Introduction to Distributed Systems and MapReduce

Game Plan

- Big data - What is it, and why is it important?
- Distributed Systems Architecture
  - Distributed filesystems
  - Distributed processing
- Hadoop MapReduce

Objectives

- Understand the differences between local and distributed systems, and when to use each
- Describe the general structure of a distributed systems architecture / storage system
- Describe the storage mechanisms behind Distributed File Systems
- Understand the Hadoop MapReduce framework
  - What is the Map step?
  - What is the Reduce step?

Game Plan

- Big data - What is it, and why is it important?
- Distributed Systems Architecture
  - Distributed filesystems
  - Distributed processing with Hadoop MapReduce

BIG
DATA
ANALYTICS
STORAGE
TECHNOLOGIES
INFORMATION
USE
ONE
SINCE
SOFTWARE
UBIQUITOUS
REQUIRING
BUSINESS
MOVING
SAN
RADIO-FREQUENCY
COMPLEXITY
SOLID
WIRELESS
TOLERABLE
PARALLEL
SIZE
NEEDED
QUALITIES
PETABYTES
PROCESSING
LOGS
INTERNET
ORGANIZATIONS
USED
DISTRIBUTED
CAPTURE
MAY
MANAGEMENT
MANAGE
GROW
SOCIAL
LARGE
STORE
DEFINING
CASE
EVERY
LARGER
CURRENT
THOUGHT
ELAPSED
EXAMPLES
APPLIED
AMOUNT
DESCRIBING
SETS
ABILITY
TARGET
ARCHIVES
SENSOR
DIFFICULTY
INDEXING
DATABASES
COMPLEX
RESEARCH
MPP
TERABYTES
CONTINUES
COST
CITATION
RECORDS
CONNECTOMICS
OPPORTUNITIES
RECONSIDER
SEARCH
DEFINITION
BIOGEOCHEMICAL
NETWORKS
PROCESS
INCLUDE
SYSTEMS
WITHIN
TOOLS
COMPUTING
SET
GENOMICS
COMBAT
SIGNIFICANT
SHARED
TIME
RELATIONAL
ZETTABYTES
PERFORMANCE
DISK
ALSO
RELATED
HUNDREDS
CREATED
BIOLOGICAL
GARTNER
MASSIVELY
USING
TYPES
WORKING
DESKTOP
CURRENTLY
TENS
CAPACITY
FORMS
PRACTITIONERS
PRESENTATIONS
WORLD'S
NOW

Big data - What is it?

The three V’s:

Big data is **high volume**, **high velocity**, and/or **high variety** information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.

-Gartner Inc.

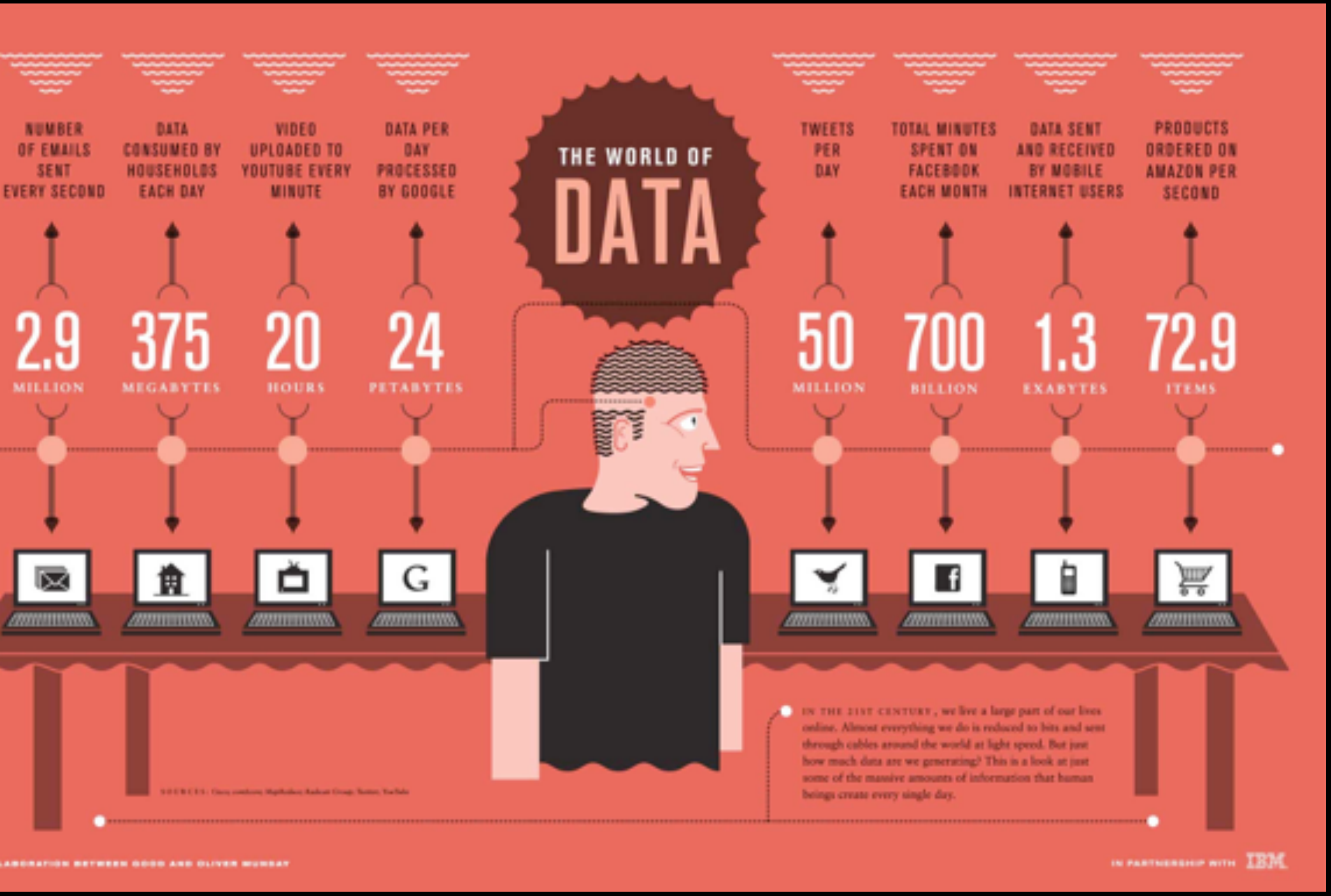
The Three V’s

- **High Volume -** Data so large that it can’t be managed by a single computer (scale out vs. scale up)
- **High Velocity -** Data coming in and going out too quickly to be processed by a single computer (think real-time processing real-fast)
- **High Variety -** Data in many different formats (largely unstructured: text, videos, log files, etc.)

