云计算与大数据应用开发

第五章：大数据处理

丁烨
dingye@dgut.edu.cn
计算机科学与技术学院

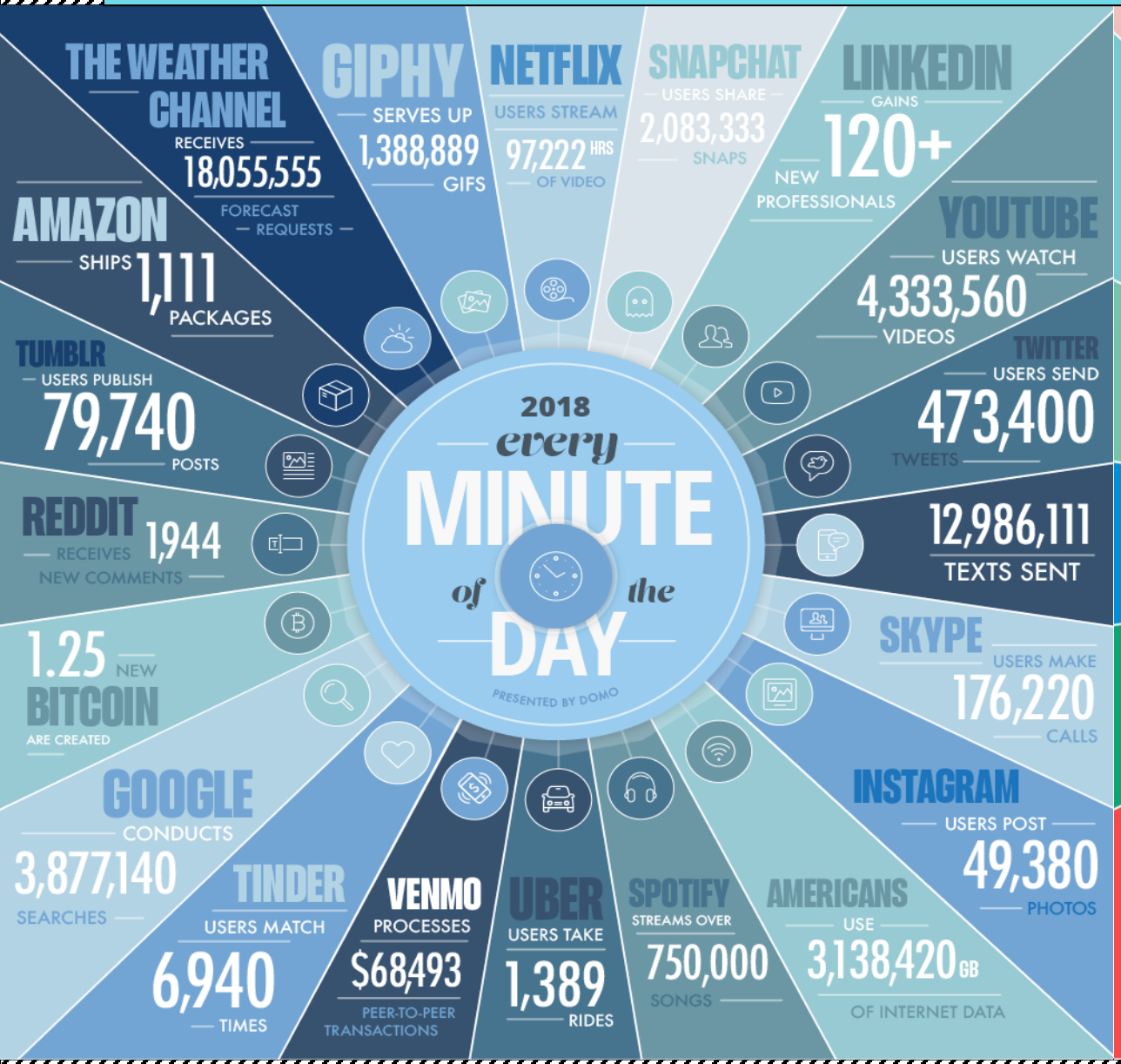
2025-04-16

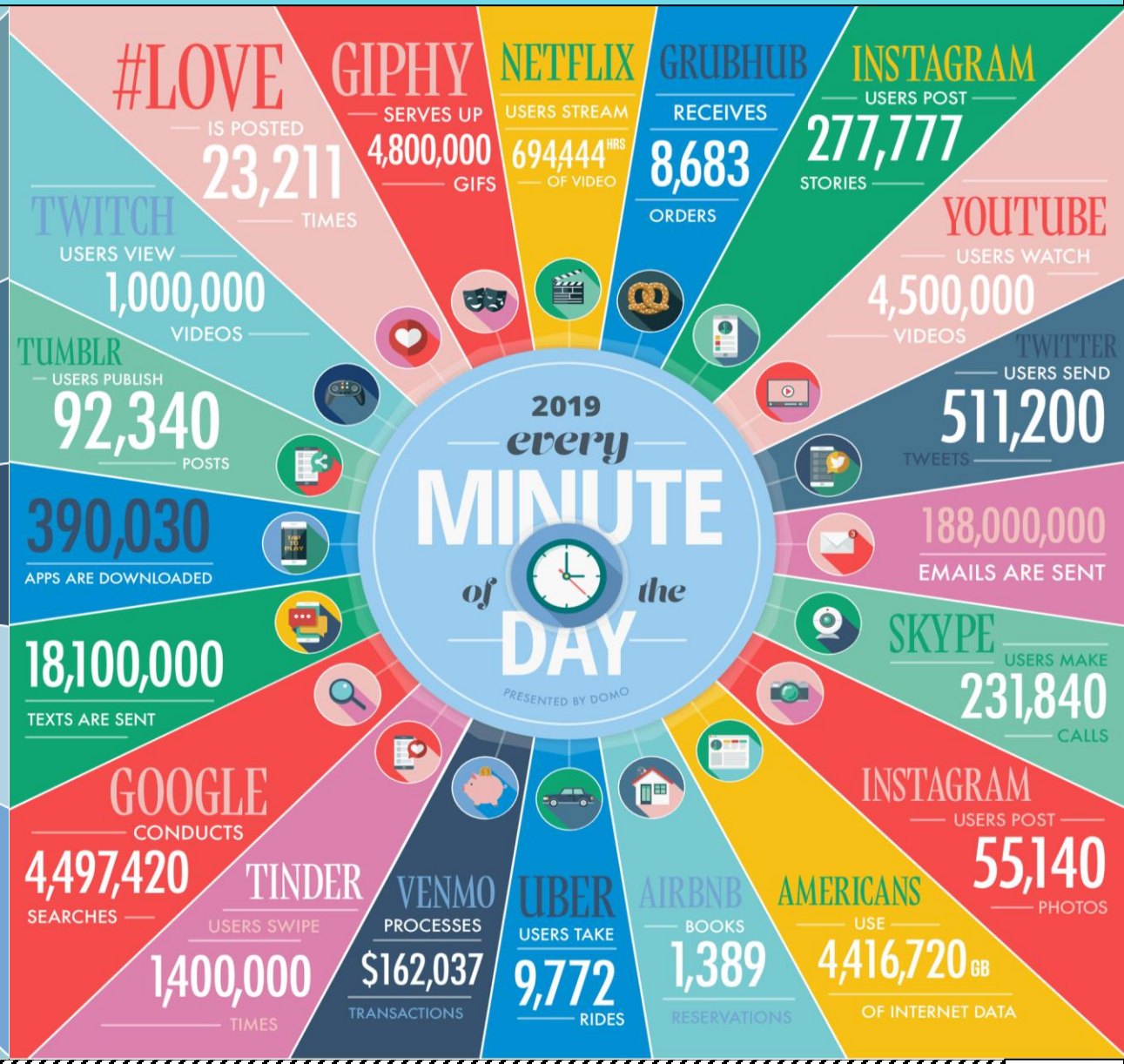
東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

目录

大数据处理技术概述

大数据行业的市场潜力





大数据行业的市场潜力

- SAP
- http://www.sap.com/

- SAP 是德国一家软件企业，总部位于德国巴登符腾堡州瓦尔多夫，主营企业资源管理软件，在 130 个国家设有办事处，在 190 个国家拥有超过 335,000 名客户
- SAP 于 1972 年在德国创立，五位创始人于德国魏恩海姆（Weinheim）初创时，公司名为“Systemanalyse and Programmentwicklung（系统分析与程式开发）”，后根据“Systeme, Anwendungen and Produkte in der Datenverarbeitung（数据处理的系统、应用与产品）”将公司名称缩写为 SAP
- 该公司股票是欧洲 Stoxx 50 股票市场指数的成分股

大数据处理技术概述

大数据行业的市场潜力

117.96 USD

+105.08 (815.84%) ↑ all time

Closed: Feb 16, 4:00 PM EST • Disclaimer

After hours 117.96 0.00 (0.00%)

大数据处理技术概述

大数据行业的市场潜力

- Palantir
- http://www.palantir.com/

- 一家美国的软件和服务公司，2003 年成立于帕罗奥图
- 以大数据分析出名，主要客户为政府机构和金融机构，“硅谷最神秘的大数据公司”

- 知名案例：
- 以大数据技术帮助美国军方成功定位和击毙基地组织首脑本·拉登
- 以大数据技术协助 BP（英国石油）钻探石油
- 以大数据技术帮助纽约警方寻找犯罪嫌疑人

Tech spooks who found bin Laden help BP find oil

Secretive data outfit Palantir is helping squeeze more from the North Sea, reports *Danny Fortson*

A fall in the oil price was the last thing Bob Dudley needed. It was 2014, BP was still clawing its way out of a $60bn financial hole created by the Deepwater Horizon disaster, and crude was set to plunge from $100 (£78) a barrel to the mid-$30s.

The chief executive knew his company needed to become leaner, more efficient and more profitable. He turned to an unusual ally: Alex Karp, the billionaire co-founder of Palantir, the secretive CIA-funded data analytics business.

The Silicon Valley outfit, chaired by co-founder Peter Thiel, had made its name working with three-letter government agencies – including the MoD. Its software combed through mountains of data to provide intelligence for planning attacks, co-ordinating disaster relief and finding "bad guys".

The most famous case of the latter was the 2011 assassination of Osama bin Laden. The CIA used Palantir to help zero in on the terrorist's location.

Palantir often sent engineers to the front lines of war zones. After Dudley's call, it sent a crack team to Aberdeen. So began an unlikely alliance. BP invested in Palantir not long after they began working together five years ago – a fact that has not been revealed until now.

The partnership catalysed a "digital transformation" that Bernard Looney, who will replace Dudley next year, claimed was the most advanced in the oil industry.

In a private speech in Paris last month, he said: "I believe we've travelled further than any other energy business." Palantir, he added, was "right at the heart of that work".

The tie-up started small: BP put Palantir together with a single reservoir engineer in Aberdeen. At the time, Palantir's only product was Gotham, a software platform that specialised in collating online and offline data into a usable form. It was adept at finding needles in haystacks, such as the locations of mines on dirt roads in Iraq.

It had not built an equivalent for corporate customers, which more often need to draw out trends from data to make business decisions. Palantir would eventually launch that product, Foundry, in 2016. When it started working with BP, it was doing one-off deals with corporates.

The goal was simple. Lamar McKay, BP's deputy group chief executive, said the company had a "garbage dump" of data that had built up over a half-century of operations in the North Sea. Could Palantir make sense of it, and turn it into something that might squeeze more juice from the company's oldest, costliest fields?

Rank-and-file workers were sceptical. A source close to the company said: "It was kind of sold to people like, 'Hey, these are the guys who found bin Laden'."

BP estimates that Palantir, led by Alex Karp, has helped to increase its North Sea oil production by 10%

The early results were promising, McKay said: "If something grass-roots happens in Aberdeen, it's probably pretty good because, you know, they [in Aberdeen] don't like to be told what to do, let's say."

BP soon invested in the private company, whose biggest shareholder was Thiel, known today as perhaps the tech industry's most prominent supporter of Donald Trump.

What began as an experiment turned into a 10-year, $1.2bn contract. McKay reckons that in the North Sea, BP produces an extra 20,000 barrels – a 10% improvement – thanks to the "digital twin" that Palantir has built. The facsimile allows it to test and make minute, constant changes to flows, extraction rates, compression and countless other variables across thousands of miles of pipes and risers and wells.

BP and Palantir kept their partnership private until very recently. For BP, it is not hard to understand why. Palantir, which is worth an estimated $40bn and rumoured to be eyeing a stock market float next year, is, perhaps, the most controversial tech company in Silicon Valley after Facebook.

Karp, 52, is an unabashed supporter of American and western governments. The company's contracts with the CIA and Immigration and Customs Enforcement (ICE) have led to protests at its Palo Alto headquarters. Several organisations this year have dropped Palantir as a sponsor of conferences.

Karp has bashed Big Tech rivals such as Google for pulling out of defence contracts in response to public outcry. He has called Silicon Valley "an island that seems to be devoid of any of the cultural norms that the rest of us tend to share". ICE this summer renewed a contract with Palantir for software that is used to track immigrants at the border.

Palantir has burrowed deep inside BP. Having started in the production arm, it has been rolled out across most of BP's operations, including trading and refining. BP will soon bring in its wind energy division as well, which is in dire need of data help: the company's 10 wind farms produce more data than the rest of its operations combined. Looney said: "We collect the data – and immediately delete it. We have neither capacity nor systems to analyse it."

