USC Viterbi
School of Engineering
Information Sciences Institute

INF 553: Foundations and Applications of Data Mining

Rafael Ferreira da Silva
rafsilva@isi.edu

http://rafaelsilva.com

Thanks for source slides and material to:
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets
http://www.mmds.org

The Instructors…

Rafael Ferreira da Silva
Research Assistant Professor
Department of Computer Science
Computer Scientist
Information Sciences Institute

Ph.D. in Computer Science, INSA-Lyon (France), 2013
rafsilva@isi.edu – *http://rafaelsilva.com*

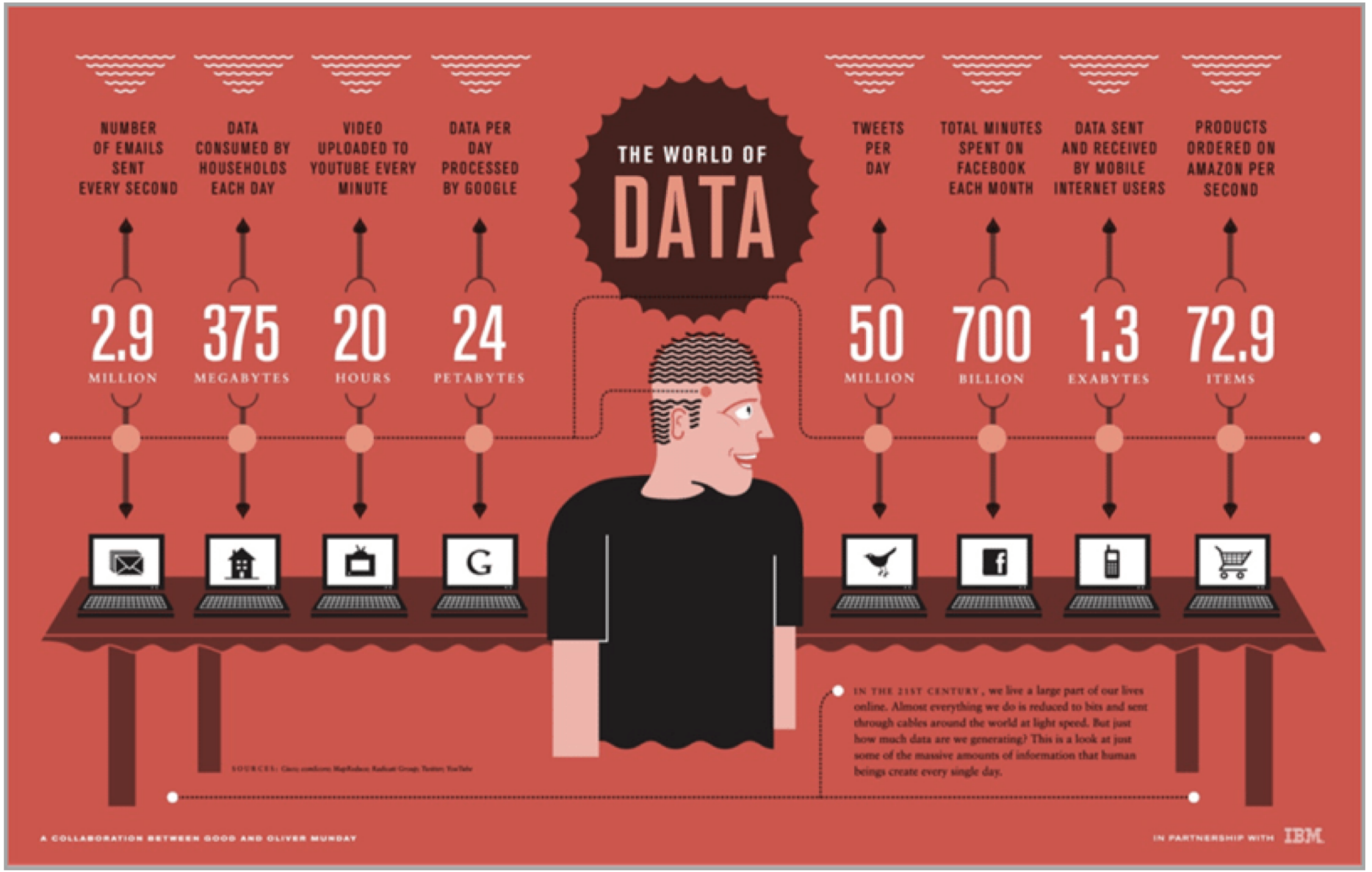
Anoop Kumar
Senior Computer Scientist
Information Sciences Institute

Ph.D. in Computer Science, Tufts University, 2010
anoopk@isi.edu – *http://usc-isi-i2.github.io/anoopkum*

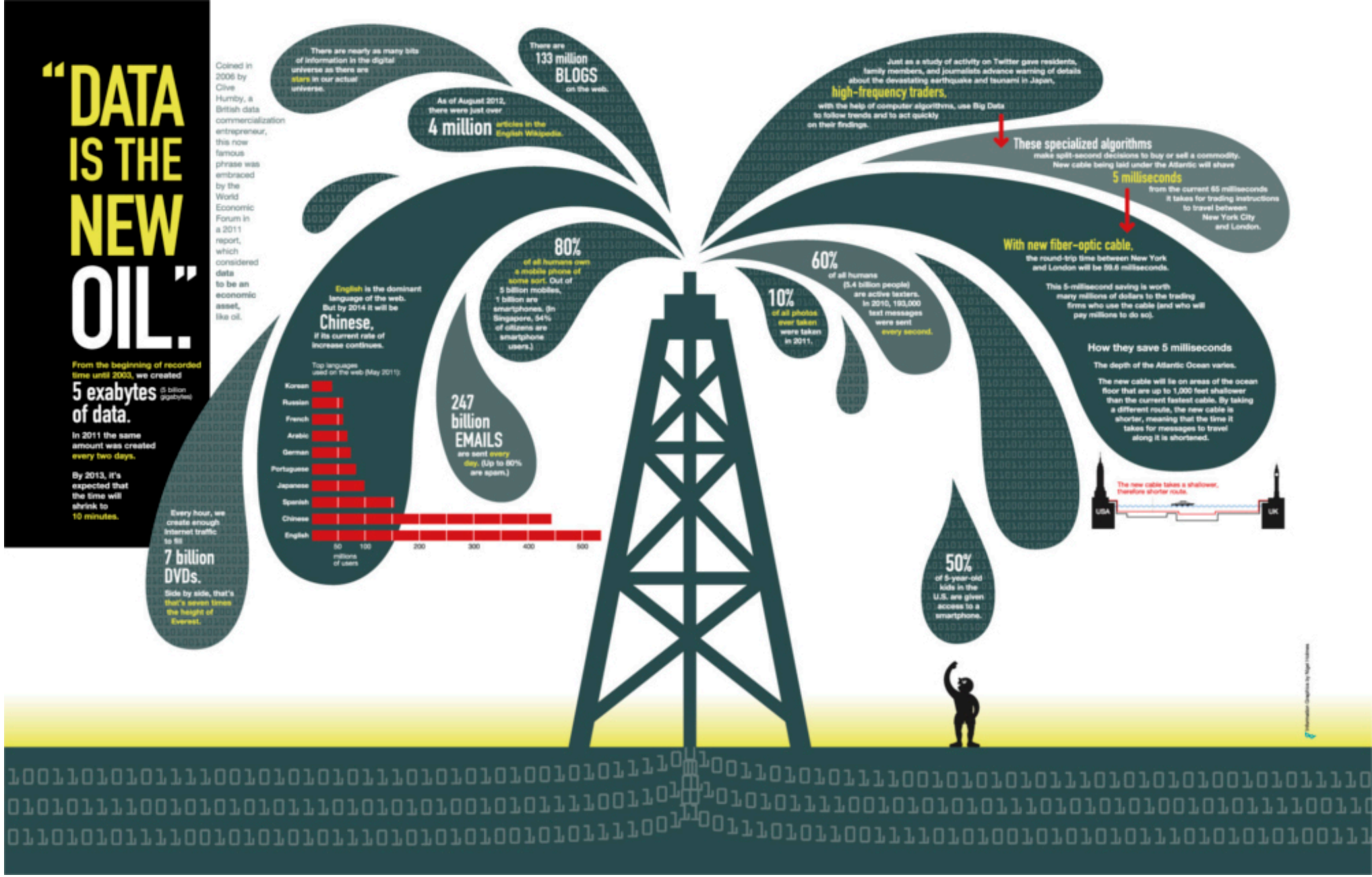
Why Data Mining?

https://www.jeffbullas.com/wp-content/uploads/2016/12/The-Basics-for-data-mining-visualization-and-infographics.png

Knowledge Discovery from Data



Data contains value and knowledge



Data Mining

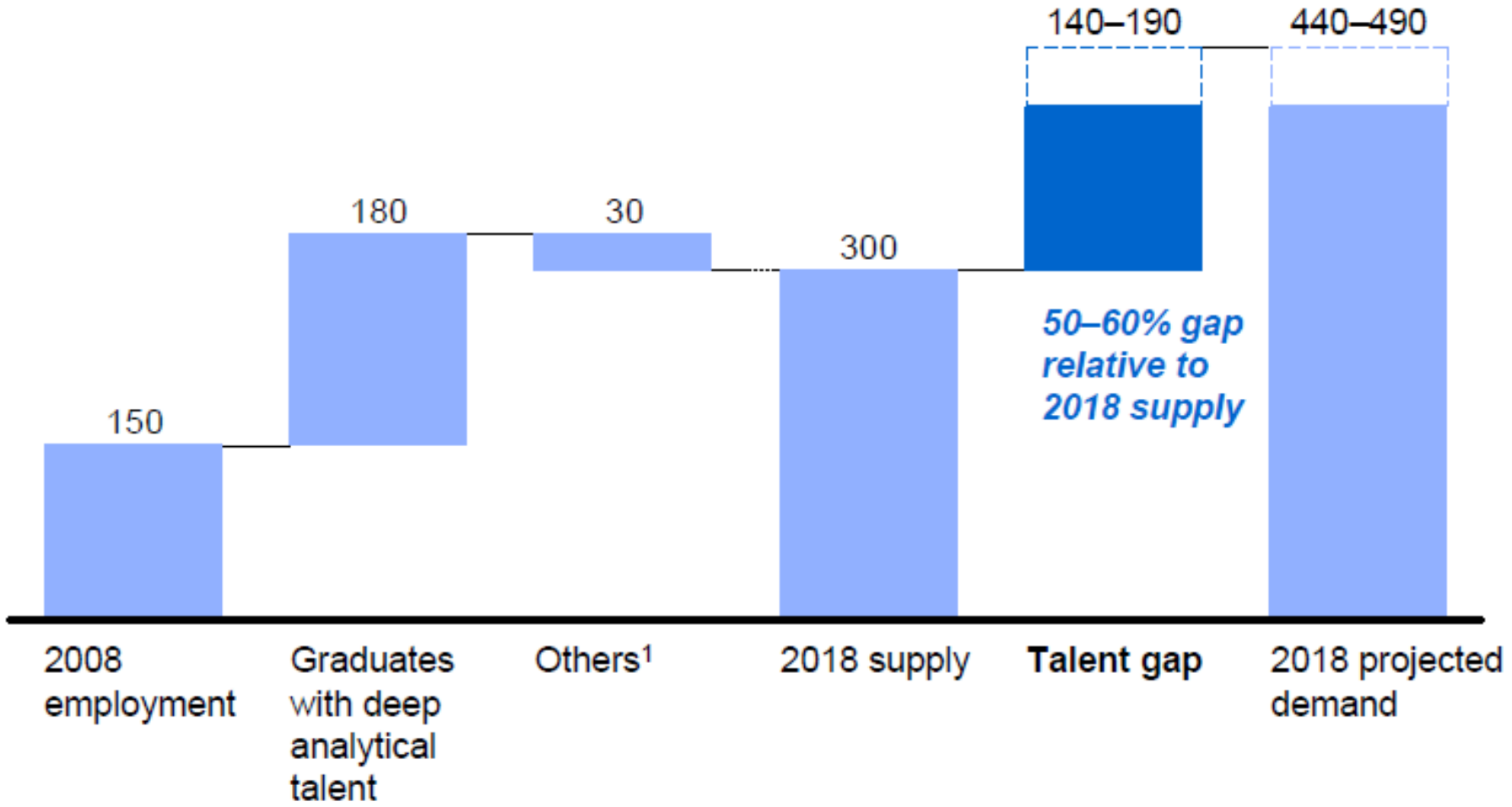
- But to extract the knowledge data needs to be
  - Stored
  - Managed
  - And **ANALYZED** ← this class

