云计算与大数据应用开发

实验四：大数据处理

丁烨
dingye@dgut.edu.cn
计算机科学与技术学院

2024-04-29

東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

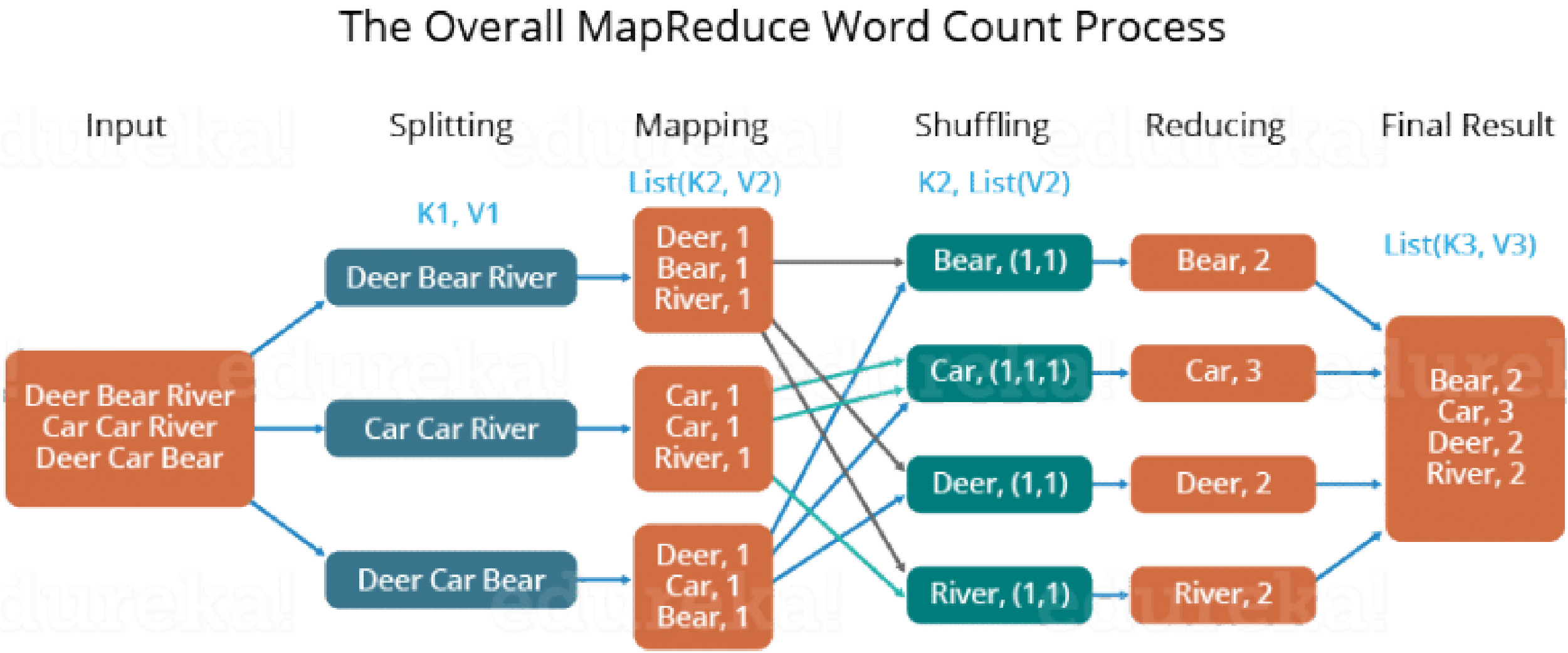
MapReduce 简介

- 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

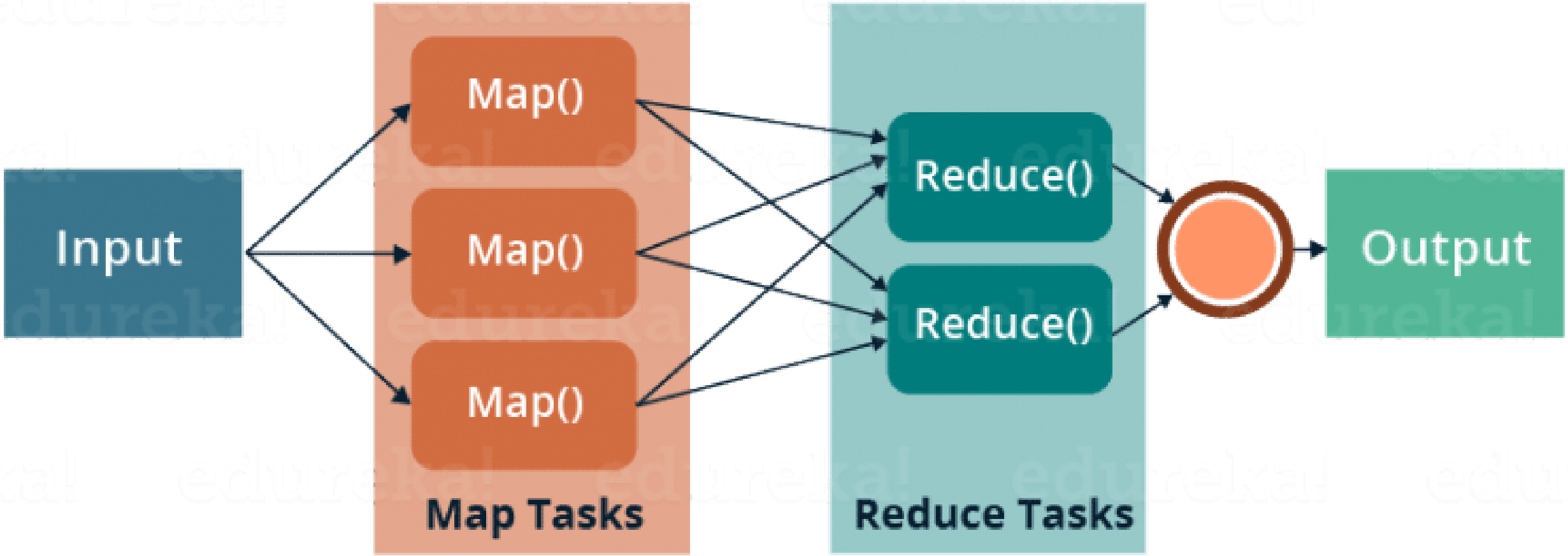
MapReduce 简介

- 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

MapReduce

MapReduce 简介



Spark 简介

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

