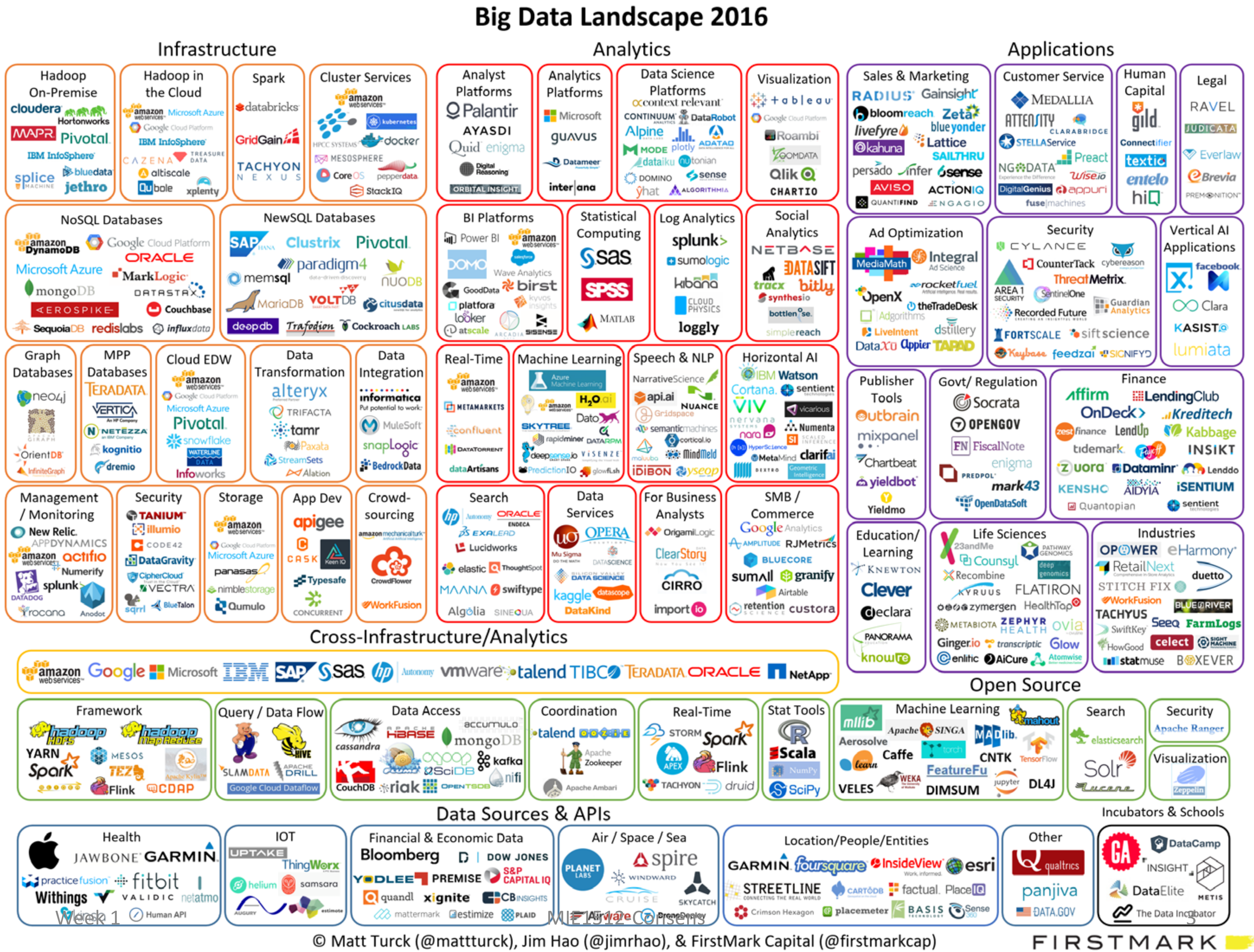
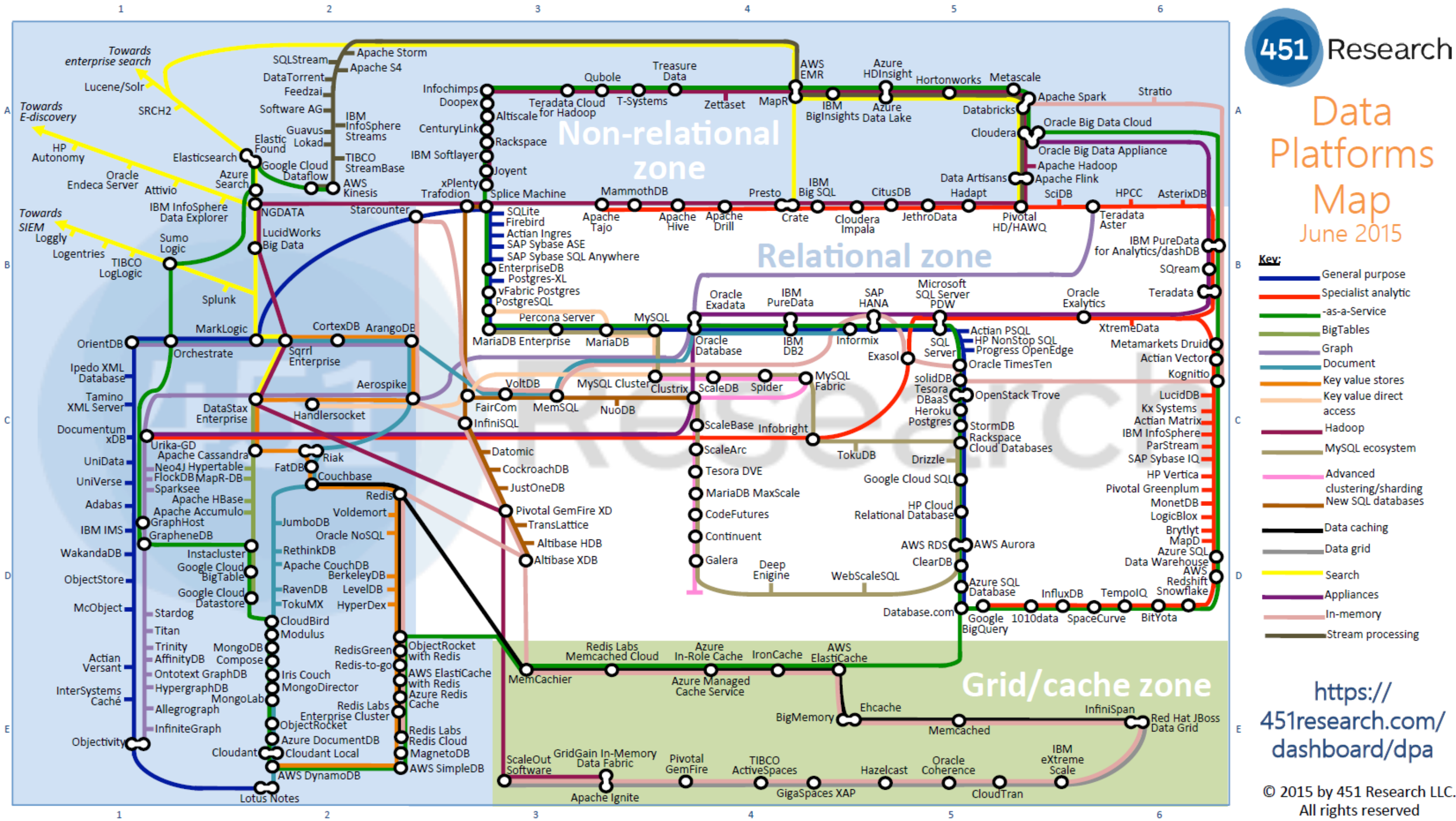
MIE1512 Data Analytics

