شرکت کلان داده پویان نوین
آناکاو

Big Data

Mahdi Samani

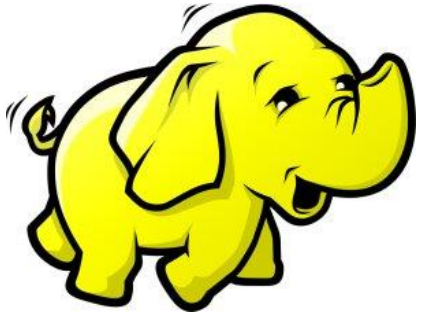
- ✓ CTO at anacav
- ✓ M.Sc at IUT

Anacav

IUT

Agenda

- What is Big Data?
- Big Data Characteristics
- Comparison between Hadoop and RDBMS
- Hadoop History and Timeline
- What is Hadoop?
- HDFS and Map Reduce
- Exercise

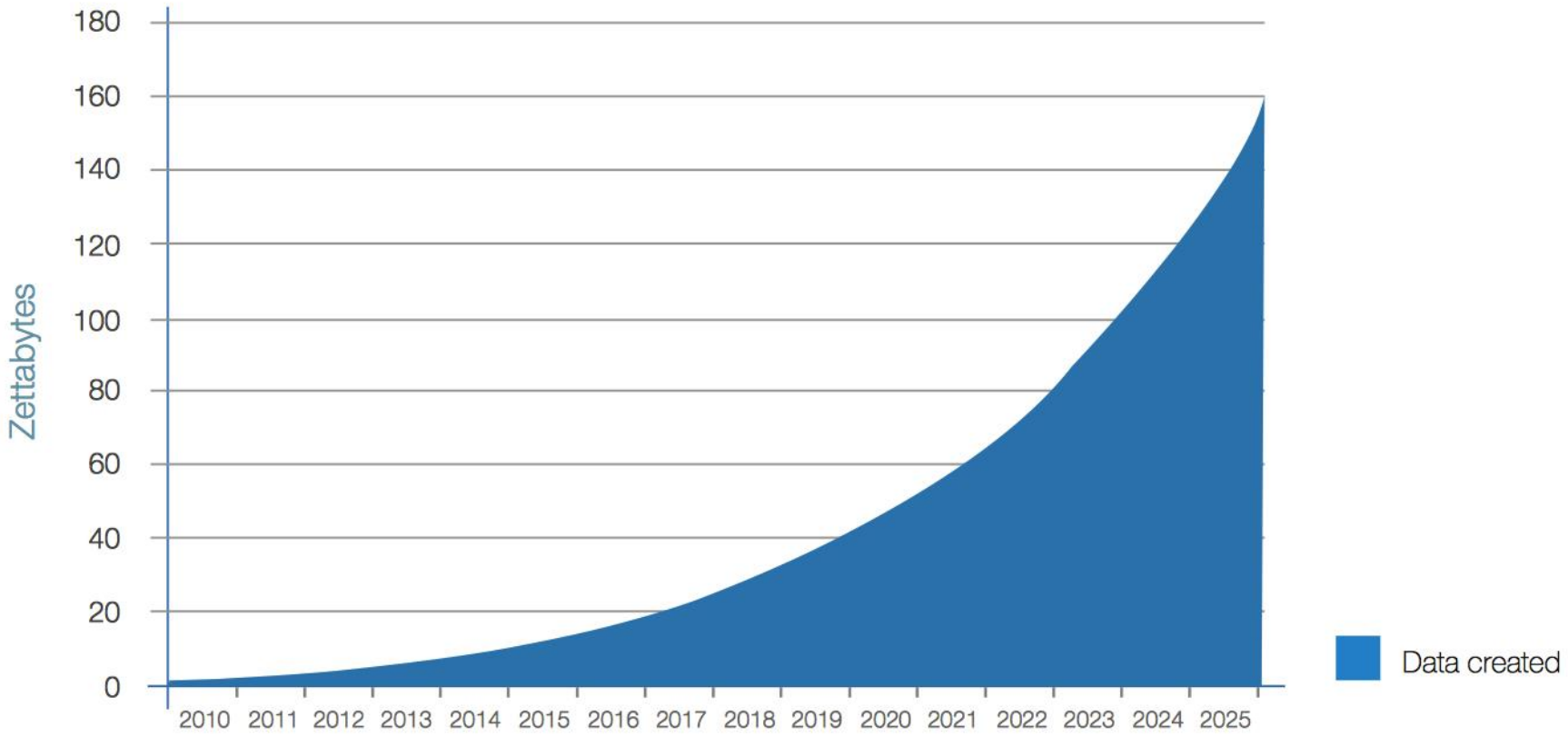


Big Data

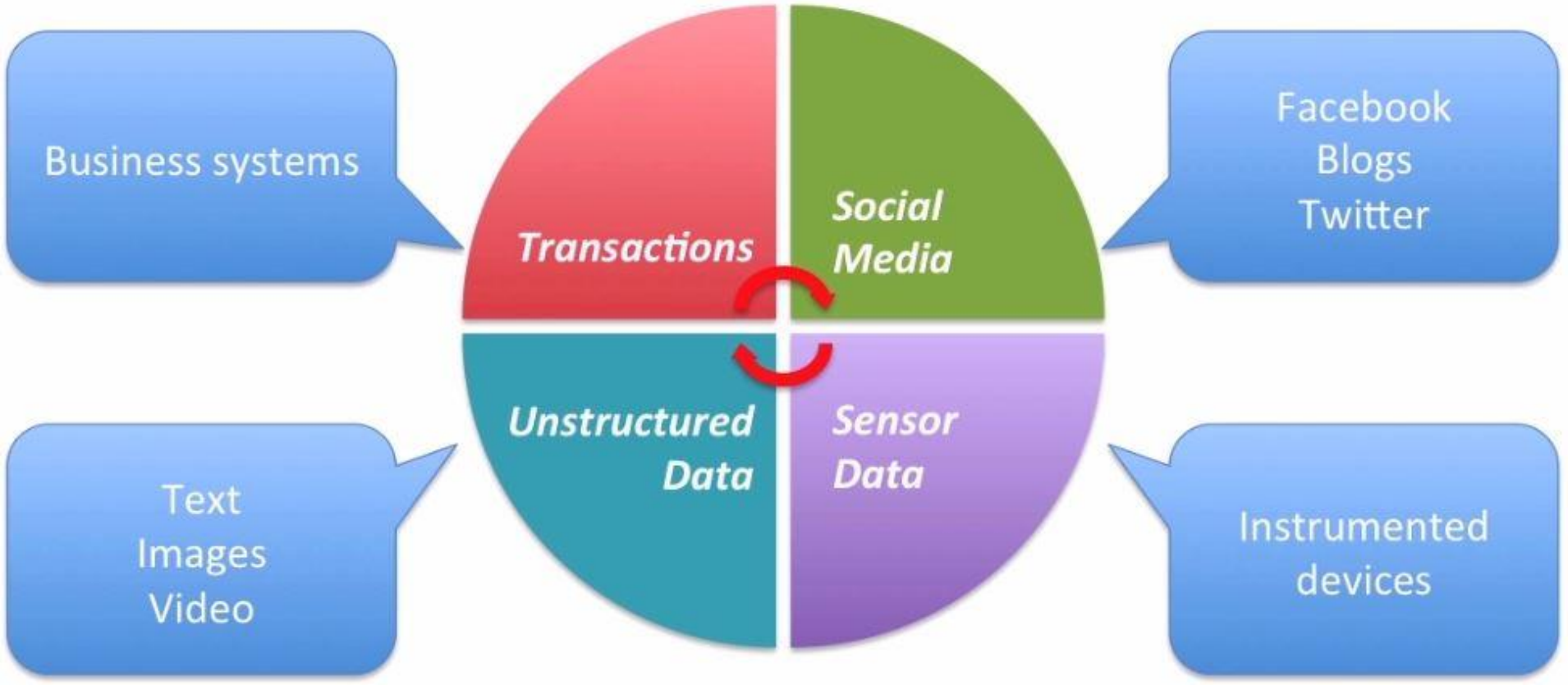
- Look at the scale!

- Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture , curation , storage, search, sharing, transfer, analysis, and visualization." Cited from Wikipedia

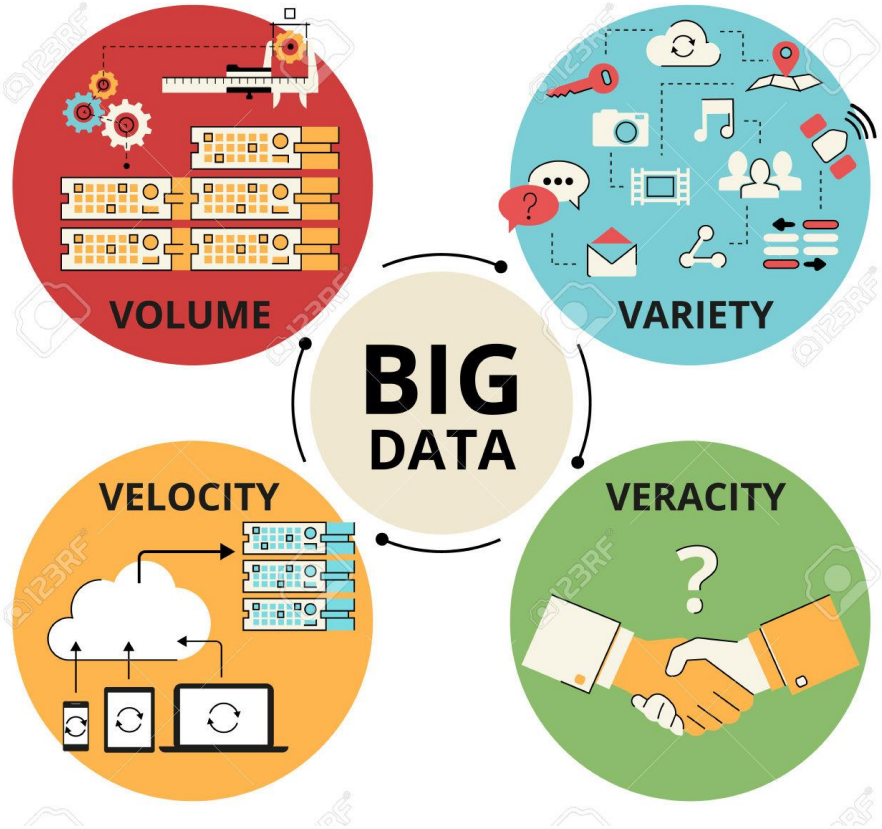
Big Data


Zettabytes
180
160
140
120
100
80
60
40
20
0
2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025
Data created

Big Data Sources



Big Data Characteristics



Hadoop VS RDBMS

RDBMS

1. Gigabytes Data size
2. Interactive and batch Access
3. Read and Write many times
4. Static schema
5. Nonlinear Scaling

HADOOP

1. Petabytes Data size
2. Batch Access
3. Write once ,read many times
   - Read and Write many times (HBase)
4. Dynamic schema
5. Linear Scaling

Hadoop!

Ask bigger questions.

Timeline

- Dec 2004: Dean/Ghemawat (Google) MapReduce paper

- 2005: Doug Cutting and Mike Cafarella (Yahoo) create Hadoop, at first only to extend Nutch

Timeline

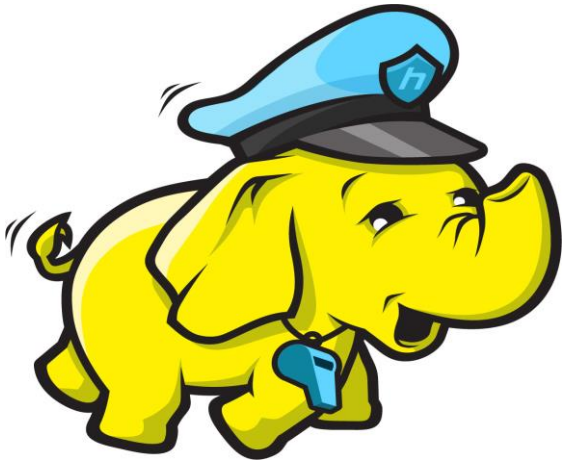
- 2006: Yahoo runs Hadoop on 5-20 nodes
- March 2008: Cloudera founded
- July 2008: Hadoop wins TeraByte sort benchmark (1st time a Java program won this competition)
- April 2009: Amazon introduce “Elastic MapReduce” as a service on S3/EC2
- June 2011: Hortonworks founded

Timeline

- 27 dec 2011: Apache Hadoop release 1.0.0
- June 2012: Facebook claim “biggest Hadoop cluster”, totalling more than 100 PetaBytes in HDFS
- 2013: Yahoo runs Hadoop on 42,000 nodes, computing about 500,000 MapReduce jobs per day
- 15 oct 2013: Apache Hadoop release 2.2.0 (YARN)

Big Data Challenge

- Problem:
  - Servers can fail
  - Data should not be lost!
- The idea:
  - Distributed, reliable storage
- Solution:
  - Hadoop Distributed File System (HDFS)



What is hadoop?

- **Open sourced**, **flexible** and **available** architecture for **large scale** computation and data processing on a network of **commodity hardware**
- Inspired by Google

Hadoop amazing name!