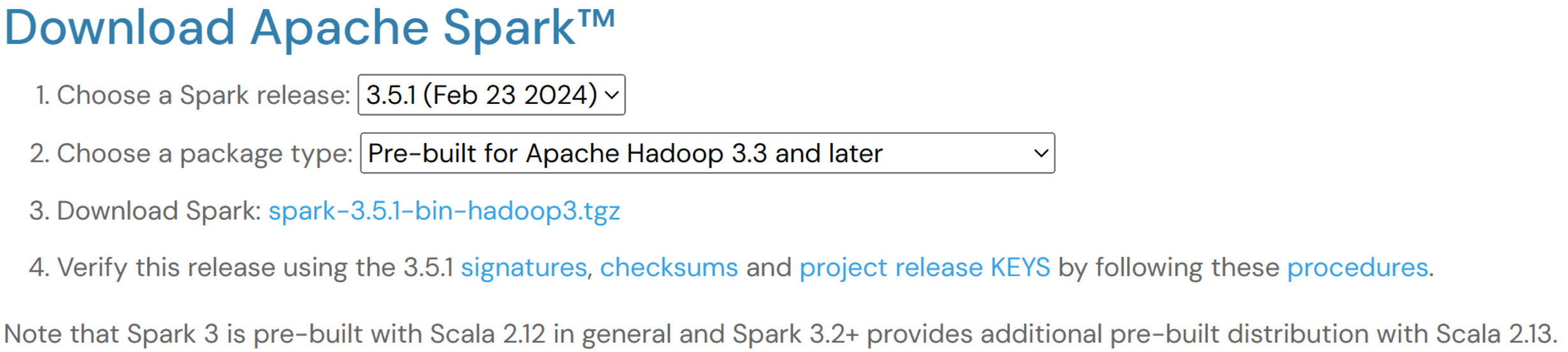
- Apache Spark 是一个开源集群运算框架
- 2009 年由 Matei Zaharia 在加州大学伯克利分校 AMPLab 开创，2010 年通过 BSD 许可协议开源发布
- 使用 Spark 需要搭配集群资源管理器和分布式存储系统
- 集群资源管理器：支持独立模式、Hadoop YARN、Apache Mesos、Kubernetes
- 分布式存储：支持 HDFS、Alluxio、Cassandra、OpenStack Swift、Amazon S3 等
- 在 2014 年有超过 465 位贡献者投入 Spark 开发，让其成为 Apache 软件基金会以及大数据众多开源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