How big is big?

Class	Size	Tools	Storage	Examples
Small	< 10 GB	R/Python	Fits in a single machine's memory	Thousands of sales figures
Medium	10GB - 1TB	R/Python with Indexed Files, Large Databases	Fits on one machine's disk	Millions of web pages
Large	> 1TB	Hadoop, Spark, Distributed Databases	Stored across multiple machines	Billions of web clicks

Why does this matter?


NUMBER OF EMAILS SENT EVERY SECOND
2.9 MILLION
DATA CONSUMED BY HOUSEHOLDS EACH DAY
375 MEGABYTES
VIDEO UPLOADED TO YOUTUBE EVERY MINUTE
20 HOURS
DATA PER DAY PROCESSED BY GOOGLE
24 PETABYTES
THE WORLD OF DATA
TWEETS PER DAY
50 MILLION
TOTAL MINUTES SPENT ON FACEBOOK EACH MONTH
700 BILLION
DATA SENT AND RECEIVED BY MOBILE INTERNET USERS
1.3 EXABYTES
PRODUCTS ORDERED ON AMAZON PER SECOND
72.9 ITEMS
IN THE 21ST CENTURY, we live a large part of our lives online. Almost everything we do is reduced to bits and sent through cables around the world at light speed. But just how much data are we generating? This is a look at just some of the massive amounts of information that human beings create every single day.
IN PARTNERSHIP WITH IBM

As more and more data is collected, the size of the average dataset will increase.

We need something that will allow us to store and process these increasingly large datasets…

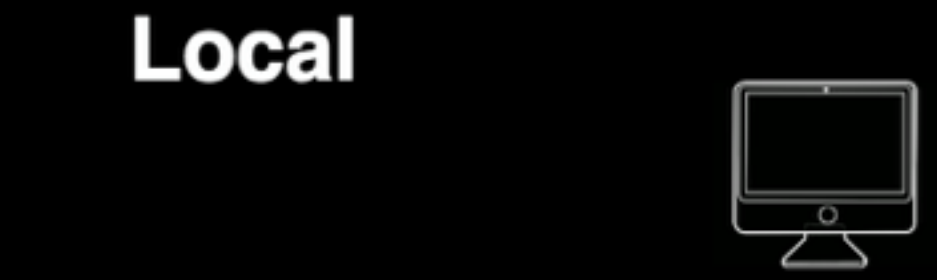
Game Plan

- Big data - What is it, and why is it important?
- Distributed Systems Architecture
    - Distributed filesystems
    - Distributed processing with Hadoop MapReduce

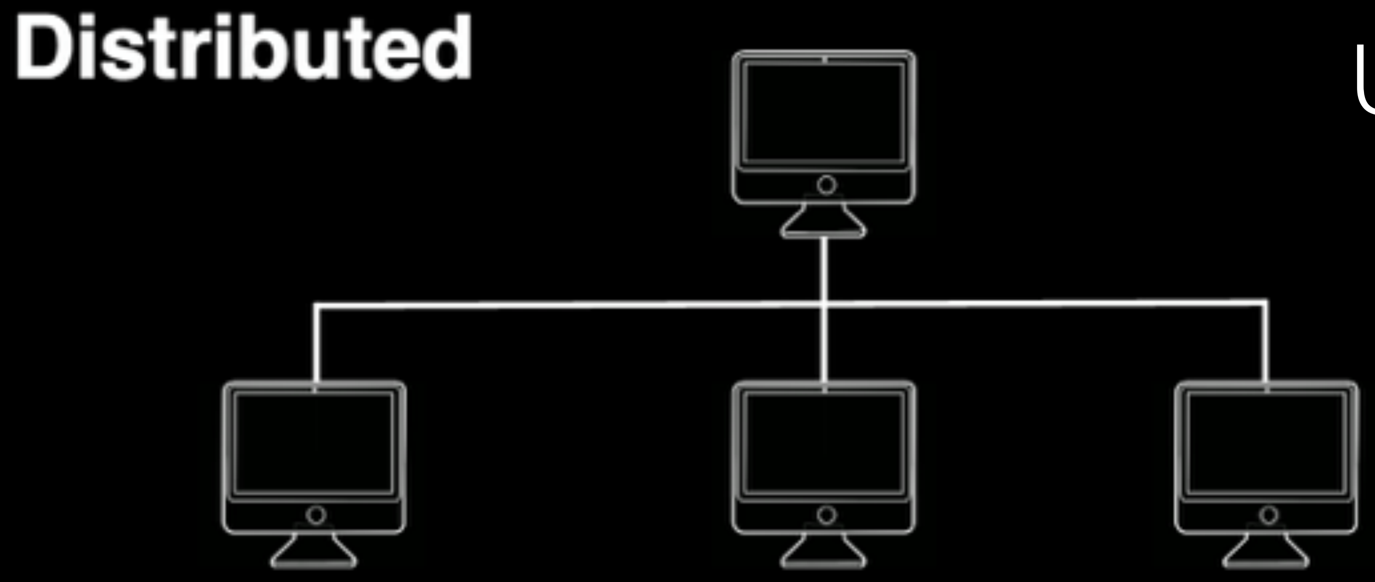
Objectives

- Understand the differences between local and distributed systems, and when to use each
- Describe the general structure of a distributed systems architecture / storage system
- Describe the storage mechanisms behind Distributed File Systems
- Understand the Hadoop MapReduce framework
  - What is the Map step?
  - What is the Reduce step?

