Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 03: Data Mining (Cont.) + Map-Reduce

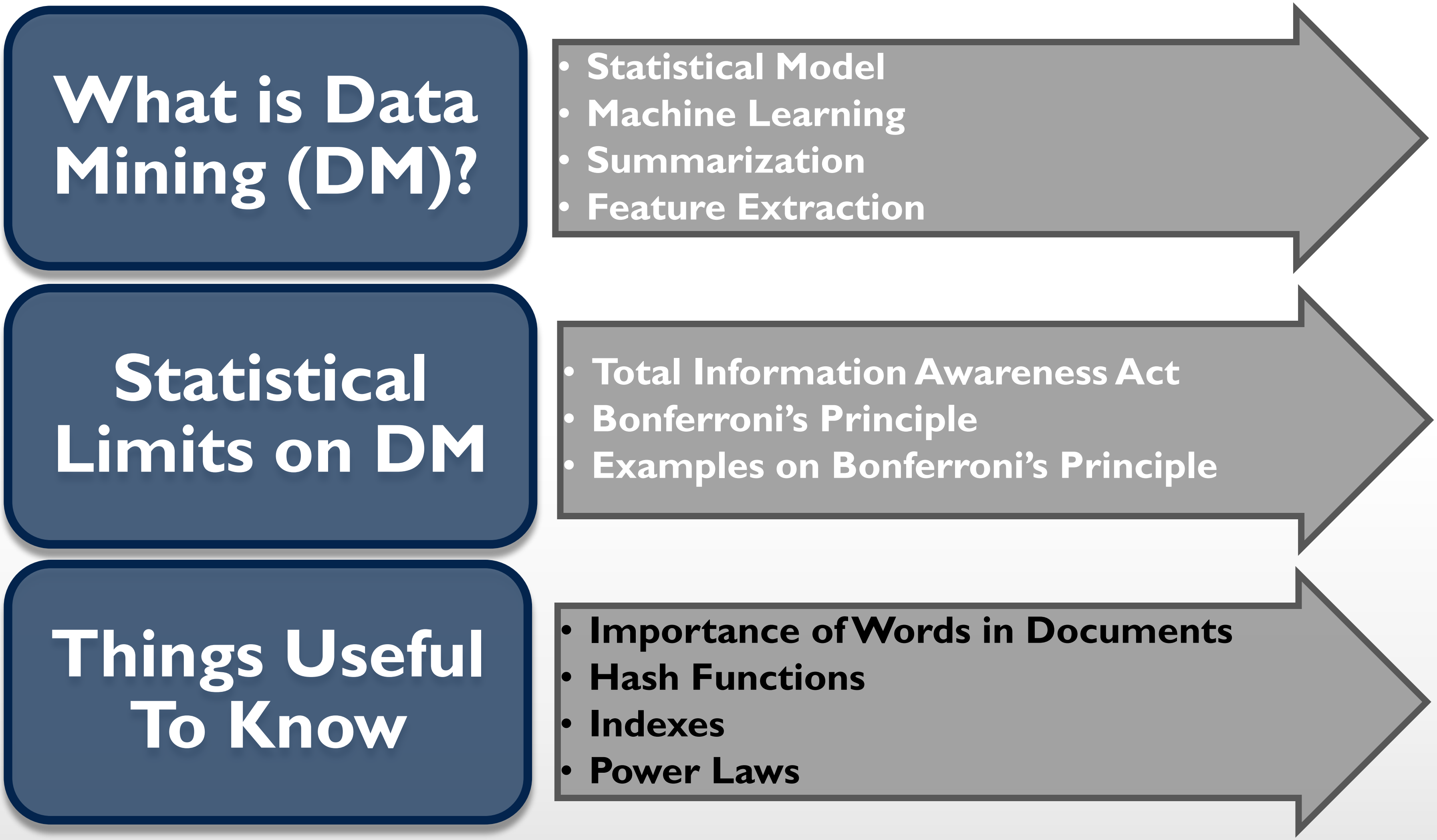
AUBURN UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13

Chapter 01: Data Mining (Cont.)



What is Data Mining (DM)?

- Statistical Model
- Machine Learning
- Summarization
- Feature Extraction

Statistical Limits on DM

- Total Information Awareness Act
- Bonferroni's Principle
- Examples on Bonferroni's Principle

Things Useful To Know

- Importance of Words in Documents
- Hash Functions
- Indexes
- Power Laws

Importance of Words in Documents

- In several applications, we will want to categorize documents by their topic.

- Typically, topics are identified by finding special words that characterize that topic!!

- Words of Caution:
  - The most frequent words in a document are typically of little value
  - Not all rare words are equally useful
    - "Notwithstanding" versus "chukker"

TF.IDF – Term Frequency × Inverse Document Frequency

- A formal way to measure how concentrated a given word in relatively few documents.