Spark 简介

- 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

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-3.5.1-bin-hadoop3.tgz`



安装 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 17：sudo apt install openjdk-17-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

- Spark 推荐使用 Scala 或 Python 来撰写代码
- https://spark.apache.org/docs/latest/rdd-programming-guide.html

- 本节实验以 Python 为主

使用 Spark

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

- Spark 的 Python 版 SDK
- 安装 PySpark：
- `pip3 install -U pyspark`

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

- 还珠格格：
- http://down.xiaoshuodaquan.com/txt/39/还珠格格_琼瑶.txt

- 中文文本和英文文本不同，词与词之间没有空格，因此要进行“分词”

使用 Spark

- “结巴”中文分词
- https://github.com/fxsjy/jieba

- 使用 Python 开发的中文分词 SDK
- 利用 PaddlePaddle 深度学习框架，训练序列标注（双向 GRU）网络模型实现分词
- 支持词性标注、繁体分词、自定义词典

- 安装“结巴”：
- `pip3 install -U jieba`

❖ 词性标注

标签	含义	标签	含义	标签	含义	标签	含义
n	普通名词	f	方位名词	s	处所名词	t	时间
nr	人名	ns	地名	nt	机构名	nw	作品名
nz	其他专名	v	普通动词	vd	动副词	vn	名动词
a	形容词	ad	副形词	an	名形词	d	副词
m	数量词	q	量词	r	代词	p	介词
c	连词	u	助词	xc	其他虚词	w	标点符号
PER	人名	LOC	地名	ORG	机构名	TIME	时间

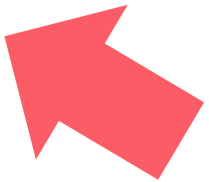
Apache Spark

使用 Spark

```
import sys
from operator import add

import jieba.posseg as posseg
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("WordCount").getOrCreate()
    ...
    spark.stop()
```

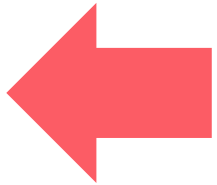


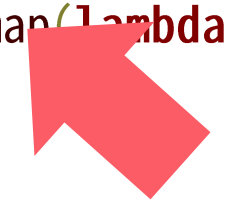
Apache Spark

使用 Spark

```
spark = SparkSession.builder.appName("WordCount").getOrCreate()
lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])
counts = lines.flatMap(
    lambda x: [w for w, f in posseg.cut(x, use_paddle=True) if f in (
        'n', 'f', 's', 'nr', 'ns', 'nt', 'nw', 'nz', 'PER', 'LOC', 'ORG'
    )]
).map(lambda x: (x, 1)).reduceByKey(add)

counts = counts.sortBy(lambda x: x[1], ascending=False)
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 hzgg.txt > result.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`

- Spark 运行 Python 代码需要指定 Python 运行环境，如使用 venv 会比较复杂
- 请尽量使用较新的 Python 版本（3.10+）

Apache Spark

使用 Spark