Local Versus Distributed

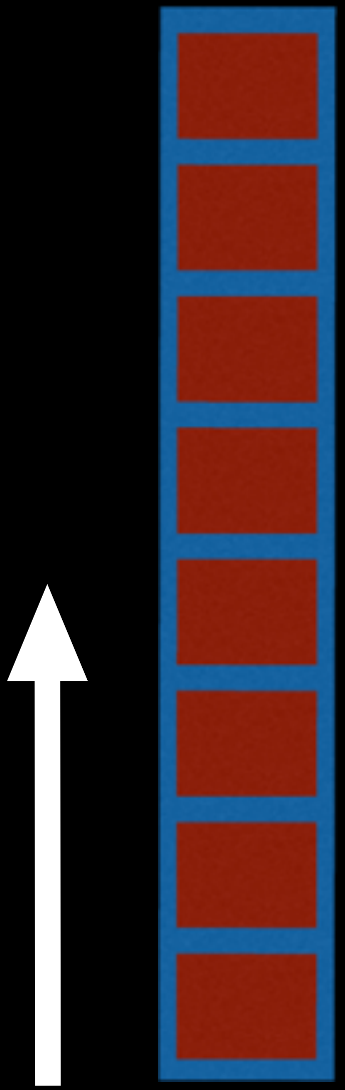


Uses the computational resources of 1 machine

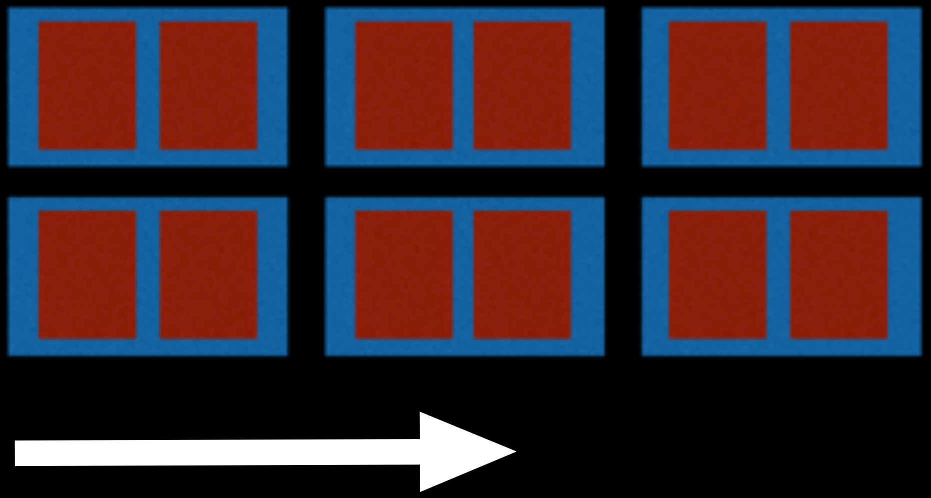


Uses the computational resources of many machines

Scaling Up Vs. Out



With a local system, we can only scale **up**



With distributed systems, we can scale **out**

Local Versus Distributed: Pros

- **Distributed:**
  - Better scalability - easier to add more machines than to add more cores to a machine
  - Fault tolerance - if one machine fails, the whole network is not down
- **Local:**
  - Simplistic - much less complex than distributed systems

Local or Distributed

- Practically speaking, stick with local processing unless you have a need to use distributed processing (see… the three V's, or something close to that)

- Throwing practicality aside for a second, use it if you want it on your resume, or to practice with it (especially the latter)