Data Mining ≈ Big Data ≈
Predictive Analytics ≈ Data Science

Good News: There is an urge for Data Mining

Demand for deep analytical talent in the United States could be 50 to 60 percent greater than its projected supply by 2018

Supply and demand of deep analytical talent by 2018
Thousand people



1 Other supply drivers include attrition (-), immigration (+), and reemploying previously unemployed deep analytical talent (+).

SOURCE: US Bureau of Labor Statistics; US Census; Dun & Bradstreet; company interviews; McKinsey Global Institute analysis

DATA Engineer

DataCamp
Learn Data Science By Doing

DATA Scientist

Develops, constructs, tests, and maintains architectures. Such as databases and large-scale processing systems.

Cleans, massages and organizes (big) data. Performs descriptive statistics and analysis to develop insights, build models and solve a business need.

Responsibilities...

ON DAILY BASIS

Develop, construct, test, and maintain architectures (such as databases and large-scale processing systems)

Ensure architecture will support the requirements of the business

Discover opportunities for **data acquisition**

Develop data set processes for data modeling, mining and production

Employ a variety of languages and tools (e.g. scripting languages) to **marry systems together**

Recommend ways to **improve data** reliability, efficiency and quality

Conduct research to **answer industry and business questions**

Leverage large volumes of data from internal and external sources to answer that business

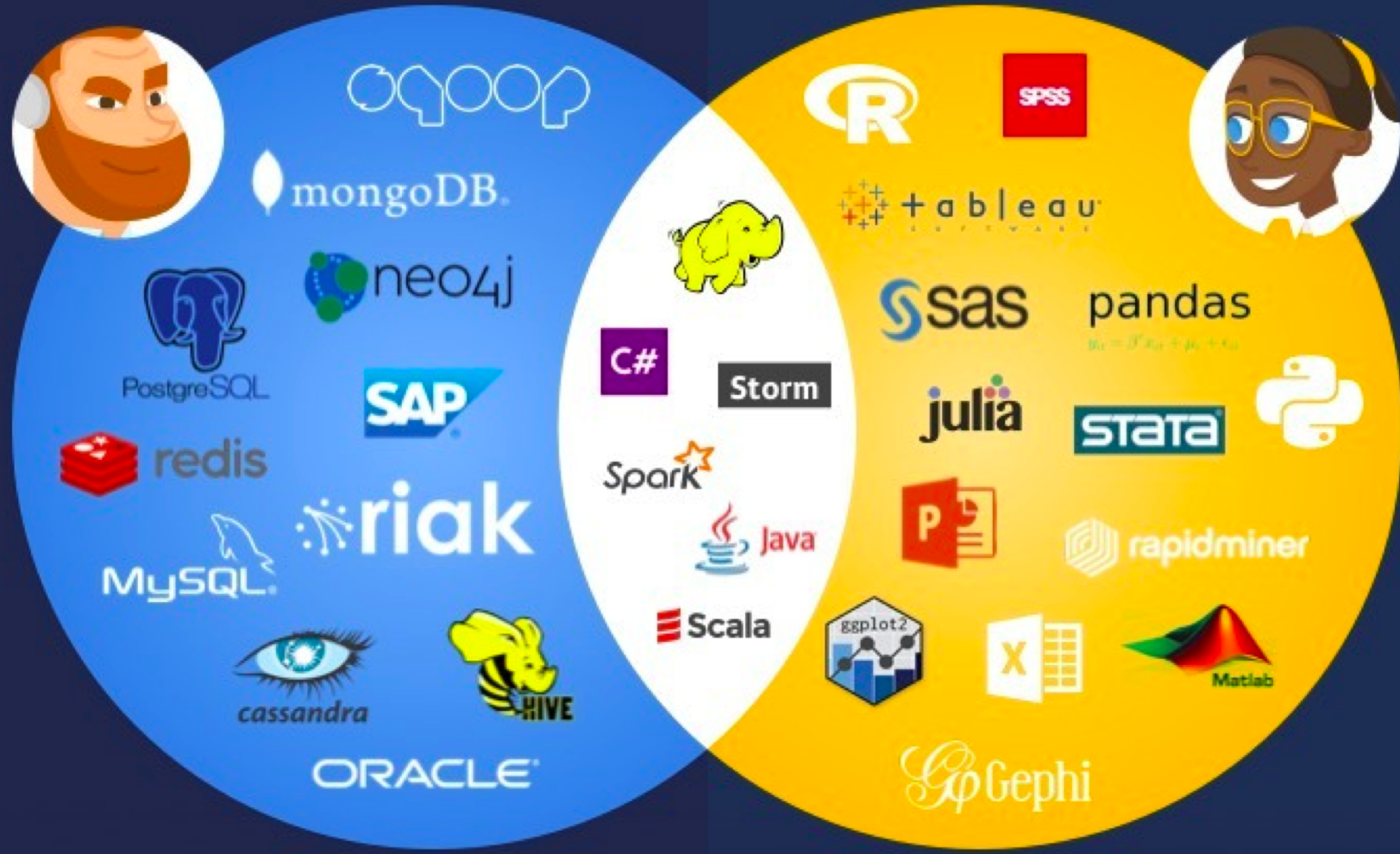
Employ sophisticated analytics programs, machine learning and statistical methods to **prepare data for use in predictive and prescriptive modeling**

Explore and examine data to **find hidden patterns**

Automate work through the use of predictive and prescriptive analytics

Tell stories to key stakeholders based on their analysis

Languages, Tools & Software

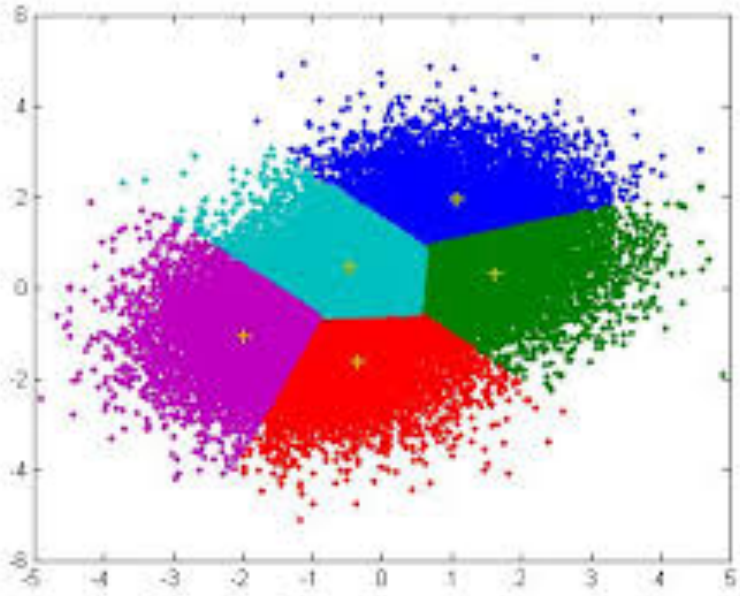


What is Data Mining?

- **Given lots of data**
- **Discover patterns and models that are**
  - **Valid:** hold on new data with **some** certainty
  - **Useful:** should be **possible** to **act** on the item
  - **Unexpected:** **non-obvious** to the system
  - **Understandable:** **humans** should be able to interpret the pattern

Data Mining Tasks

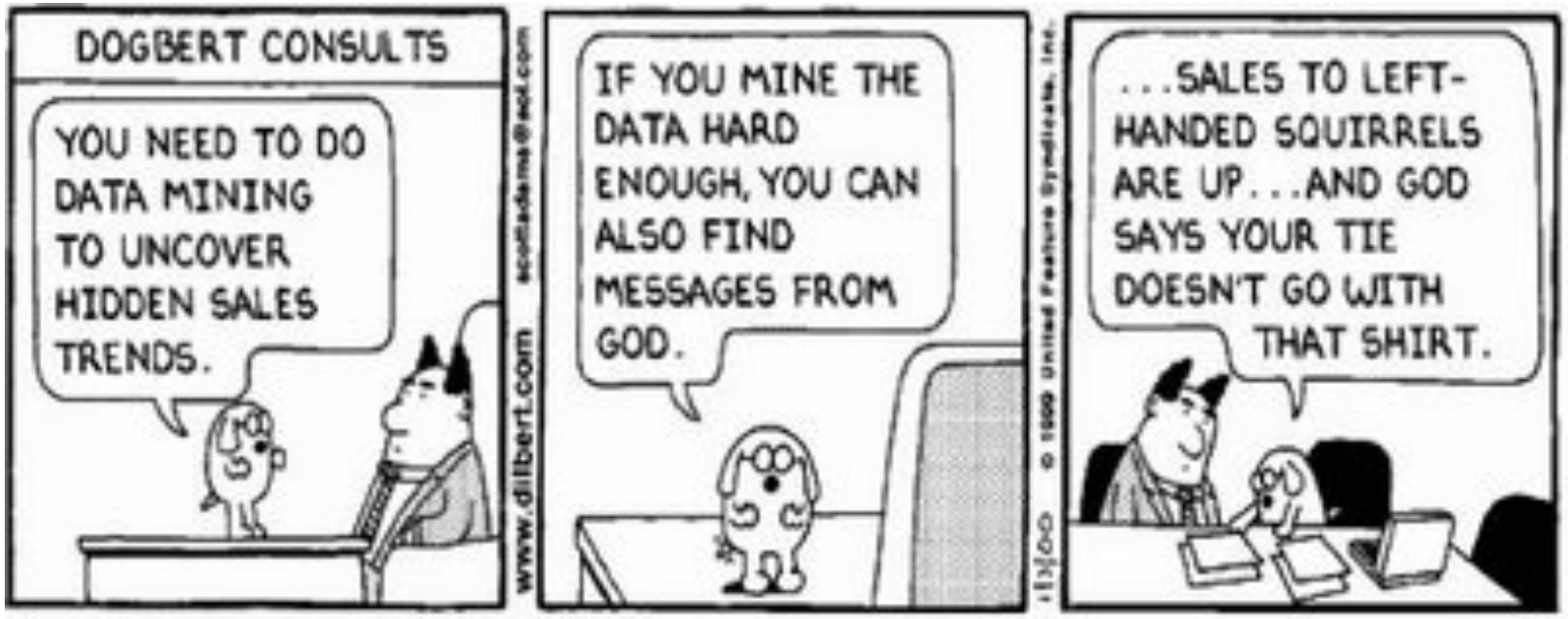
- **Descriptive methods**
  - Find human-interpretable patterns that describe the data
    - **Example:** Clustering