```
~/Workspace/spark-2.4.5-bin-hadoop2.7 » ./bin/spark-submit wordcount.py hzgg.txt > result.txt
20/04/25 17:39:27 WARN Utils: Your hostname, elementary-os resolves to a loopba address: 127.0.1.1; usi
20/04/25 17:39:27 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to anothe dress
20/04/25 17:39:27 WARN NativeCodeLoader: Unable to load native-hadoo ibrary fo ur platform... using
Using Spark's default log4j profile: org/apache/spark/log4j-defau roperties
20/04/25 17:39:28 INFO SparkContext: Running Spark version 2.4.5
20/04/25 17:39:28 INFO SparkContext: Submitted application: WordCount
20/04/25 17:39:28 INFO SecurityManager: Changing view acls to: valency
20/04/25 17:39:28 INFO SecurityManager: Changing modify acls to: valency
20/04/25 17:39:28 INFO SecurityManager: Changing view acls groups to:
20/04/25 17:39:28 INFO SecurityManager: Changing modify acls groups to:
20/04/25 17:39:28 INFO SecurityManager: SecurityManager: authentication disabl di acls disabled; users
 with modify permissions: Set(valency); groups with modify permissions: Set()
20/04/25 17:39:29 INFO Utils: Successfully started service 'sparkDriver' on port 37549.
20/04/25 17:39:29 INFO SparkEnv: Registering MapOutputTracker
20/04/25 17:39:29 INFO SparkEnv: Registering BlockManagerMaster
20/04/25 17:39:29 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper f
20/04/25 17:39:29 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
```

Apache Spark

使用 Spark

```
20/04/25 17:41:46 INFO Executor: Running task 0.0 in stage 1.0 (TID 1)
20/04/25 17:41:46 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks including 1 local blocks a
20/04/25 17:41:46 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 8 ms
20/04/25 17:41:46 INFO PythonRunner: Times: total = 24, boot = -404, init = 421, finish = 7
20/04/25 17:41:46 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 99430 bytes result sent to drive
20/04/25 17:41:46 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 88 ms on localhost (exec
20/04/25 17:41:46 INFO DAGScheduler: ResultStage 1 (collect at /mnt/hgfs/Workspace/spark-2.4.5-bin-hadoop
20/04/25 17:41:46 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
20/04/25 17:41:46 INFO DAGScheduler: Job 0 finished: collect at /mnt/hgfs/Workspace/spark-2.4.5-bin-hadoo
20/04/25 17:41:46 INFO SparkUI: Stopped Spark web UI at http://192.168.148.130:4040
20/04/25 17:41:46 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
20/04/25 17:41:46 INFO MemoryStore: MemoryStore cleared
20/04/25 17:41:46 INFO BlockManager: BlockManager stopped
20/04/25 17:41:46 INFO BlockManagerMaster: BlockManagerMaster stopped
20/04/25 17:41:46 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator s
20/04/25 17:41:46 INFO SparkContext: Successfully stopped SparkContext
--------------------------------------------------------------------------------
~/Workspace/spark-2.4.5-bin-hadoop2.7 »                    valency@elementary-os
```

使用 Spark

result.txt | wordcount.py | hzgg.txt