- Calculation:

$$TF_{ij} = \frac{f_{ij}}{\max_k \; f_{kj}}$$

f_{ij} is the no. of occurrences of term *i* in document *j*

f_{kj} is the no. of occurrences of term *k* in document *j*

$$IDF_i = \log_2\left(\frac{N}{n_i}\right)$$

N is the total number of documents examined

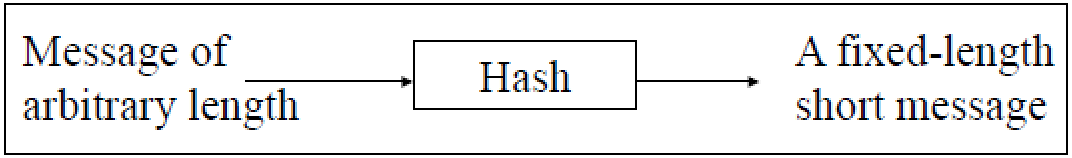
n_i is the total # documents that term *i* appeared in

- Score: $\text{TF.IDF} = TF_{ij} \times IDF_i$

- The terms with the highest score are often the best characterize the topic of the document.

Hash Functions

- Essential components of many DM algorithms



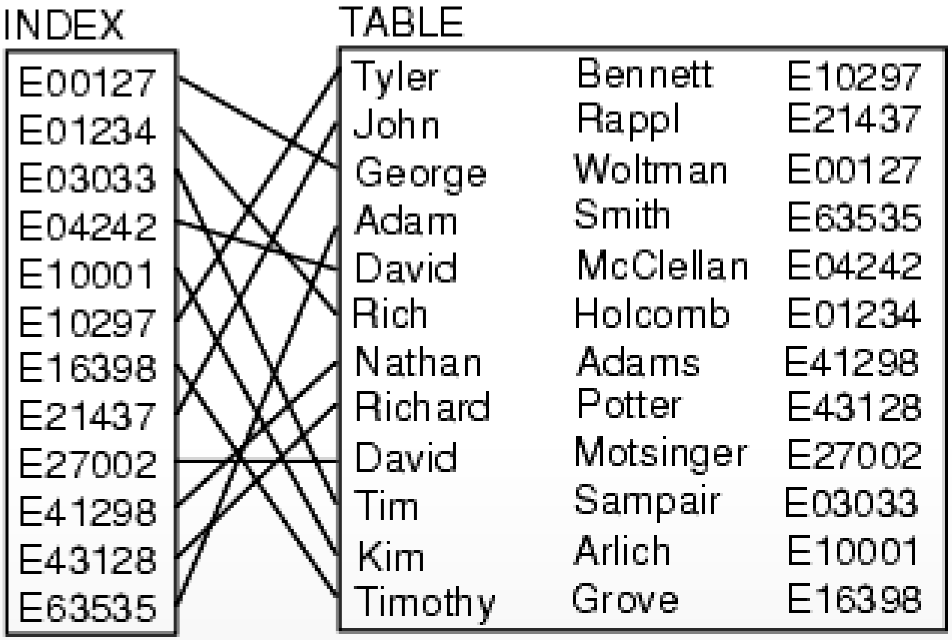
- Also known as:
  - Message digest
  - One-way function
  - Hash

- Length of $H(m)$ much shorter then length of m

- Usually fixed lengths: 128 or 160 bits

Index

- A data structure that makes it efficient to retrieve objects given one or more value of the object.
- The most common situation is one where the objects are records and the index is one of the fields of that record.



INDEX

E00127
E01234
E03033
E04242
E10001
E10297
E16398
E21437
E27002
E41298
E43128
E63535

TABLE

Tyler	Bennett	E10297
John	Rappl	E21437
George	Woltman	E00127
Adam	Smith	E63535
David	McClellan	E04242
Rich	Holcomb	E01234
Nathan	Adams	E41298
Richard	Potter	E43128
David	Motsinger	E27002
Tim	Sampair	E03033
Kim	Arlich	E10001
Timothy	Grove	E16398

Source: http://www.datadirect.com/resources/odbc/using-indexes/index.html

Power Laws

- Many phenomena can be represented by a linear relationship between the logarithms of the variables.
- General form:

$$\boxed{\begin{aligned} &\log y = b + a\log x \\ &y = cx^{a} \end{aligned}}$$

- Power law is handy in many applications, see P.14-15 in book

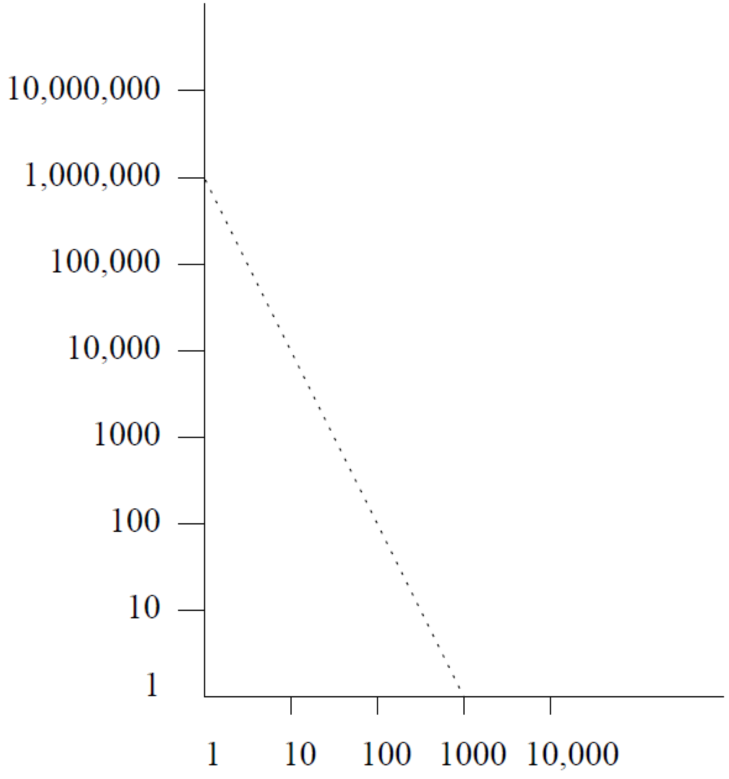
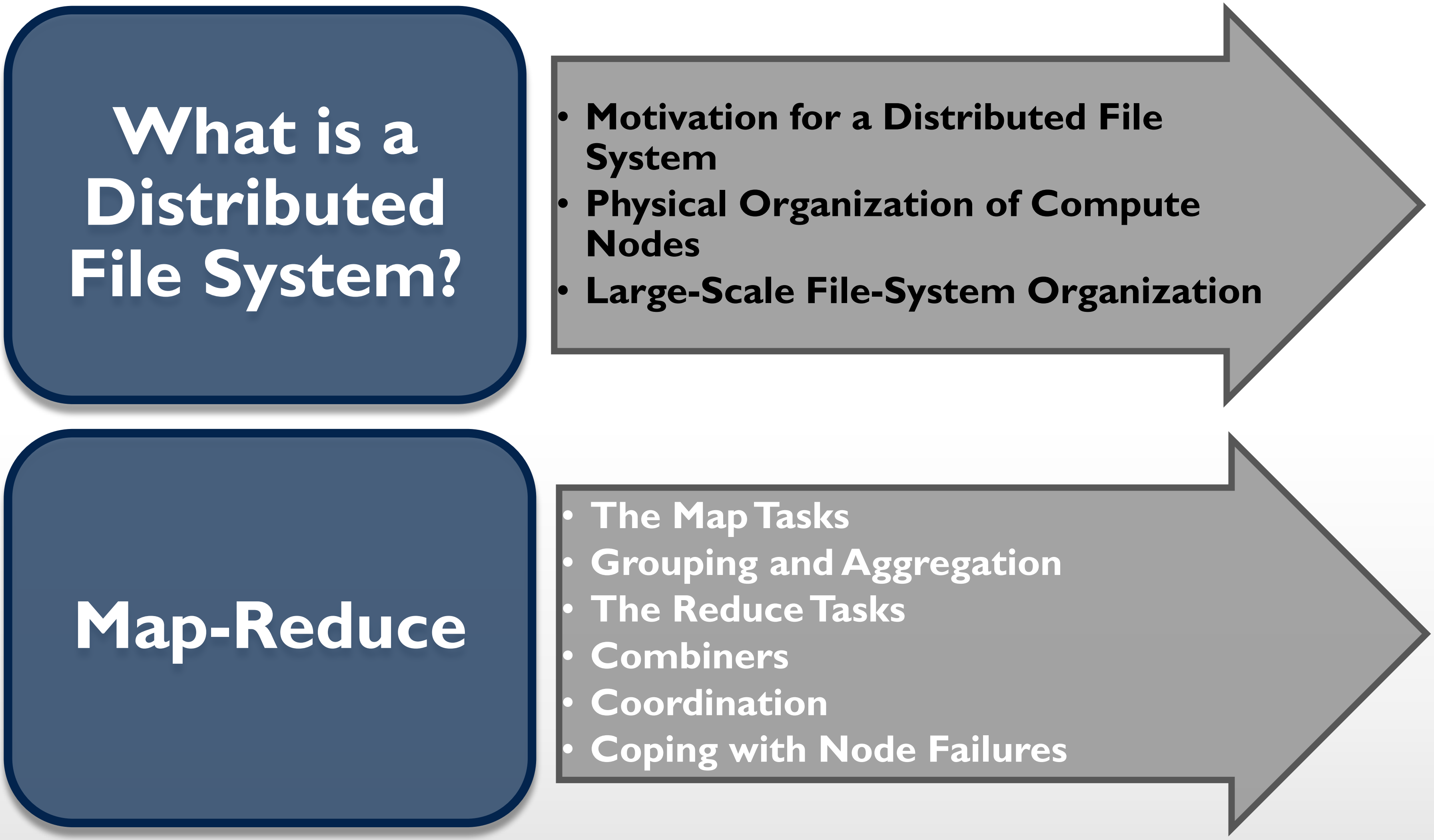


Figure 1.3: A power law with a slope of −2

Source: A. Rajaraman, J. Leskovec, J.D. Ullman. (2012). "Mining of Massive Datasets". http://i.stanford.edu/~ullman/mmds.html