Game Plan

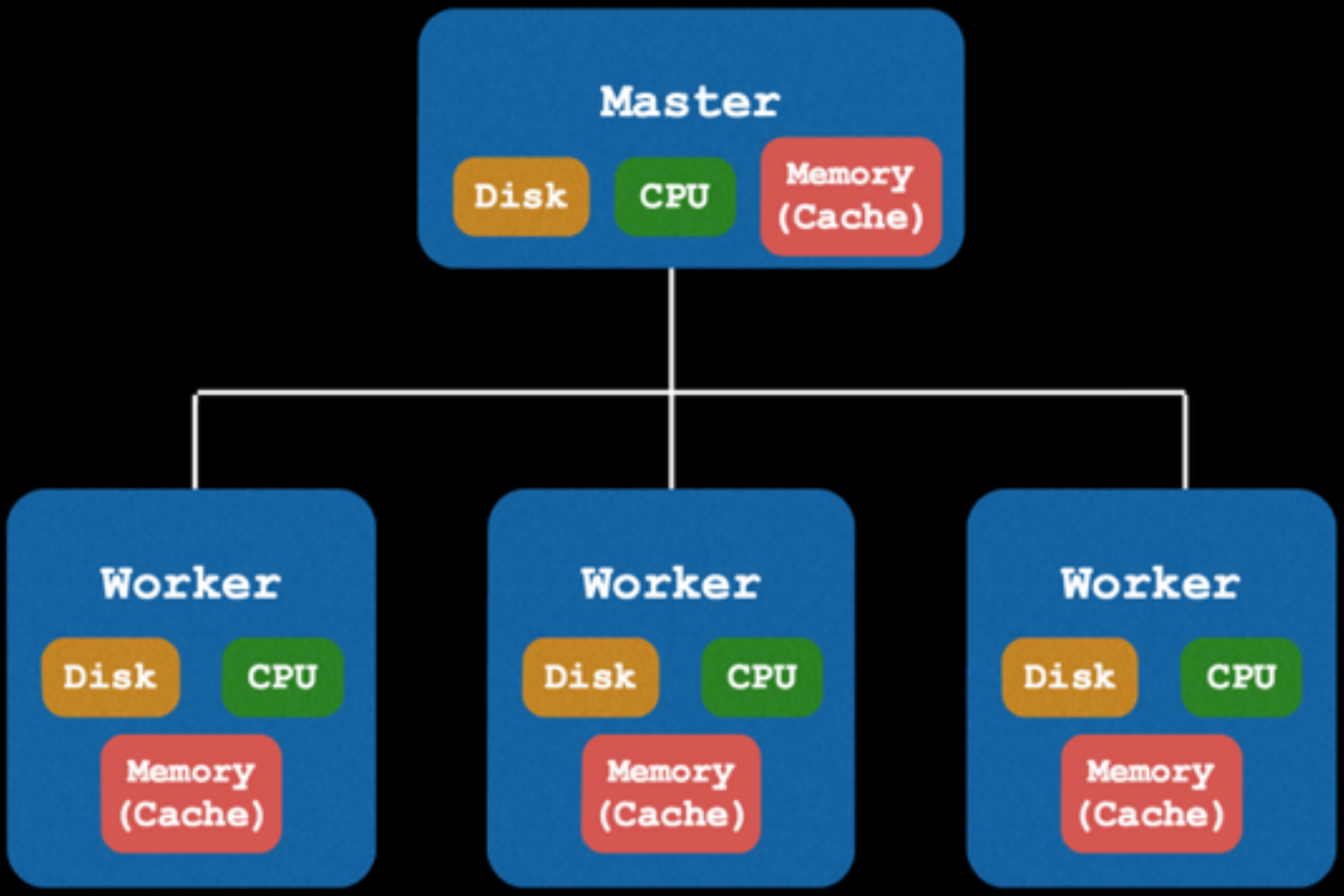
- Big data - What is it, and why is it important?
- Distributed Systems Architecture
  - Distributed filesystems
  - Distributed processing with Hadoop MapReduce

Objectives

- Understand the differences between local and distributed systems, and when to use each
- Describe the general structure of a distributed systems architecture / storage system
- Describe the storage mechanisms behind Distributed File Systems
- Understand the Hadoop MapReduce framework
  - What is the Map step?
  - What is the Reduce step?

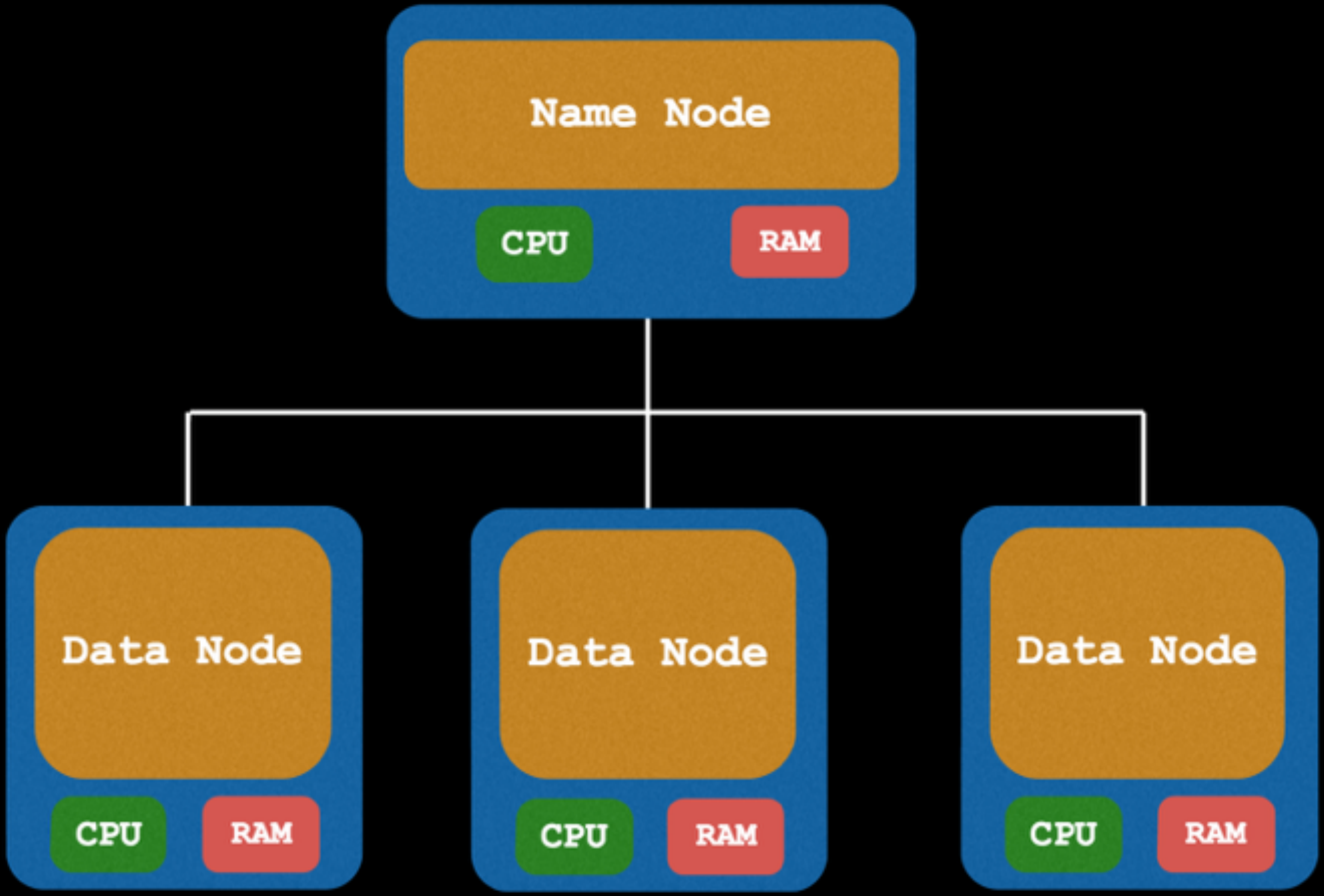
General Distributed Systems Architecture