Welcome!
Course Overview

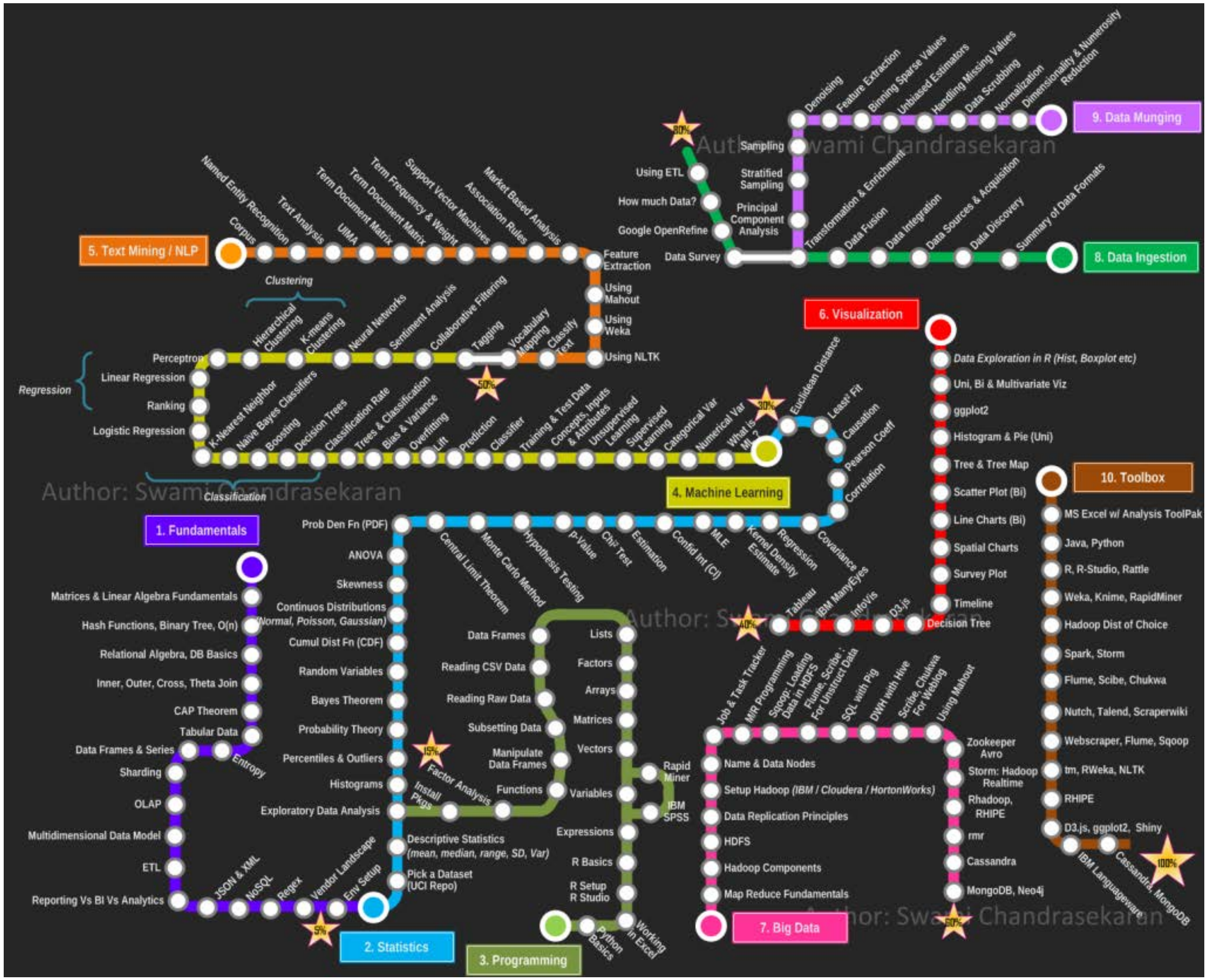
Overview

- This course is a research seminar that focuses on recent developments in the area of Data Analytics.
- Science, businesses, society and government are been revolutionized by data-driven methods. The increased access to large quantities of digital information has provided new opportunities for innovation.
- A new area of Data Analytics, known as Big Data, is made possible thanks to novel affordable techniques for processing huge amounts of data
  - Focus on Scalable Data Management


