实验七：Spark 编程实践

“大数据工程”课程实验报告		
题目：Spark 编程实践	学号姓名：郭加璐	日期：2024.5.23

实验环境：

虚拟机软件：VirtualBox 7.0.14

Linux 操作系统：Ubuntu Kylin 22.04.4，虚拟机名称 UbuntuRita

Java 版本：Oracle JDK 1.8

Java IDE：Eclipse

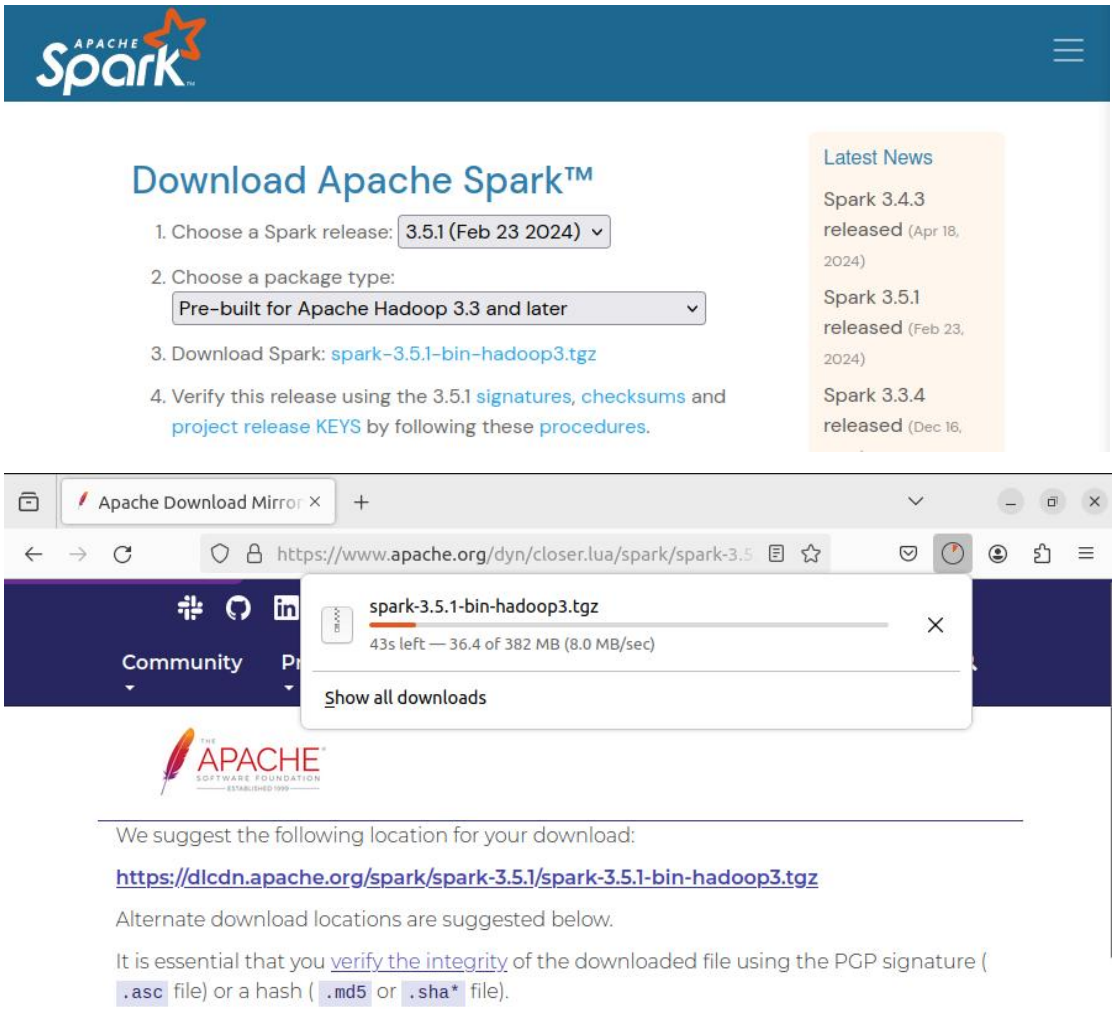
Hadoop：3.1.3

Spark：3.5.1

实验内容与完成情况：

一、安装部署 Spark

1. 下载 Spark 安装文件到本地~/Downloads/目录下：



2. 解压安装文件并修改权限

```
hadoop@UbuntuRita:~$ sudo tar -zxf ~/Downloads/spark-3.5.1-bin-hadoop3.tgz -C /usr/local/
[sudo] password for hadoop:
```

```
hadoop@UbuntuRita:/usr/local$ ls
bin              etc    hadoop  hive     kafka  man  sbin   spark-3.5.1-bin-hadoop3
eclipse-installer  games  hbase   include  lib    n    share  src
hadoop@UbuntuRita:/usr/local$ sudo mv ./spark-3.5.1-bin-hadoop3/ ./spark
hadoop@UbuntuRita:/usr/local$ sudo chown -R hadoop:hadoop ./spark
hadoop@UbuntuRita:/usr/local$
```

3. 配置文件 spark-env.sh

```
hadoop@UbuntuRita:/usr/local$ cd /usr/local/spark
hadoop@UbuntuRita:/usr/local/spark$ cp ./conf/spark-env.sh.template ./conf/spark-env.sh
hadoop@UbuntuRita:/usr/local/spark$ vim ./conf/spark-env.sh
hadoop@UbuntuRita:/usr/local/spark$
```

```
hadoop@UbuntuRita: /usr/local/spark
export SPARK_DIST_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)
#!/usr/bin/env bash

#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
```

4. 运行命令 bin/run-example SparkPi，检查 Spark 是否安装成功：

```
hadoop@UbuntuRita:/usr/local/spark$ bin/run-example SparkPi
24/05/23 19:20:54 WARN Utils: Your hostname, UbuntuRita resolves to a loopback address: 127.0.1.1;
 using 10.0.2.15 instead (on interface enp0s3)
24/05/23 19:20:54 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
24/05/23 19:20:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform...
 using builtin-java classes where applicable