> **"We've come further than any other energy business and Palantir is at the heart of it**

大数据处理技术概述

大数据行业的市场潜力

Google 对用户搜索、邮件等数据进行分析实现
在线广告的精准投放

Amazon 利用 Kindle 数据分析用户阅读习惯
向用户推荐其可能购买的书籍

美国总统的竞选团队依据选民的微博
实时分析选民喜好

对城市交通数据进行深度分析与挖掘
对城市道路规划提出建议

美国疾病控制和预防中心依据网民搜索
分析全球范围内流感等病疫的传播状况

对冲基金依据购物网站的顾客评论
分析企业产品销售状况

欧洲粒子物理研究中心（CERN）分析大型强子对撞机（LHC）记录的
超过 100 PB 的数据，“几乎”发现了上帝粒子

……

数据　　知识　　价值

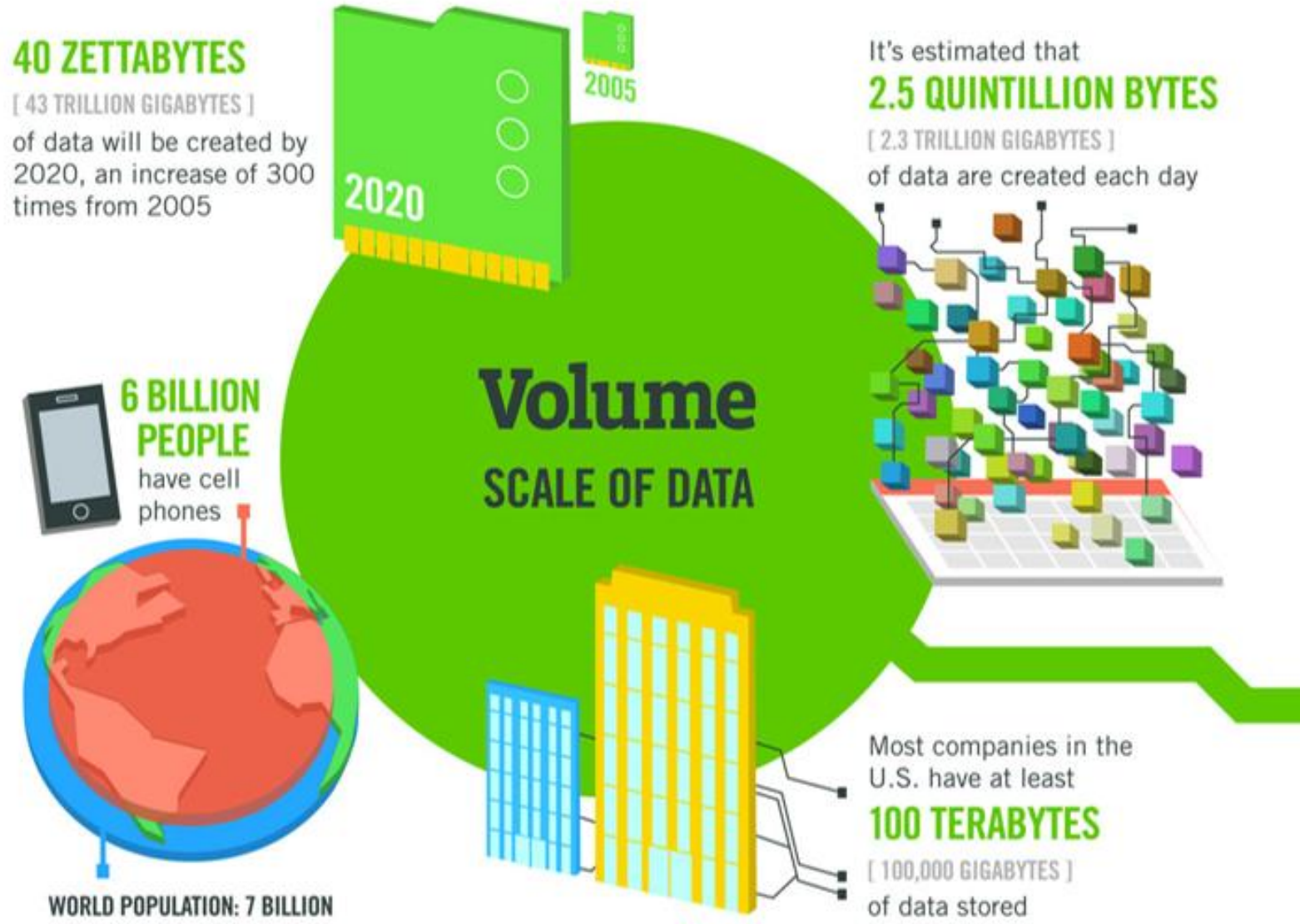
大数据存储、管理、处理、挖掘、应用

数据是未来的新石油

大数据处理技术概述

大数据处理的难点

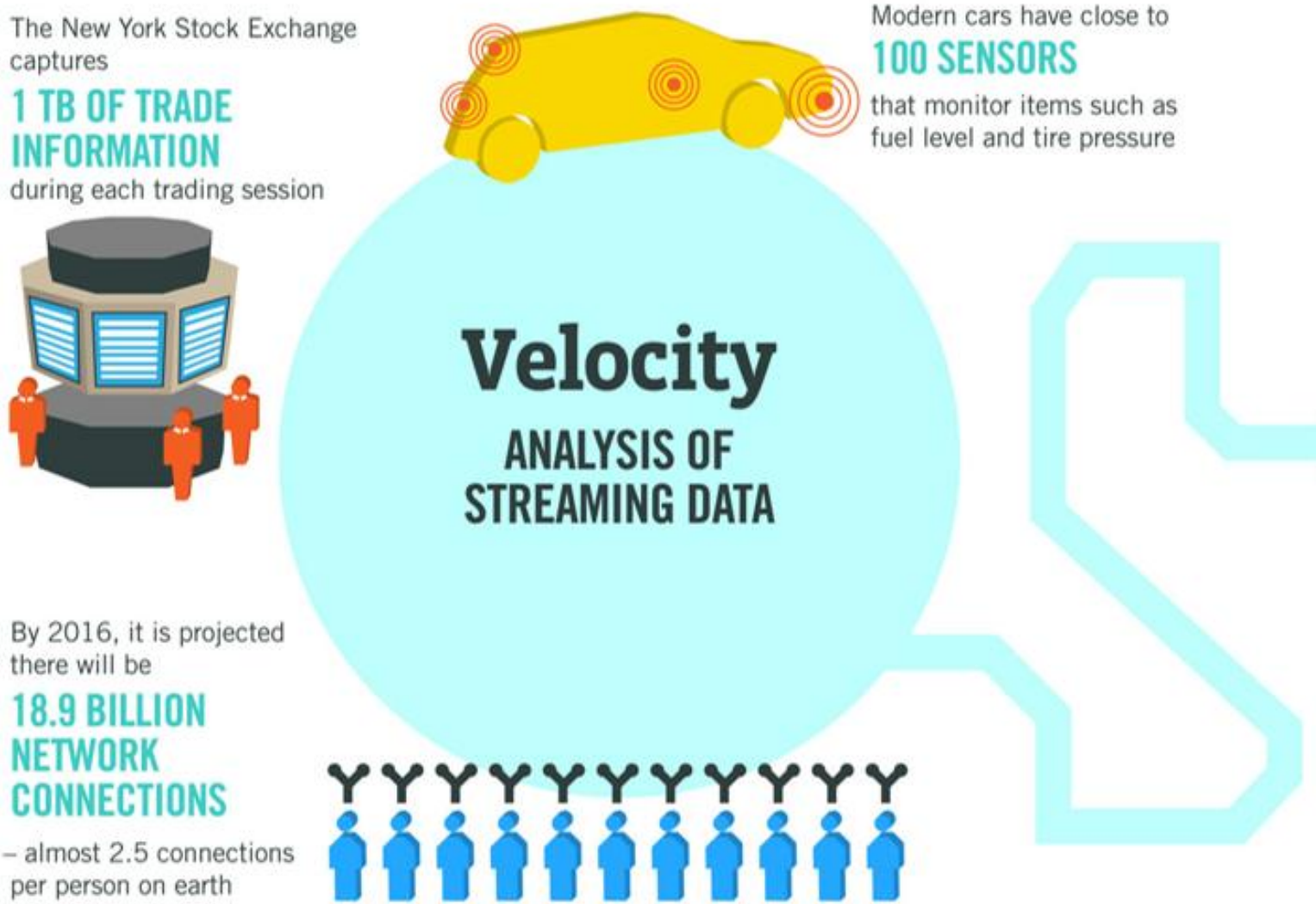
- 数据量（Volume）庞大
- 单数据集数据量往往超过一块最大容量硬盘的存储空间
- 传统的 RDBMS 数据库甚至无法将数据读入到内存中
- 工业计算（Industrial Computing）数据量太大，导致计算速度缓慢，尤其是：气象学、基因组学、神经网络体学、复杂的物理模拟，以及生物和环境研究等领域



大数据处理技术概述

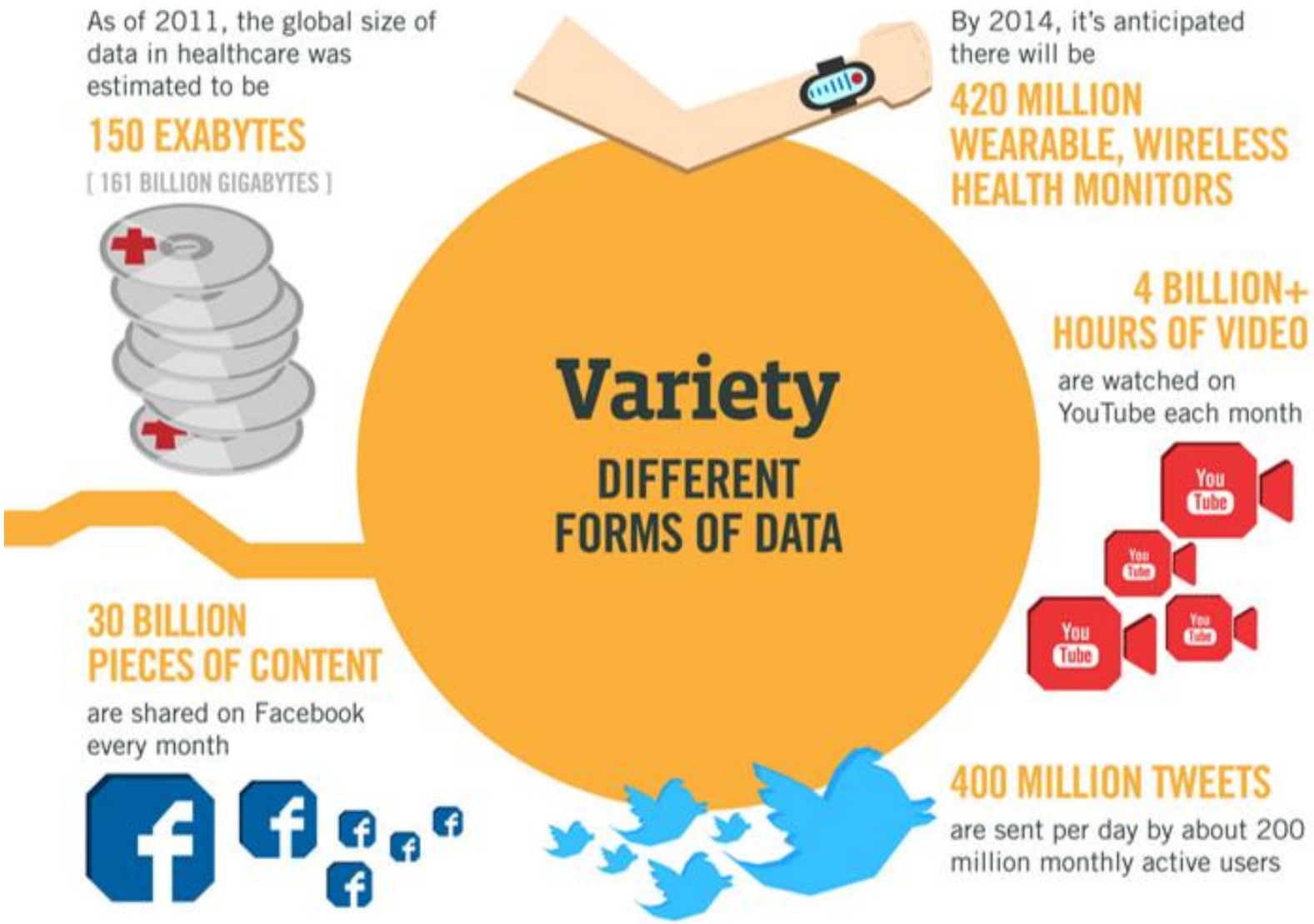
大数据处理的难点

- 数据产生速度（Velocity）快
- 上一条数据还没有处理完，下一条数据已经产生
- 在社交媒体、量化金融等行业尤其明显
- 数据产生的速度甚至超过了网络吞吐量，导致网络系统崩溃