Master Node



Worker Node **Worker Node** **Worker Node**

Distributed Storage - HDFS



Game Plan

- Big data - What is it, and why is it important?
- Distributed Systems Architecture
    - Distributed filesystems
    - Distributed processing with Hadoop MapReduce

Objectives

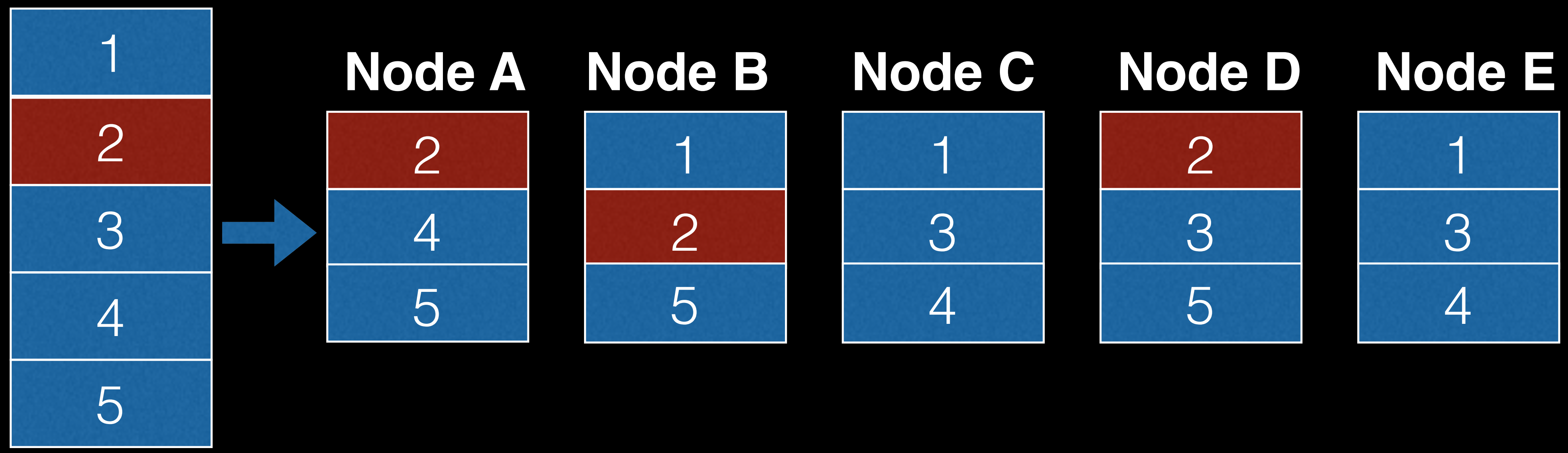
- Understand the differences between local and distributed systems, and when to use each
- Describe the general structure of a distributed systems architecture / storage system
- Describe the storage mechanisms behind Distributed File Systems
- Understand the Hadoop MapReduce framework
  - What is the Map step?
  - What is the Reduce step?

How does Distributed Storage Work?

- Individual files are broken up into blocks
- A given number of replicas of each block are stored across the cluster (on **data nodes)**
- If any of those replicas is lost, the **name node** replicates one of the remaining copies so that the replication factor remains the same

How does Distributed Storage Work?

HDFS Data Distribution



Input File

- Individual files are broken up into blocks (here 5)
- A given number of replicas (here 3) of each block are stored across the cluster (on **data nodes)**

HDFS Defaults

- There are customizable defaults for everything in HDFS
    - Individual files are broken up into 128 MB blocks
    - 3 replicas of each block are stored across the cluster (on **data nodes)**

Game Plan

- Big data - What is it, and why is it important?
- Distributed Systems Architecture
  - Distributed filesystems
  - Distributed processing with Hadoop MapReduce

Objectives

- Understand the differences between local and distributed systems, and when to use each
- Describe the general structure of a distributed systems architecture / storage system
- Describe the storage mechanisms behind Distributed File Systems
- Understand the Hadoop MapReduce framework
  - What is the Map step?
  - What is the Reduce step?

Distributed Processing Motivation

When working with distributed storage, we need to solve the following problem:

How do I process this data?

There are two options:

1. **Move the data to the code (and processing power)**
2. **Move the code to the data (MapReduce)**

We need this little guy…

Hadoop MapReduce handles a lot of the details of distributed computing for us and does most of the heavy lifting around:

- Parallelization & Distribution (input splits)
- Partitioning (shuffle & sort)
- Fault-tolerance
- Resource Management
- Status and monitoring