- **Predictive methods**
  - Use some variables to predict unknown or future values of other variables
    - **Example:** Recommender systems

Meaningfulness of Analytic Answers

- **A risk with "Data mining" is that a scientist can "discover" patterns that are meaningless**
- **Bonferroni's principle**:
  - If you look in more places for interesting patterns than your amount of data will support, you are bound to find crap



Meaningfulness of Analytic Answers

Example:

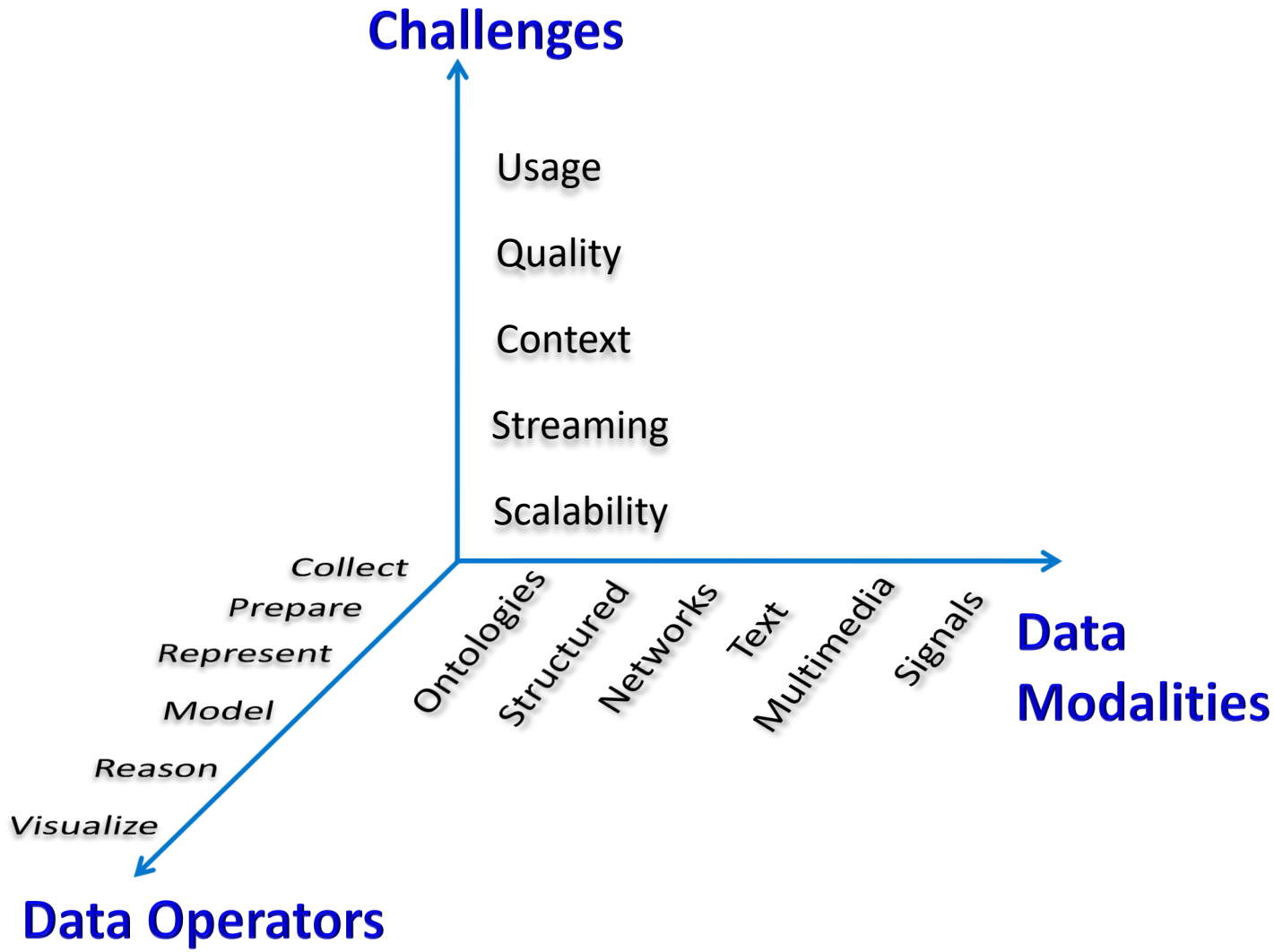
- We want to find (unrelated) people who **at least twice have stayed at the same hotel on the same day**
  - 10^9 people being tracked – 1 billion
  - 1,000 days ~ 3 years
  - Each person stays in a hotel 1% of time (1 day out of 100)
  - Hotels hold 100 people (so 10^5 hotels)
    - enough to hold the 1% of a billion people who visit a hotel on any given day
  - **If everyone behaves randomly will the data mining detect anything suspicious?**

Meaningfulness of Analytic Answers (Cont'd)

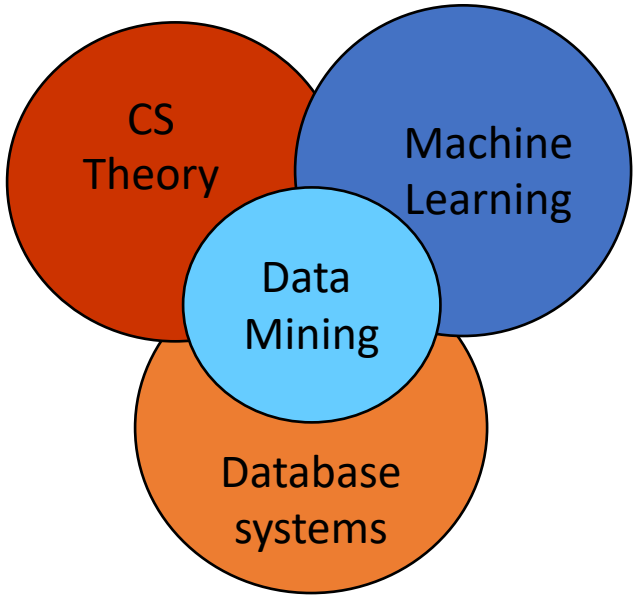
- 10^9 people, 1,000 days, 1% hotel stay, 10^5 hotels
  - The probability of any two people both deciding to visit a hotel on any given day is 0.0001 (i.e., 1%*1%)
  - The chance that they will visit the same hotel for one day is $0.0001 / 10^5 = 10^{-9}$; for two given days = 10^{-18}

  - The number of pairs of people is $C(10^9, 2) = 5 \times 10^{17}$
  - The number of pairs of days is $C(10^3, 2) = 5 \times 10^5$
  - **Expected number of "suspicious" pairs of people:**
    - $5 \times 10^{17} \times 5 \times 10^5 \times 10^{-18} = 250{,}000$!
    - … too many combinations to check – we need to have some additional evidence to find "suspicious" pairs of people in some more efficient way

What matters when dealing with data?



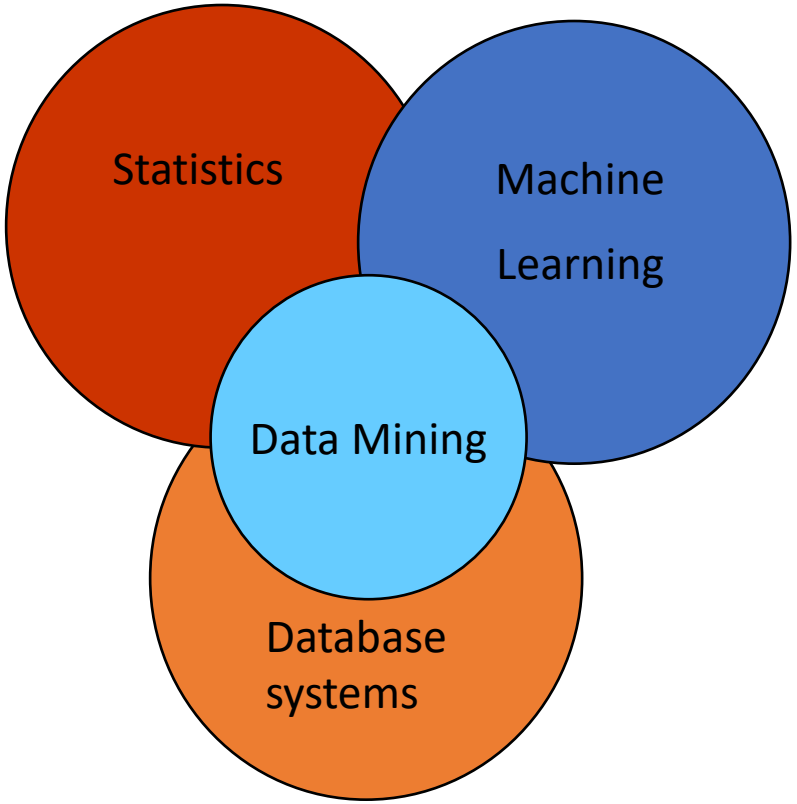
Data Mining: Cultures



- **Data mining overlaps with:**
  - **Databases:** Large-scale data, simple(r) queries
  - **Machine learning:** Small data, Complex models
  - **CS Theory:** (Randomized) Algorithms
- **Different cultures:**
  - To a DB person, data mining is an extreme form of **analytic processing** – queries that examine **large amounts of data**
    - Result is the query answer
  - To a ML person, data-mining is the **inference of models**
    - Result is the parameters of the model
- **In this class we will do both!**

This Course

- This course overlaps with machine learning, statistics, artificial intelligence, databases but more stress on
  - **Scalability** (big data)
  - **Algorithms**
  - **Computing architectures**
  - Automation for handling **large data**



What will we learn?