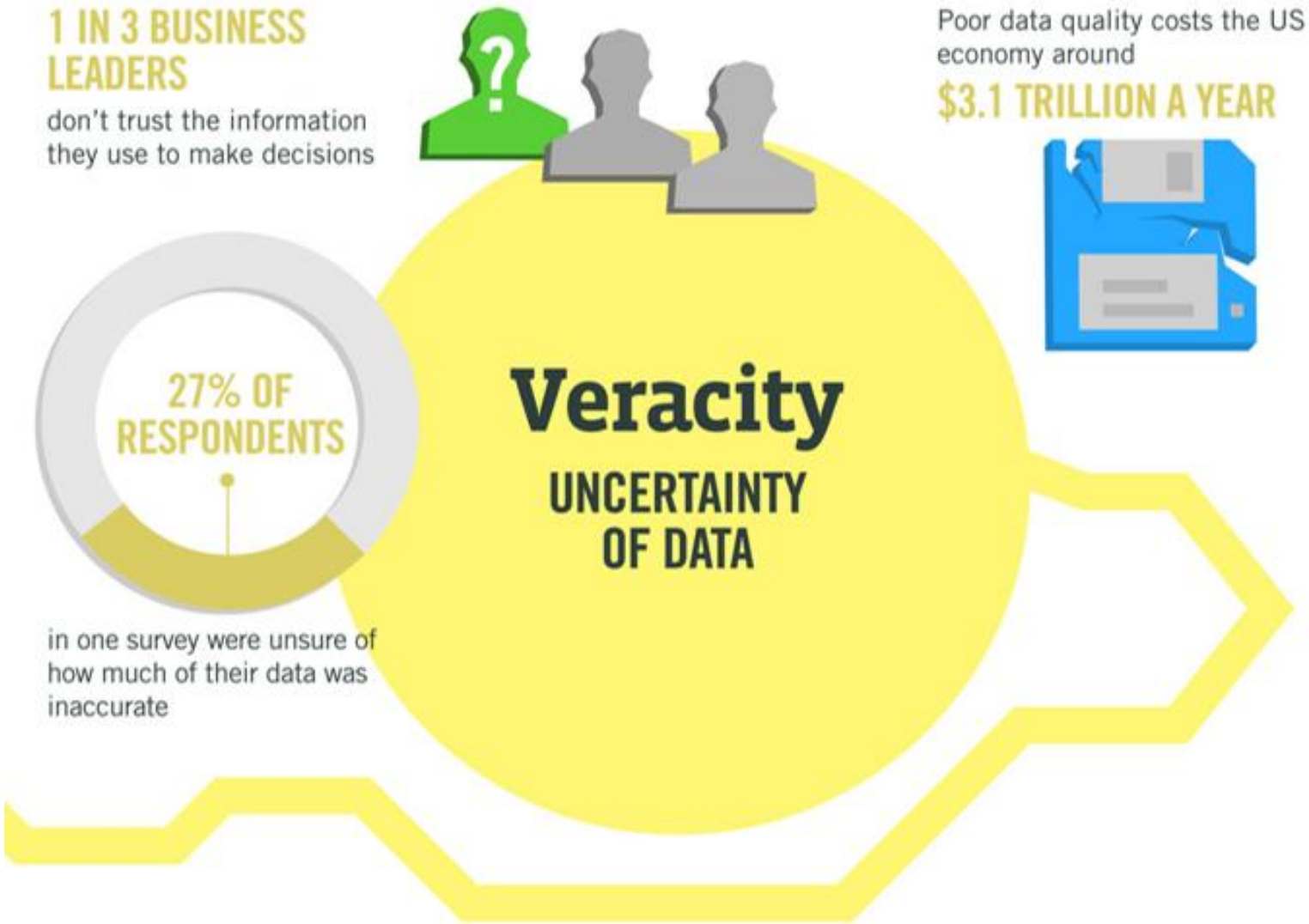
大数据处理的难点

- 数据格式多变（Variety）
- 数据来源广泛，且格式不一
- 在自然语言处理（Natural Language Processing, NLP）、计算机视觉（Computer Vision，CV）、物联网（Internet of Things，IoT）领域尤其明显
- “数据清洗”：统一、泛化不同格式的数据是个非常复杂的问题



大数据处理的难点

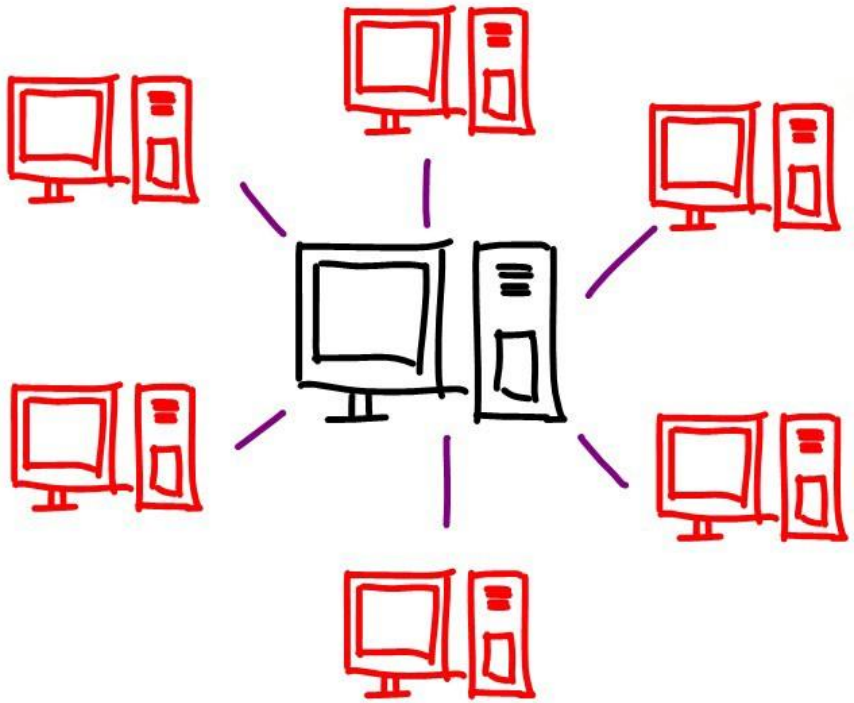
- 数据真实性（Veracity）难辨
- 数据不是 100% 准确的（Uncertain）
- 众包（Crowdsourcing）、社交媒体、地理系统（Geographic Information System，GIS）产生的数据尤其明显
- 处理不准确的数据一直是大数据和人工智能领域一个重要的问题



大数据解决方案

Virtualization

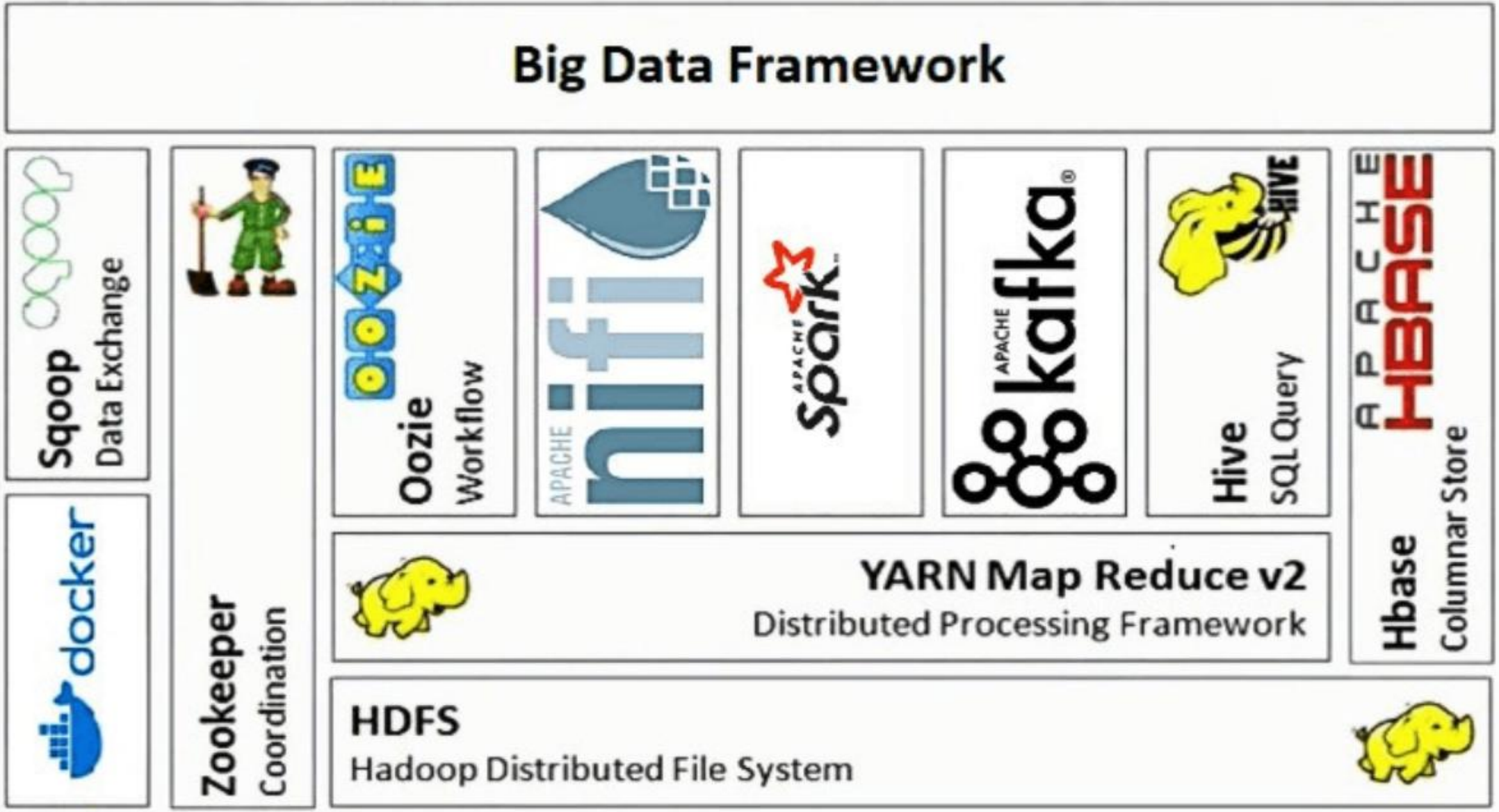
大数据解决方案



Distributed Computing

大数据处理技术概述

Hadoop 大数据处理技术栈



Hadoop is Dead. Long live Hadoop.

Arun C Murthy Follow
Sep 10, 2019 · 7 min read ★

Personally, "Hadoop" is a *philosophy* — a movement towards a modern *architecture for managing and analyzing data*.

Hadoop 大数据处理技术栈

- Hadoop 正在逐渐退出历史舞台
- Hadoop 的三个核心均被云计算或更高级的工具所替代
- YARN：Hadoop 的核心资源管理器，已被 Docker + Kubernetes 替代
- HDFS：Hadoop 的存储核心，已被云存储（例如 AWS S3、Ceph、Alluxio 等）替代
- MapReduce：Hadoop 的核心算法，先进的大数据处理平台如 Spark、Flink、Beam 等已经可以完全独立运行 MapReduce，不再需要依赖 Hadoop