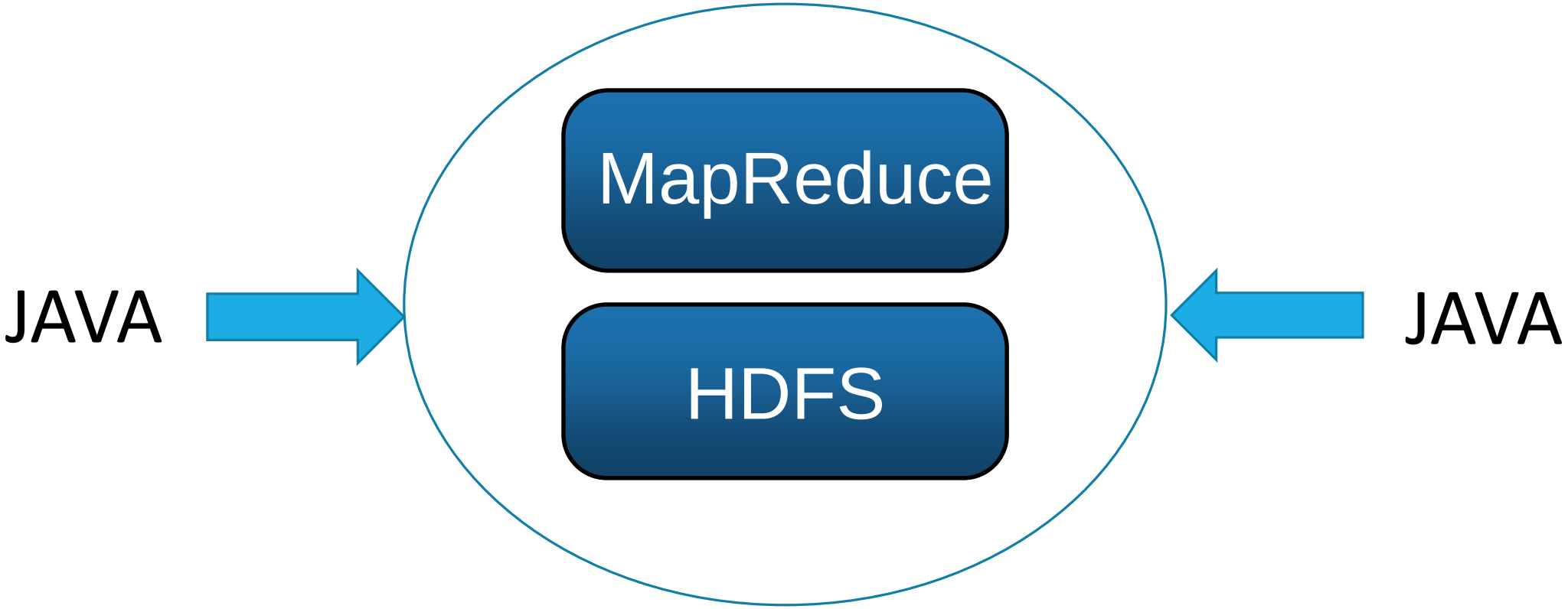
Who uses Hadoop?