Chapter 02: Map-Reduce and the New Software Stack



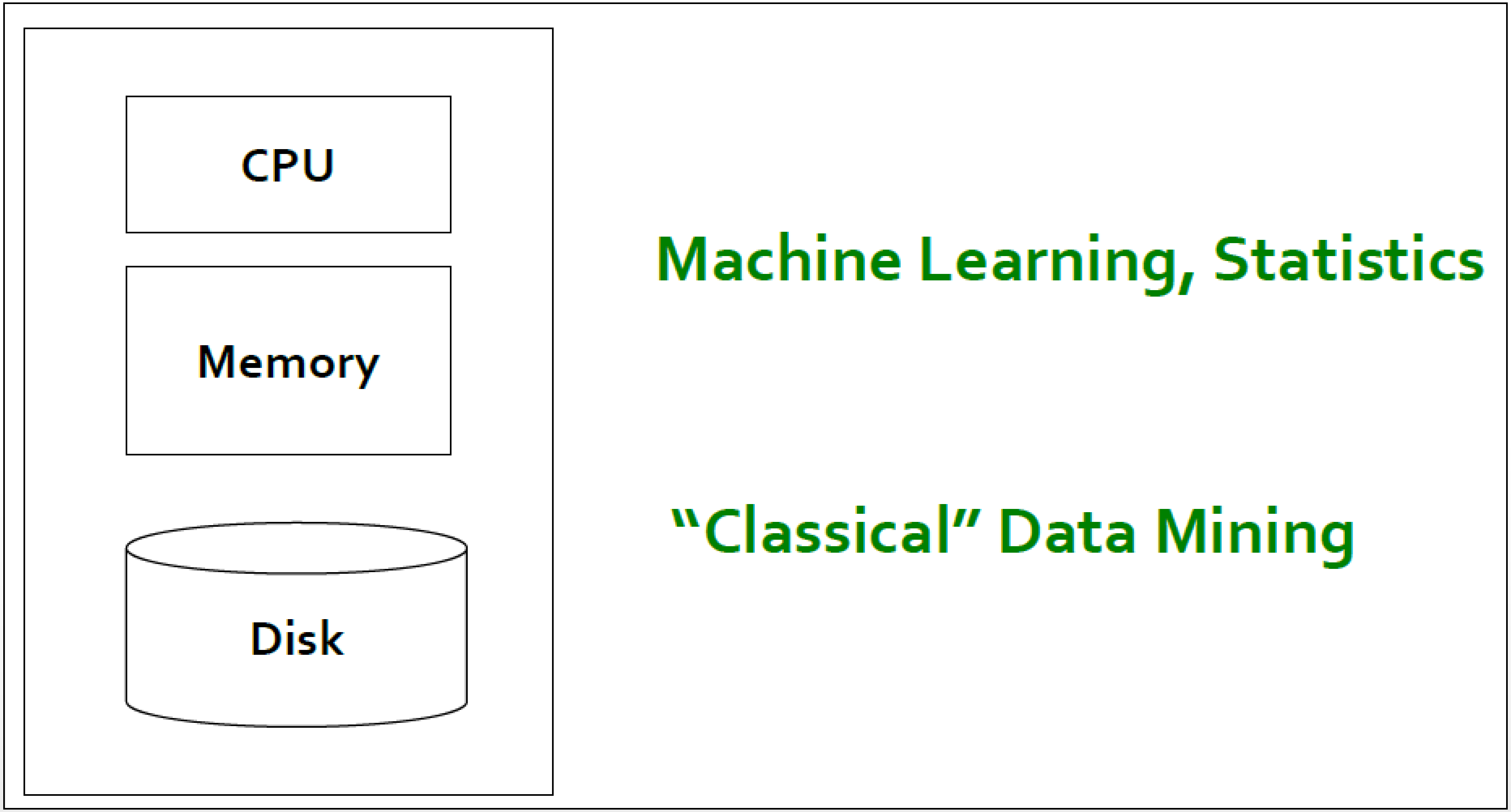
What is a Distributed File System?

- Motivation for a Distributed File System
- Physical Organization of Compute Nodes
- Large-Scale File-System Organization

Map-Reduce

- The Map Tasks
- Grouping and Aggregation
- The Reduce Tasks
- Combiners
- Coordination
- Coping with Node Failures

Single Node Architecture



Machine Learning, Statistics

“Classical” Data Mining

Source: The figure is adapted from Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

The Need for a Different Architecture: An Exercise

- Currently, there are ~20 billion webpages that need to be indexed and searched by Google. The average size of a webpage is 20KB. In addition, let us assume that the average size of a Google computer is 1 TB. Based on this information, calculate:
  - The number of days that is needed to read the web; typically, 1 computer reads 30 MB/sec from disk.
  - The number of hard drives needed to store the web.