- Hadoop 为大数据处理带来了先进的架构和理念，已经完成了历史使命

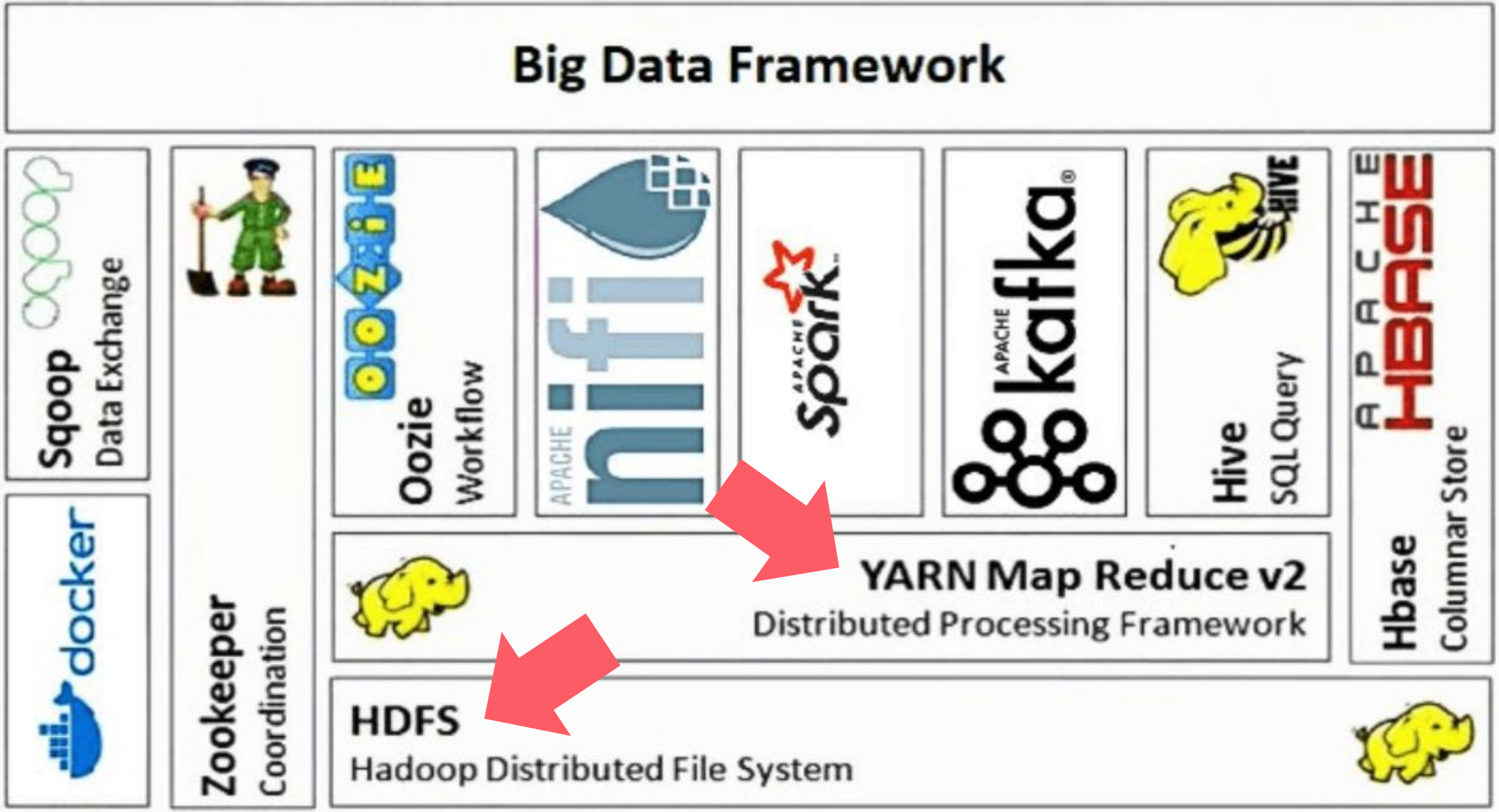
目录

Apache Hadoop

Hadoop 简介

- Apache Hadoop
- http://hadoop.apache.org/

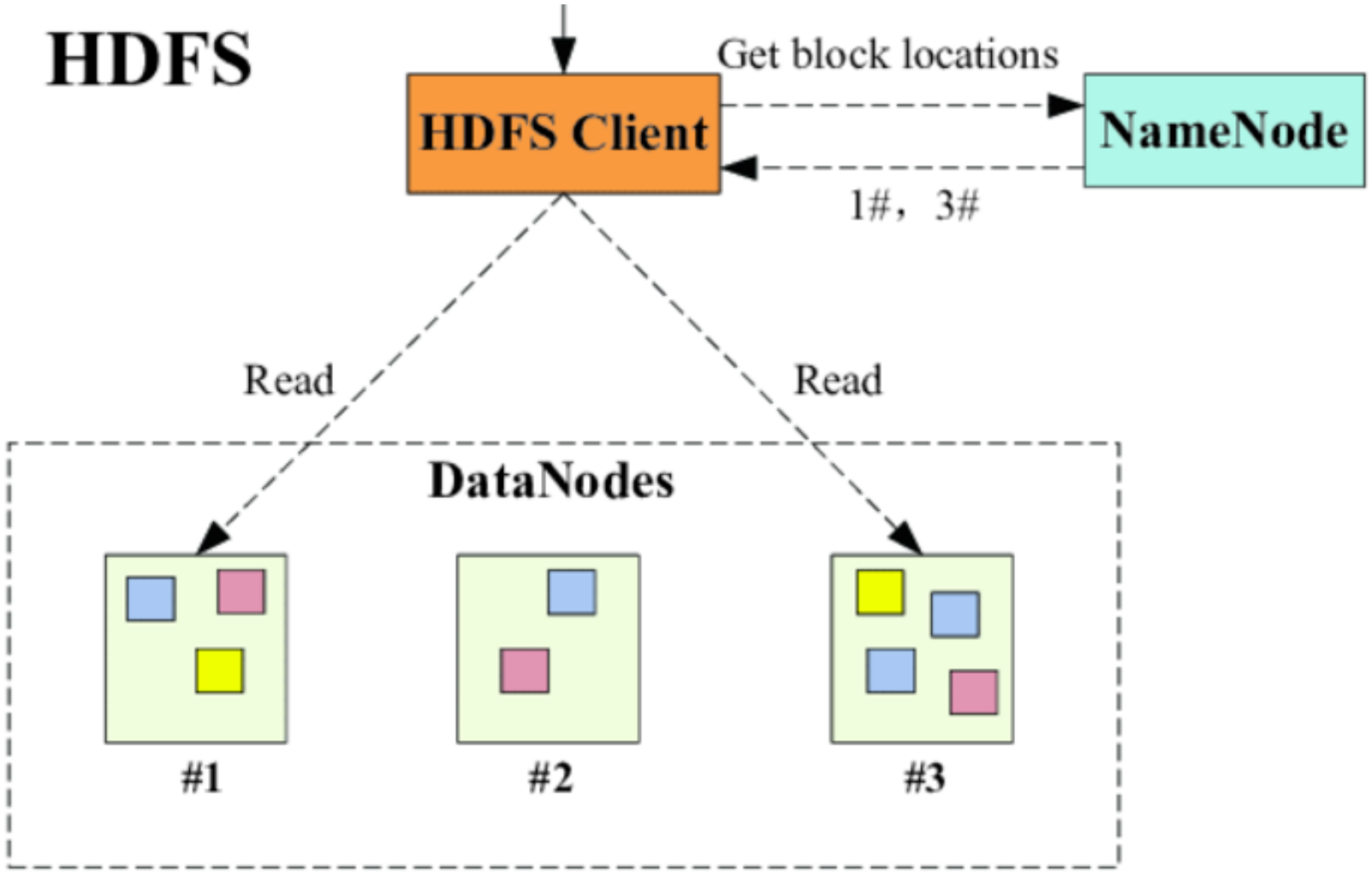
- 一款支持数据密集型分布式应用程序并以 Apache 2.0 许可协议发布的开源软件框架
- Hadoop 是根据 Google 发表的 MapReduce 和 Google 文件系统（GFS）的论文自行实现而成：https://research.google.com/archive/gfs-sosp2003.pdf
- 所有的 Hadoop 模块都有一个基本假设，即硬件故障是常见情况，应该由框架自动处理
- Hadoop 框架透明的为应用提供可靠性和数据移动
- 它实现了名为 MapReduce 的编程范式：应用程序被分割成许多小部分，而每个部分都能在集群中的任意节点上运行或重新运行


Big Data Framework
Sqoop
Data Exchange
Zookeeper
Coordination
Oozie
Workflow
APACHE nifi
APACHE spark
APACHE kafka
Hive
SQL Query
APACHE HBASE
Hbase
Columnar Store
docker
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

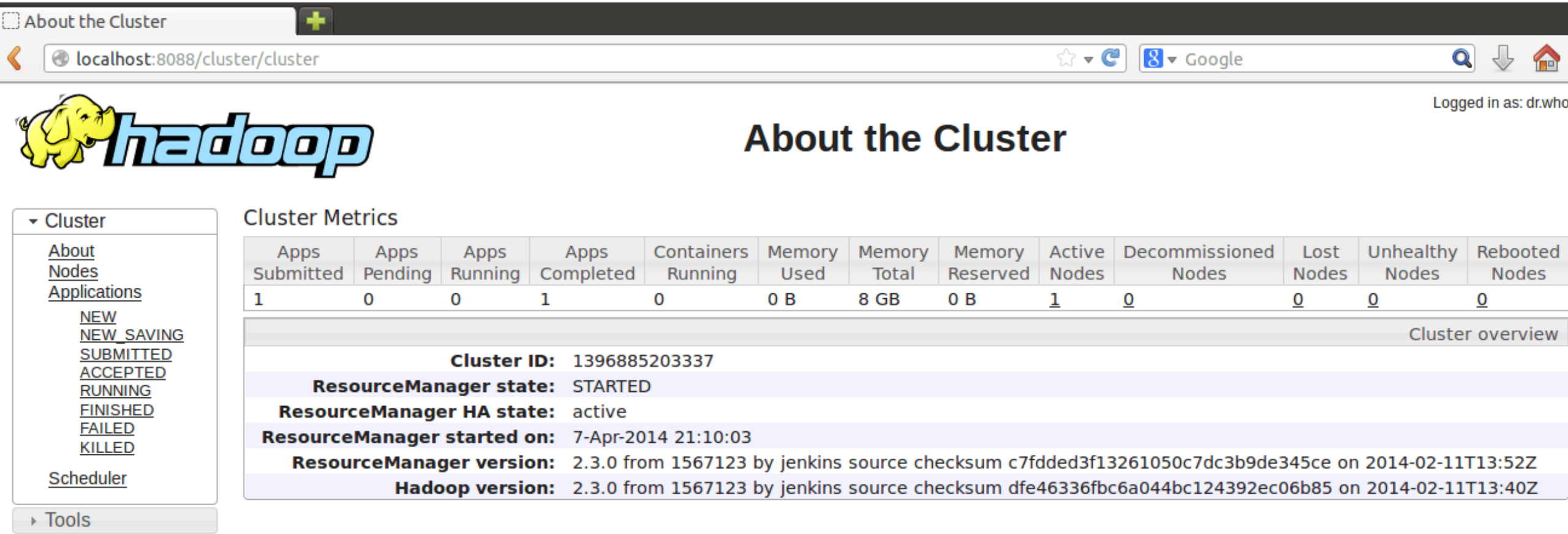
Hadoop 简介

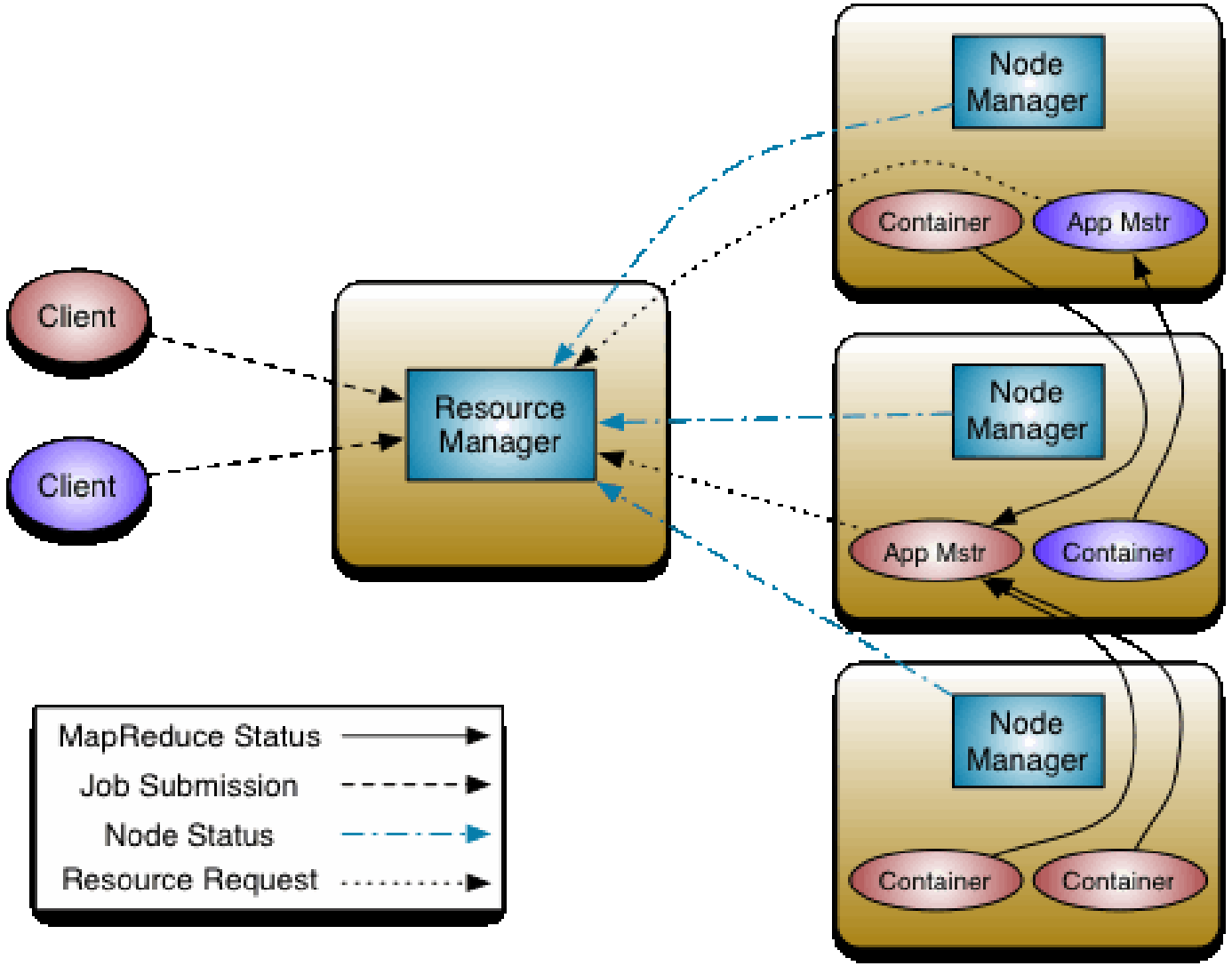
- Hadoop 分布式文件系统（Hadoop Distributed File System，HDFS）
- Hadoop 提供的分布式文件系统，用以存储所有计算节点的数据
- 这为整个集群带来了非常高的带宽
- MapReduce 和分布式文件系统的设计，使得整个框架能够自动处理节点故障
- 它使应用程序与成千上万的独立计算的计算机和 PB 级的数据连接起来

- HDFS 的架构
- HDFS 由一个 Name Node 和多个 Data Node 组成
- Name Node 为管理节点
- Data Node 为数据节点



- YARN（Yet Another Resource Negotiator）
- Hadoop 的资源管理系统，和 Docker / Kubernetes 理念类似

