Introduction to Big Data Systems

CS 448 - Spring 2019
March 18th
Thamir Qadah

Overview

- **Discussion on:**
  - *Motivation for Big Data*
  - *The MapReduce Model*
  - *Hadoop distributed file system*
  - *Spark data processing framework*
- **Think-Pair-Share Sessions, given a few discussion question:**
  - *2 minutes of thinking*
  - *2-4 minutes discuss with partner*
  - *2-4 minutes class-wide discussion*

Discussion on Big Data

What are the characteristics of Big Data?

How are they different from traditional database applications?

Why do we need different data management systems for them?

What are the characteristics of Big Data?

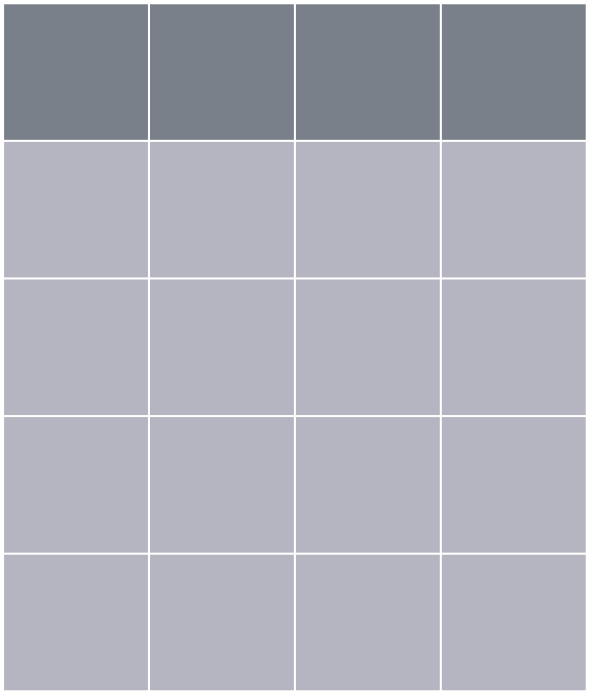
Volume: Size of data

Velocity: Rate of data

Variety: Types of data

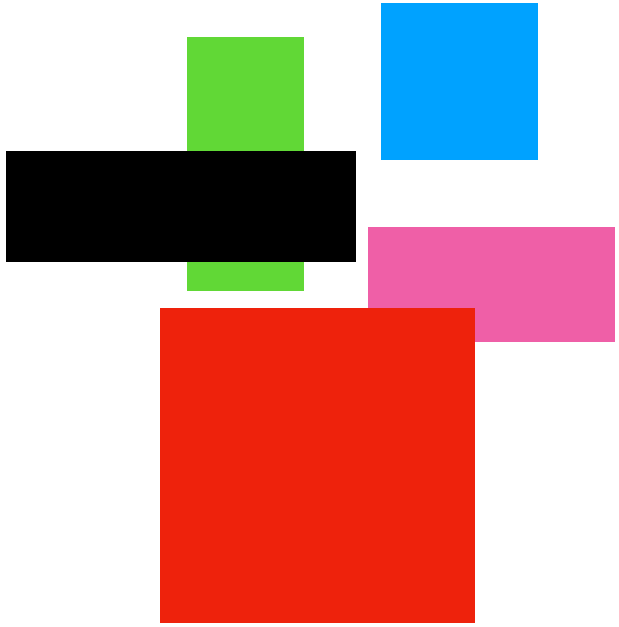
Veracity: Quality of data

How are they different from traditional database applications?



