigm (Google)
  - Hadoop MapReduce is an open-source software

Spark

- One popular answer to "What's beyond MapReduce?"
- Open-source engine for large-scale data processing
  - Supports generalized dataflows
  - Written in Scala, with bindings in Java and Python
- Brief history:
  - Developed at UC Berkeley AMPLab in 2009
  - Open-sourced in 2010
  - Became top-level Apache project in February 2014
  - Commercial support provided by DataBricks



Spark

- Fast and expressive cluster computing system interoperable with Apache Hadoop
- Improves efficiency through:
  - **In-memory** computing primitives
  - General computation graphs

→ Up to 100× faster (2-10× on disk)

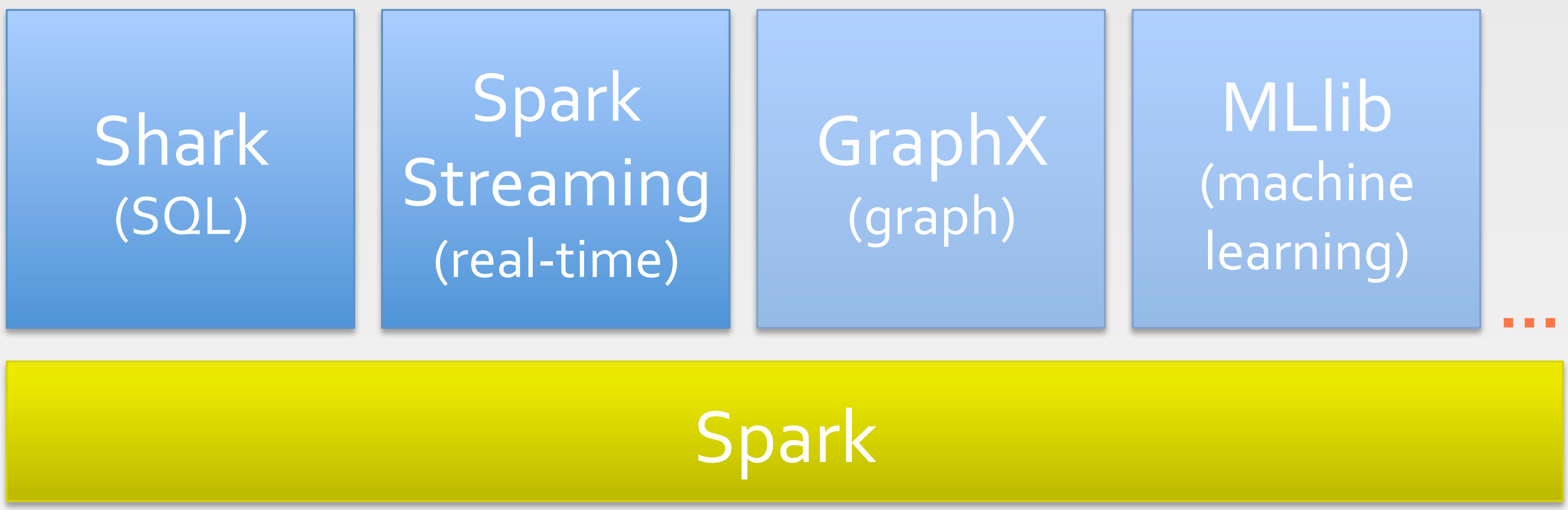
- Improves usability through:
  - Rich APIs in Scala, Java, Python
  - Interactive shell

→ Often 5× less code

- **Spark is not**
  - a modified version of Hadoop
  - dependent on Hadoop because it has its own cluster management
  - Spark uses Hadoop for storage purpose only

Spark Platform

- Spark is the basis of a wide set of projects in the Berkeley Data Analytics Stack (BDAS)



- Spark SQL (SQL on Spark)
- Spark Streaming (stream processing)
- GraphX (graph processing)
- MLlib (machine learning library)

Spark


Steven Luscher
@steveluscher
Map/filter/reduce in a tweet:
map([🌽, 🐮, 🐔], cook)
=> [🍿, 🍔, 🍳]
filter([🍿, 🍔, 🍳], isVegetarian)
=> [🍿, 🍳]
reduce([🍿, 🍳], eat)
=> 💩
RETWEETS
6,472
LIKES
6,357

WordCount in Spark (Scala)