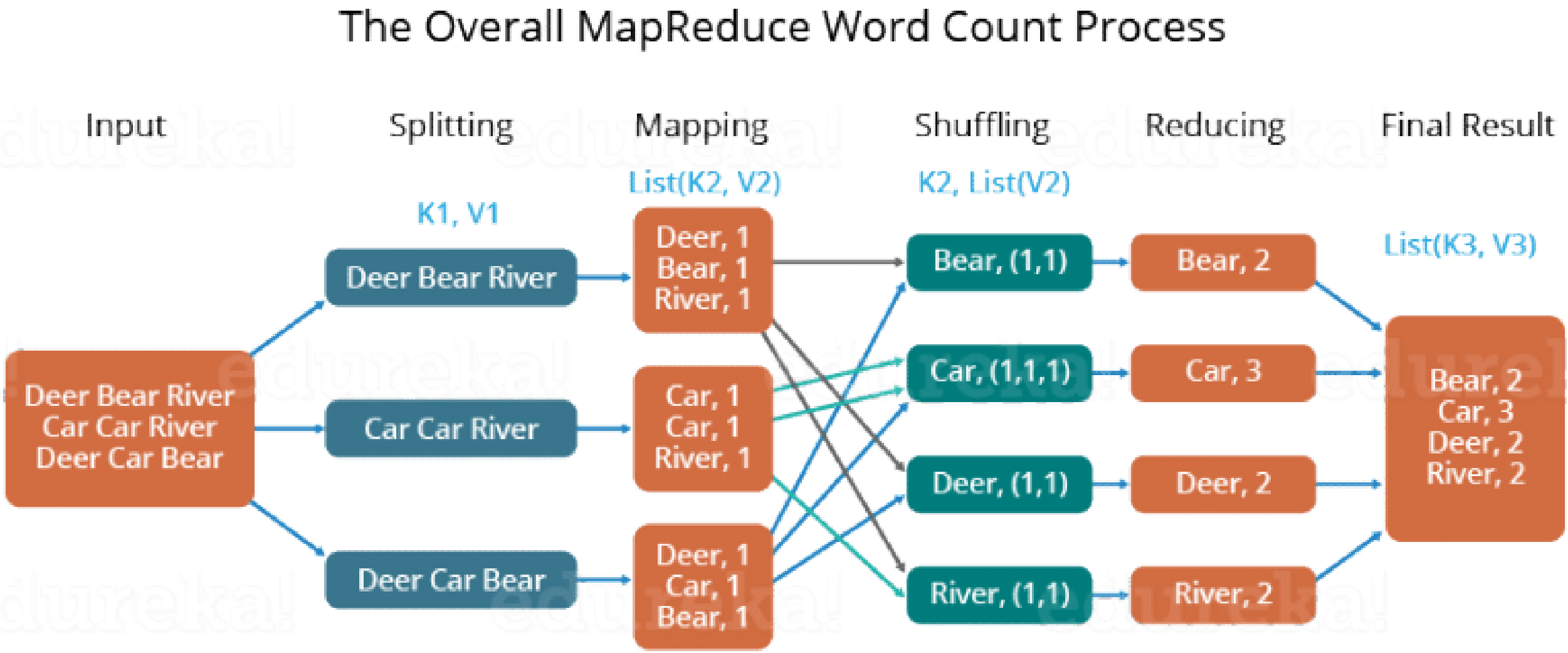


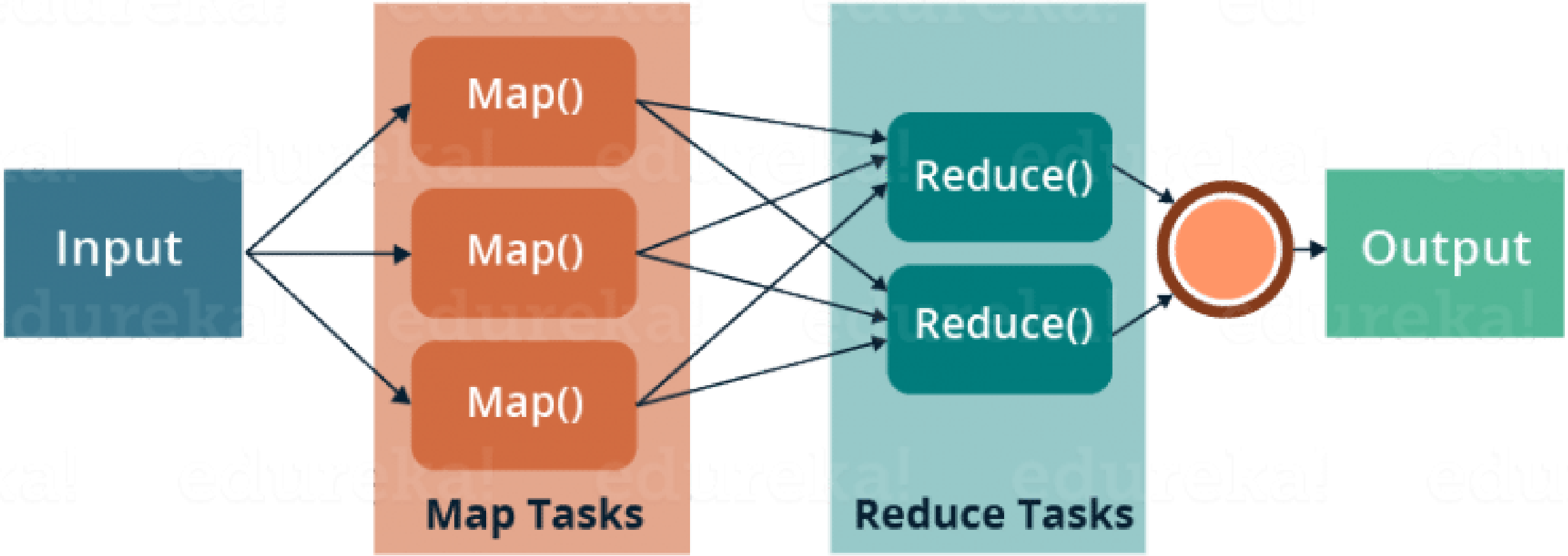

Client
Client
Resource Manager
Node Manager
Container
App Mstr
Node Manager
App Mstr
Container
Node Manager
Container
Container
MapReduce Status
Job Submission
Node Status
Resource Request

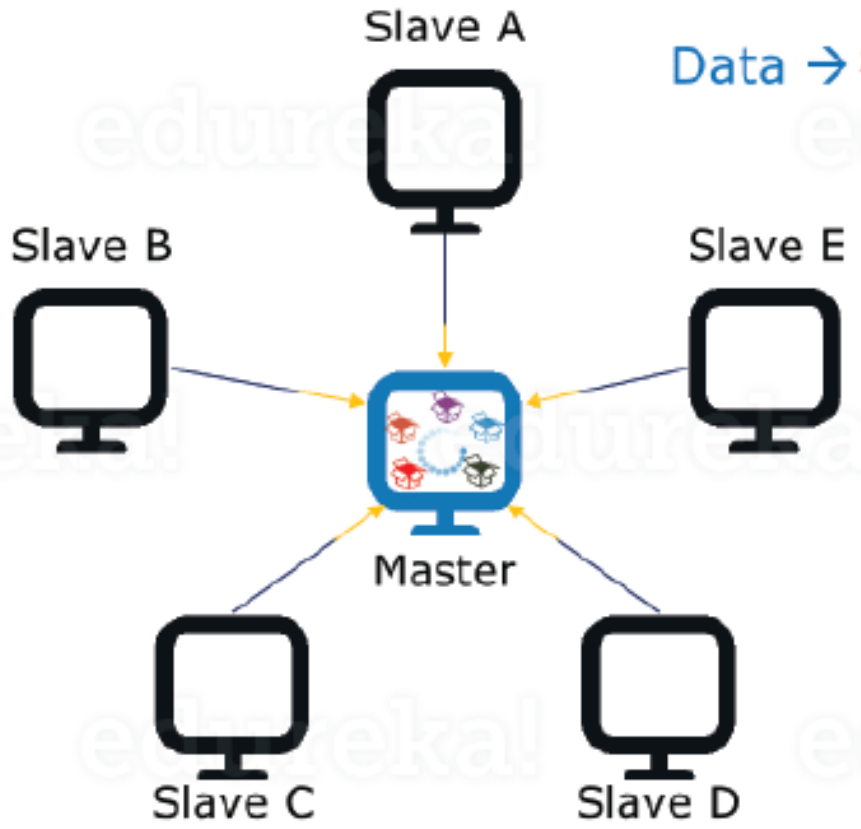
- MapReduce
- Google 提出的一个软件架构，用于大规模数据集（大于 1 TB）的并行运算
- MapReduce 包括两个主要组成部分：Map（映射）和 Reduce（归纳）
- 这两个概念主要来自函数式编程语言

- Map：每个 Mapper 节点通过一定计算将原始数据处理后，产生临时输出（一般是键-值（Key-Value）形式）
- Shuffle：资源管理器将临时输出分配给 Reducer 节点，通常来说，同一个键（Key）会分配给同一个（或同一组）Reducer
- Reduce：每个 Reducer 节点将收到的临时数据进一步计算，得到输出

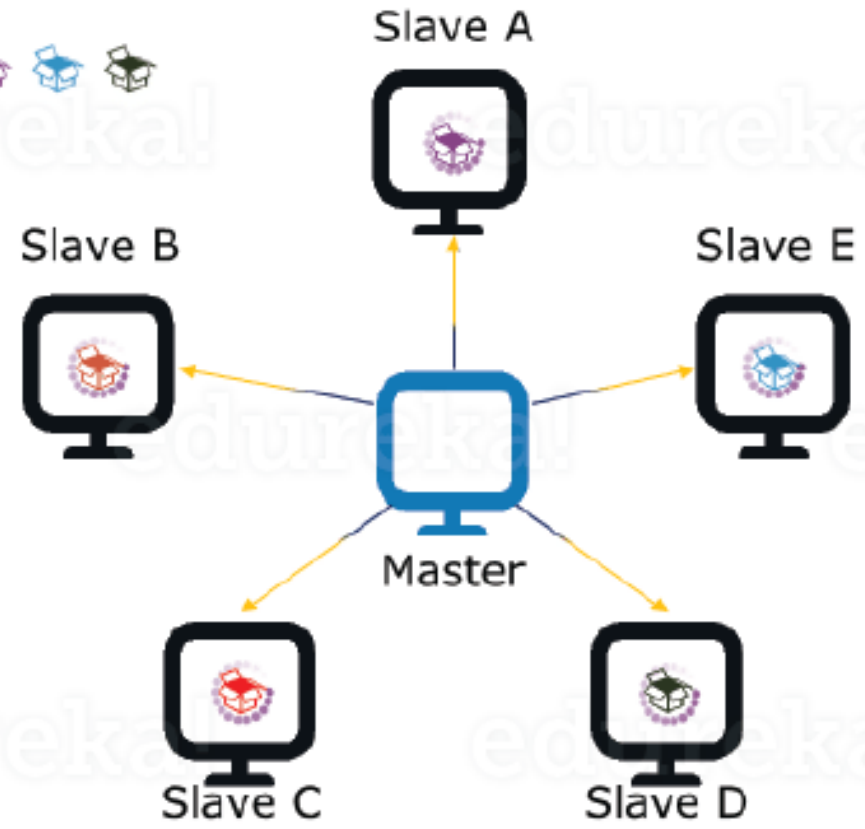
- Word Count
- 统计一篇或多篇文章中出现的词及其出现次数
- 非常适合测试分布式系统的一个经典算法问题
- 当文章数量（长度）非常庞大时，单机计算几乎不可能
- 例如，2019 年，平均每分钟有 511,200 条 Tweet 被发送，一天的数据量超过 4 TB
- MapReduce 非常适合解决此类问题


The Overall MapReduce Word Count Process
Input
Splitting
Mapping
Shuffling
Reducing
Final Result
K1, V1
List(K2, V2)
K2, List(V2)
List(K3, V3)
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, (1,1)
Car, (1,1,1)
Deer, (1,1)
River, (1,1)
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2


Input
Map()
Map()
Map()
Map Tasks
Reduce()
Reduce()
Reduce Tasks
Output



1. Moving data to the Processing Unit
(Traditional Approach)



2. Moving Processing Unit to the data
(MapReduce Approach)

Apache Hadoop

安装 Hadoop

- Hadoop 的安装较为复杂，可以参考 Hadoop 官方教程：
- https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

Apache Hadoop

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

Learn more » Download » Getting started »

安装 Hadoop

- Hadoop Docker Image by Big Data Europe
- https://www.big-data-europe.eu/
- https://github.com/big-data-europe/docker-hadoop

- Hadoop 的 Docker Image 很多，Big Data Europe 只是其中一个
- 由于 Hadoop 为分布式系统，使用 Hadoop 需要同时启动多个 Docker Container
- Big Data Europe 采用了 Docker Compose 进行更方便的部署

- Docker Compose
- https://docs.docker.com/compose/

- Docker Compose 是一个用来管理 Docker Container 的工具
- 使用 Docker Compose 可以同时部署、启动、关闭多个 Docker Container
- Docker Compose 相当于管理 Docker Container 的“脚本工具”