Hadoop Map-Reduce

Divide & Conquer

1. Split task into many subtasks
2. Solve these tasks independently
3. Recombine the subtasks into a final result

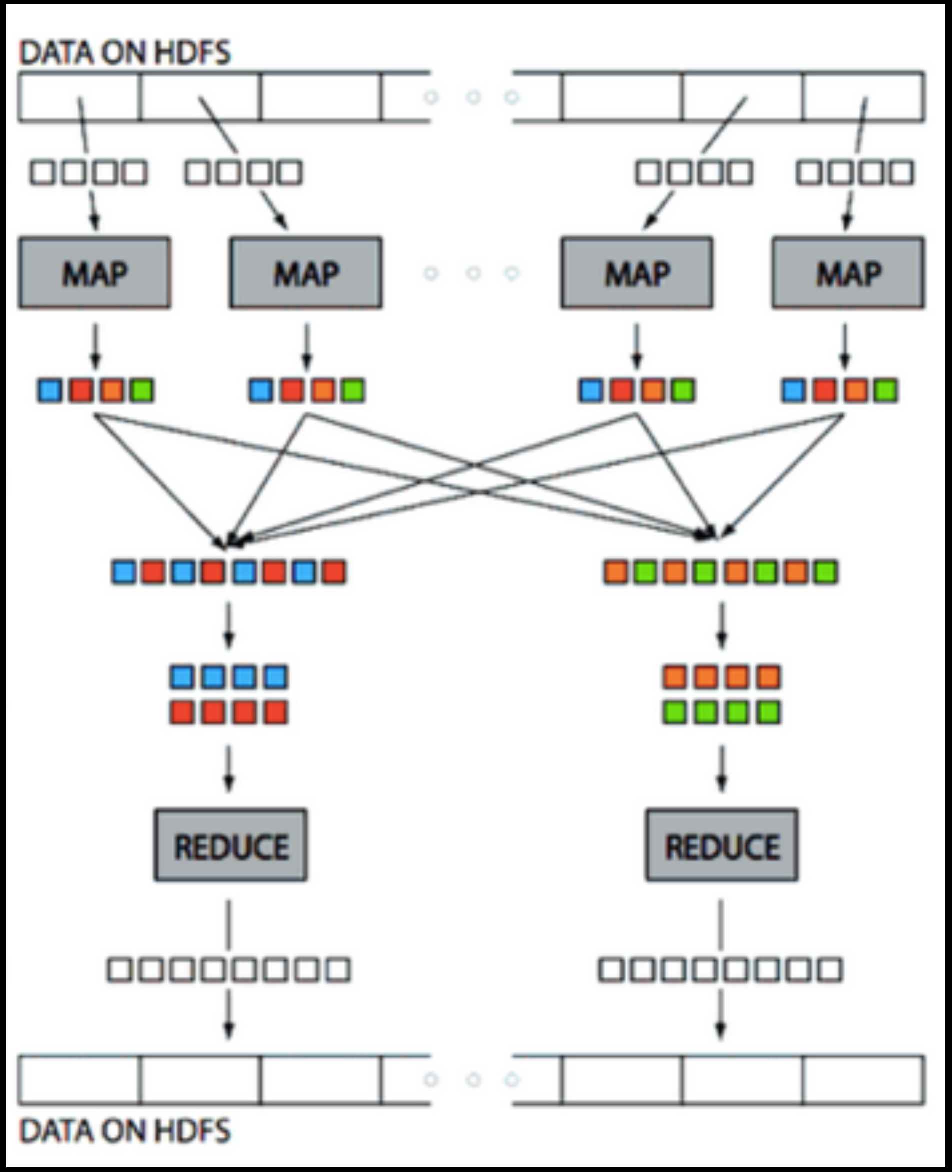
Divide & Conquer

- MapReduce splits a problem into subtasks to be processed in parallel
- This happens in two steps:
  1. The **map** step
  2. The **reduce** step
- All processing happens on the data nodes

  (which remember, are just computers)