Big Data Landscape 2016
Infrastructure
Hadoop On-Premise
Hadoop in the Cloud
Spark
Cluster Services
NoSQL Databases
NewSQL Databases
Graph Databases
MPP Databases
Cloud EDW
Data Transformation
Data Integration
Management / Monitoring
Security
Storage
App Dev
Crowd-sourcing
Analytics
Analyst Platforms
Analytics Platforms
Data Science Platforms
Visualization
BI Platforms
Statistical Computing
Log Analytics
Social Analytics
Real-Time
Machine Learning
Speech & NLP
Horizontal AI
Search
Data Services
For Business Analysts
SMB / Commerce
Applications
Sales & Marketing
Customer Service
Human Capital
Legal
Ad Optimization
Security
Vertical AI Applications
Publisher Tools
Govt/ Regulation
Finance
Education/ Learning
Life Sciences
Industries
Cross-Infrastructure/Analytics
Open Source
Framework
Query / Data Flow
Data Access
Coordination
Real-Time
Stat Tools
Machine Learning
Search
Security
Visualization
Data Sources & APIs
Health
IOT
Financial & Economic Data
Air / Space / Sea
Location/People/Entities
Other
Incubators & Schools
© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)
FIRSTMARK



451 Research
Data Platforms Map
June 2015
Non-relational zone
Relational zone
Grid/cache zone
Towards enterprise search
Towards E-discovery
Towards SIEM
Key:
General purpose
Specialist analytic
-as-a-Service
BigTables
Graph
Document
Key value stores
Key value direct access
Hadoop
MySQL ecosystem
Advanced clustering/sharding
New SQL databases
Data caching
Data grid
Search
Appliances
In-memory
Stream processing
https://451research.com/dashboard/dpa
© 2015 by 451 Research LLC. All rights reserved

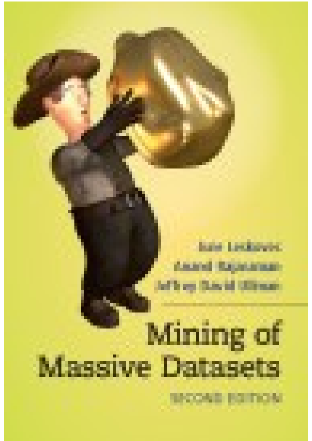


1. Fundamentals
Matrices & Linear Algebra Fundamentals
Hash Functions, Binary Tree, O(n)
Relational Algebra, DB Basics
Inner, Outer, Cross, Theta Join
CAP Theorem
Tabular Data
Entropy
Data Frames & Series
Sharding
OLAP
Multidimensional Data Model
ETL
Reporting Vs BI Vs Analytics
JSON & XML
NoSQL
Regex
Vendor Landscape
Env Setup
5%
2. Statistics
Pick a Dataset (UCI Repo)
Descriptive Statistics (mean, median, range, SD, Var)
Exploratory Data Analysis
Histograms
Percentiles & Outliers
Probability Theory
Bayes Theorem
Random Variables
Cumul Dist Fn (CDF)
Continuos Distributions (Normal, Poisson, Gaussian)
Skewness
ANOVA
Prob Den Fn (PDF)
Central Limit Theorem
Monte Carlo Method
Hypothesis Testing
p-Value
Chi² Test
Estimation
Confid Int (CI)
MLE
Kernel Density Estimate
Regression
Covariance
Correlation
Pearson Coeff
Causation
Least² Fit
Euclidean Distance
15%
Factor Analysis
3. Programming
Python Basics
Working in Excel
R Setup R Studio
R Basics
Expressions
Variables
IBM SPSS
Rapid Miner
Vectors
Matrices
Arrays
Factors
Lists
Data Frames
Reading CSV Data
Reading Raw Data
Subsetting Data
Manipulate Data Frames
Functions
Install Pkgs
4. Machine Learning
What is ML?
Numerical Var
Categorical Var
Supervised Learning
Unsupervised Learning
Concepts, Inputs & Attributes
Training & Test Data
Classifier
Prediction
Lift
Overfitting
Bias & Variance
Trees & Classification
Classification Rate
Decision Trees
Boosting
Naive Bayes Classifiers
K-Nearest Neighbor
Classification
Logistic Regression
Ranking
Linear Regression
Regression
Perceptron
Hierarchical Clustering
K-means Clustering
Clustering
Neural Networks
Sentiment Analysis
Collaborative Filtering
Tagging
30%
50%
5. Text Mining / NLP
Corpus
Named Entity Recognition
Text Analysis
UIMA
Term Document Matrix
Term Frequency & Weight
Support Vector Machines
Association Rules
Market Based Analysis
Feature Extraction
Using Mahout
Using Weka
Using NLTK
Classify Text
Vocabulary Mapping
6. Visualization
Data Exploration in R (Hist, Boxplot etc)
Uni, Bi & Multivariate Viz
ggplot2
Histogram & Pie (Uni)
Tree & Tree Map
Scatter Plot (Bi)
Line Charts (Bi)
Spatial Charts
Survey Plot
Timeline
Decision Tree
D3.js
InfoVis
IBM ManyEyes
Tableau
40%
7. Big Data
Map Reduce Fundamentals
Hadoop Components
HDFS
Data Replication Principles
Setup Hadoop (IBM / Cloudera / HortonWorks)
Name & Data Nodes
Job & Task Tracker
M/R Programming
Sqoop: Loading Data in HDFS
Flume, Scribe: For Unstruct Data
SQL with Pig
DWH with Hive
Scribe, Chukwa For Weblog
Using Mahout
Zookeeper Avro
Storm: Hadoop Realtime
Rhadoop, RHIPE
rmr
Cassandra
MongoDB, Neo4j
60%
8. Data Ingestion
Summary of Data Formats
Data Discovery
Data Sources & Acquisition
Data Integration
Data Fusion
Transformation & Enrichment
Data Survey
Google OpenRefine
How much Data?
Using ETL
80%
9. Data Munging
Dimensionality & Numerosity Reduction
Normalization
Data Scrubbing
Handling Missing Values
Unbiased Estimators
Binning Sparse Values
Feature Extraction
Denoising
Sampling
Stratified Sampling
Principal Component Analysis
10. Toolbox
MS Excel w/ Analysis ToolPak
Java, Python
R, R-Studio, Rattle
Weka, Knime, RapidMiner
Hadoop Dist of Choice
Spark, Storm
Flume, Scibe, Chukwa
Nutch, Talend, Scraperwiki
Webscraper, Flume, Sqoop
tm, RWeka, NLTK
RHIPE
D3.js, ggplot2, Shiny
IBM Languageware
Cassandra, MongoDB
100%
Author: Swami Chandrasekaran

Overview

- This seminar provides an overview of data analytics concepts, approaches, and techniques, including distributed computations on massive datasets and frameworks for enabling large-scale parallel data processing on clusters of commodity servers. Emphasis is given to algorithmic techniques for analyzing Web Data and Open Data.
- The course evaluation is based on and invigilated lab test, course participation and presentations, and a project.
- The project goal is to prepare reproducible (and potentially publishable) contributions in the area of data analytics, with an emphasis in scalable data preparation and exploration.