A `docker-compose.yml` looks like this:

```
version: '3'
services:
  web:
    build: .
    ports:
    - "5000:5000"
    volumes:
    - .:/code
    - logvolume01:/var/log
    links:
    - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

❖ 安装 Docker Compose

❖ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

❖ sudo chmod +x /usr/local/bin/docker-compose

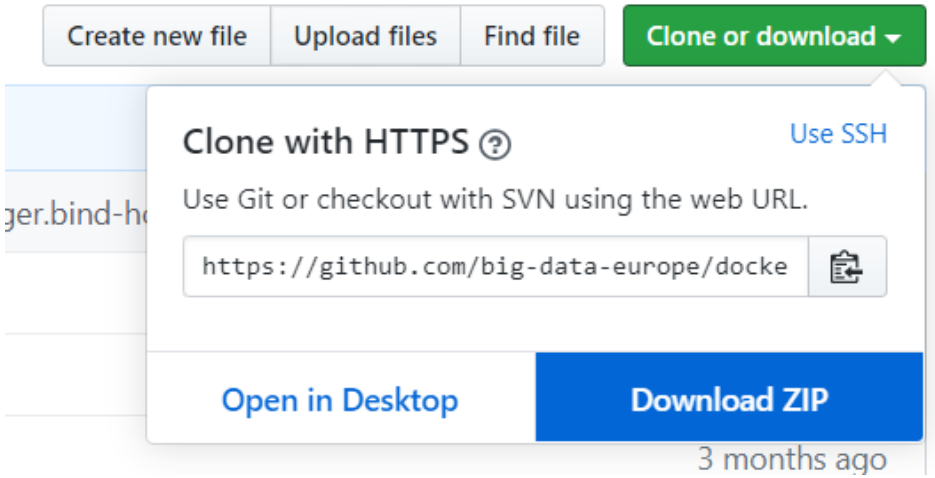
```
~/Workspace/docker-hadoop(master) » sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-
$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   638  100   638    0     0    746      0 --:--:-- --:--:-- --:--:--   745
100 16.7M  100 16.7M    0     0   188k      0  0:01:31  0:01:31 --:--:--  142k
--------------------------------------------------------------------------------------------------------------------------------
~/Workspace/docker-hadoop(master) » sudo chmod +x /usr/local/bin/docker-compose                                  valency@dev-server
[sudo] password for valency:
--------------------------------------------------------------------------------------------------------------------------------
~/Workspace/docker-hadoop(master) » docker-compose --version                                                    valency@dev-server
docker-compose version 1.25.5, build 8a1c60f6
```

安装 Hadoop

- 使用 Big Data Europe 的 Hadoop Docker Image
- https://github.com/big-data-europe/docker-hadoop

- 首先使用 Git 将本项目 Clone 到本地：
- `git clone git@github.com:big-data-europe/docker-hadoop.git`

- 如果不熟悉 Git，也可以点击 Download ZIP 下载



安装 Hadoop

- 通过 Docker Compose 启动本地 Hadoop 集群：
- `cd docker-hadoop`
- `docker-compose up -d`

```
~/Workspace/docker-hadoop(master) » docker-compose up -d
Creating network "docker-hadoop_default" with the default driver
Creating nodemanager     ... done
Creating namenode        ... done
Creating resourcemanager ... done
Creating datanode        ... done
Creating historyserver   ... done
```

```
~/Workspace/docker-hadoop(master) » docker ps -a
CONTAINER ID        IMAGE                                                COMMAND                  CREATED             STATUS
0c587dd9fc3d        bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8    "/entrypoint.sh /run…"   5 seconds ago       Up 2 seconds (healt
b2288fd92084        bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8         "/entrypoint.sh /run…"   5 seconds ago       Up 2 seconds (healt
6323736595f2        bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   5 seconds ago       Up 2 seconds (healt
8a41126ded7c        bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   5 seconds ago       Up 2 seconds (healt
822cf1aa55e3        bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8       "/entrypoint.sh /run…"   5 seconds ago       Up 3 seconds (healt
```

- 查看 YARN 状态
- http://localhost:9870

Overview 'namenode:9000' (active)

Started:	Fri Apr 17 18:48:44 +0800 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 23:56:00 +0800 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-ea901893-aef1-4f53-a8a5-5ce8b4e2b022
Block Pool ID:	BP-1042871402-172.18.0.2-1587120298015

- 测试 HDFS
- 进入 Hadoop 的 Name Node Container：
- `docker exec -it namenode bash`

- 创建两个测试文件：
- `mkdir input`
- `echo "Hello World" >input/f1.txt`
- `echo "Hello Docker" >input/f2.txt`

使用 Hadoop

- 在 HDFS 中创建目录：
- `hadoop fs -mkdir -p input`

- 查看 HDFS 的目录和文件：
- `hadoop fs -ls`

```
~/Workspace/docker-hadoop(master) » docker exec -it namenode bash
root@8a41126ded7c:/# mkdir input
root@8a41126ded7c:/# echo "Hello World" >input/f1.txt
root@8a41126ded7c:/# echo "Hello Docker" >input/f2.txt
root@8a41126ded7c:/# hadoop fs -mkdir -p input
root@8a41126ded7c:/# hadoop fs -ls
Found 1 items
drwxr-xr-x   - root supergroup          0 2020-04-17 10:51 input
root@8a41126ded7c:/#
```

使用 Hadoop

- 将文件传入 HDFS:
- `hadoop fs -put ./input/* input`

- 查看传入的文件:
- `hadoop fs –ls input`

```
root@8a41126ded7c:/# hadoop fs -put ./input/* input
root@8a41126ded7c:/# hadoop fs -ls input
Found 2 items
-rw-r--r--   3 root supergroup         12 2020-04-17 10:54 input/f1.txt
-rw-r--r--   3 root supergroup         13 2020-04-17 10:54 input/f2.txt
root@8a41126ded7c:/#
```

使用 Hadoop

- 将文件从 HDFS 复制到本地：
- hadoop fs -get -r input input2

```
root@8a41126ded7c:/# hadoop fs -get input input2
2020-04-17 10:59:43,958 INFO sasl.SaslDataTransferClient: SASL encryption trust check:
root@8a41126ded7c:/# ls input*
input:
f1.txt  f2.txt

input2:
f1.txt  f2.txt
root@8a41126ded7c:/#
```

使用 Hadoop

- 将文件从 HDFS 删除：
- `hadoop fs -rm -r input`

```
root@8a41126ded7c:/# hadoop fs -rm -r input
Deleted input
root@8a41126ded7c:/# hadoop fs -ls
root@8a41126ded7c:/#
```

使用 Hadoop

- 如需将文件传入 Data Node，则需使用：
- `hdfs dfs -mkdir –p input`
- `hdfs dfs -put ./input/* input`

```
root@8a41126ded7c:/# hdfs dfs -mkdir -p input
root@8a41126ded7c:/# hdfs dfs -put ./input/* input
root@8a41126ded7c:/# hdfs dfs -ls input
Found 2 items
-rw-r--r--   3 root supergroup         12 2020-04-17 11:04 input/f1.txt
-rw-r--r--   3 root supergroup         13 2020-04-17 11:04 input/f2.txt
root@8a41126ded7c:/#
```

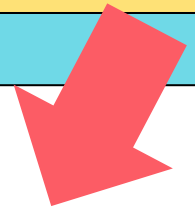
- 测试 MapReduce
- Hadoop 的“官方”客户端 SDK 仅支持 Java
- 也可以通过 Hadoop Streaming 连接任何一种编程语言
- 关于 Hadoop Streaming：
- https://hadoop.apache.org/docs/current/hadoop-streaming/HadoopStreaming.html
- 使用 Python 通过 Hadoop Streaming 运行 MapReduce 任务的例子：
- https://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/

❖ Word Count 示例代码

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import java.io.IOException;
import java.util.StringTokenizer;
```

使用 Hadoop



```
public class WordCount {
    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
        ...
    }
    public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        ...
    }
    public static void main(String[] args) throws Exception {
        ...
    }
}
```

```
public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}
```

```
public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

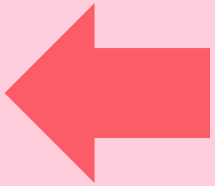
使用 Hadoop

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

使用 Hadoop

- 使用 Hadoop 编译 WordCount 源代码：
- `hadoop com.sun.tools.javac.Main WordCount.java`
- 将编译好的 Class 打包成 Jar：
- `jar cf wc.jar WordCount*.class`

```
root@8a41126ded7c:~# hadoop com.sun.tools.javac.Main WordCount.java
/opt/hadoop-3.2.1/libexec/hadoop-functions.sh: line 2401: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_USER: bad substitution
/opt/hadoop-3.2.1/libexec/hadoop-functions.sh: line 2366: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_USER: bad substitution
/opt/hadoop-3.2.1/libexec/hadoop-functions.sh: line 2461: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_OPTS: bad substitution
root@8a41126ded7c:~# jar cf wc.jar WordCount*.class
root@8a41126ded7c:~# ls -al WordCount.*
-rw-r--r-- 1 root root 1491 Apr 17 19:03 WordCount.class
-rw-rw-r-- 1 1000 1000 2291 Apr 17 18:58 WordCount.java
root@8a41126ded7c:~# ls -al *.jar
-rw-r--r-- 1 root root 3075 Apr 17 19:03 wc.jar
root@8a41126ded7c:~#
```



使用 Hadoop

- 如果在编译中遇到以下错误:
- Error: Could not find or load main class com.sun.tools.javac.Main
- 可以尝试将 tools.jar 加入到 HADOOP_CLASSPATH 中:
- export HADOOP_CLASSPATH=$JAVA_HOME/lib/tools.jar
- 或者参考以下环境变量要求:

Assuming environment variables are set as follows:

```
export JAVA_HOME=/usr/java/default
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

使用 Hadoop

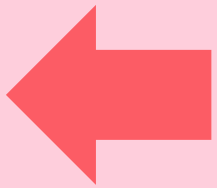
- 向 Hadoop 集群提交任务：
- `hadoop jar wc.jar WordCount input output`

```
root@8a41126ded7c:~# hadoop jar wc.jar WordCount input output
2020-04-17 19:05:01,144 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/172.19.0.6:8032
2020-04-17 19:05:01,281 INFO client.AHSProxy: Connecting to Application History server at historyserver/172.19.0.3:
10200
2020-04-17 19:05:02,830 INFO mapreduce.Job: The url to track the job: http://resourcemanager:8088/proxy/application
_1587120561209_0001/
2020-04-17 19:05:02,836 INFO mapreduce.Job: Running job: job_1587120561209_0001
2020-04-17 19:05:12,071 INFO mapreduce.Job: Job job_1587120561209_0001 running in uber mode : false
2020-04-17 19:05:12,073 INFO mapreduce.Job:  map 0% reduce 0%
2020-04-17 19:05:21,150 INFO mapreduce.Job:  map 100% reduce 0%
2020-04-17 19:05:26,182 INFO mapreduce.Job:  map 100% reduce 100%
2020-04-17 19:05:26,213 INFO mapreduce.Job: Job job_1587120561209_0001 completed successfully
2020-04-17 19:05:26,399 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=52
                FILE: Number of bytes written=687208
```

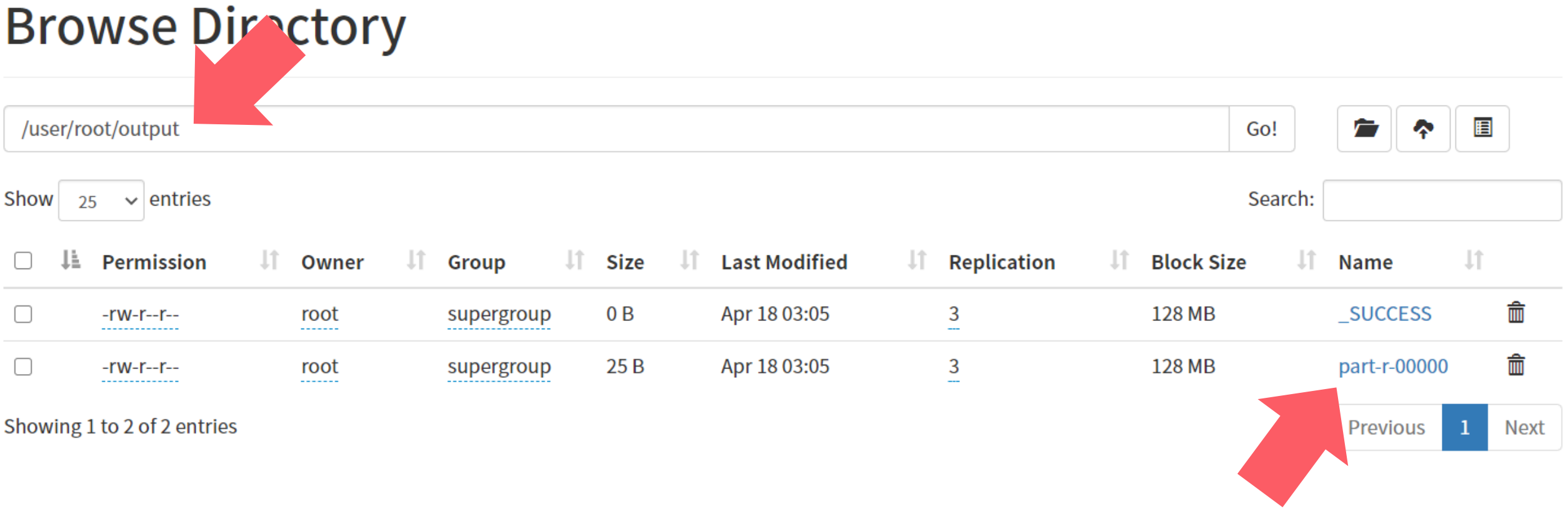
使用 Hadoop

- 查看运行结果：
- hadoop fs -cat output/part-r-00000
- 在测试 HDFS 时，两个测试文件的内容为：
- echo "Hello World" >input/f1.txt
- echo "Hello Docker" >input/f2.txt

```
root@8a41126ded7c:~# hadoop fs -cat output/part-r-00000
2020-04-17 19:11:38,280 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, re
moteHostTrusted = false
Docker  1
Hello   2
World   1
root@8a41126ded7c:~#
```