Structured

e.g. Database tables



Semi- or Un-structured

e.g. JSON, XML,
Images, Videos …

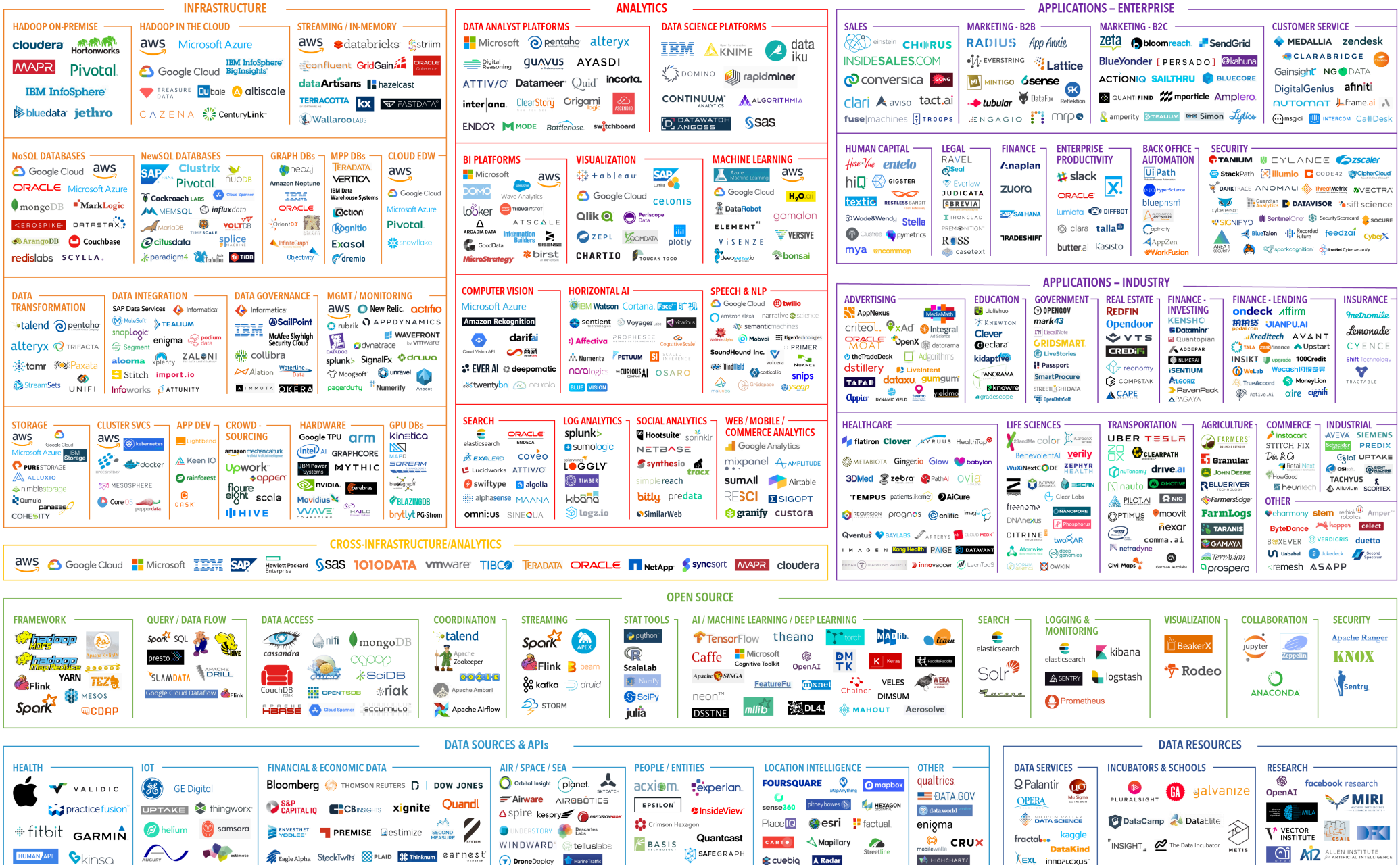
Why do we need different data management systems for Big Data?

Traditional DBMSs require some form of ETL

Not ideal for certain use-cases (e.g., Build an inverted index of webpages, Page-rank of web-pages)

One size **does not** fit all

BIG DATA & AI LANDSCAPE 2018



FIRSTMARK
EARLY STAGE VENTURE CAPITAL

Discussion on MapReduce

What are the main pieces of logic a programmer needs to specify?

What are the benefits of the MapReduce and Hadoop?

What are the main pieces of logic a programmer needs to specify?