General Textbook Reference



Mining of Massive Datasets

[Jure Leskovec](), [Anand Rajaraman](), [Jeff Ullman]()

http://infolab.stanford.edu/~ullman/mmds/book.pdf

Course Grading

Invigilated Lab Test (notebooks)	20%
Class participation and presentation	15%
Course project	65%

- The presentation is based on the bibliography selected for the project
- All the project deliverables contribute to the project grade (individual grading within a group project)
- The analytical techniques must be selected from the literature, and then applied to an open dataset (the focus is on data preparation and exploration)
- Originality constraint: the project cannot use a dataset AND analytical techniques if already described elsewhere

Project Schedule

	Week
Form Groups	4-5
Bibliography + PlanV1	5-7
Validation + PlanV2	8-9
Progress + PlanV3	10-11
Final Report	12-13
Presentations	6-13

Presentation

- Select techniques described in the project literature (see Bibliography deliverable)
- Motivate the problem
- Present the approach (using examples)
- Describe related work, contributions, and relevance

Form Groups

- Groups must have 5 to 7 members
- The project is graded on a **individual** basis
- Project plans (V1 to V3) represent the evolving group deliverables
- Individual grade is based on the contributions of each member to the group deliverables
  - Planned contribution vs. actual contribution
- Warning: group chemistry and the quality of the group deliverables will affect the individual grades

Bibliography

- Each group member must review 5 papers in the literature relevant to the project
  - Emphasis on quality of venue, authors, publication
- Each group member must focus on covering a selected paper in depth (directly relevant to the project)
- Suggestions
  - Bibliographic portals (ACM, IEEE, dblp)
  - Look at aminer.org and the tutorial

    www.wsdm-conference.org/2016/invited-speakers.html#pe-tang

Project Plan V1

- Project summary describing techniques and datasets, with references to the bibliography
- Initial (V1) four week long plan of activities, describing the tasks (with estimate of hours) for each project member
  - Select relevant subset of CRISP-DM
- Planning should focus on de-risking dataset selection and data preparation
  - Accuracy of task/estimates for first week is critical
- Must include notebooks with individual work

CRISP-DM

Phases and Tasks

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives	Collect Initial Data	Select Data	Select Modeling Technique	Evaluate Results	Plan Deployment
Assess Situation	Describe Data	Clean Data	Generate Test Design	Review Process	Plan Monitering & Maintenance
Determine Data Mining Goals	Explore Data	Construct Data	Build Model	Determine Next Steps	Produce Final Report
Produce Project Plan	Verify Data Quality	Integrate Data	Assess Model		Review Project
		Format Data			

UofT

Suggestions

- Understand the Metadata of the Open Data available in a given Domain
- Browse around
  - datatau.com
  - XML Standard Data (xml.coverpages.org)
  - Social Data (datakind.org), Urban Data
  - Developers Data (github.com)
  - commoncrawl.org
  - ckan.org
  - schema.org
  - dataverse.org
- Explore Data Challenges
  - kaggle.com

UofT

Project Plans V2,V3

- Updates to project summary
  - Opportunity for incorporating instructor feedback
- Updates to project plan
  - Planned vs. Actual
  - Revisions to earlier plans
- Must include notebooks with individual work
- For V2 notebooks should include a full validation of the data preparation
- For V3 you should be almost done

Final Project Report

- The results of the project should be reported in a 5-10 page long manuscript following ACM or IEEE conference/journal style guidelines
  - Single space, single column, 8.5 X 11 paper, 11pt font
- All the project data manipulations **must be reproducible** and (mostly) completed using notebooks (including documentation and datasets)

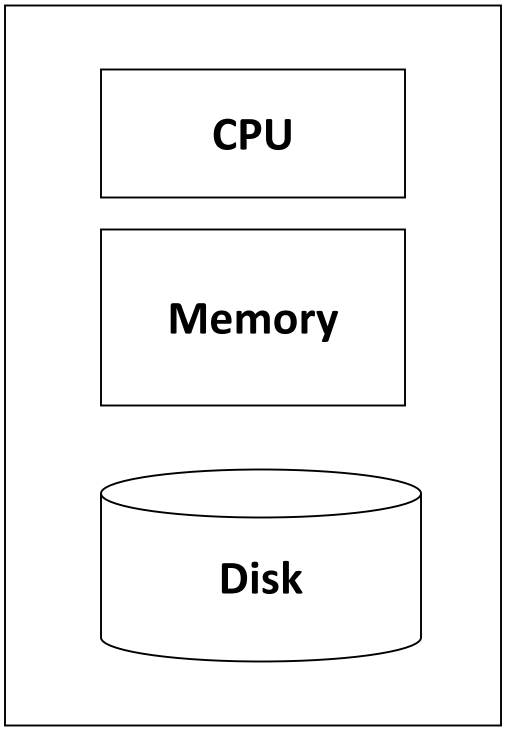
Course Prerequisite

- MIE253 or equivalent data management course
- Suggested MOOC Option
  - https://cs.stanford.edu/people/widom/DB-mooc.html
  - https://lagunita.stanford.edu/courses/DB/2014/SelfPaced/about