- 也可以通过 HDFS 的 File Explorer 查看文件：
- http://localhost:9870/explorer.html



- 使用完毕后，可以通过 Docker Compose 停止并销毁本地 Hadoop 集群：
- `cd docker-hadoop`
- `docker-compose down`

- 注意：销毁 Container 会同时销毁 HDFS 中的全部文件，请注意备份

- Hadoop Java SDK
- https://hadoop.apache.org/releases.html
- 注意：SDK 版本需要与 Hadoop 的部署版本一致
- 如果使用 Big Data Europe 的 Hadoop Docker Image，请使用 3.2.1 版本

Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
2.10.0	2019 Oct 29	source (checksum signature)	binary (checksum signature)	Announcement
3.1.3	2019 Oct 21	source (checksum signature)	binary (checksum signature)	Announcement
3.2.1	2019 Sep 22	source (checksum signature)	binary (checksum signature)	Announcement
3.1.2	2019 Feb 6	source (checksum signature)	binary (checksum signature)	Announcement
2.9.2	2018 Nov 19	source (checksum signature)	binary (checksum signature)	Announcement

- Hadoop 官方文档：
- https://hadoop.apache.org/docs/current/index.html

- MapReduce 简明教程：
- https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

- 更多 Hadoop Docker Container 的使用方法和功能：
- https://clubhouse.io/developer-how-to/how-to-set-up-a-hadoop-cluster-in-docker/

- Cloudera
- https://www.cloudera.com/

- 由于 Hadoop 安装过于复杂，工业界普遍使用更为简单的 Cloudera 数据平台
- Cloudera 提供了一整套自动化安装和配置系统，安装过程全程可视化
- 另外还提供了一套监控（Provision & Monitor）系统，可以远程监控和管理 Hadoop 及其相关的应用平台
- Cloudera 还提供了多种多样的人工智能（AI）及大数据处理平台工具

- Hadoop Platform and Application Framework
- https://www.coursera.org/learn/hadoop

UC San Diego

目录

Spark 简介

- Apache Spark
- http://spark.apache.org/

- Apache Spark 是一个开源集群运算框架
- 2009 年由 Matei Zaharia 在加州大学伯克利分校 AMPLab 开创，2010 年通过 BSD 许可协议开源发布
- 使用 Spark 需要搭配集群资源管理器和分布式存储系统
- 集群资源管理器：支持独立模式、Hadoop YARN、Apache Mesos、Kubernetes
- 分布式存储：支持 HDFS、Alluxio、Cassandra、OpenStack Swift、Amazon S3 等
- 在 2014 年有超过 465 位贡献者投入 Spark 开发，让其成为 Apache 软件基金会以及大数据众多开源项目中最为活跃的项目

Spark 简介

- UC Berkeley AMPLab
- https://amplab.cs.berkeley.edu/

amplab UC BERKELEY

- 加州大学伯克利分校的一个大数据分析实验室
- AMP = Algorithms、Machines、People
- 主要组织者：Michael J. Franklin、Michael I. Jordan、Ion Stoica
- 主要项目：Apache Mesos、Apache Spark、Alluxio

Spark 简介

- 弹性分布式数据集（Resilient Distributed Dataset，RDD）
- 相对于 Hadoop 的 MapReduce 以磁盘作为数据中介，Spark 使用内存作为数据中介，能在数据尚未写入磁盘时即在内存中分析运算
- Spark 核心提供了分布式任务调度和基本的 I/O 功能，其应用程序抽象（Abstract）被称为弹性分布式数据集（RDD），是一个可以并行操作、有容错机制的数据集合
- RDD 可以通过引用外部分布式存储系统的数据集创建，或者是通过对现有 RDD 的转换而创建（例如 Map、Filter、Reduce、Join 等）
- Spark RDD 的运算速度能做到比 Hadoop MapReduce 快 100 倍，即便是在磁盘中运行应用程序，Spark 也能快上 10 倍速度
- 通过高效率的 RDD，Spark 可以在用户多次对数据进行查询时保持高效的 I/O，非常适合用于机器学习算法

- Spark SQL & DataFrame
- Spark SQL 允许用户直接使用类似 SQL 的语法查询大规模数据
- Spark DataFrame 允许用户直接对不同格式的数据源进行数据库操作（例如 Join），支持的数据格式包括 Hive、Avro、Parquet、ORC、JSON、JDBC 等

Integrated

Seamlessly mix SQL queries with Spark programs.

Spark SQL lets you query structured data inside Spark programs, using either SQL or a familiar DataFrame API. Usable in Java, Scala, Python and R.

```
results = spark.sql(
  "SELECT * FROM people")
names = results.map(lambda p: p.name)
```

Apply functions to results of SQL queries.

Spark 简介

- MLlib
- MLlib 是 Spark 的分布式机器学习框架
- Spark 的 MLlib 比 Hadoop 的 Mahout 处理速度快近 10 倍，且扩展性更好
- MLlib 内置了许多常见的机器学习和统计算法

- 目前开发者们正在努力让 MLlib 支持分布式 GPU 加速：
- https://medium.com/rapids-ai/nvidia-gpus-and-apache-spark-one-step-closer-2d99e37ac8fd

Algorithms

MLlib contains many algorithms and utilities.

ML algorithms include:

- Classification: logistic regression, naive Bayes,...
- Regression: generalized linear regression, survival regression,...
- Decision trees, random forests, and gradient-boosted trees
- Recommendation: alternating least squares (ALS)
- Clustering: K-means, Gaussian mixtures (GMMs),...
- Topic modeling: latent Dirichlet allocation (LDA)
- Frequent itemsets, association rules, and sequential pattern mining

ML workflow utilities include:

- Feature transformations: standardization, normalization, hashing,...
- ML Pipeline construction
- Model evaluation and hyper-parameter tuning
- ML persistence: saving and loading models and Pipelines

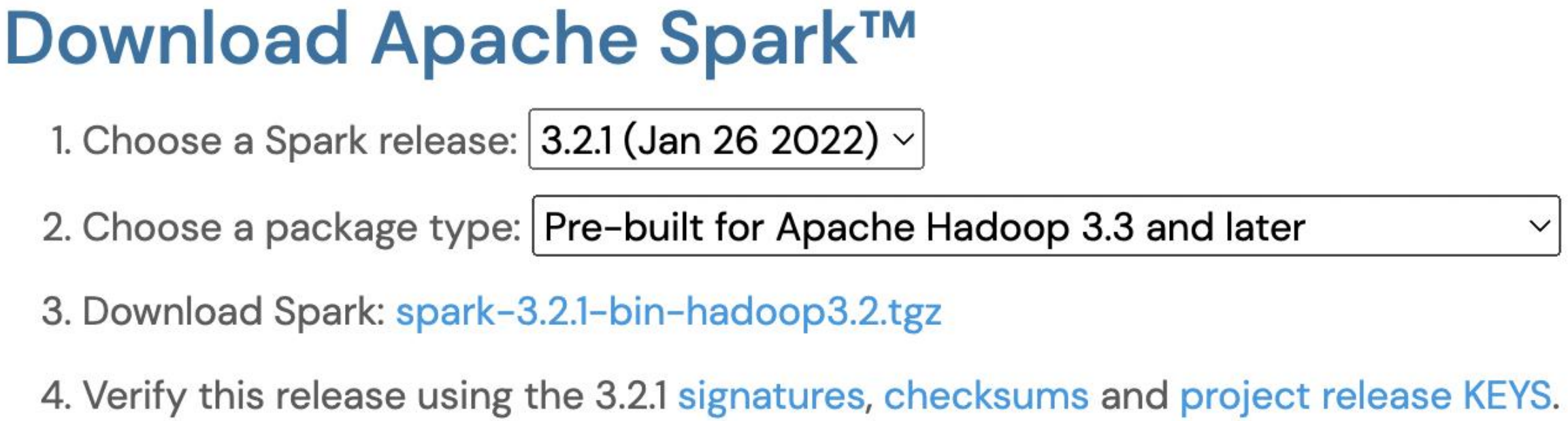
Other utilities include:

- Distributed linear algebra: SVD, PCA,...
- Statistics: summary statistics, hypothesis testing,...

Refer to the MLlib guide for usage examples.

安装 Spark

- 下载并安装独立模式的 Spark
- https://spark.apache.org/downloads.html
- 下载完毕后解压即可：`tar -zxvf spark-3.2.1-bin-hadoop3.2.tgz`



使用 Spark

- Spark 独立模式不需要部署服务，可以直接向其提交任务
- 运行 Spark 测试代码（计算圆周率）：./bin/run-example SparkPi 10
- 如果没有安装 Java，可以安装 JDK 8：sudo apt install openjdk-8-jdk

```
~/Workspace/spark-2.4.5-bin-hadoop2.7 » ./bin/run-example SparkPi 10
20/04/18 16:53:47 INFO SparkContext: Running Spark version 2.4.5
20/04/18 16:53:47 INFO SparkContext: Submitted application: Spark Pi
20/04/18 16:53:50 INFO TaskSetManager: Starting task 9.0 in stage 0.0 (TID 9, localhost, executor driver, partition 9, PROCESS_L
20/04/18 16:53:50 INFO Executor: Running task 9.0 in stage 0.0 (TID 9)
20/04/18 16:53:50 INFO TaskSetManager: Finished task 8.0 in stage 0.0 (TID 8) in 29 ms on localhost (executor driver) (9/10)
20/04/18 16:53:50 INFO Executor: Finished task 9.0 in stage 0.0 (TID 9). 824 bytes result sent to driver
20/04/18 16:53:50 INFO TaskSetManager: Finished task 9.0 in stage 0.0 (TID 9) in 21 ms on localhost (executor driver) (10/10)
20/04/18 16:53:51 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
20/04/18 16:53:51 INFO DAGScheduler: ResultStage 0 (reduce at SparkPi.scala:38) finished in 0.989 s
20/04/18 16:53:51 INFO DAGScheduler: Job 0 finished: reduce at SparkPi.scala:38, took 1.094654 s
Pi is roughly 3.141347141347141
20/04/18 16:53:51 INFO SparkUI: Stopped Spark web UI at http://dev-server:4040
20/04/18 16:53:51 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
20/04/18 16:53:51 INFO MemoryStore: MemoryStore cleared
```

使用 Spark

- PySpark
- https://pypi.org/project/pyspark/

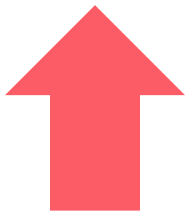
- Spark 推荐使用 Scala 或 Python 来撰写代码，PySpark 为 Spark 的 Python 版 SDK
- 安装 PySpark:
- `pip3 install -U pyspark`

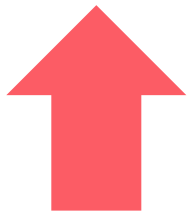
- 如果使用 Spark / PySpark 遇到类似以下的错误:
- `Unsupported class file major version 55`
- 这是因为 Spark 必须使用 Java 8/11，因此需要配置当前 JDK 版本为 8/11

使用 Spark

❖ Word Count 示例代码

```
import sys
from operator import add
from pyspark.sql import SparkSession
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: wordcount <file>", file=sys.stderr)
        sys.exit(-1)
    spark = SparkSession.builder.appName("PythonWordCount").getOrCreate()
    lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])
    counts = lines.flatMap(lambda x: x.split(' ')).map(lambda x: (x, 1)).reduceByKey(add)
    output = counts.collect()
    for (word, count) in output:
        print("%s: %i" % (word, count))
    spark.stop()
```





使用 Spark

- 向 Spark 提交任务：
- `./bin/spark-submit wordcount.py pg10.txt > pg10.log`

- 国王詹姆斯版圣经（The King James Version of the Bible）：
- http://www.gutenberg.org/cache/epub/10/pg10.txt

- 如果提示 Python 版本错误，需要将 Python 3 配置到环境变量：
- `export PYSPARK_PYTHON=python3`

- PySpark 支持 Python 3.7 以上版本，但 Python 3.9 偶尔会有问题，请特别留意

Apache Spark

使用 Spark

```
~/Workspace/spark-2.4.5-bin-hadoop2.7 » ./bin/spark-submit wordcount.py pg10.txt > pg10.log                valency@dev-server
20/04/18 17:56:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platfo    using builtin-java classes wh
ere applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
20/04/18 17:56:39 INFO SparkContext: Running Spark version 2.4.5
20/04/18 17:56:39 INFO SparkContext: Submitted application: PythonWordCount
```

```
20/04/18 17:56:48 INFO DAGScheduler: Job 0 finished: collect at /mnt/hgfs/Workspace/spark-2.4.5-bin-hadoop2.7/wordcount.py:13, t
ook 3.137409 s
20/04/18 17:56:48 INFO SparkUI: Stopped Spar    UI at http://dev-server:4040
20/04/18 17:56:48 INFO MapOutputTrackerMasterEn    : MapOutputTrackerMasterEndpoint stopped!
20/04/18 17:56:48 INFO MemoryStore: MemoryStore cleared
```

```
~/Workspace/spark-2.4.5-bin-hadoop2.7 » head pg10.log                                    1 □ valency@dev-server
The: 1914
Project: 79
Gutenberg: 22
EBook: 2
of: 34553
King: 56
James: 27
```