Hadoop core

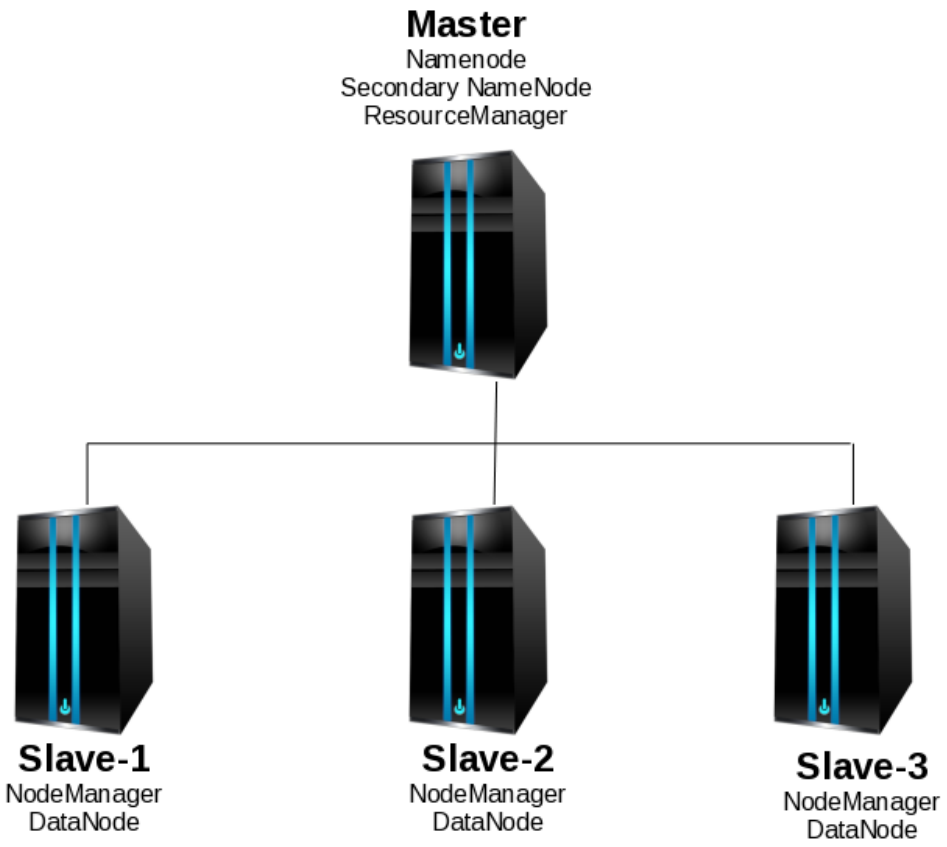
MapReduce

HDFS

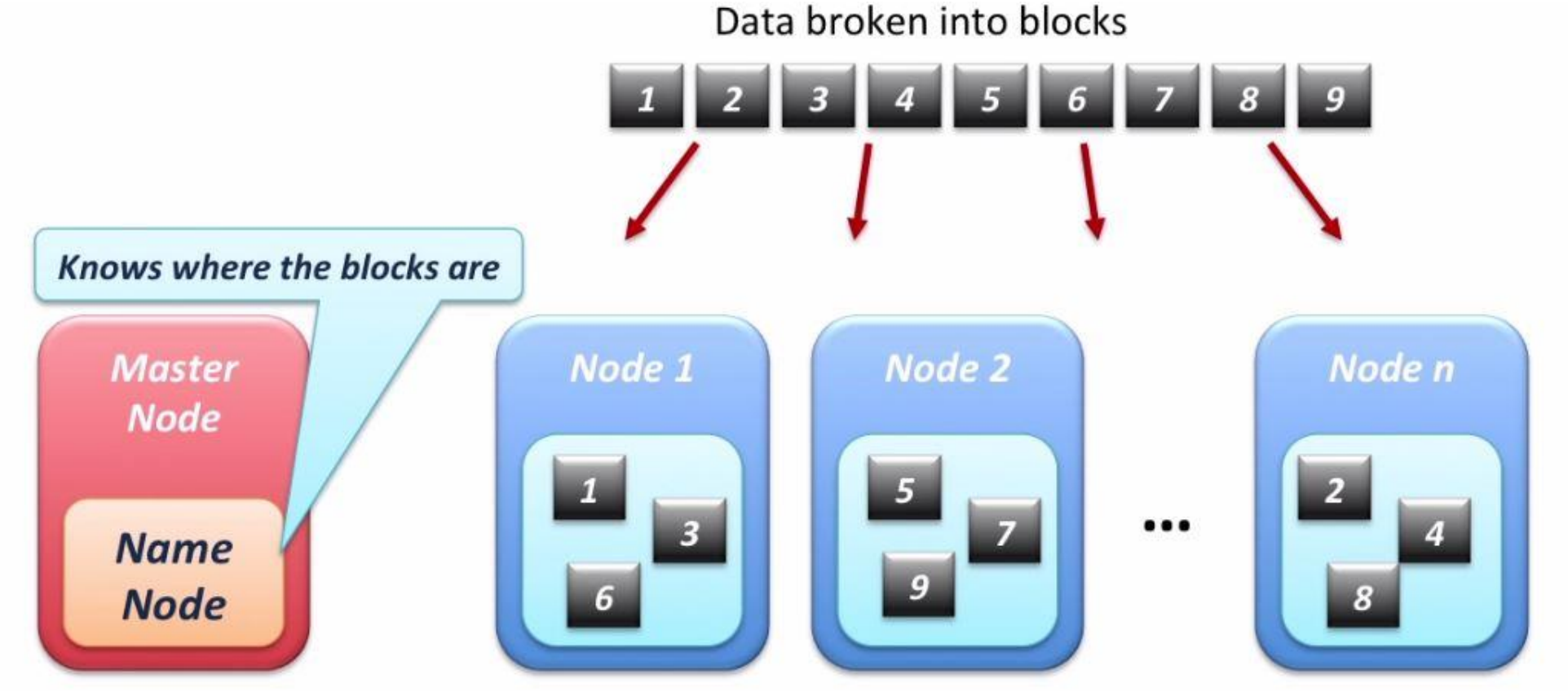
Hadoop core



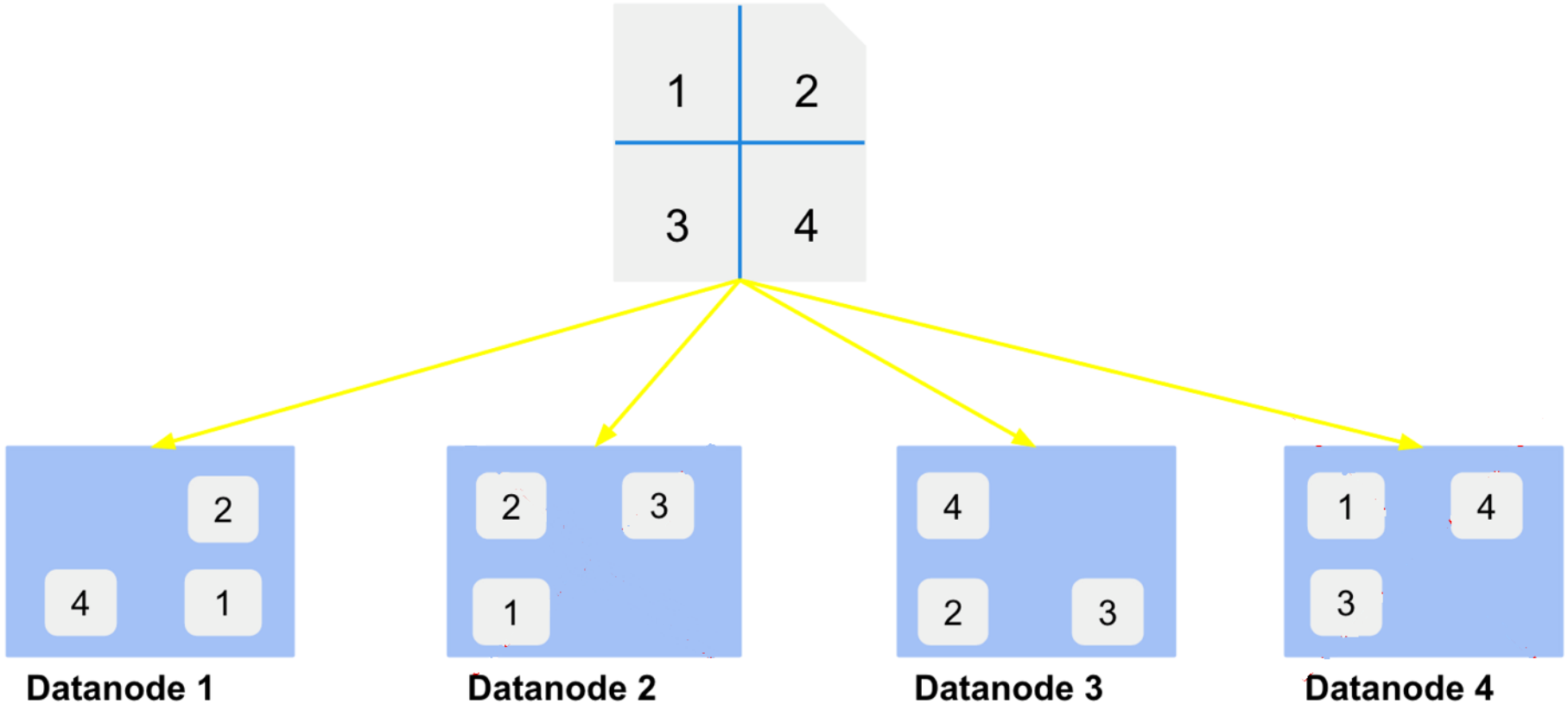
Hadoop architecture



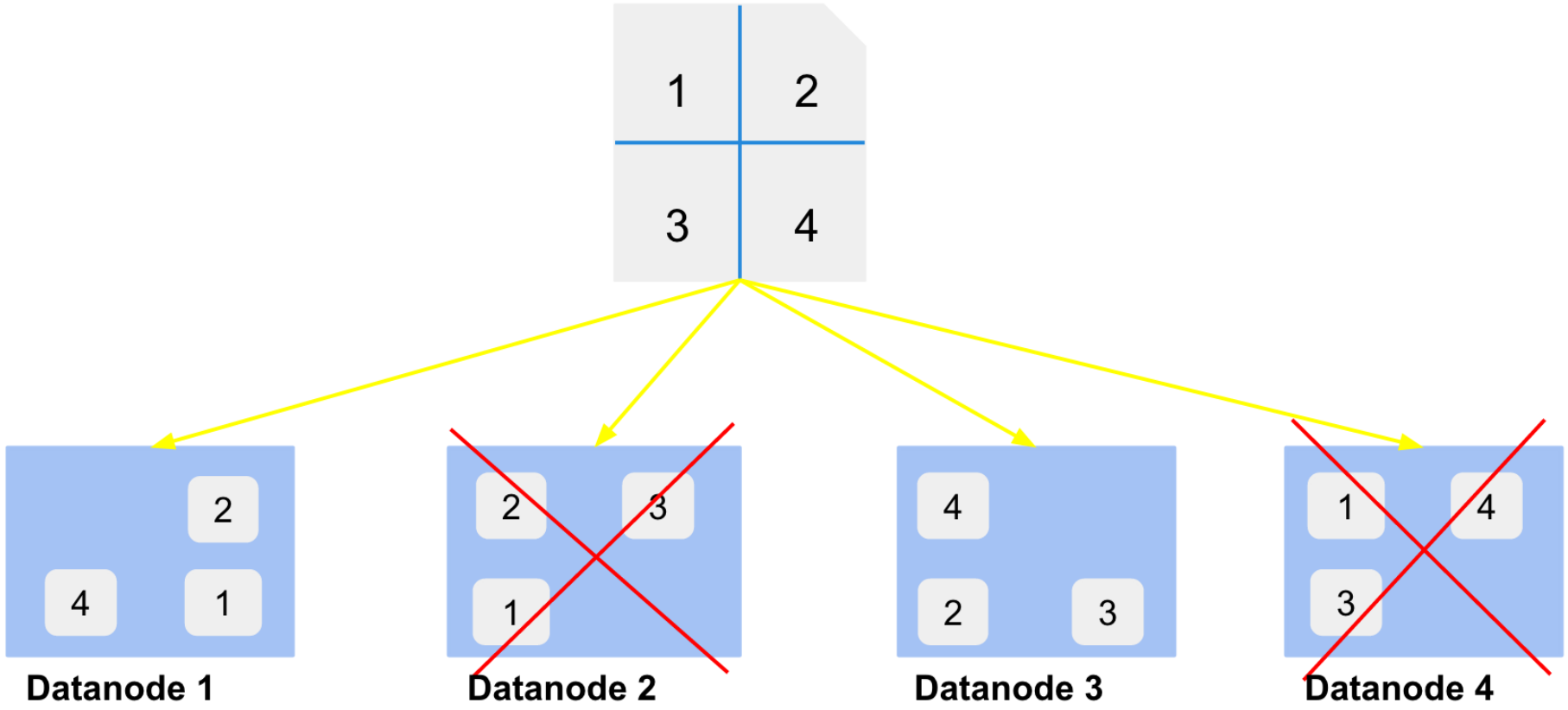
Hadoop Distributed File System



HDFS Replication



HDFS Replication



Hadoop Distributed File System

Data is :

- Split into blocks
- Distributed across the machines in the cluster
- Each block is replicated **3** times

- Machines in the cluster are cheap and unreliable
- HDFS resides on top of a native file system
- Block size is typically 64 or 128 MB
- Follows the idea of the Google File System (GFS)

HDFS Features

- Smaller number of large files
  - Files typically > 100 MB
- Ideal applications read the data from the beginning to the end
- Files are typically not updated
- No random access
- Default replication : 3

Map Reduce

- Is a framework for performing high performance distributed data processing using the divide and aggregate programming paradigm.