- **We will learn to mine different types of data**
  - Data is high dimensional
  - Data is a graph
  - Data is infinite/never-ending
  - Data is labeled
- **We will learn to use different models of computation**
  - MapReduce
  - Streams and online algorithms
  - Single machine in-memory

What will we learn?

- **We will learn to solve real-world problems**
  - Recommender systems
  - Market Basket Analysis
  - Spam detection
  - Duplicate document detection
- **We will learn various "tools"**
  - Linear algebra (Rec. Sys., Communities)
  - Optimization (stochastic gradient descent)
  - Dynamic programming (frequent itemsets)
  - Hashing (LSH, Bloom filters)

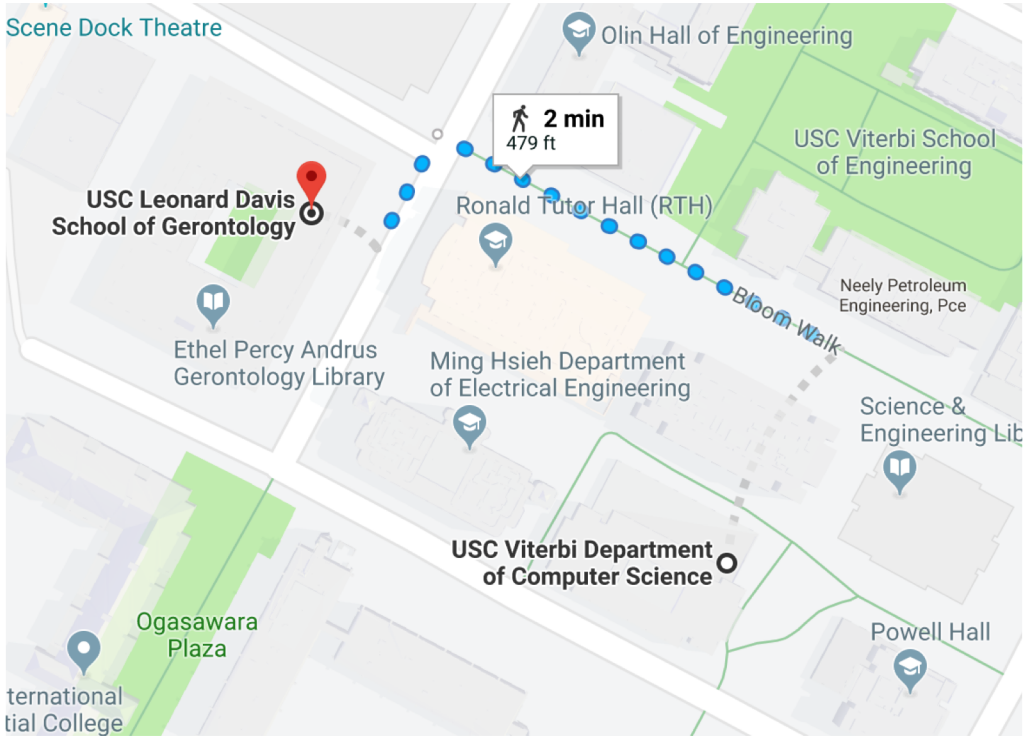
How It All Fits Together

High dim. data	Graph data	Infinite data	Machine learning	Apps
Locality sensitive hashing	PageRank, SimRank	Filtering data streams	SVM	Recommender systems
Clustering	Community Detection	Web advertising	Decision Trees	Association Rules
Dimensionality reduction	Spam Detection	Queries on streams	Perceptron, kNN	Duplicate document detection

Fall 2018 INF 553 Course Staff

- **3 CPs**
  - Anirudh Kashi
  - Prasad Bhagwat
  - Yuanbin Cheng
- **2 Graders**
  - Rasika Guru
  - Roshani Mangalore

- **Office Hours**
  - Instructors: Fridays after class (GRE 207)



  - CPs and Graders
    - See Blackboard for hours and locations

Course Logistics

- **Course website:**

  https://blackboard.usc.edu/

  - Lecture slides
  - Additional Material (e.g., examples) may be posted into my website: http://rafaelsilva.com
  - Homework
  - Readings
    - **Mining of Massive Datasets**
      - J. Leskovec, A. Rajaraman and J. Ullman
      - Free online: http://www.mmds.org or on Blackboard
    - Other relevant papers

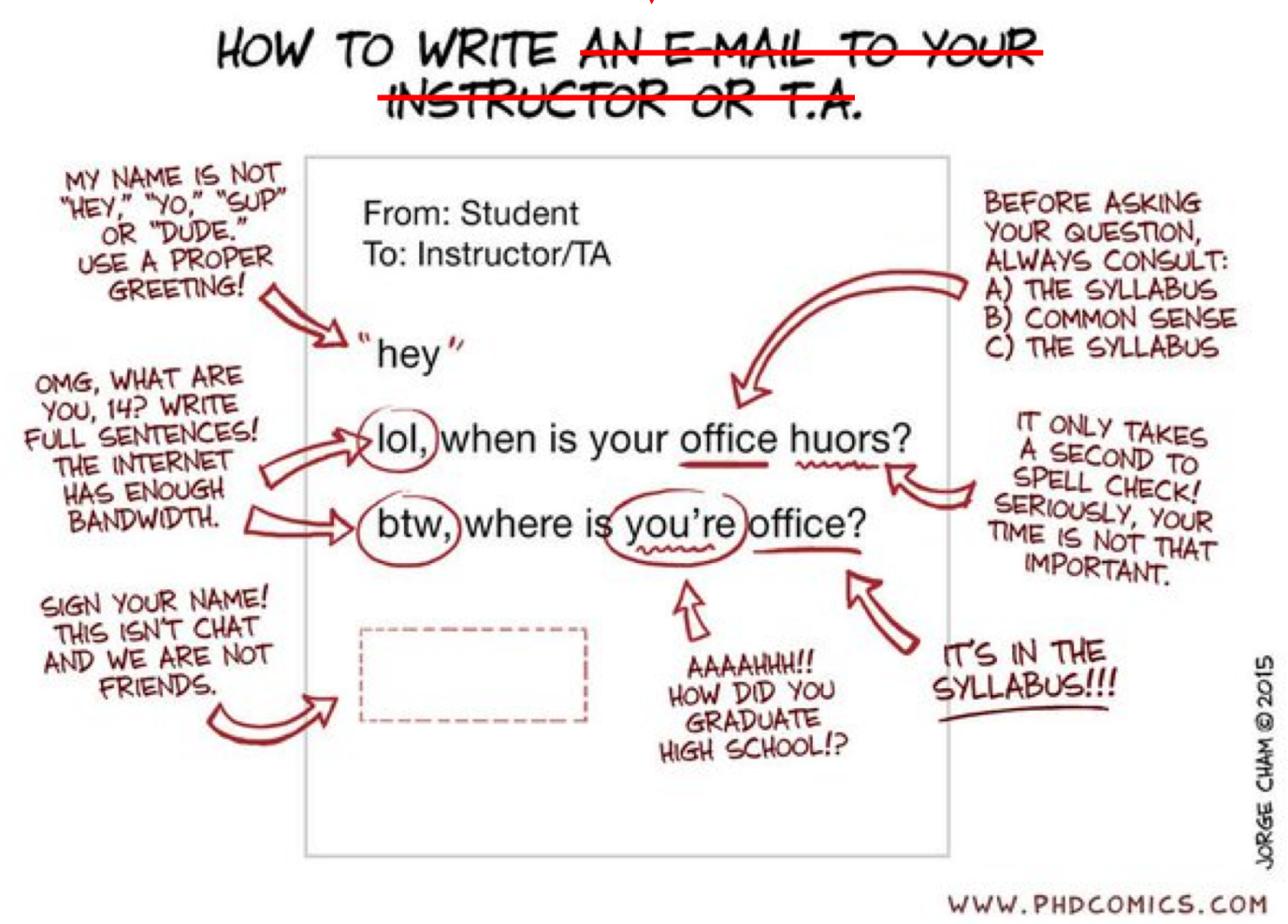
Logistics: Communication

- **Discussion board on Blackboard:**
  - https://blackboard.usc.edu/
    - Use the **discussion board** on Blackboard for all questions and public communication with the course staff

Forum	Description
Assignments	Discussions of homework assignments
General	General discussions and announcements
Project	This thread is devoted to discussion related to the course project

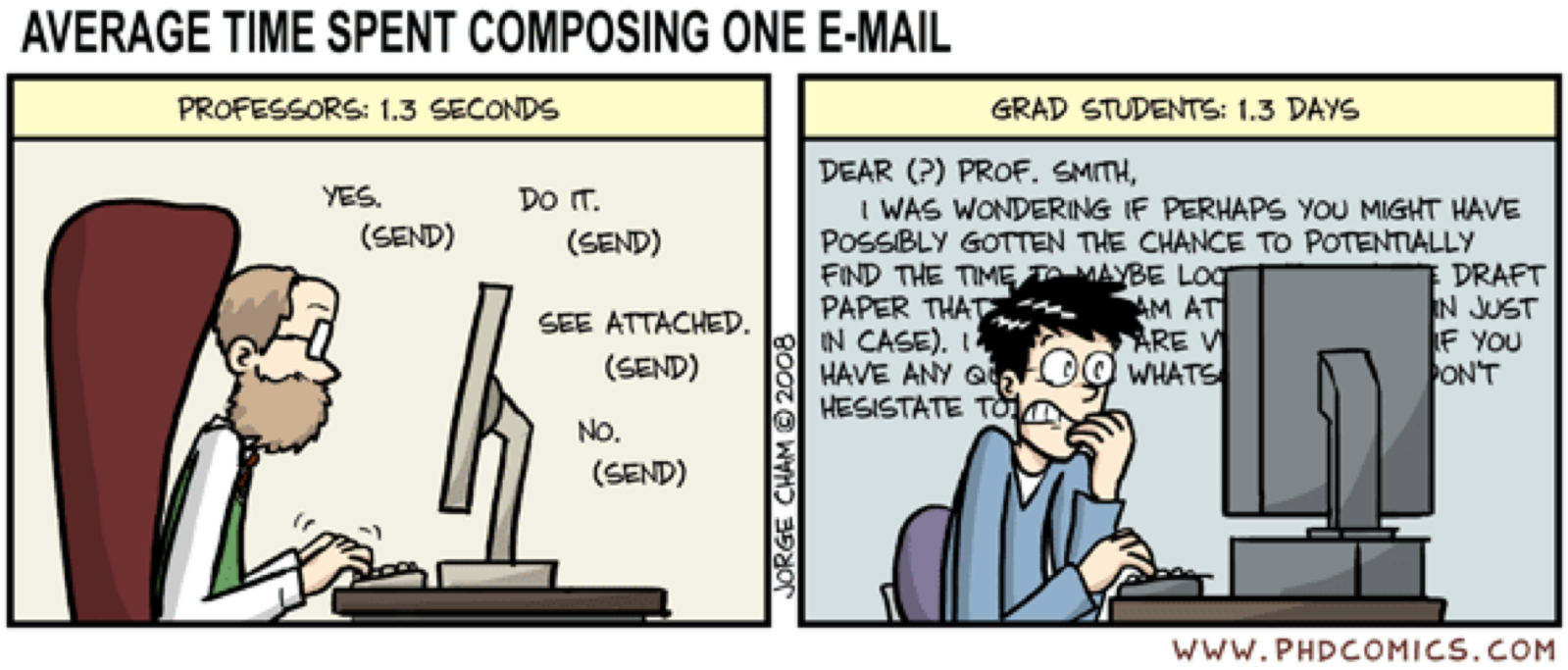
subscribe

a post on the discussion board



Logistics: Communication

- We will post course announcements to Blackboard (make sure you check it regularly)
- **Emails:**
  - Do not use emails unless it's personal!