- Spark 官方文档：
- http://spark.apache.org/docs/latest/

- Spark 简明教程：
- http://spark.apache.org/docs/latest/quick-start.html

- 配置 Spark 集群：
- http://spark.apache.org/docs/latest/cluster-overview.html

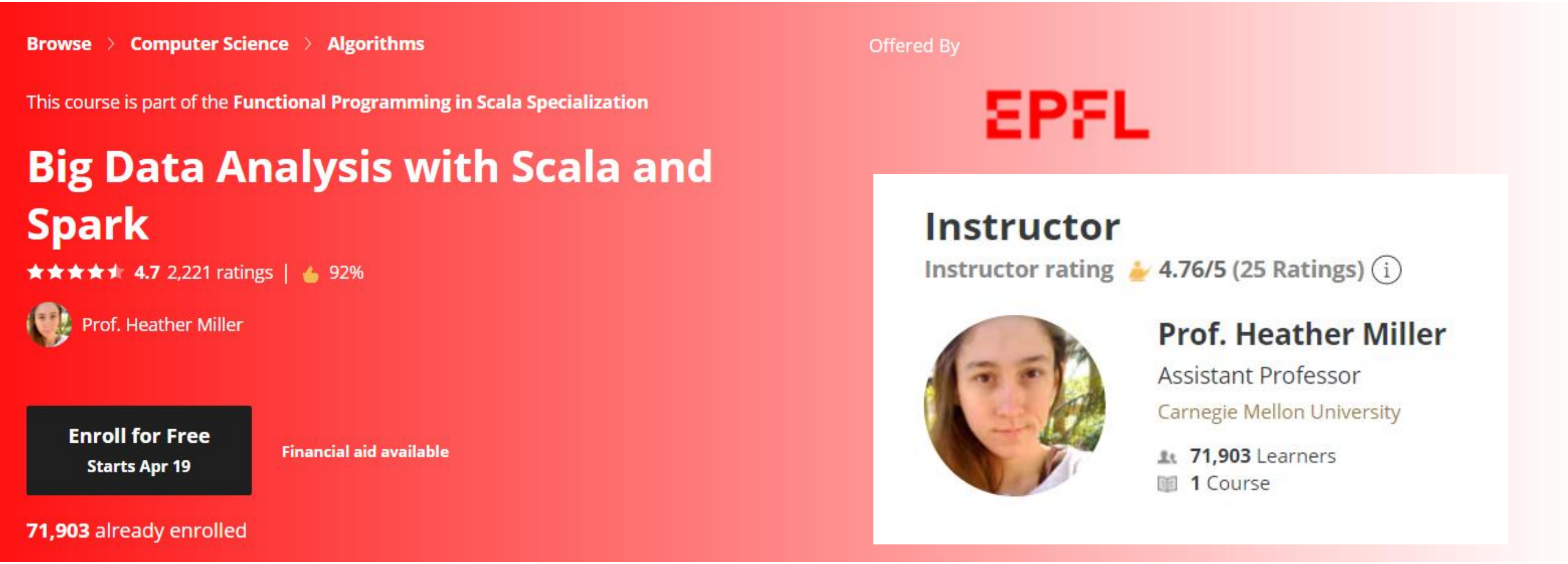
- IBM Data Engineering Professional Certificate
- https://www.coursera.org/professional-certificates/ibm-data-engineer/

- Course 1：Introduction to Data Engineering
- Course 2：Python for Data Science, AI & Development
- Course 3：Python Project for Data Engineering

- …

- Course 11：Introduction to Big Data with Spark and Hadoop
- Course 12：Data Engineering and Machine Learning using Spark
- Course 13：Data Engineering Capstone Project

- Big Data Analysis with Scala and Spark
- https://www.coursera.org/learn/scala-spark-big-data

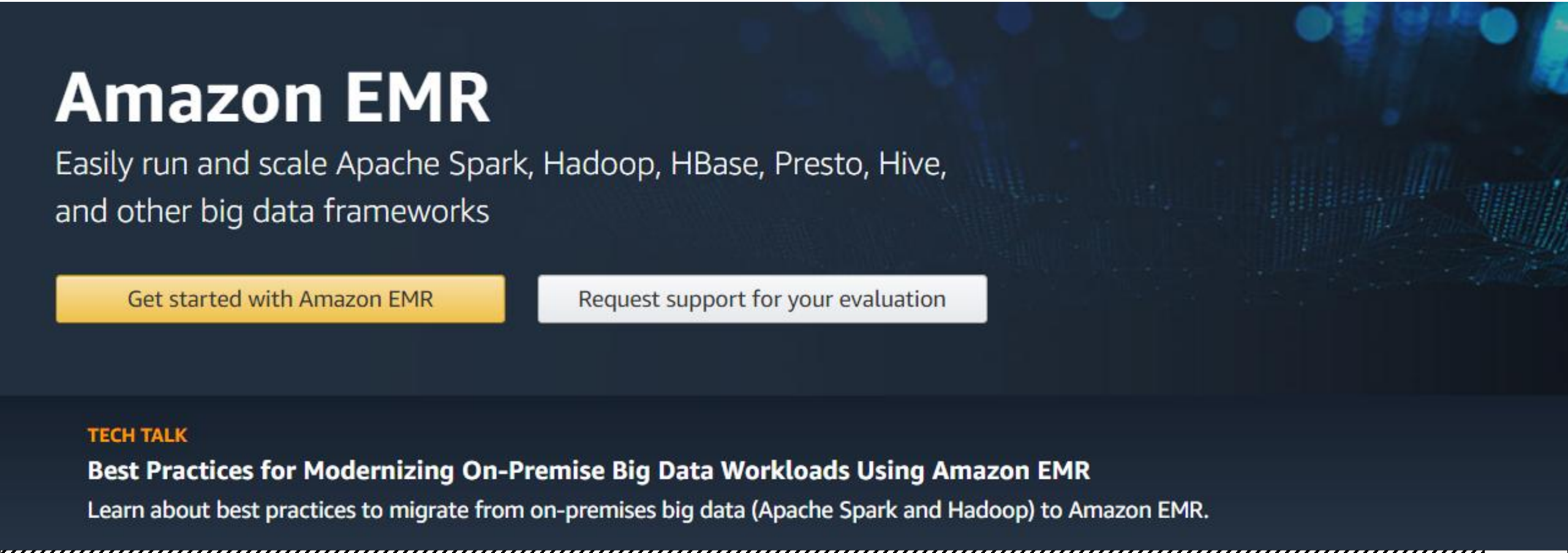


总结

- 其他常见的大数据处理平台及工具（基本上都属于 Apache 基金会管理）
- Avro：数据序列化工具
- Flume：分布式流数据处理系统
- Kafka：高效的分布式消息队列（目前最高效的分布式消息队列）
- Cassandra：高效的分布式 NoSQL 数据库（和 HBase 类似，但架构不同）
- HBase：高效的分布式 NoSQL 数据库（和 Cassandra 类似，但架构不同）
- Impala：高效的分布式 RDBMS 数据库（替代了 Hive）
- Mahout：基于 Hadoop 的机器学习库（效率不及 Spark MLlib）
- Tez：处理复杂有向无环图（DAG）应用程序的框架
- ZooKeeper：高效的应用程序协调框架
- …

总结

- Amazon Elastic MapReduce
- https://aws.amazon.com/emr/
- AWS 提供的 MapReduce 服务，按需收费



总结

- 阿里云 E-MapReduce
- https://www.aliyun.com/product/emapreduce
- 阿里云提供的 MapReduce 服务，按需收费

阿里云

总结

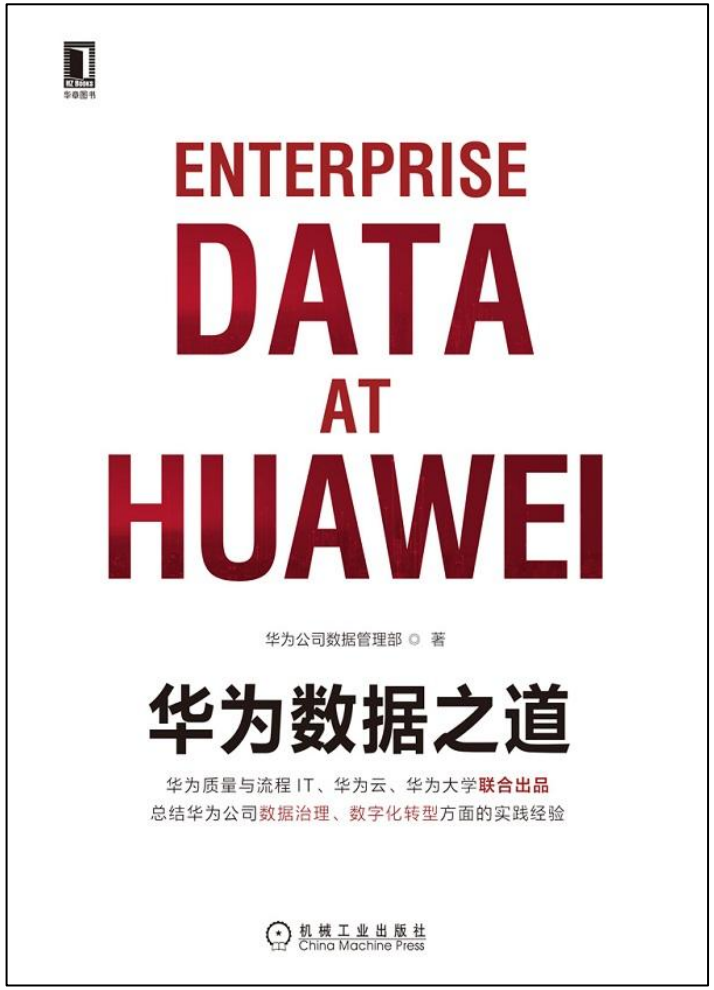
Google Cloud
cloud.google.com

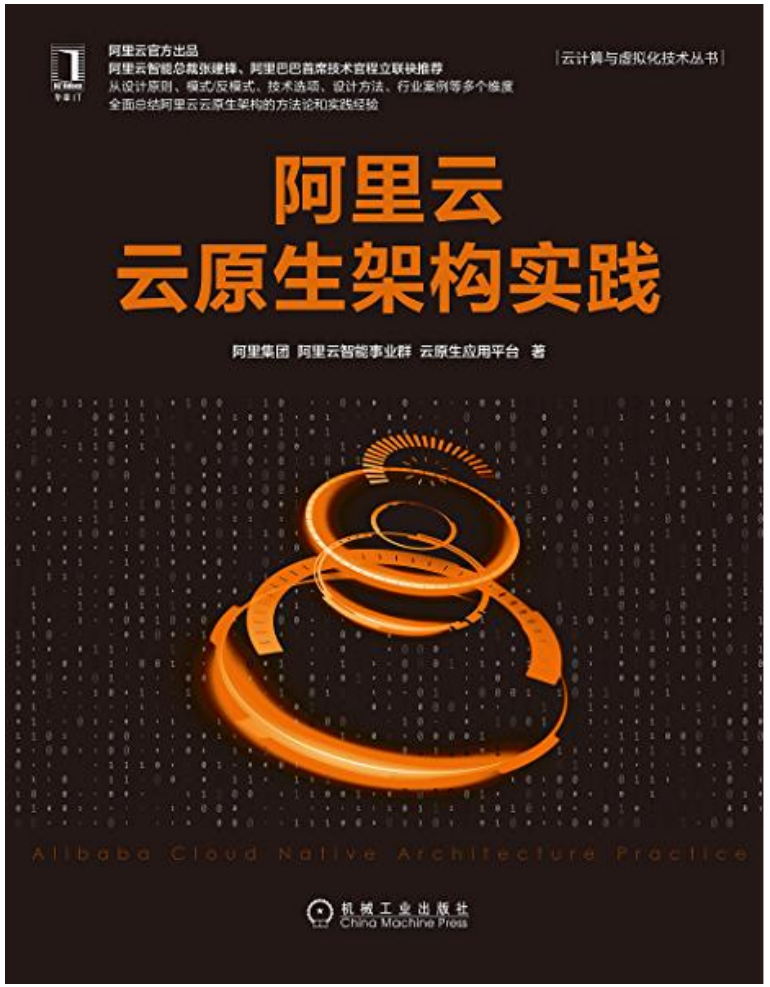
Amazon Web Services
aws.amazon.com

阿里云
aliyun.com

总结


大数据技术与应用专业规划教材
AWS-教育部-AWS产学合作专业综合改革项目规划教材
云计算与大数据
技术理论及应用
林伟伟 彭绍亮 编著
清华大学出版社


ENTERPRISE
DATA
AT
HUAWEI
华为公司数据管理部 著
华为数据之道
华为质量与流程IT、华为云、华为大学联合出品
总结华为公司数据治理、数字化转型方面的实践经验
机械工业出版社
China Machine Press


阿里云官方出品
阿里云智能总裁张建锋、阿里巴巴首席技术官程立联袂推荐
从设计原则、模式/反模式、技术选项、设计方法、行业案例等多个维度
全面总结阿里云云原生架构的方法论和实践经验
云计算与虚拟化技术丛书
阿里云
云原生架构实践
阿里集团 阿里云智能事业群 云原生应用平台 著
Alibaba Cloud Native Architecture Practice
机械工业出版社
China Machine Press

THANK YOU