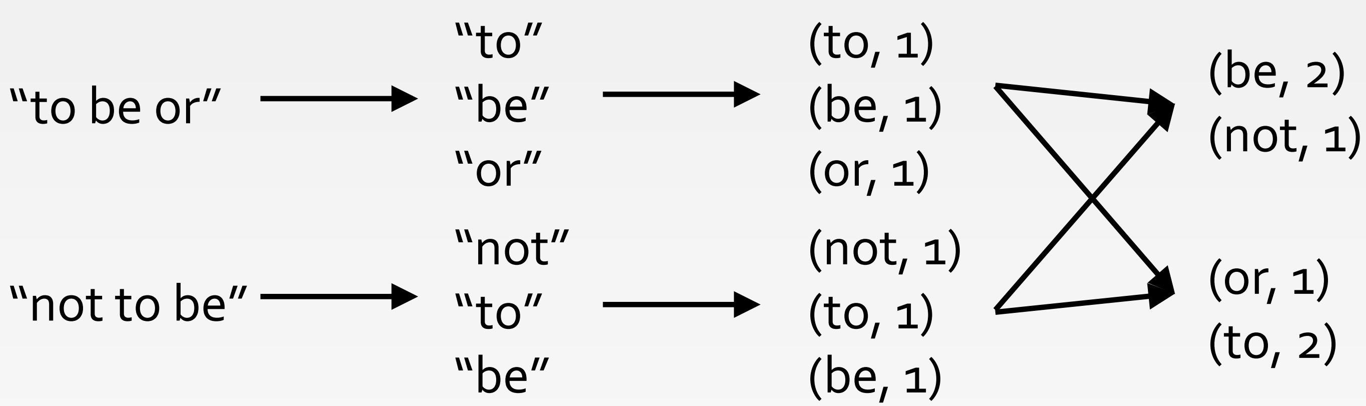
Transformation

```
val file = sc.textFile("hdfs://...")

val counts = file.flatMap(line => line.split(" "))
    .map(word => (word,1))
    .reduceByKey(_ + _)
```

Action

```
counts.saveAsTextFile("hdfs://...")
```



AWS (Amazon Web Services)

- Amazon

From Wikipedia 2006

Article | Talk | Read

Amazon.com

From Wikipedia, the free encyclopedia

This is an old revision of this page
05:10, 20 March 2006 (→Customer
permanent link to this revision, wh
revision.

(diff) ← Previous revision | Latest revisi

Amazon.com (NASDAQ: AMZN) is an American electronic commerce company based in Seattle, Washington. It was one of the

From Wikipedia 2017

Amazon.com

From Wikipedia, the free encyclopedia

Amazon.com, Inc. (/ˈæməzɒn/) is an American electronic commerce and cloud computing company that was founded on July 5, 1994 by Jeff Bezos and is based in Seattle, Washington. The tech giant is the largest Internet-based retailer in the world by total sales and market capitalization.[4]

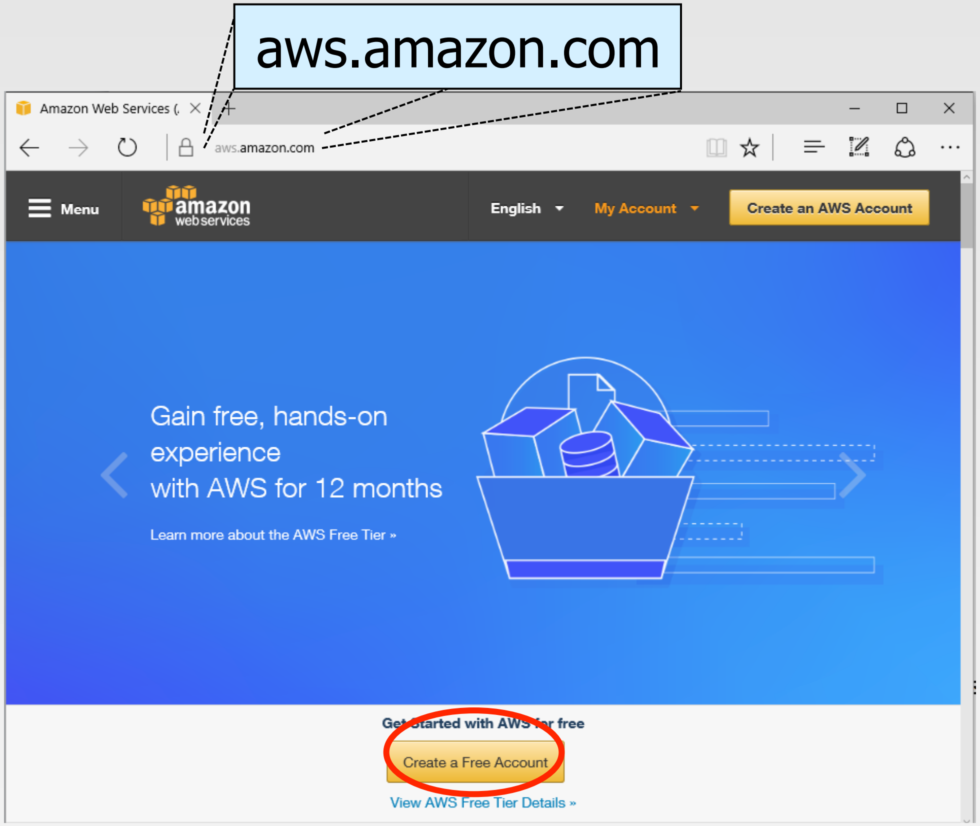
AWS (Amazon Web Services)