Work for the Course

- **Four homework: 40%**
  - Theoretical and programming questions
    - HW1: MapReduce (with Spark)
    - HW2: Recommendation Systems
    - HW3: Finding Frequent Itemsets
    - HW4: Clustering
    - HW5: Streaming
      - *Optional for this section: may replace the lowest scored homework*

- **Assignments take lots of time. Start early!!**
- **Please work on your own code!**

Work for the Course

- Homework policy:
  - No regrading
  - One week late penalty – 20%
  - 0 points after one week
  - Free five-day extensions
    - You can use these five days on homework however you want
    - No more extension days will be given for any reason

Work for the Course

- **Project Proposal: 10%**
  - **Living document**
  - **Students are expected to send updates to the project proposal weekly**
    - A simple paragraph explaining the progress made on the project, and changes/improvements to the proposal

- **Project: 50%**
  - Real and large dataset
  - Team work
  - Several algorithms/techniques should be used to **unveil novel knowledge**
  - Deliverables: project **source code** and a **presentation** in the last day of class
  - *Detailed guidelines will be provided soon*

Prerequisites

- **Algorithms**
  - Dynamic programming, basic data structures
- **Basic probability**
  - Moments, typical distributions, MLE, …
- **Programming**
  - Your choice, but Python will be very useful
  - Some of the homework will require you to use Scala only

- **We provide some background, but the class will be fast paced**

Course Grade

Grades will range from A through F. The following is the breakdown for grading:

[93 – 100] = A	[73 – 77) = C
[90 – 93) = A-	[70 –73) = C-
[87 – 90) = B+	[67 – 70) = D+
[83 – 87) = B	[63 – 67) = D
[80 – 83) = B-	[60 – 63) = D-
[77 – 80) =C+	Below 60 is an F

What's after the class

- Directed Research
- Course producer or grader positions
- Paid student worker positions

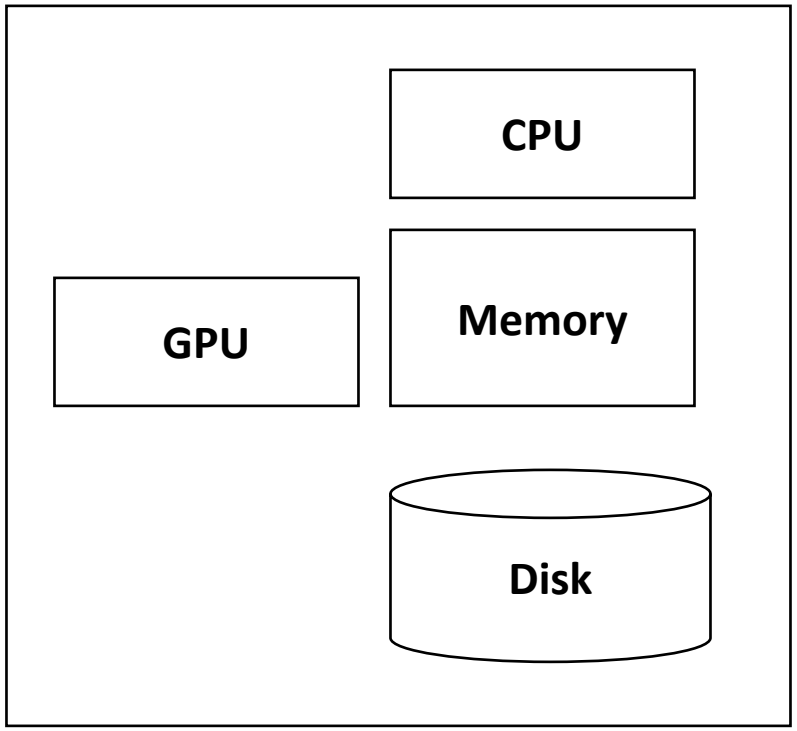
USC Viterbi
School of Engineering
Information Sciences Institute

Introduction to MapReduce

Rafael Ferreira da Silva
rafsilva@isi.edu

http://rafaelsilva.com

Single-node architecture



Machine Learning, Statistics

"Classical" Data Mining

MapReduce

- Much of the course will be devoted to **large-scale computing for data mining**

- Challenges
  - How to distribute computation?
  - Distributed/parallel programming is hard

- **MapReduce** addresses all of the above (to some extent)
  - Google's computational/data manipulation model
  - Elegant way to work with Big Data

Motivation: Google Example

- 20+ billion web pages x 20KB = 400+ TB
- 1 computer reads 30-35 MB/sec from disk
  - ~4 months to read the web
  - ~1,000 hard drives to store the web

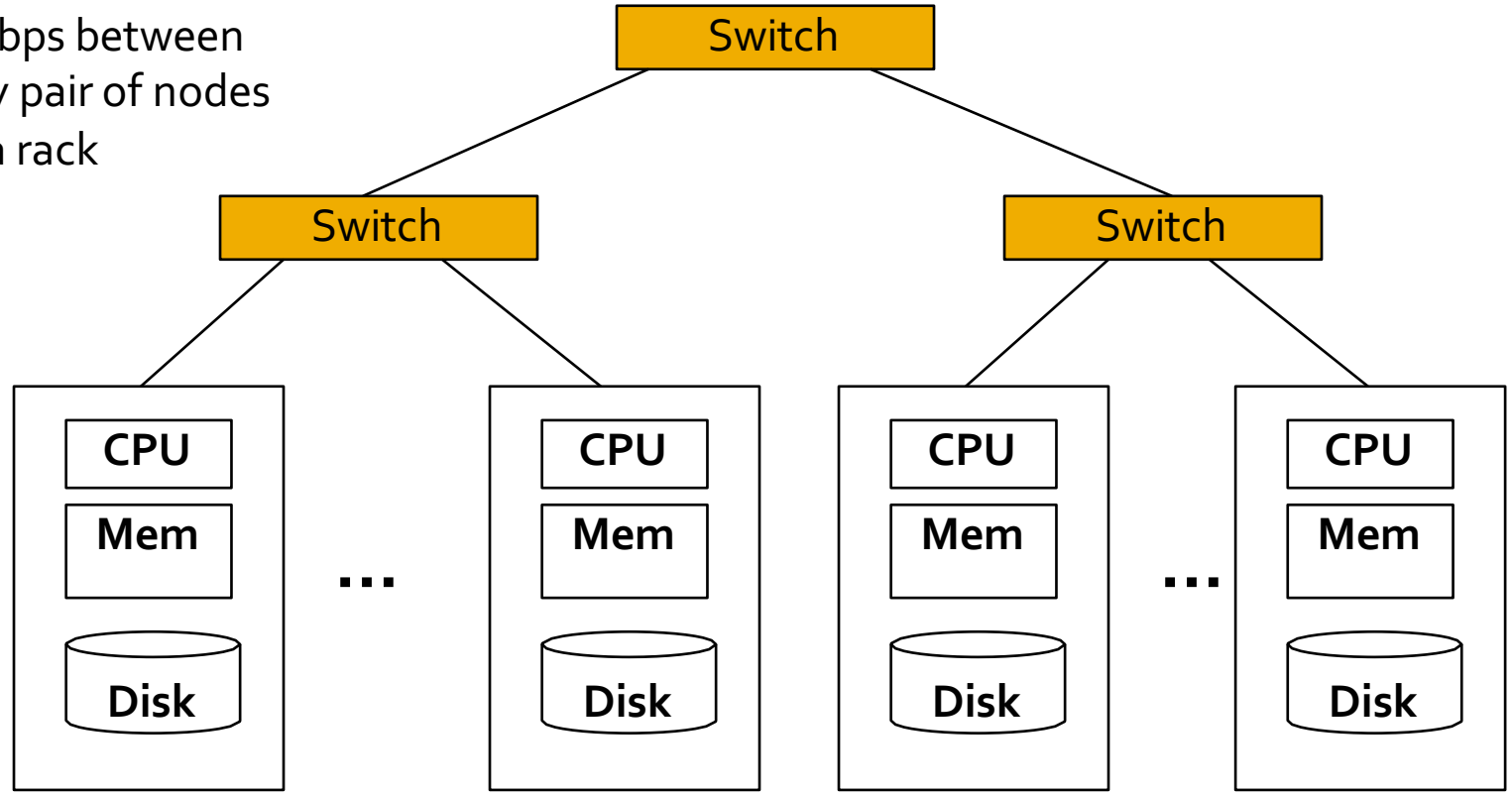
- Takes even more to do something useful with the data!

- Recently standard architecture for such problems emerged:
  - Cluster of commodity (Linux) nodes
  - Commodity network (ethernet) to connect them

Cluster Architecture

2-10 Gbps backbone between racks

1 Gbps between any pair of nodes in a rack



Each rack contains 16-64 nodes

In 2011 it was guestimated that Google had 1M machines, http://bit.ly/Shh0RO

Large-scale Computing

- Large-scale computing for data mining problems on commodity hardware

- Challenges:
  - How to **distribute** computation
  - How to make it easy to **write distributed programs**
  - Machines **fail**
    - One server may stay up 3 years (1,000 days)
    - If you have 1,000 servers, expect to lose 1/day
    - With 1M machines 1,000 machines fail every day!

Idea and Solution

- Issue: Copying data over a network takes time
- Idea:
  - **Bring computation to data**
  - Store files **multiple times** for reliability

- MapReduce addresses these problems
  - Storage Infrastructure – File system
    - Google: GFS; Apache Hadoop: HDFS
  - Programming Model
    - MapReduce

Storage Infrastructure

- Problem
  - If nodes fail, how to store data persistently?