The Need for a Different Architecture: Insights

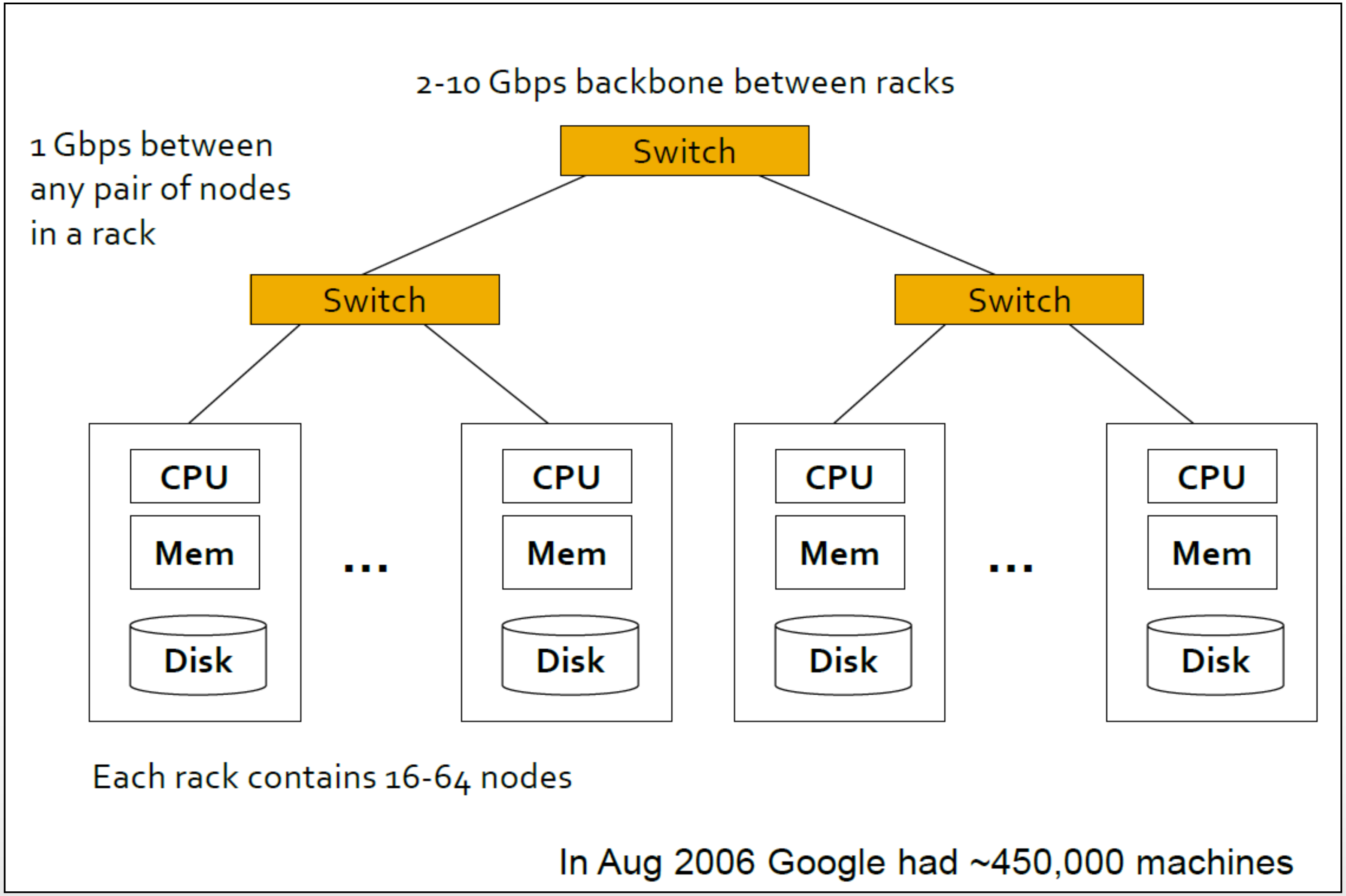
- **Takes even more to do something useful with the data!!**
  - The ranking of web pages by importance
  - Crawling the web and social networks to identify emerging public health threats

- **To deal with such applications, we are now using computing clusters that are characterized by:**
  - A large # of conventional hardware, connected via Gigabit Ethernet cables
  - A new form of file system, called a distributed file system (DFS)
  - Programming methods that can utilize/exploit the DFS

Cluster Architecture: the New Parallel Computing Paradigm

Source: http://www4.uwm.edu/projects/cluster/

Cluster Architecture: the New Parallel Computing Paradigm



2-10 Gbps backbone between racks

1 Gbps between any pair of nodes in a rack

Each rack contains 16-64 nodes

In Aug 2006 Google had ~450,000 machines

Source: Figure from Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Challenges Associated with Cluster Computing

- How do you distribute computation?

- How can we make it easy to write distributed programs?

- Machines fail:
  - Compute nodes may stay up for 3 years (~1,000 days)
  - It is not untypical to loose 1 machine/day
  - Google had ~0.5 Million machines in 2006 ☺

If we had to restart the computation every time one component failed, then the computation might not never complete successfully!!

Approach to Challenges in Large Scale Computing

- **Idea:**
  - Store files multiple times for reliability
  - Computations must be divided into tasks such that if any one fails, it can be restarted without affecting other tasks.