- AWS is a subsidiary of Amazon.com, which offers a suite of cloud computing services that make up an on-demand computing platform.
- 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 Elastic MapReduce (EMR)
    Scalable MapReduce computation
  - Amazon DynamoDB
    Distributed NoSQL database, one of several in AWS
  - Amazon SimpleDB
    Simple NoSQL database
  - ...

Cloud Computing Services in AWS

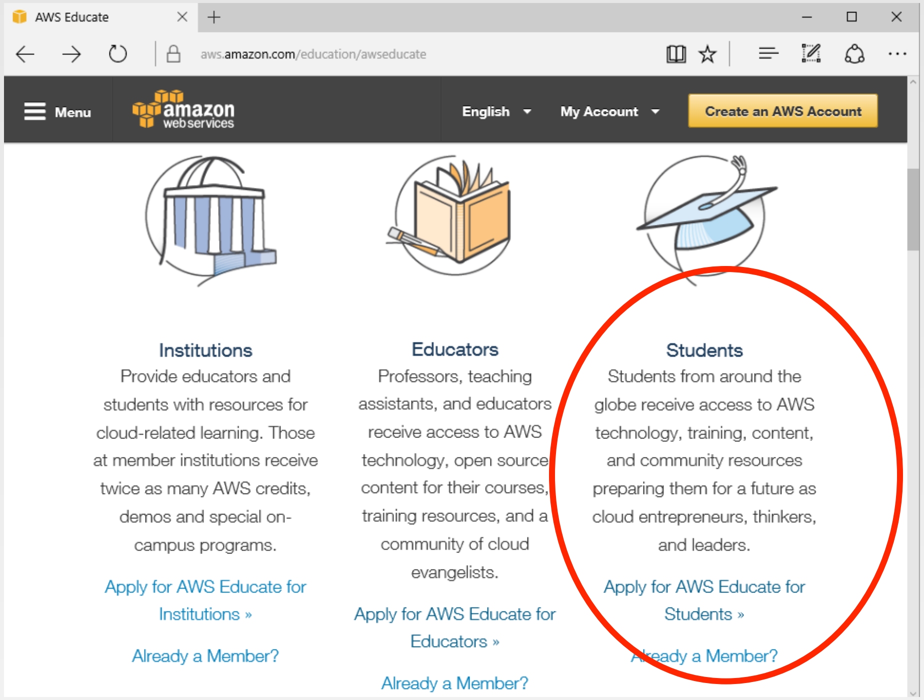
- IaaS
  - EC2, S3, …
  - *Highlight:* EC2 and S3 are two of the earliest products in AWS
- PaaS
  - Aurora, Redshift, …
  - *Highlight:* Aurora and Redshift are two of the fastest growing products in AWS
- SaaS
  - WorkDocs, WorkMail
  - *Highlight:* May not be the main focus of AWS

Setting up an AWS account



- Sign up for an account on aws.amazon.com
  - You need to choose an 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

Signing up for AWS Educate



- Complete the web form on https://aws.amazon.com/education/awseducate/
  - Assumes you already have an AWS account
  - Use your UNSW email address!
  - Amazon says it should only take 2-5 minutes (but don't rely on this!!)
- This should give you $100/year in AWS credits. **Be careful!!!**

Big Data Applications

- Finding similar items
- Graph data processing
- Data stream mining
- Recommender Systems

End of Chapter 1