MIE1512 Data Analytics

MapReduce

Single-node architecture



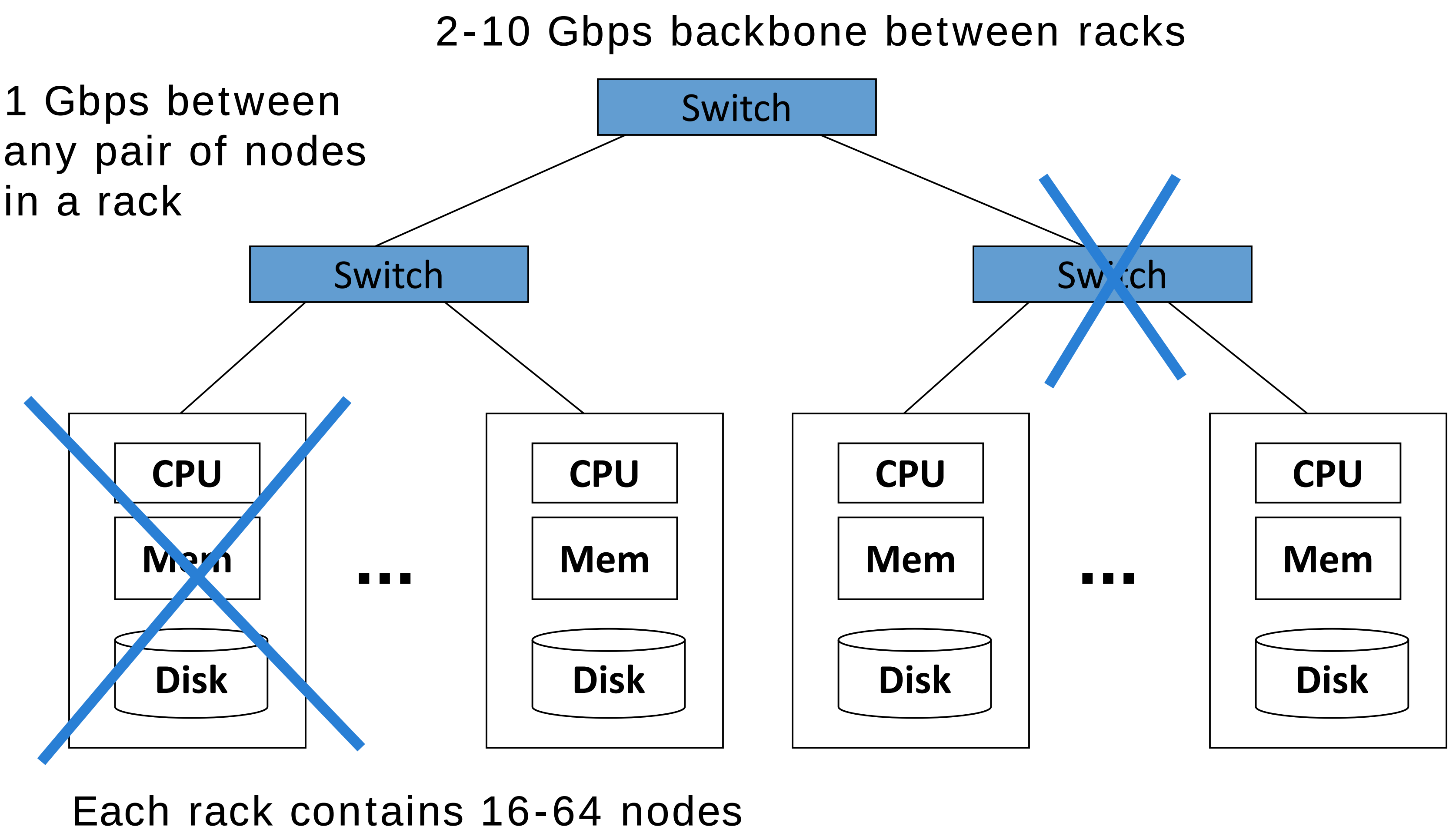
Machine Learning, Statistics

“Classical” Data Mining

Commodity Clusters

- Web data sets can be very large
  - Tens to hundreds of terabytes
- Cannot mine on a single server (why?)
- Standard architecture emerging:
  - Cluster of commodity Linux nodes
  - Gigabit ethernet interconnect
- How to organize computations on this architecture?
  - Mask issues such as hardware failure

Cluster Architecture



Stable storage

- First order problem: if nodes can fail, how can we store data persistently?
- Answer: Distributed File System
  – Provides global file namespace
  – Google GFS; Hadoop HDFS; Kosmix KFS
- Typical usage pattern
  – Huge files (100s of GB to TB)
  – Data is rarely updated in place
  – Reads and appends are common

Distributed File System

- Chunk Servers
  - File is split into contiguous chunks
  - Typically each chunk is 16-64MB
  - Each chunk replicated (usually 2x or 3x)
  - Try to keep replicas in different racks
- Master node
  - a.k.a. Name Nodes in HDFS
  - Stores metadata
  - Might be replicated
- Client library for file access
  - Talks to master to find chunk servers
  - Connects directly to chunkservers to access data

Reading

- Jeffrey Dean and Sanjay Ghemawat,

  MapReduce: Simplified Data Processing on Large Clusters, CACM 2008 (OSDI 2004)

  https://dl.acm.org/citation.cfm?doid=1327452.1327492

- Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, **The Google File System, SOSP 2003**

  https://dl.acm.org/citation.cfm?doid=1165389.945450

Warm up: Word Count

- We have a large file of words, one word to a line
- Count the number of times each distinct word appears in the file
- Sample application: analyze web server logs to find popular URLs

Word Count (2)