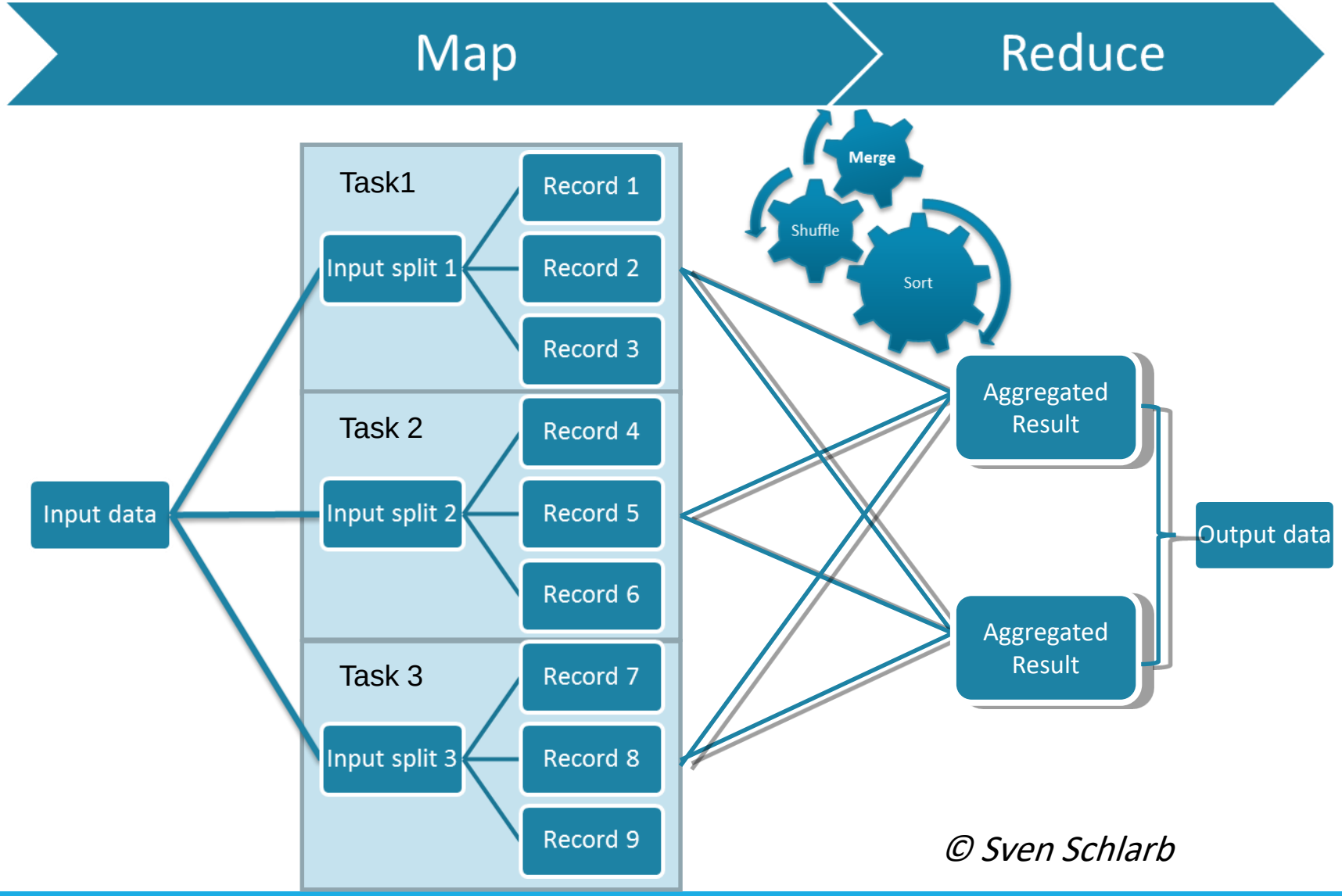
Simple example

- **Goal**
  - Count the number of each word

- **Map Function**
  - It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair)

- **Reduce Function**
  - Takes the output from Map as an input and combines those data tuples into a smaller set of tuples

MapReduce in a nutshell



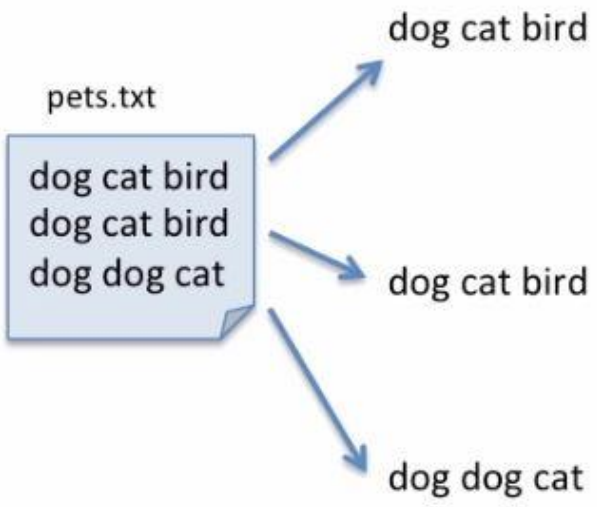
© Sven Schlarb

Word count example

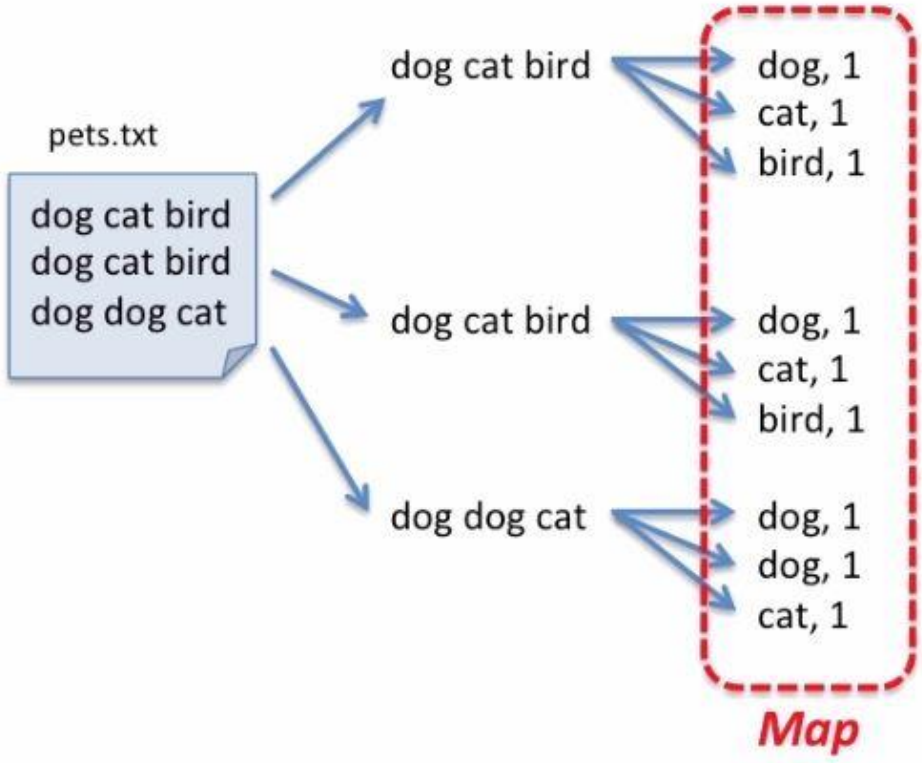
pets.txt

```
dog cat bird
dog cat bird
dog dog cat
```