MapReduce Framework

Input data - stored on HDFS data nodes

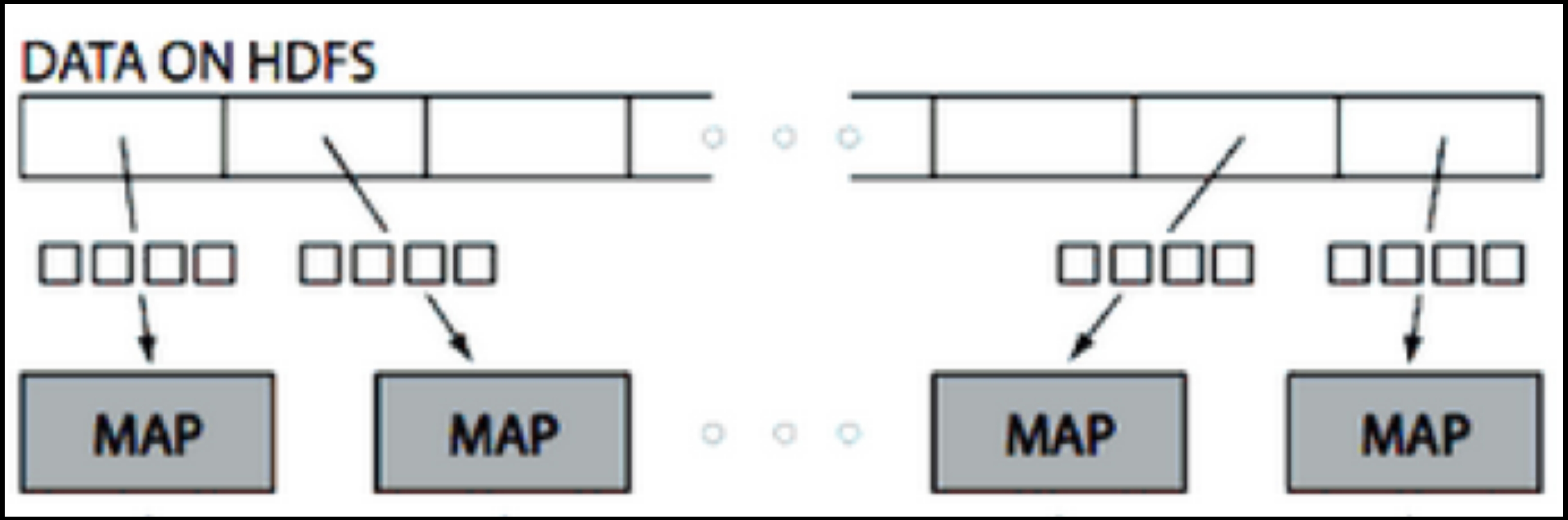


Map step

Reduce step

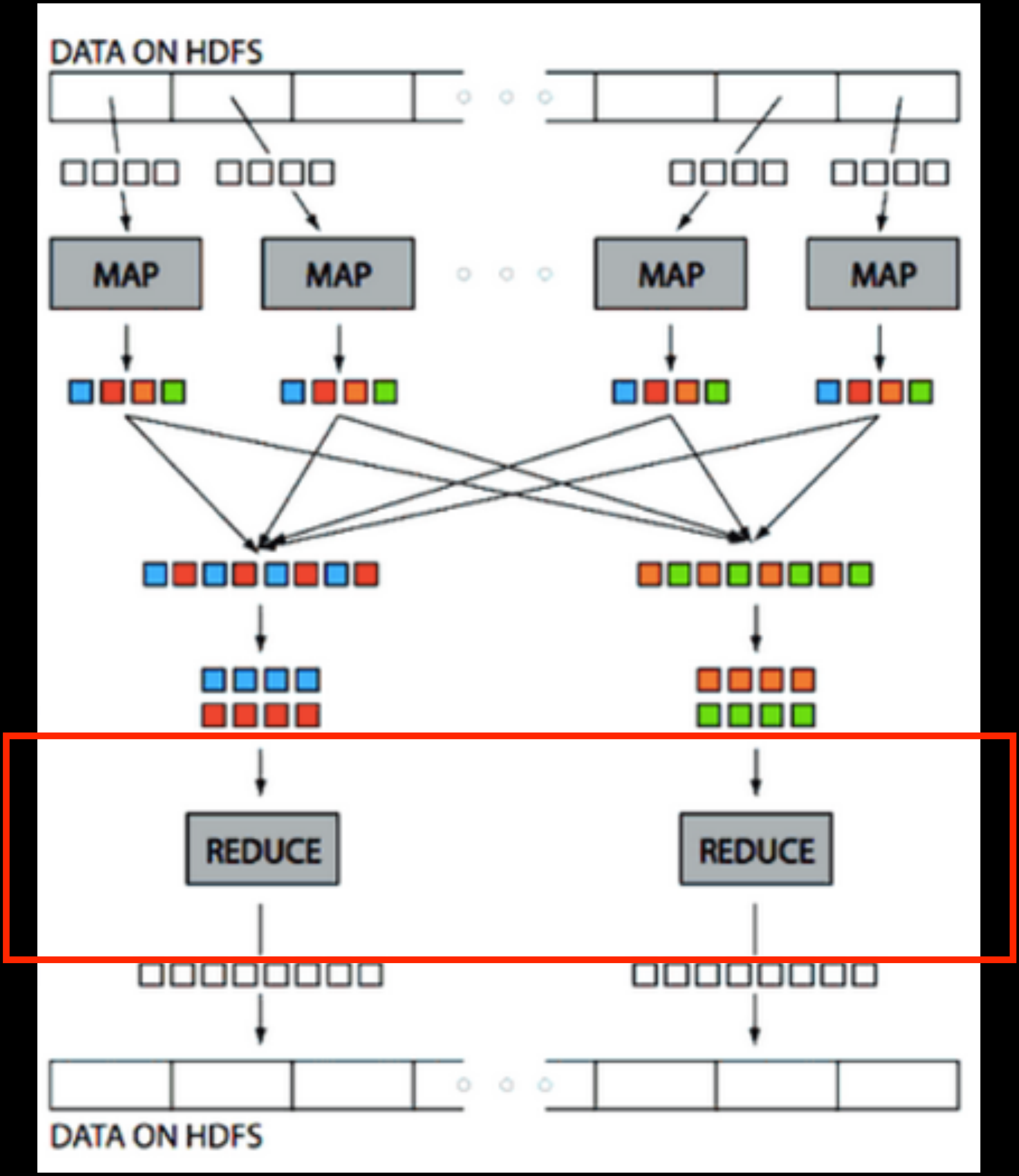
Output data - stored on HDFS data nodes

Map Step



1. Data is read in as **key-value** pairs
2. Data is **filtered & transformed**
3. Data is passed on to the reduce step as **key-value** pairs