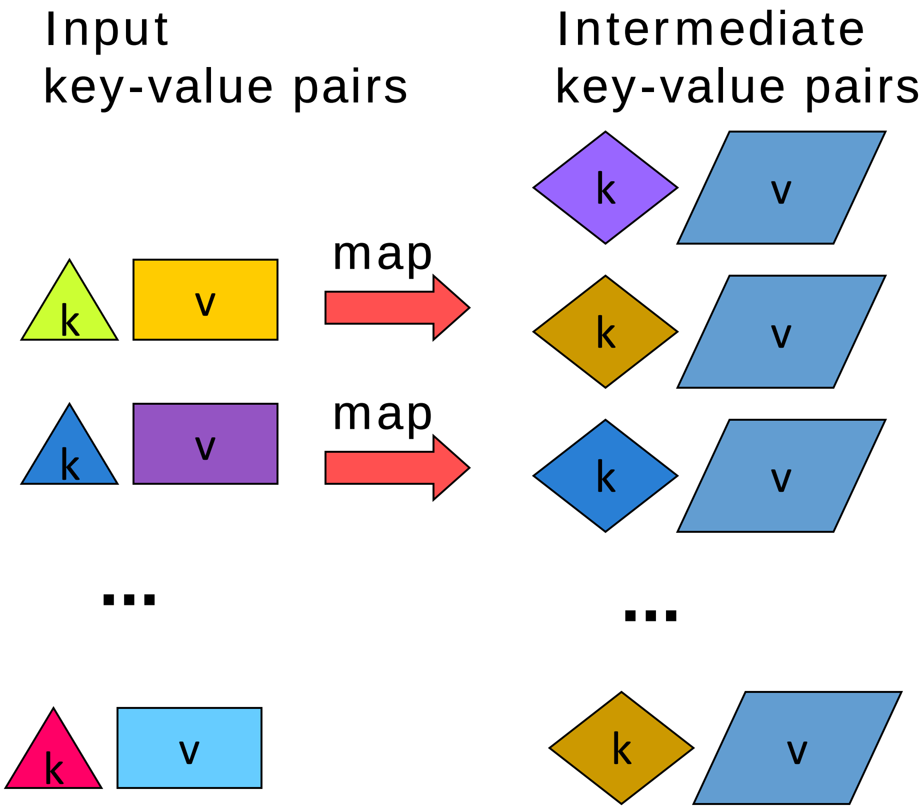
- Case 1: Entire file fits in memory
- Case 2: File too large for mem, but all <word, count> pairs fit in mem
- Case 3: File on disk, too many distinct words to fit in memory
  - **`sort datafile | uniq –c`**

UofT

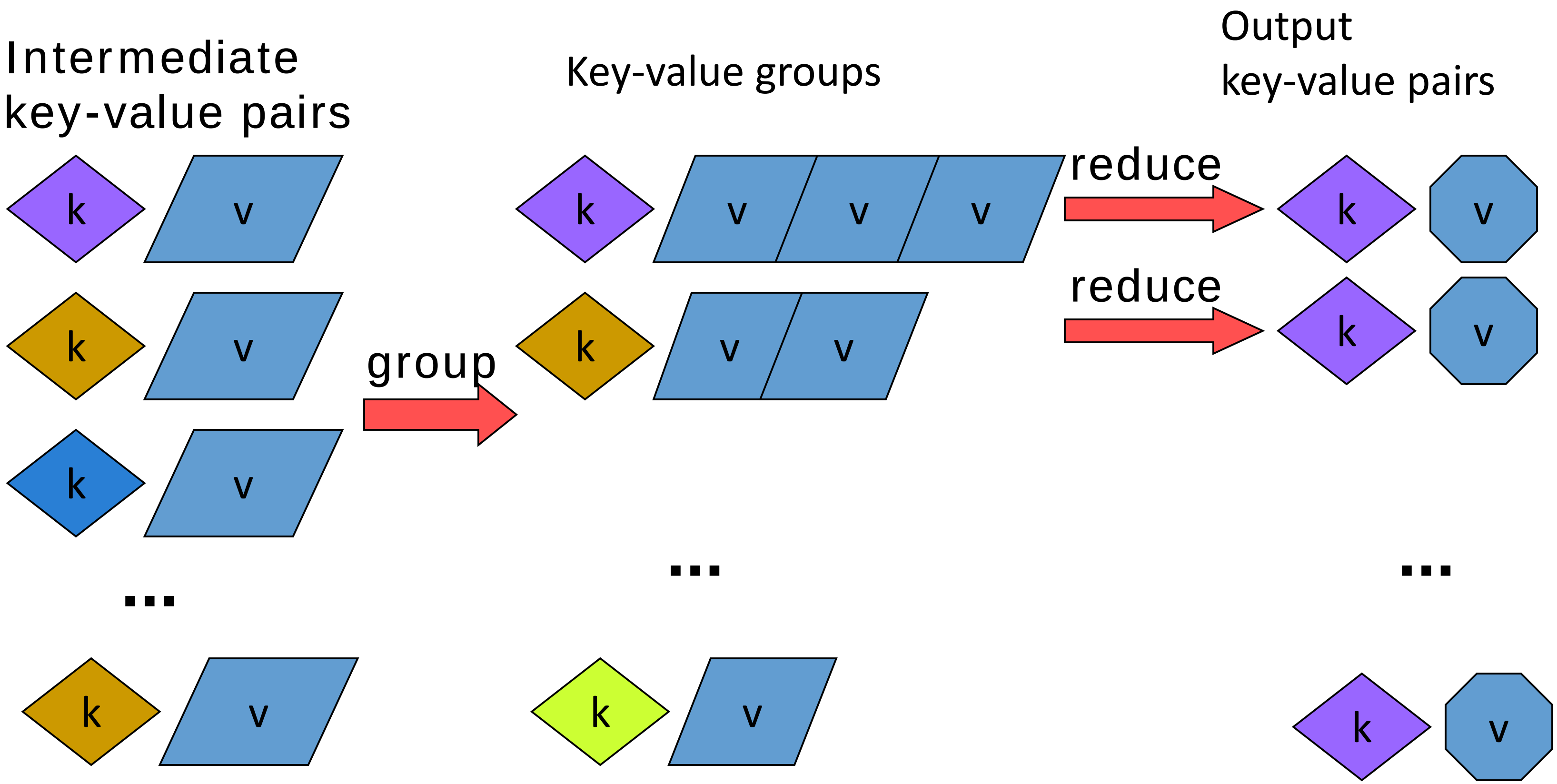
Word Count (3)

- To make it slightly harder, suppose we have a large corpus of documents
- Count the number of times each distinct word occurs in the corpus
  - **`words(docs/*) | sort | uniq -c`**
  - where **`words`** takes a file and outputs the words in it, one to a line
- The above captures the essence of MapReduce
  - Great thing is it is naturally parallelizable