- **Map-reduce** addresses these problems
  - Google's computational/data manipulation model
  - Elegant way to work with big data

- Storage Infrastructure – **File system**

- Programming model – **Map-Reduce**

Large-Scale File System Organization

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

- Distributed File System:
  - Provides global file namespace
  - Google GFS; Hadoop HDFS; Kosmix’s Cloudstore

- Typical usage pattern
  - Huge files (100s of GB to TB)
  - Data is rarely updated in place
  - Reads and appends are common

Characteristics of Files that are managed by a DFS

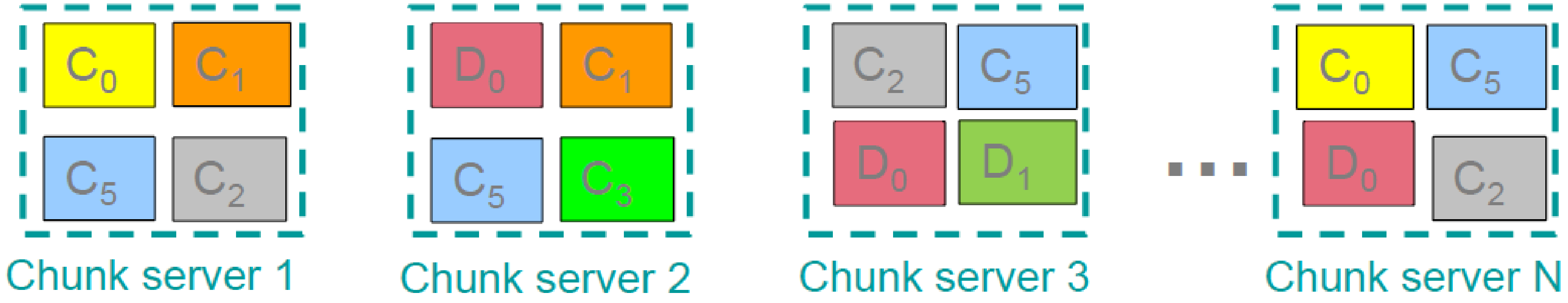
Large-Scale File System Organization (Cont.)

- 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 (Name) node
  - Stores metadata
  - Might be replicated
- Client library for file access
  - Talks to master to find chunk servers
  - Connects directly to chunk servers to access data

Source: The slide is adapted from Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Large-Scale File System Organization (Cont.)

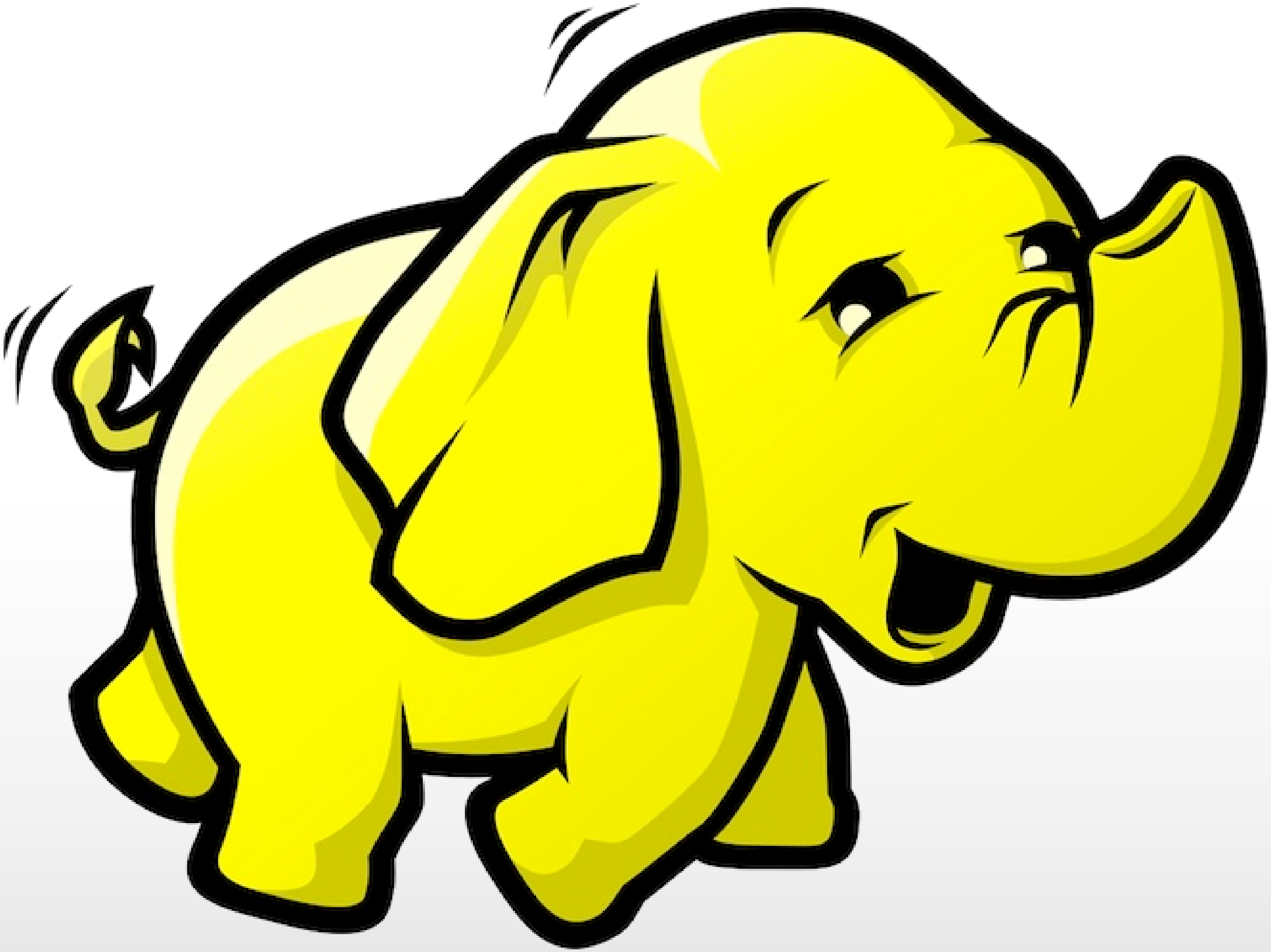
- **Reliable distributed file system**
  - Data kept in “chunks” spread across machines
  - Each chunk **replicated** on different machines
    - Seamless recovery from disk or machine failure