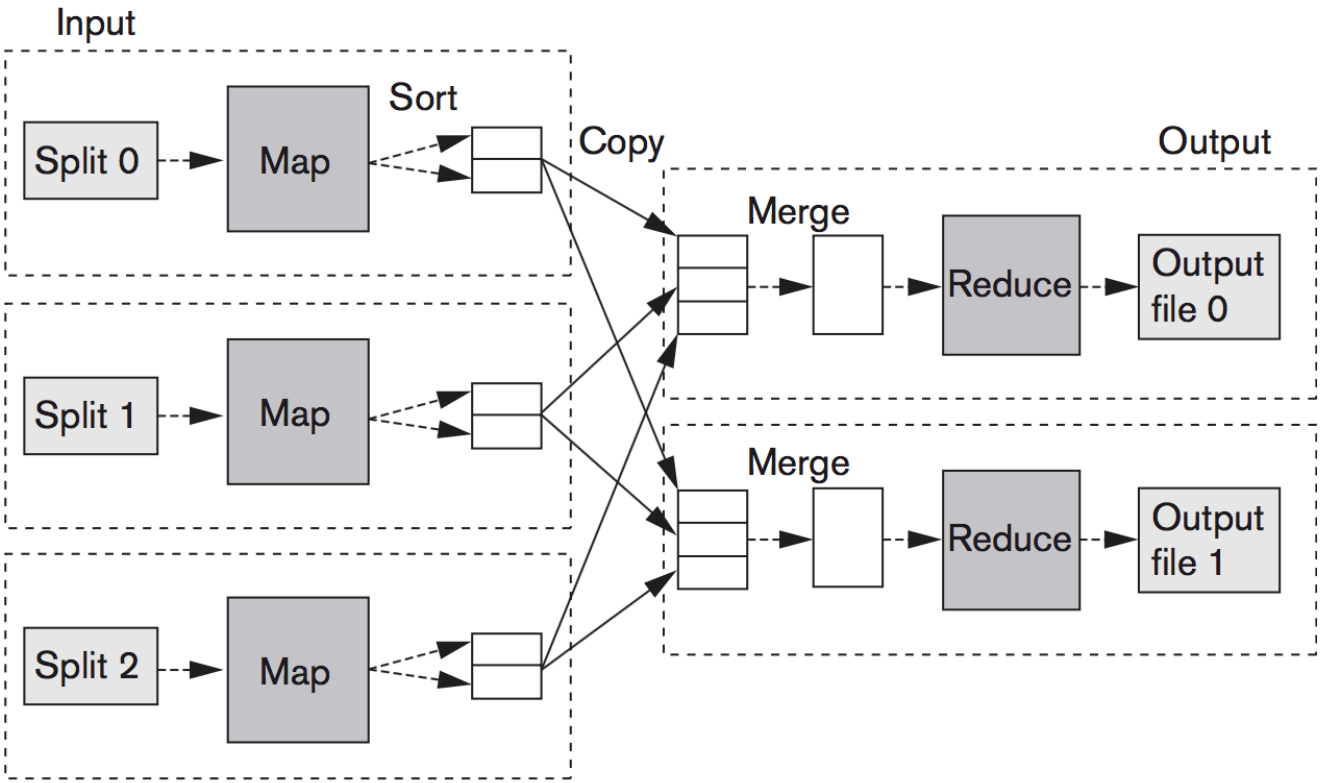


Figure 25.1
Overview of MapReduce execution. (Adapted from T. White, 2012)

MapReduce Model

```
map(K1,V1) : List[K2,V2]

reduce(K2, List[V2]) : List[K3,V3]
```

MapReduce Example

```
map[LongWritable,Text](key, value) : List[Text, LongWritable] = {
    String[] words = split(value)
    for(word : words) {
        context.out(Text(word), LongWritable(1))
    }
}
reduce[Text, Iterable[LongWritable]](key, values) : List[Text, LongWritable] = {
    LongWritable c = 0
    for( v : values) {
        c += v
    }
    context.out(key,c)
    }
```

What does this code compute?

What are the benefits of the MapReduce and Hadoop?

Simple distributed programming

Allows for highly parallel and distributed and reliable data processing

Free and open source

Discussion on HDFS

What are the design goals for HDFS?

What are the main architectural components of HDFS?

What are the design goals for HDFS?

Fault-tolerance

Throughput-optimized

Support for large files

Append-only data write model

What are the main architectural components of HDFS?

Name Node (+ secondary)