- Answer:
  - Distributed File System:
    - Provides **global file namespace**

- Typical **usage pattern**
  - Huge files (100s of Gigabytes to Terabytes)
  - Data are rarely updated in place
  - Reads and appends are common

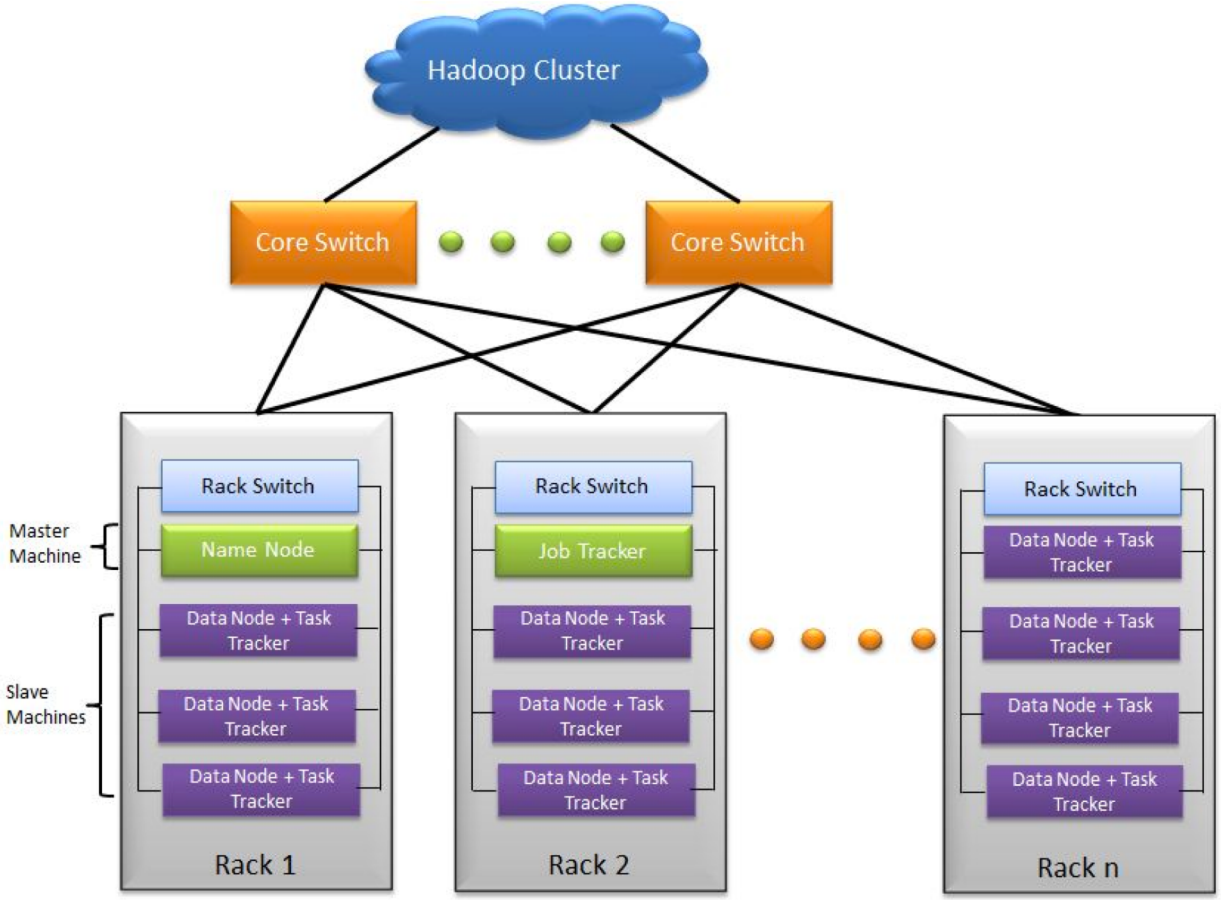
Distributed File System

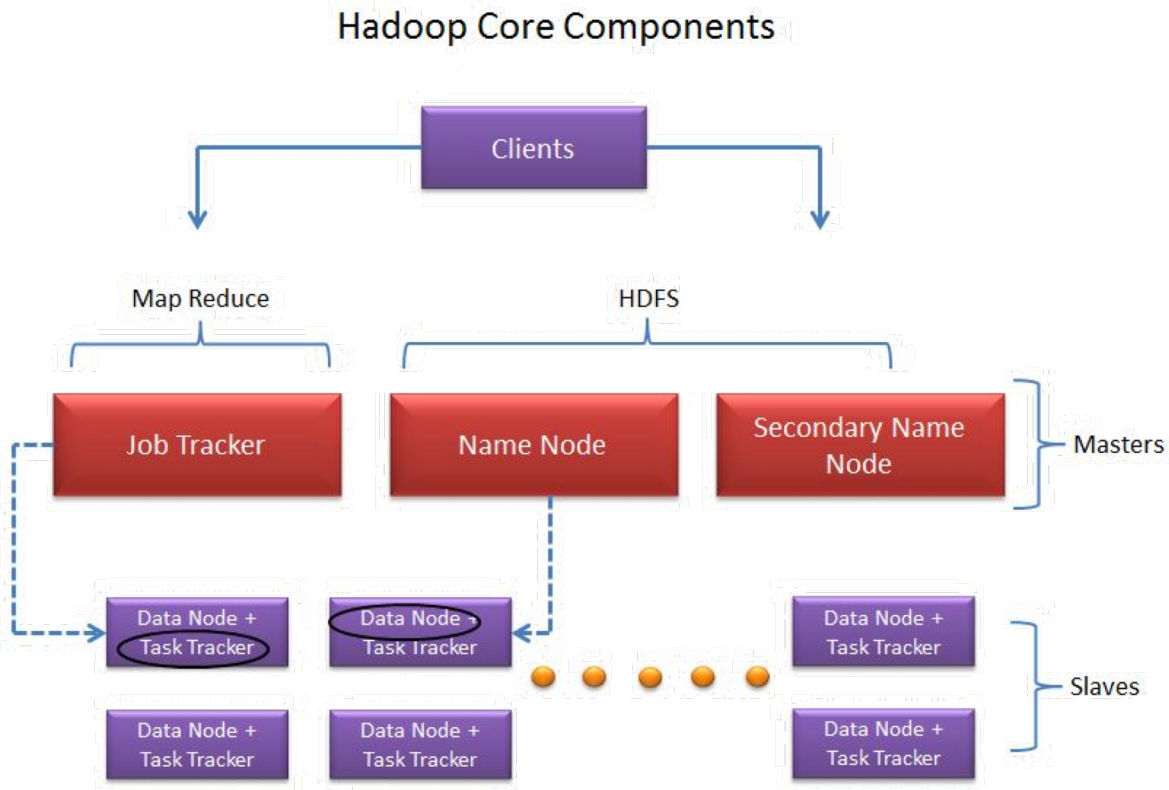
- Chunk Servers
  - File is split into **contiguous chunks**
  - Typically each chunk (file block) is 16-64MB
  - Each **chunk replicated** (usually 2x or 3x)
  - Try to keep replicas in different racks

- Master node
  - a.k.a. **Name Node** in Hadoop's HDFS
  - Stores **metadata** about where files are stored
  - Might be replicated

- **Client library** for file access
  - Talks to master to find chunk servers
  - Connects directly to chunk servers to access data

Hadoop Cluster

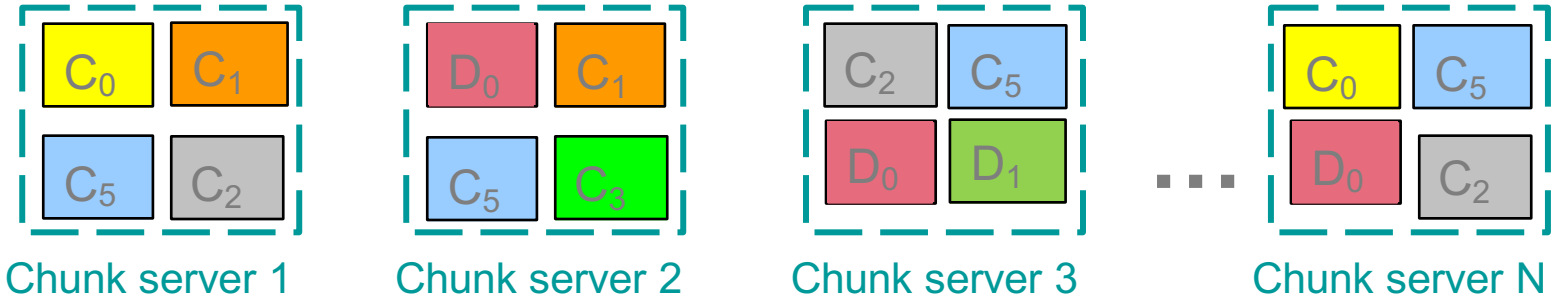




http://saphanatutorial.com/hadoop-cluster-architecture-and-core-components/

Distributed File System

- Reliable distributed file system
- Data kept in "**chunks**" spread across machines
- Each chunk **replicated** on different machines
  - Seamless recovery from disk or machine failure



Bring computation directly to the data!

Chunk servers also serve as compute servers

Programming Model: MapReduce

- **Warm-up task**
  - We have a huge text document
  - Count the number of times each distinct word appears in the file