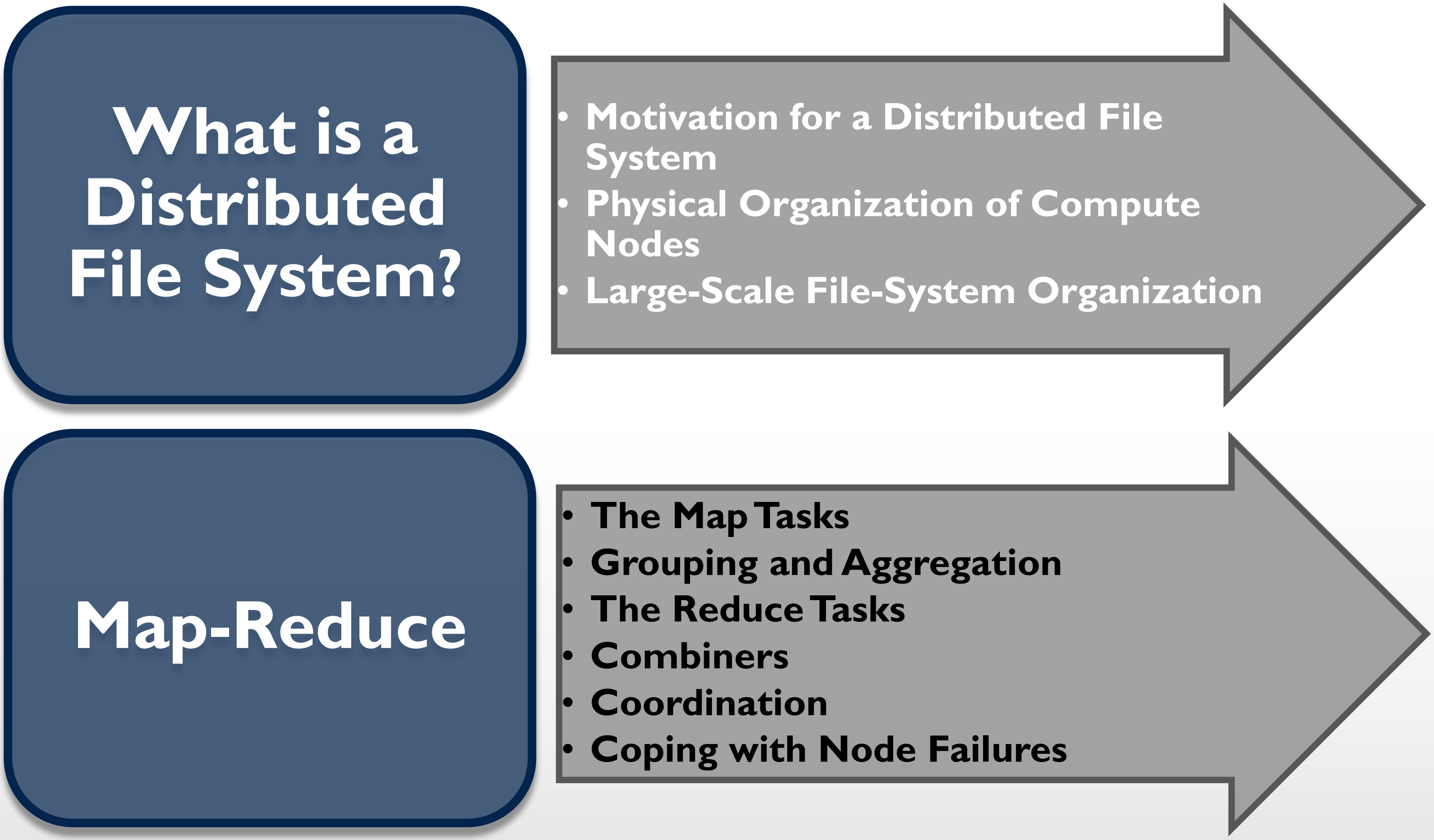


Bring computation directly to the data!

Source: The slide is adapted from Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Class Activity – A Simplistic Representation of a DFS





What is a Distributed File System?