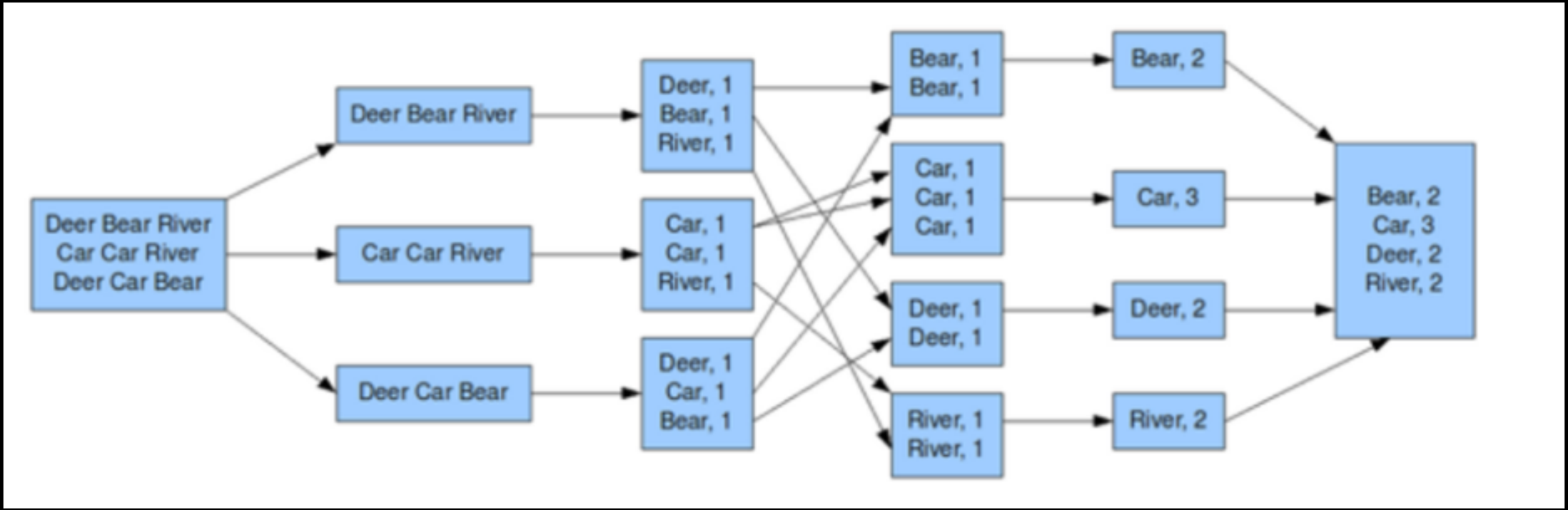
Reduce Step



1. Data is read in as **key-value** pairs, where every value for a given key has been aggregated into a list (i.e. all the records with the same key end up on the same reducer).
2. All values in the value list are combined (reduced) in some way.
3. Data is output as **key-value** pairs, stored in multiple files.

Word Count Example

Inputs **Map Step** **Reduce Step** **Outputs**

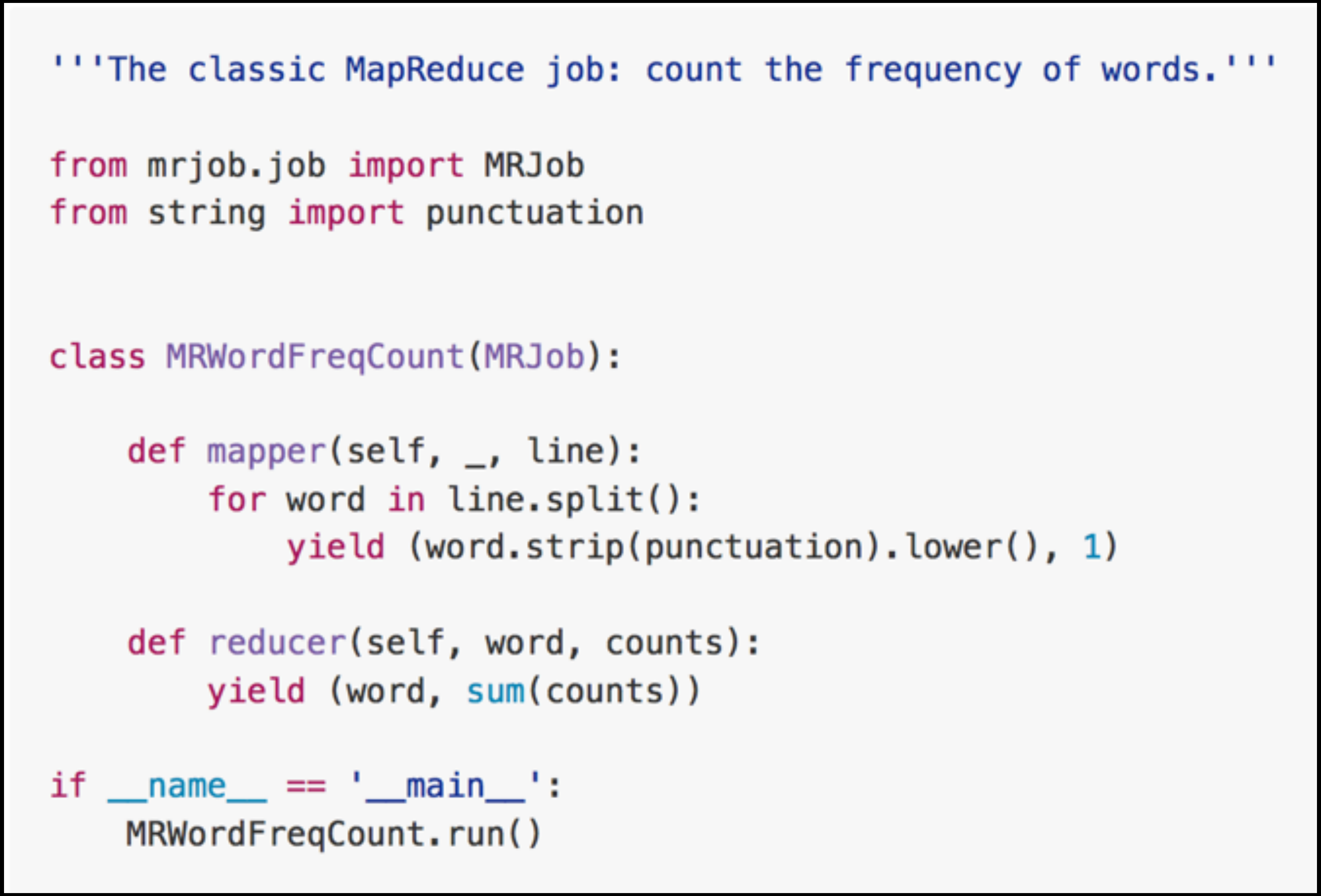


River, 1
River, 1

is read in as

River, [1, 1]

Word Count Code



```python
'''The classic MapReduce job: count the frequency of words.'''

from mrjob.job import MRJob
from string import punctuation


class MRWordFreqCount(MRJob):

    def mapper(self, _, line):
        for word in line.split():
            yield (word.strip(punctuation).lower(), 1)

    def reducer(self, word, counts):
        yield (word, sum(counts))

if __name__ == '__main__':
    MRWordFreqCount.run()
```

Game Plan

- Big data - What is it, and why is it important?
- Distributed Systems Architecture
  - Distributed filesystems
  - Distributed processing with Hadoop MapReduce

Objectives

- Understand the differences between local and distributed systems, and when to use each
- Describe the general structure of a distributed systems architecture / storage system
- Describe the storage mechanisms behind Distributed File Systems
- Understand the Hadoop MapReduce framework
  - What is the Map step?
  - What is the Reduce step?