- **Sample application**
  - Analyze web server logs to find popular URLs

MapReduce Overview

- 3 steps of MapReduce
  - Sequentially read a lot of data

- **Map**
  - Extract something you care about

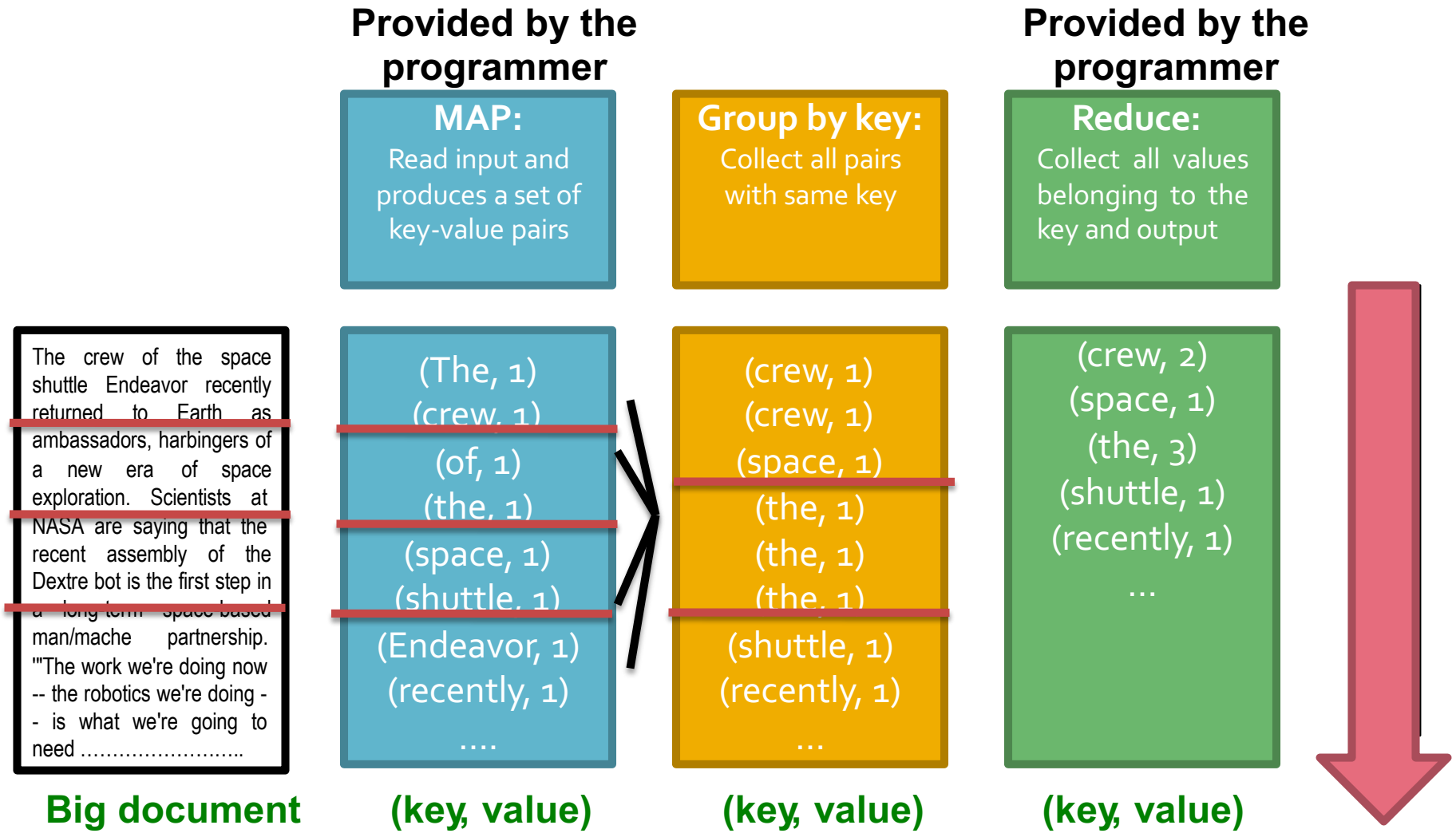
- **Group by key**
  - Sort and shuffle

- **Reduce**
  - Aggregate, summarize, filter or transform

- Output the result

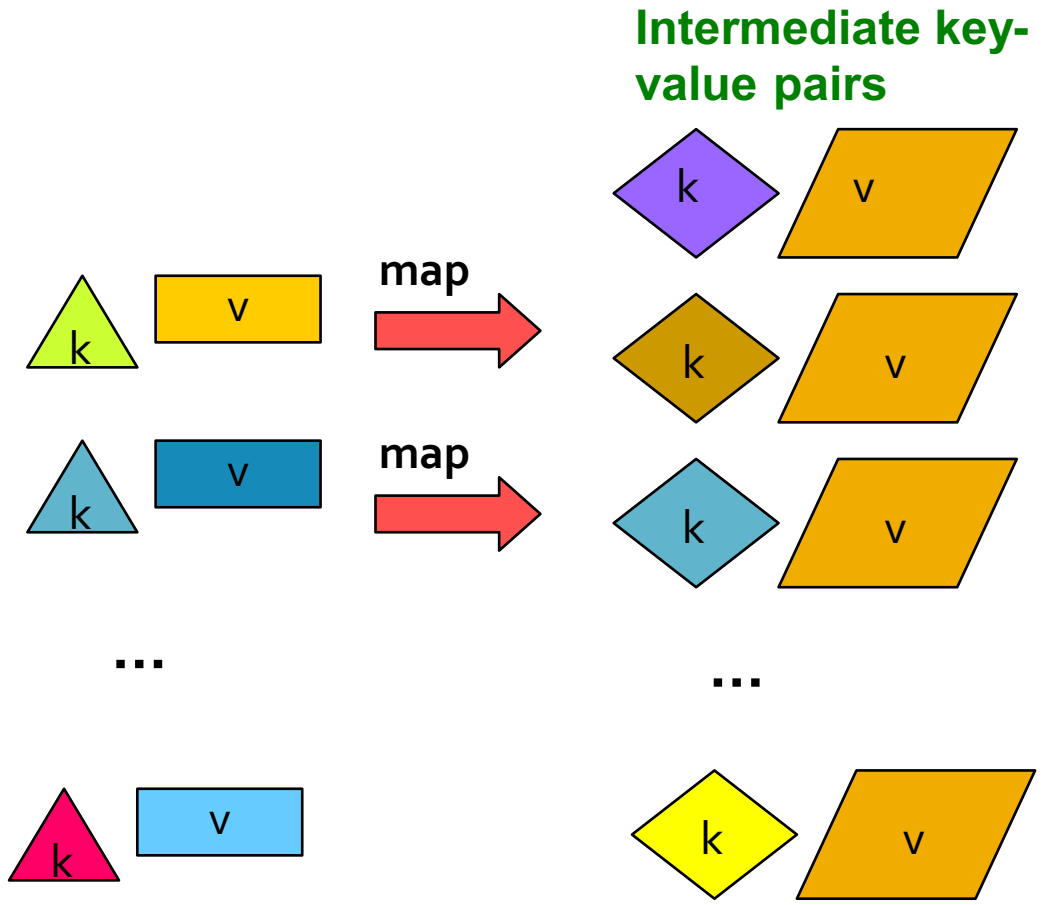
Outline stays the same, **Map** and **Reduce** change to fit the problem