Data Nodes

Discussion on YARN

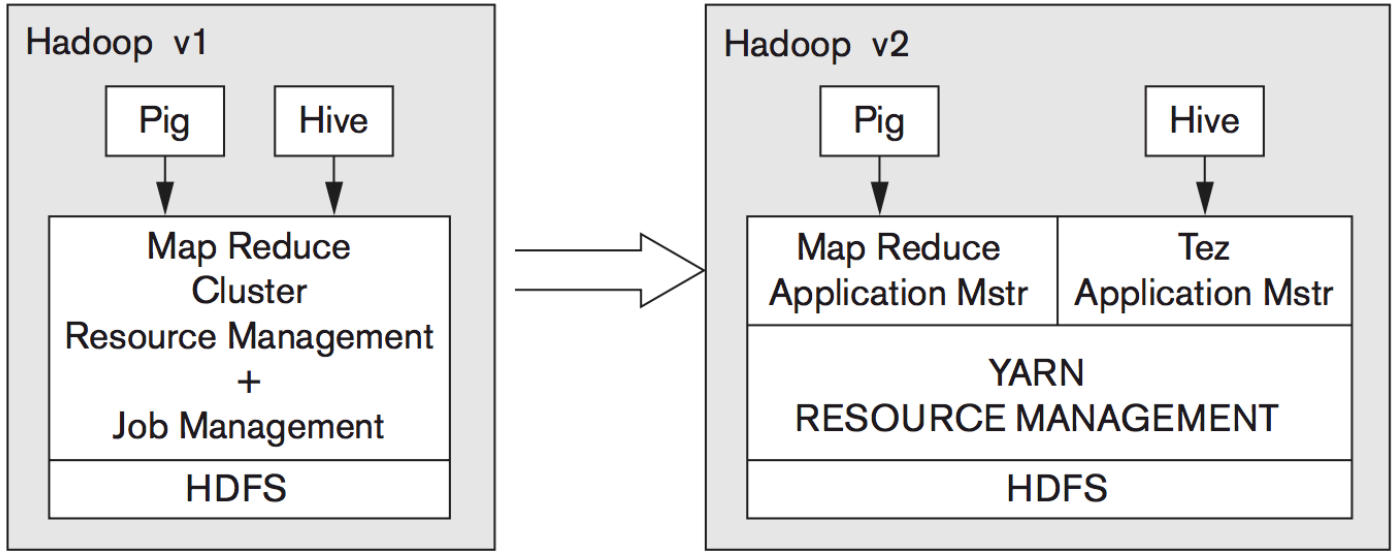


Figure 25.3
The Hadoop v1 vs. Hadoop v2 schematic.

What is the key concept behind YARN?

What are the benefits?

Discussion on YARN

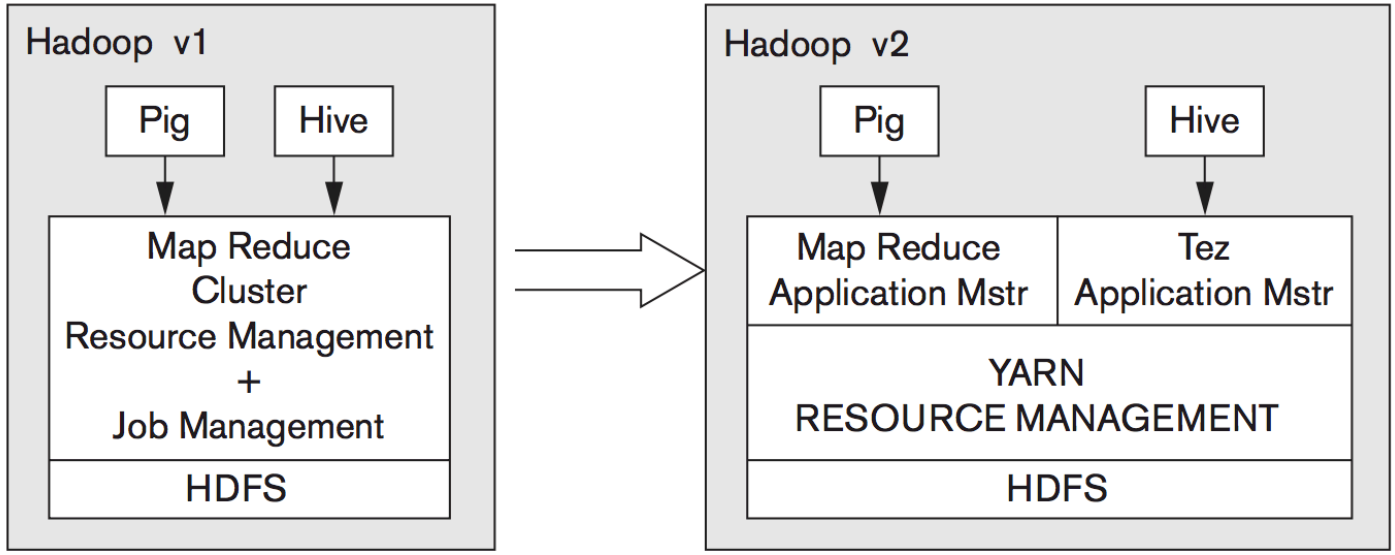


Figure 25.3
The Hadoop v1 vs. Hadoop v2 schematic.

Separation of Concerns

Improved resource utilization

Allow other applications to run on cluster

Zaharia et al. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. NSDI 2012

Shi et al. Clash of the Titans: MapReduce vs. Spark for Large Scale Data Analytics, VLDB 2015

What are the elements of the vision behind Spark?

What is the key feature introduced in Spark 2.0?

What are the elements of the vision behind Spark?

Functional High-level API to support data scientists workflows

Unified data processing

What is the key feature introduced in Spark 2.0?

Structured APIs

What technology is better?

	Parallel Databases	MapReduce
Structured Data		
Unstructured Data		
Fault-tolerance		
Query Expressiveness		
Simple Usage		
Support for Novel Applications		

Project 4

Use a real cluster environment (RCAC Scholar)

Practice with HDFS

Practice with Spark and Spark-SQL (possibly Spark-Streaming too!)