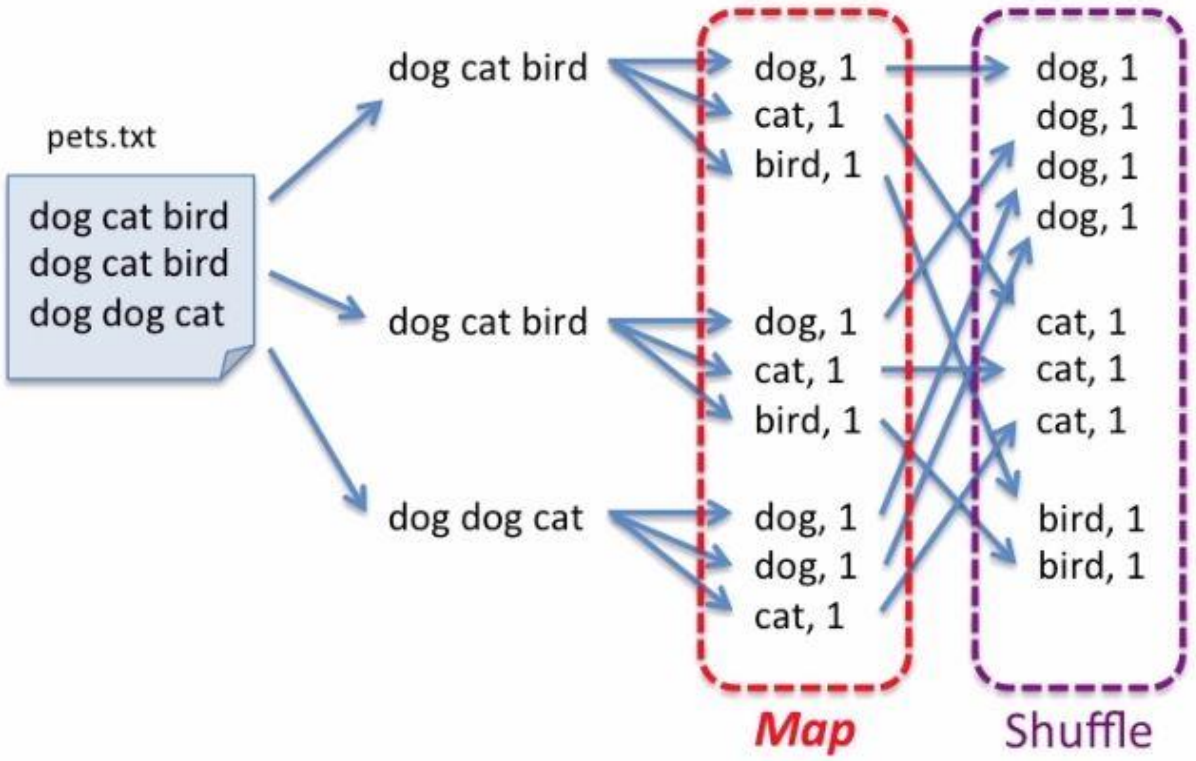
Word count example



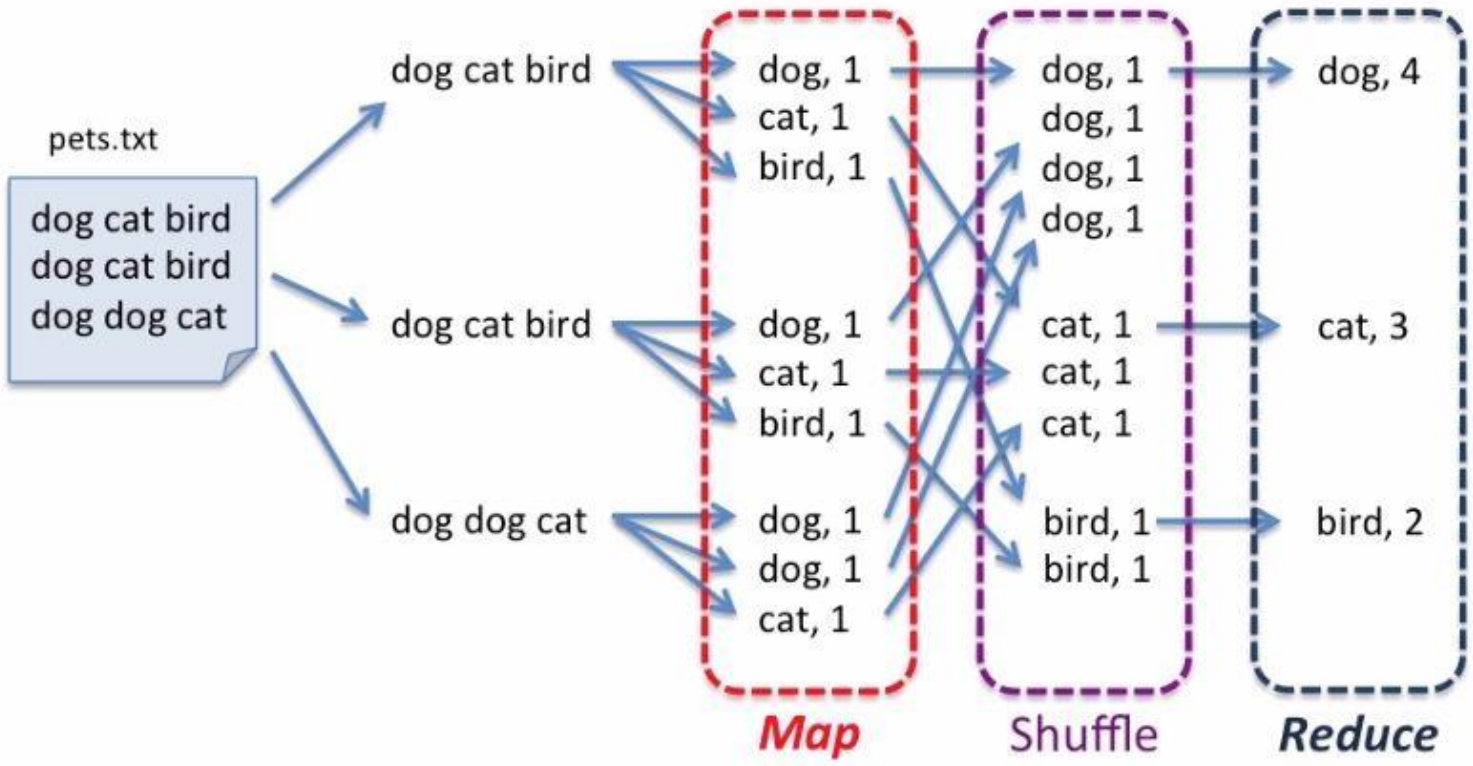
Word count example



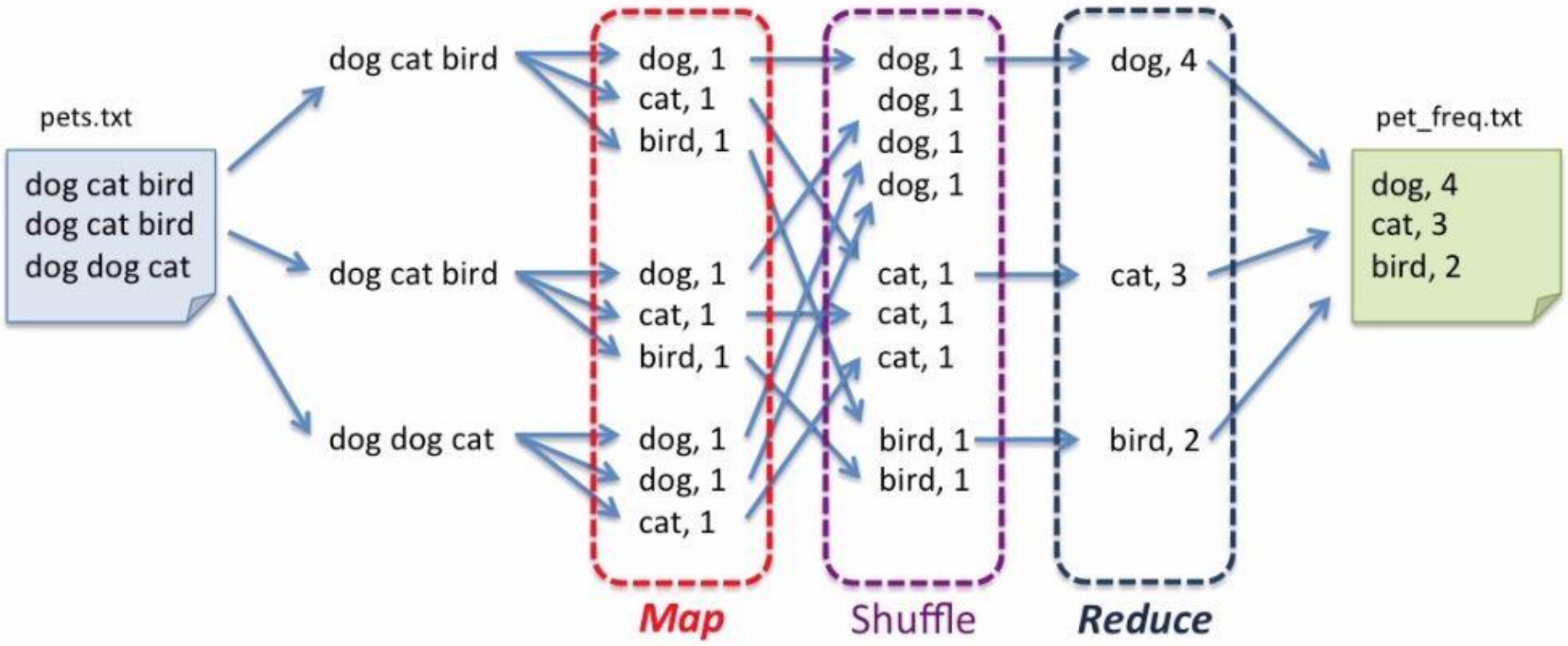
Word count example



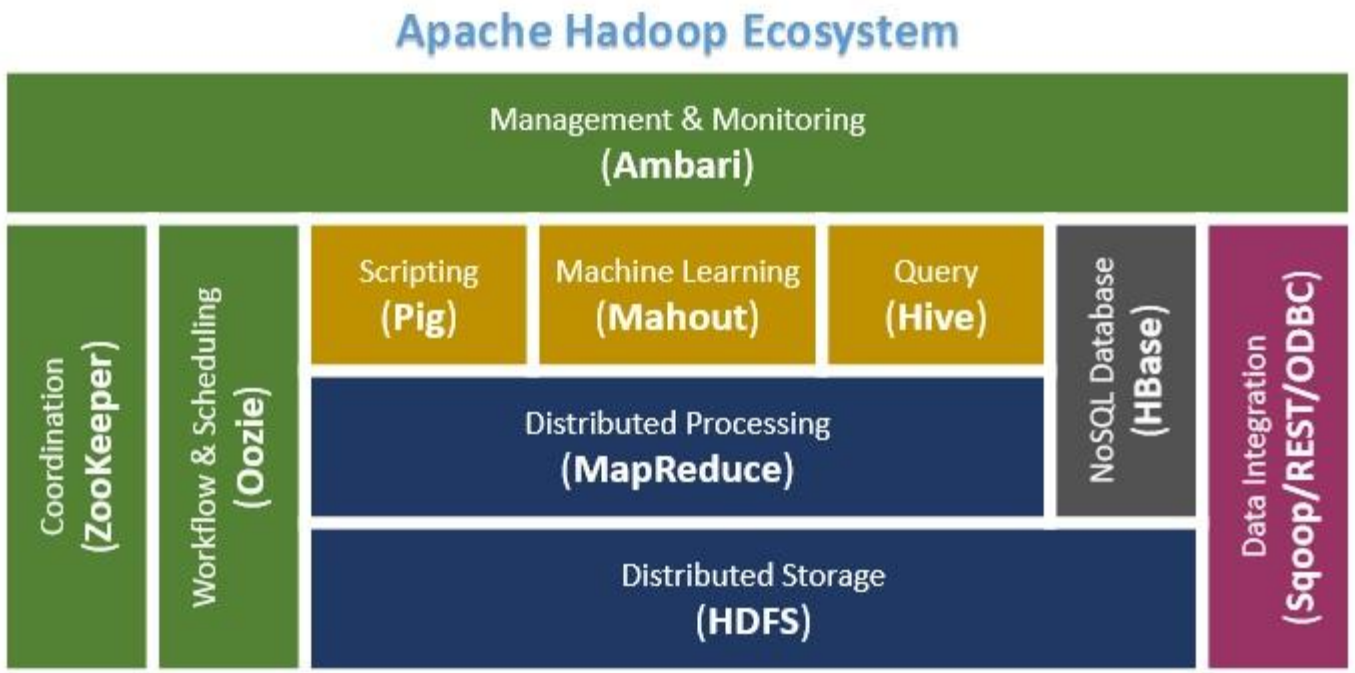
Word count example



Word count example



Hadoop Ecosystem



Tools and Libraries

- Apache Mahout
  - a machine learning library designed to run on data stored in Hadoop
- GIS Tools
  - a set of tools to help manage geographical components of your data
- Apache Hbase
  - a table-oriented database built on top of Hadoop
- Apache Hive
  - a data warehouse built on top of Hadoop that makes data accessible through an SQL-like language
- Apache Ambari
  - a software package for managing and monitoring Hadoop clusters