MapReduce: Word Counting

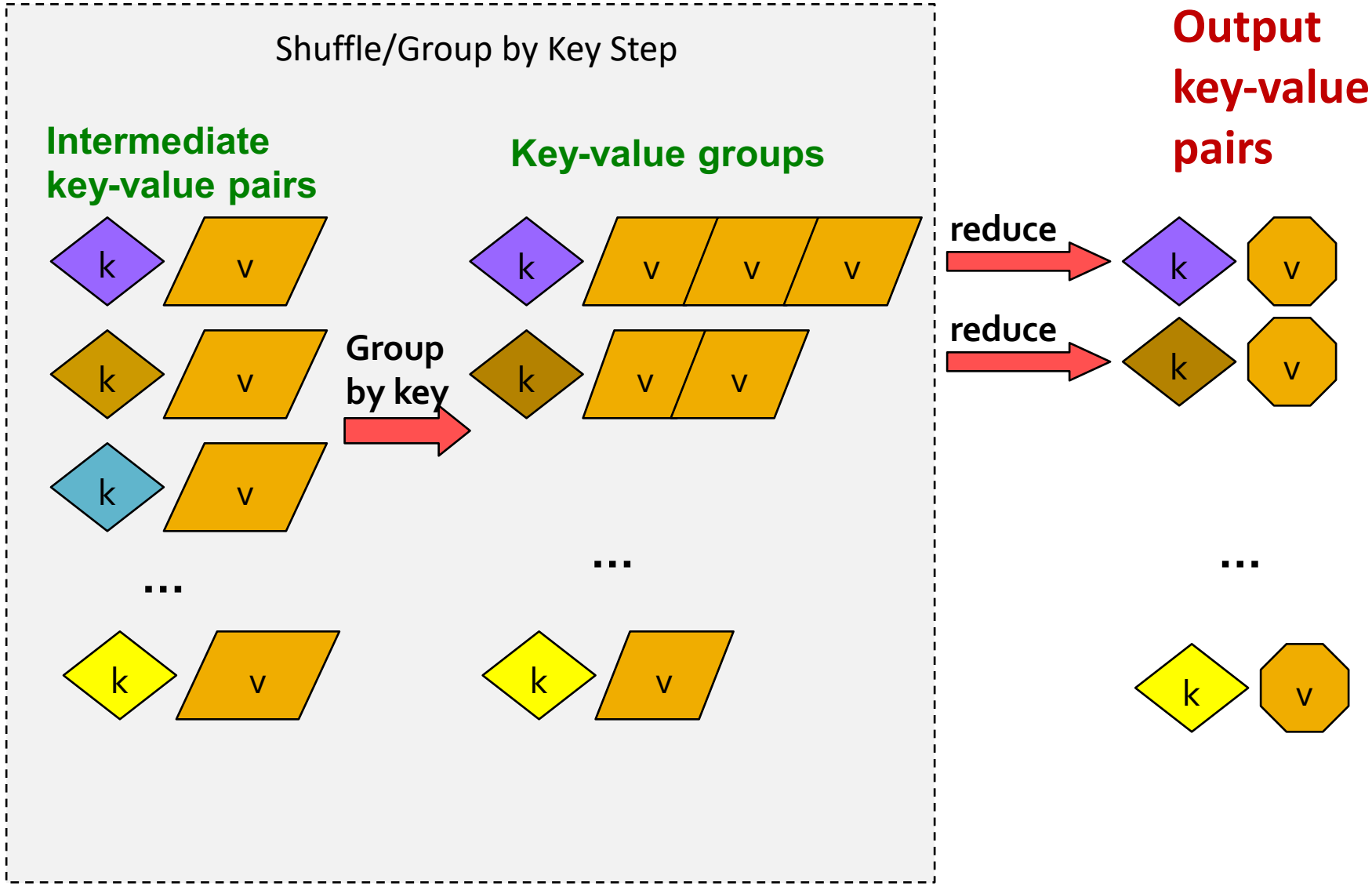


MapReduce: The Map Step

- **Input**
  - Key-value pairs
- **Output**
  - (intermediate) key-value pairs



MapReduce: The Reduce Step



More Specifically

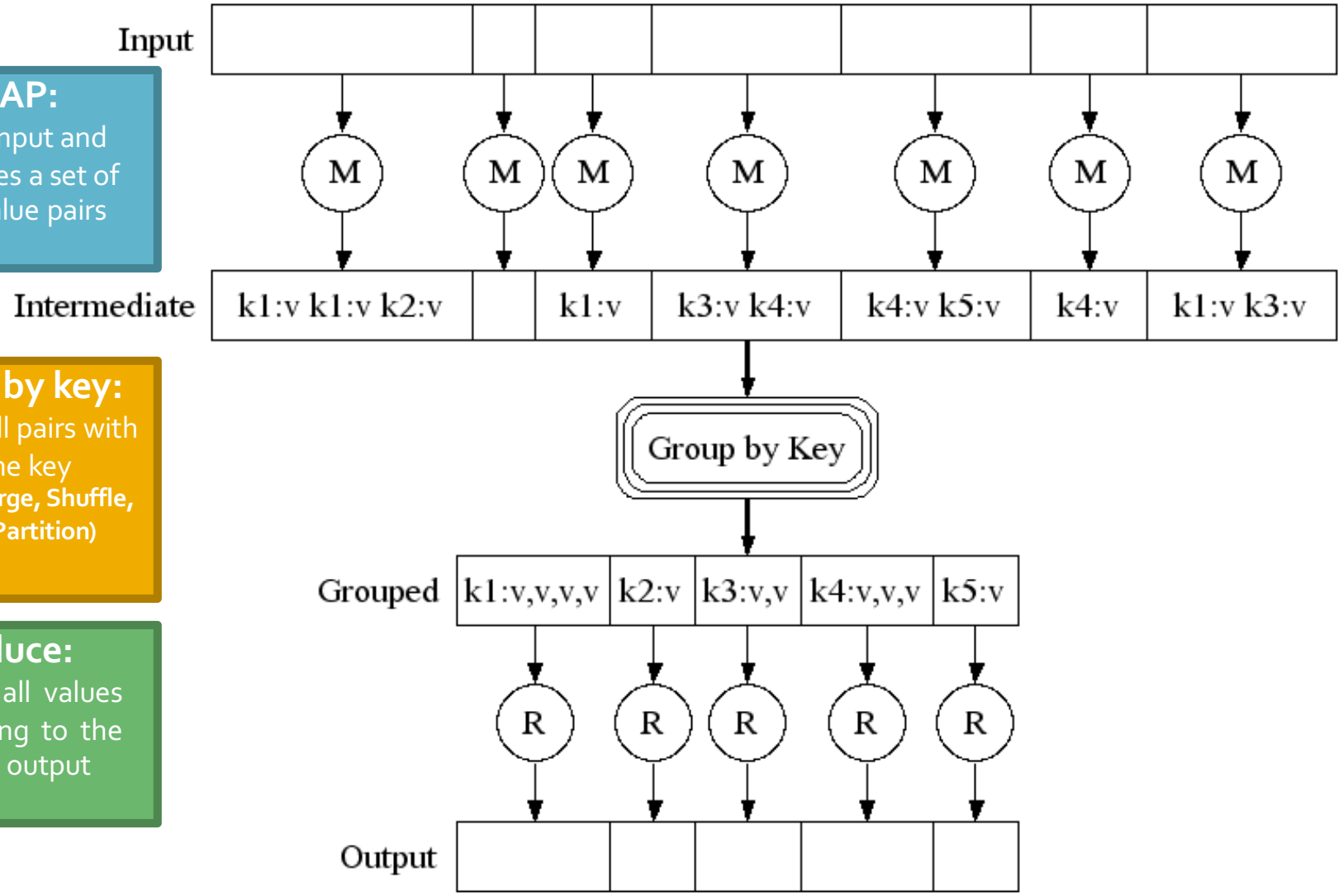
- Input: a set of key-value pairs
- Programmer specifies two methods
  - **Map(k, v) → <k', v'>***
    - **Takes a key-value pair and outputs a set of key-value pairs**
      - E.g., key is the filename, value is a single line in the file
    - There is **one Map call for every (k,v)** input pair
  - **Reduce(k', <v'>*) → <k', v''>***
    - **Takes a key-value list as input, outputs key-value pairs**
    - All values v' with same key k' are **reduced together** and processed in v' order
    - There is **one Reduce function call per unique key k'**

MapReduce: A Diagram

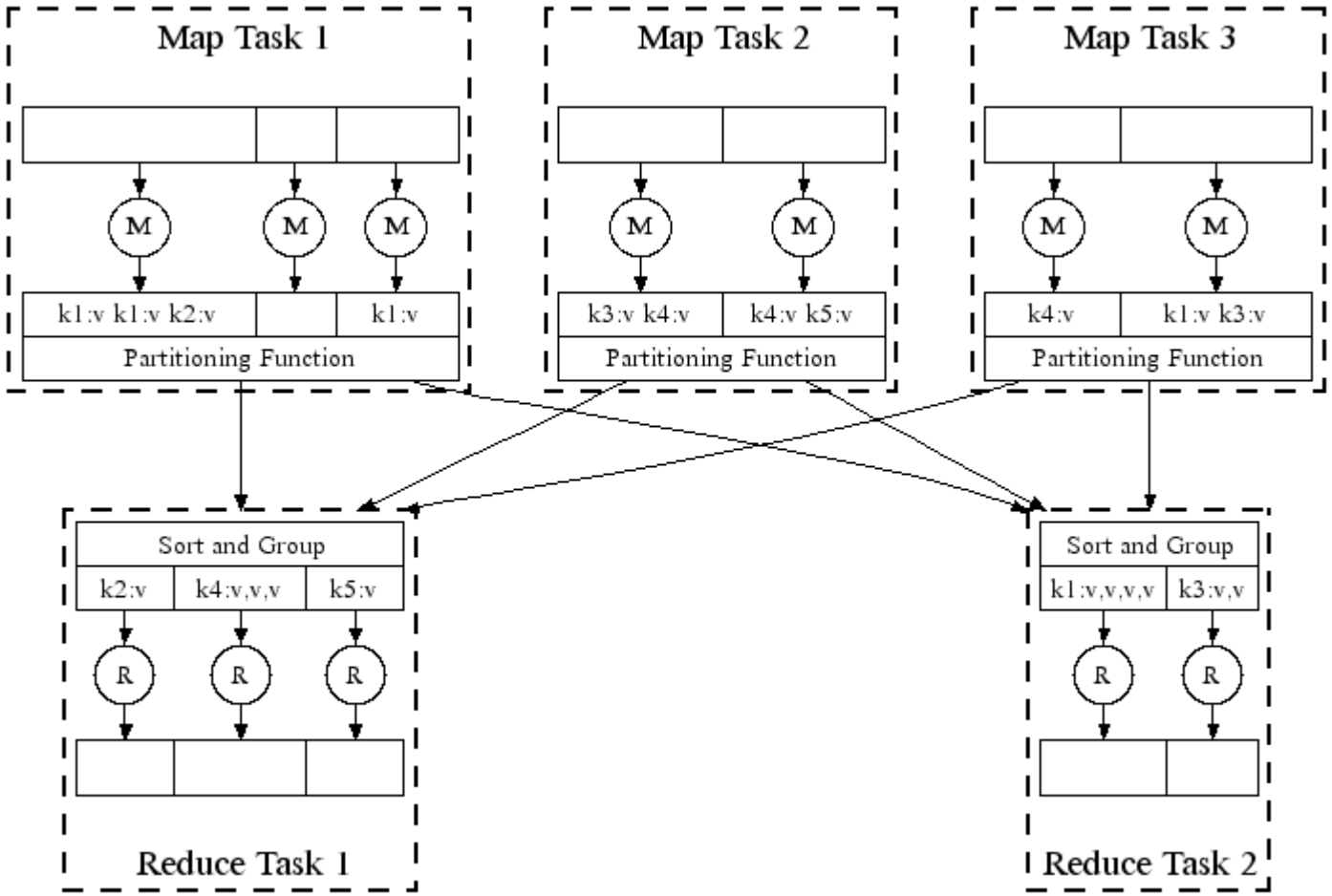
MAP:
Read input and produces a set of key-value pairs

Group by key:
Collect all pairs with same key
(Hash merge, Shuffle, Sort, Partition)

Reduce:
Collect all values belonging to the key and output



MapReduce: In Parallel



All phases are distributed with many tasks doing the work

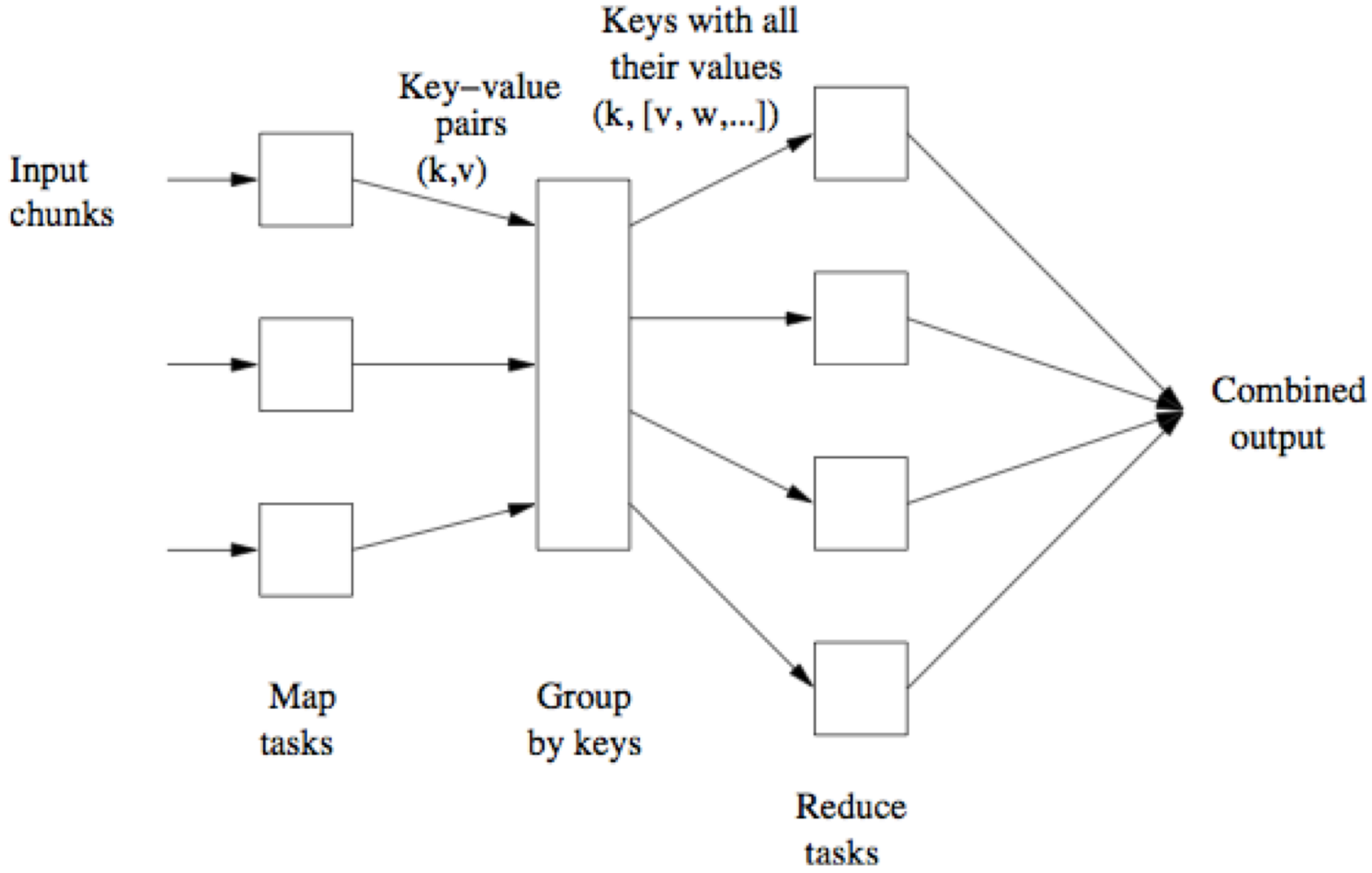
MapReduce: Summary

- **Map** tasks
  - Some number of Map tasks each are given one or more chunks from a distributed file system
  - **Map code written by the user**
  - Processes chunks and produces sequence of key-value pairs

- Master controller: **Group by Key/Shuffle**
  - Collects key-value pairs from each Map task
  - Divides keys among all Reduce tasks
  - All key-value pairs with same key go to same Reduce task

- **Reduce** task
  - Works on one key at a time
  - **Reduce code written by user**
    - Combines all values associated with that key in some way
  - Produces output key-value pairs

MapReduce: Summary



MapReduce

- Examples

```
$ docker run -it --rm -p 8888:8888 –p 4040:4040 jupyter/pyspark-notebook
```