```
1  小燕子: 4458
2  紫薇: 3566
3  乾隆: 2216
4  人: 1672
5  尔康: 1252
6  皇上: 1210
7  永琪: 1098
8  皇后: 906
9  格格: 888
10 大家: 734
11 尔泰: 700
12 金琐: 612
13 皇阿玛: 564
14 里: 516
15 上: 484
16 令妃: 444
17 阿哥: 418
```

使用 Spark

- 如果遇到文本编码问题导致分析失败，可以尝试将 txt 文件从 GBK 转为 UTF-8
- `iconv -f gbk -t utf-8 source.txt > result.txt`
- 其中:
- `source.txt` 是原始文件（GBK 编码）
- `result.txt` 是结果文件（UTF-8 编码）
- 你可以用文本编辑器（例如 Sublime）打开文件看看是否正确

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

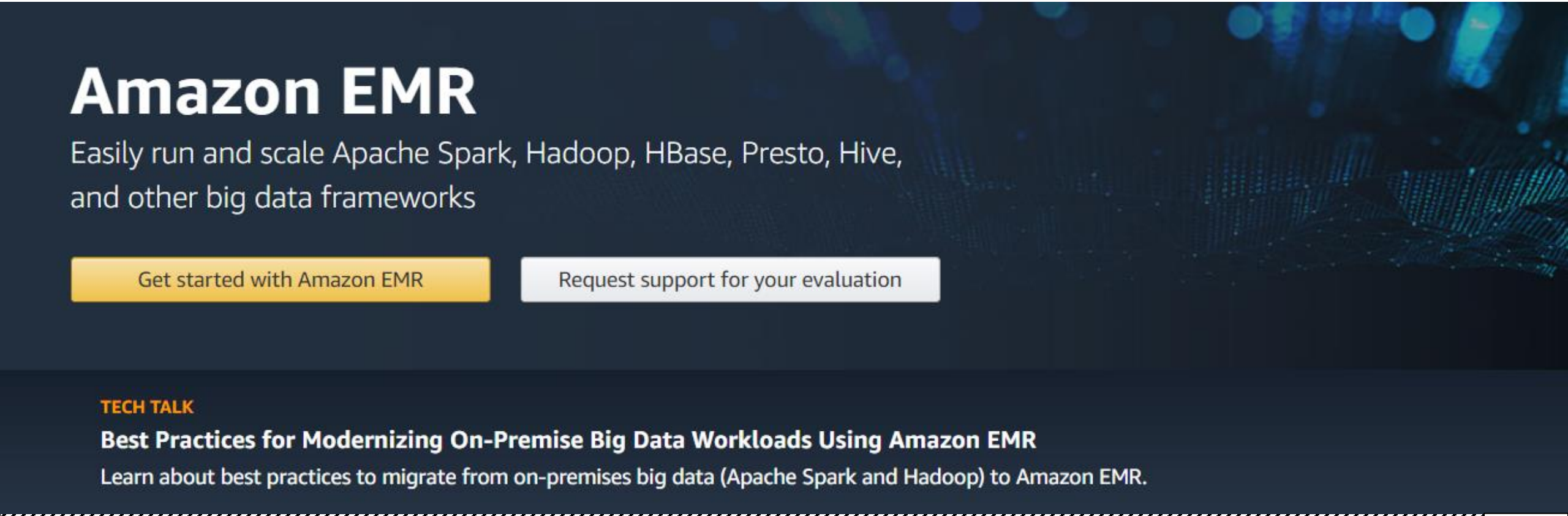
- Spark 简明教程：
- http://spark.apache.org/docs/latest/quick-start.html

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

PaaS MapReduce 服务

Amazon Elastic MapReduce

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



PaaS MapReduce 服务

阿里云 E-MapReduce

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

作业

- 使用 Hadoop 或 Spark 对任意一份长篇幅文本统计词频（Word Count）
- 不限制编程语言，可以使用 Java、Python、Scala 等
- 如果认为 Spark 或 Hadoop 安装繁琐，也可以使用 PaaS Spark / Hadoop 服务

- 输出结果必须按词频倒序排序，例如：
- 小燕子 4458
- 紫薇 3566
- …

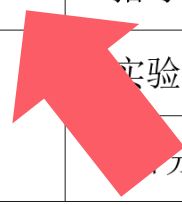
- 不要使用实验教程中的文本，请自行准备一份长篇幅文本，例如：
- https://www.bookben.net/
- 两份相同过程、相同结果的实验报告将被视为作弊

作业

- 在作业系统中下载并完成本实验课对应实验报告
- https://hw.dgut.edu.cn/
- 注意：所有标识为 * 的地方都需要填写
- 截止日期：2024-05-06 23:59:59

课程名称：云计算与大数据应用开发　　学期：2023 年春季

实验名称	虚拟化技术			实验序号	1
姓　　名	***	学　　号	***	班　　级	***
实验地点	***	实验日期	***	指导老师	丁烨
教师评语	-			实验成绩	-
				分制	100
同组同学	无				



四、　实验作业及分析

4.1　实验过程

1) *** 请将详细实验过程填写在此处 ***

4.2　实验结果

*** 请将实验结果截图填写在此处 ***

五、　实验总结

*** 请撰写一段 200 字左右的实验总结 ***


GOOD
LUCK!