Tools and Libraries

- Apache Sqoop
  - a tool for transferring data between Hadoop and other data stores
- ZooKeeper
  - a tool for configuring and synchronizing Hadoop clusters
- Apache Flume
  - a system for collecting log data using HDFS
- Apache Spark
  - a new way to run algorithms even faster on Hadoop data

Install hadoop

- Installing Java on Master and Slaves
- Setting up a Hadoop User (hduser)
- Setup passwordless SSH for communication between nodes
- Download and Install Hadoop binaries on Master and Slave nodes

Configuration

- Setup Hadoop Environment on Master and Slave Nodes
- Update hadoop-env.sh on Master and Slave Nodes
- Core-site.xml (Master and Slave nodes)
- Mapred-site.xml (Master node only)
- Hdfs-site.xml (Master and Slave Nodes)
- Yarn-site.xml (Master and Slave Nodes)
- Slaves (Master node only)

Mapper

```
public class MapClass extends Mapper{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    @Override
    protected void map(LongWritable key, Text value,
            Context context)
            throws IOException, InterruptedException {

        //Get the text and tokenize the word using space as separator.
        String line = value.toString();
        StringTokenizer st = new StringTokenizer(line," ");

        //For each token aka word, write a key value pair with
        //word and 1 as value to context
        while(st.hasMoreTokens()){
            word.set(st.nextToken());
            context.write(word,one);
        }

    }
}
```

Reducer

```
public class ReduceClass extends Reducer{

    @Override
    protected void reduce(Text key, Iterable values,
            Context context)
            throws IOException, InterruptedException {

        int sum = 0;
        Iterator valuesIt = values.iterator();

        //For each key value pair, get the value and adds to the sum
        //to get the total occurances of a word
        while(valuesIt.hasNext()){
            sum = sum + valuesIt.next().get();
        }

        //Writes the word and total occurances as key-value pair to the context
        context.write(key, new IntWritable(sum));
    }
}
```

Driver Class

```
public class WordCount extends Configured implements Tool{

    public static void main(String[] args) throws Exception{
        int exitCode = ToolRunner.run(new WordCount(), args);
        System.exit(exitCode);
    }

    public int run(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.printf("Usage: %s needs two arguments, input and output
files\n", getClass().getSimpleName());
            return -1;
        }

        //Create a new Jar and set the driver class(this class) as the main class of jar
        Job job = new Job();
        job.setJarByClass(WordCount.class);
        job.setJobName("WordCounter");

        //Set the input and the output path from the arguments
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
```

Driver Class

```
        //Set the map and reduce classes in the job
        job.setMapperClass(MapClass.class);
        job.setReducerClass(ReduceClass.class);

        //Run the job and wait for its completion
        int returnValue = job.waitForCompletion(true) ? 0:1;

        if(job.isSuccessful()) {
            System.out.println("Job was successful");
        } else if(!job.isSuccessful()) {
            System.out.println("Job was not successful");
        }

        return returnValue;
    }
}
```

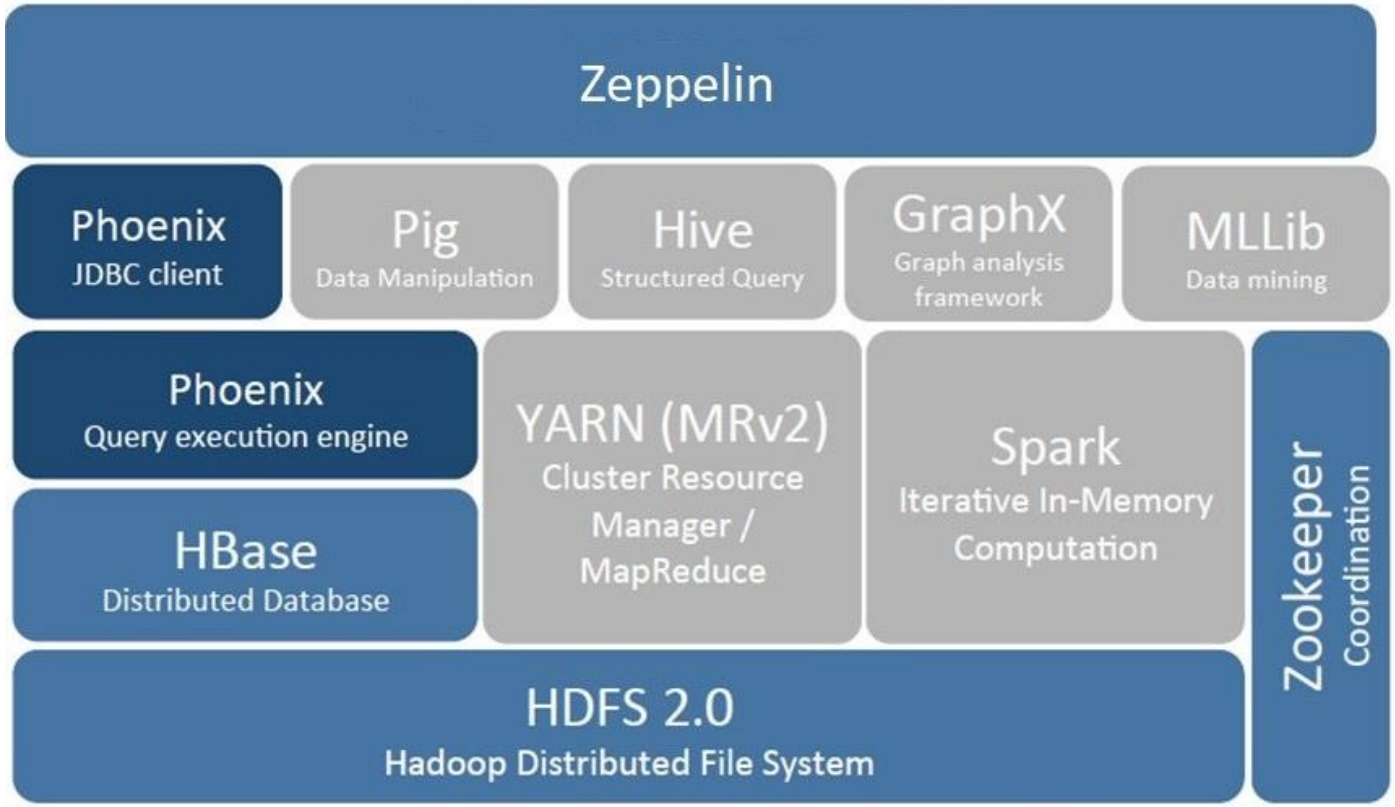
Apache Spark

- A unified analytics engine for large-scale data processing
- 100x faster than Hadoop

- Word count in 3 lines of code

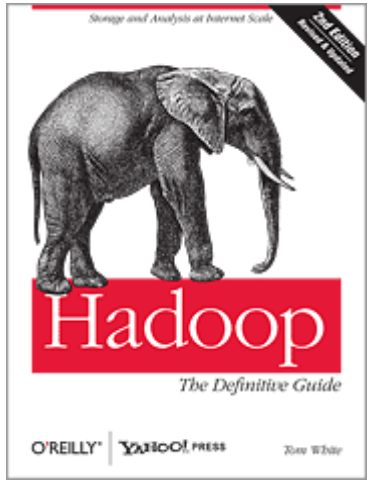
```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://...")
```

Hadoop Ecosystem



Resources

- Tom White: Hadoop. The Definitive Guide



- Udacity Big Data course

Questions?

Telegram : @mahdimd5
Mail : m.samani@anacav.com