MapReduce: The Map Step



MapReduce: The Reduce Step



MapReduce

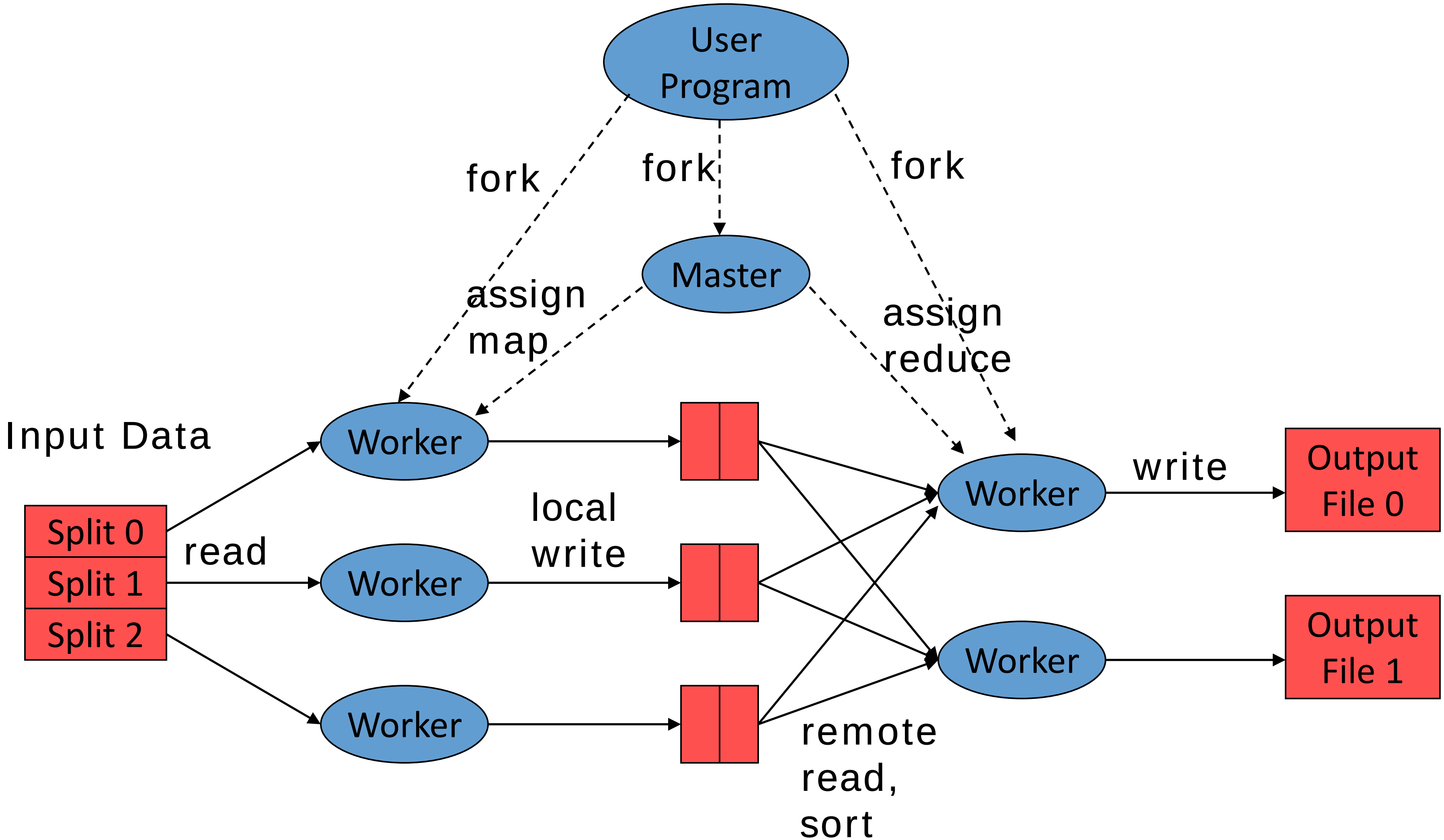
- Input: a set of key/value pairs
- User supplies two functions:
  - map(k,v) → list(k1,v1)
  - reduce(k1, list(v1)) → v2
- (k1,v1) is an intermediate key/value pair
- Output is the set of (k1,v2) pairs

Word Count using MapReduce

```
map(key, value):
// key: document name; value: text of document
   for each word w in value:
    emit(w, 1)


reduce(key, values):
// key: a word; value: an iterator over counts
    result = 0
    for each count v in values:
        result += v
    emit(result)
```

Distributed Execution Overview



Partition Function

- Inputs to map tasks are created by contiguous splits of input file
- For reduce, we need to ensure that records with the same intermediate key end up at the same worker
- System uses a default partition function e.g., hash(key) mod R
- Sometimes useful to override
  - E.g., hash(hostname(URL)) mod R ensures URLs from a host end up in the same output file

Implementations

- Created by Google
  - Not available outside Google
- Hadoop
  - An open-source implementation in Java
  - Uses HDFS for stable storage
  - Available from http://lucene.apache.org/hadoop/
- Several commercial vendors
  - Cloudera
  - Hortonworks
    - Sandbox (VM image) is the easiest way to get started http://hortonworks.com/products/hortonworks-sandbox/
    - Supports Hadoop 2.3, many tutorials

Hadoop Evolution

Ongoing Innovation in Apache

	Hadoop & YARN	Pig	Hive	Tez	Solr	Spark	Zeppelin	Slider	HBase	Phoenix	Accumulo	Storm	Falcon	Atlas	Sqoop	Flume	Kafka	Ambari	Cloudbreak	Zookeeper	Oozie	Knox	Ranger
HDP 2.5 2H2016	2.7.3	0.16.0	1.2.1+ 2.1*	0.7.0	5.2.2	1.6.2+ 2.0*	0.6.0	0.91.0	1.1.2	4.7.0	1.7.0	1.0.1	0.10.0	0.7.0	1.4.6	1.5.2	0.10.0	2.4.0	1.3.0	3.4.6	4.2.0	0.9.0	0.6.0
HDP 2.4 Mar 2016	2.7.1	0.15.0	1.2.1	0.7.0	5.2.1	1.6.0		0.80.0	1.1.2	4.4.0	1.7.0	0.10.0	0.6.1	0.5.0	1.4.6	1.5.2	0.9.0	2.2.1	1.2.0	3.4.6	4.2.0	0.6.0	0.5.0
HDP 2.3 Oct 2015	2.7.1	0.15.0	1.2.1	0.7.0	5.2.1	1.4.1		0.80.0	1.1.2	4.4.0	1.7.0	0.10.0	0.6.1	0.5.0	1.4.6	1.5.2	0.8.2	2.1.0	1.0.0	3.4.6	4.2.0	0.6.0	0.5.0
HDP 2.2 Dec 2014	2.6.0	0.14.0	0.14.0	0.5.2	4.10.2	1.2.1		0.60.0	0.98.4	4.2.0	1.6.1	0.9.3	0.6.0		1.4.5	1.5.2	0.8.1	2.0.0		3.4.6	4.1.0	0.5.0	0.4.0
HDP 2.1 April 2014	2.4.0	0.12.1	0.13.0	0.4.0	4.7.2				0.98.0	4.0.0	1.5.1	0.9.1	0.5.0		1.4.4	1.4.0		1.5.1		3.4.5	4.0.0	0.4.0	
HDP 2.0 Oct 2013	2.2.0	0.12.0	0.12.0						0.96.1						1.4.4	1.3.1		1.4.4		3.4.5	3.3.2		
	DATA MGMT	DATA ACCESS											GOVERNANCE & INTEGRATION					OPERATIONS				SECURITY	