- Motivation for a Distributed File System
- Physical Organization of Compute Nodes
- Large-Scale File-System Organization

Map-Reduce

- The Map Tasks
- Grouping and Aggregation
- The Reduce Tasks
- Combiners
- Coordination
- Coping with Node Failures

An Overview of Map-Reduce Programming

- It is a style of computing that has been implemented in several systems:
  - Google's own implementation → map-reduce
  - Several popular open-source implementations → Hadoop

- Map-reduce can manage large-scale computations in a way that is tolerant to hardware failures.

- All you need to write are two functions **Map** and **Reduce**, while the system:
  - Manages the parallel execution
  - Coordinates the tasks that execute Map and Reduce
  - Deals with the possibility that one of the tasks will fail

An Overview of Map-Reduce Programming

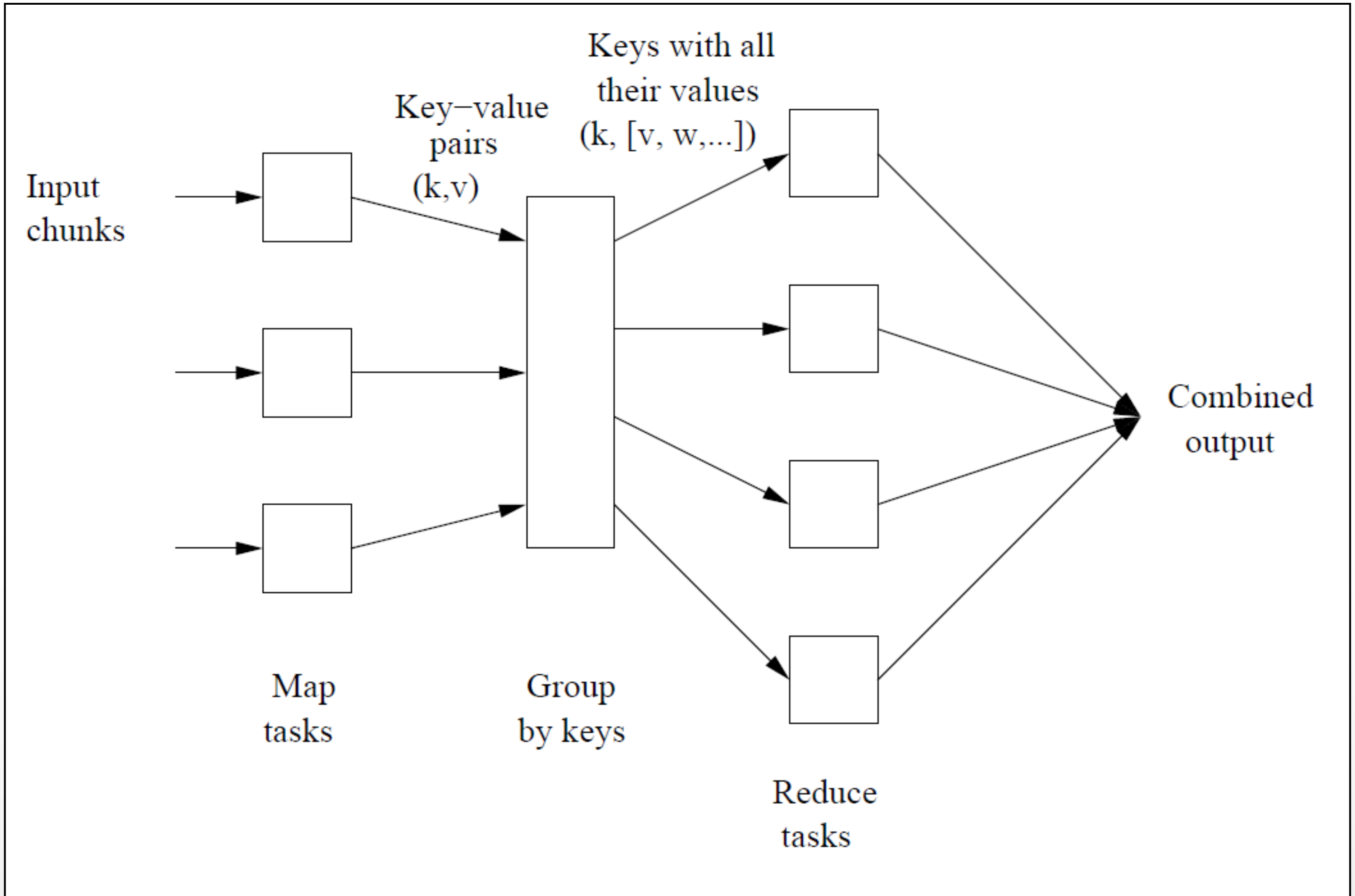


Figure 2.2: Schematic of a map-reduce computation

Source: A. Rajaraman, J. Leskovec, J.D. Ullman. (2012). "Mining of Massive Datasets". http://i.stanford.edu/~ullman/mmds.html

Recommended Readings Before Next Class

- Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters http://static.usenix.org/event/osdi04/tech/full_papers/dean/dean.pdf

- Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung: The Google File System http://www.cs.rochester.edu/meetings/sosp2003/papers/p125-ghemawat.pdf

- Wikipedia has a reasonably good overview as well http://en.wikipedia.org/wiki/MapReduce

Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 03: Data Mining (Cont.) + Map-Reduce

AUBURN UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13