HORTONWORKS DATA PLATFORM

** Spark 1.6.2+ Spark 2.0 – HDP 2.5 support installation of both Spark 1.6.2 and Spark 2.0. Spark 2.0 is Technical Preview within HDP 2.5.*
Hive 1.2.1+ Hive 2.1 – Hive 2.1 is Technical Preview within HDP 2.5.

Cloud Computing

- Ability to rent cluster computing by the hour
  – Additional services, e.g., persistent storage
  – Eg, Amazon’s “Elastic Compute Cloud” (EC2), MS Azure, IBM Bluemix, Google Cloud Platform

- Facilitates scalability and elasticity

Map-Reduce: Environment

Map-Reduce environment takes care of:

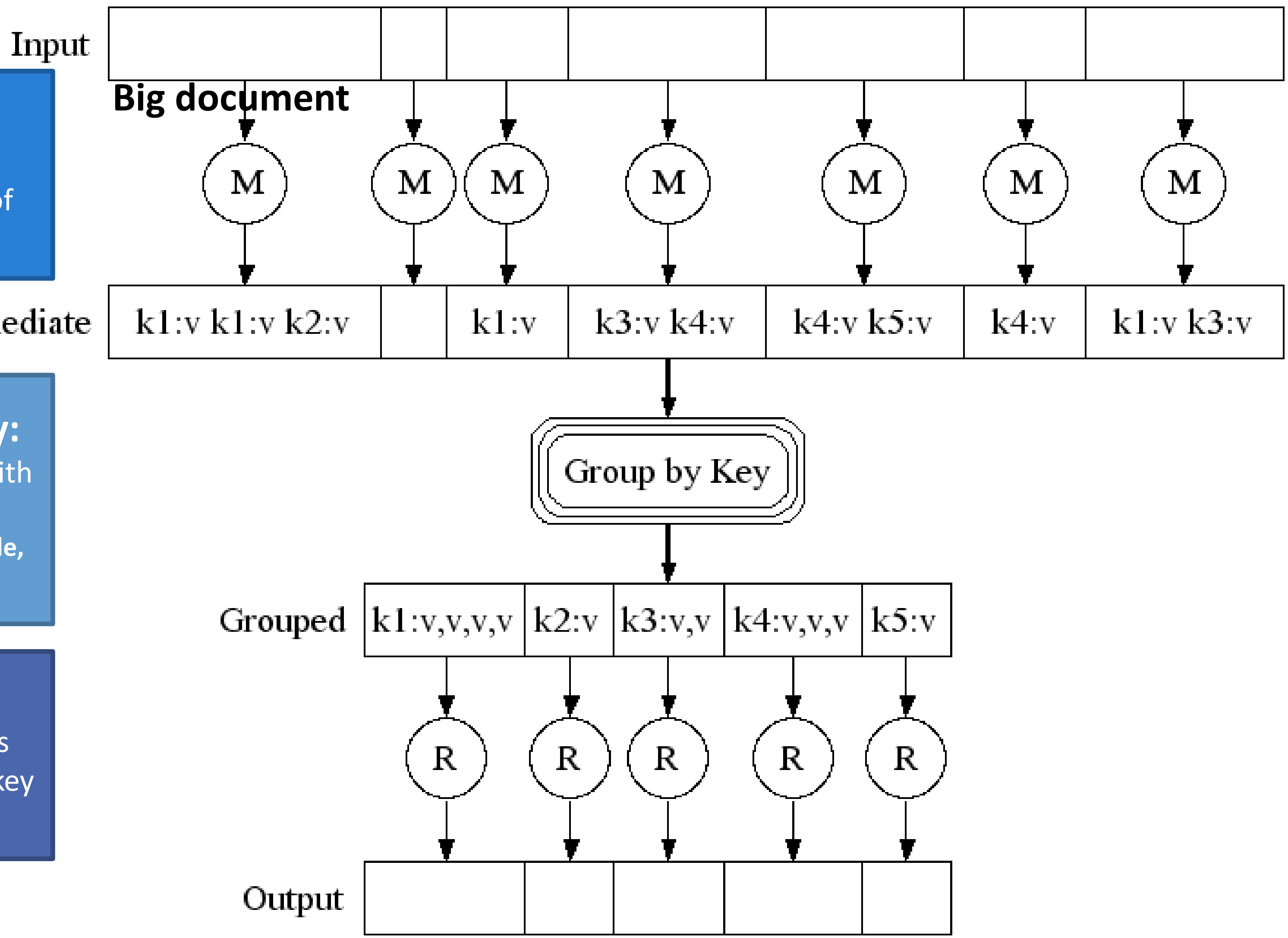
- Partitioning the input data
- Scheduling the program's execution across a set of machines
- Performing the **group by key** step
- Handling machine failures
- Managing required inter-machine communication

Map-Reduce: 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



MR Exercise 1: Host size

- Suppose we have a large web corpus
- Let's look at the metadata file
  - Lines of the form (URL, size, date, …)
- For each host, find the total number of bytes
  - i.e., the sum of the page sizes for all URLs from that host

MR Exercise 2: Distributed Grep

- Find all occurrences of the given pattern in a very large set of files

Exercise 3: Graph reversal

- Given a directed graph as an adjacency list:

src1: dest11, dest12, …

src2: dest21, dest22, …

- Construct the graph in which all the links are reversed

Exercise 4: Frequent Pairs

- Given a large set of market baskets, find all frequent pairs
  - Remember definitions from Association Rules lectures