24/05/23 19:20:55 INFO SparkContext: Running Spark version 3.5.1
24/05/23 19:20:55 INFO SparkContext: OS info Linux, 6.5.0-28-generic, amd64
24/05/23 19:20:55 INFO SparkContext: Java version 1.8.0_162
24/05/23 19:20:55 INFO ResourceUtils: ==============================================================
==
24/05/23 19:20:55 INFO ResourceUtils: No custom resources configured for spark.driver.
24/05/23 19:20:55 INFO ResourceUtils: ==============================================================
==
24/05/23 19:20:55 INFO SparkContext: Submitted application: Spark Pi
24/05/23 19:20:55 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(c
ores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script:
```

使用命令 bin/run-example SparkPi 2>&1 | grep "Pi is"进行过滤：

```
hadoop@UbuntuRita:/usr/local/spark$ bin/run-example SparkPi 2>&1 | grep "Pi is"
Pi is roughly 3.142435712178561
hadoop@UbuntuRita:/usr/local/spark$
```

5. 启动 Spark Shell：

```
hadoop@UbuntuRita:/usr/local/spark$ bin/spark-shell
24/05/23 19:22:55 WARN Utils: Your hostname, UbuntuRita resolves to a loopback address: 127.0.1.1;
 using 10.0.2.15 instead (on interface enp0s3)
24/05/23 19:22:55 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/05/23 19:23:00 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform...
 using builtin-java classes where applicable
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1716463381944).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.1
      /_/

Using Scala version 2.12.18 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_162)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

二、加载数据文件到本机的 HDFS

1. 下载文件 export.csv

2. 加载到 HDFS 中新建的/user/hadoop/spark 目录下

```
hadoop@UbuntuRita:~$ cd /usr/local/hadoop
hadoop@UbuntuRita:/usr/local/hadoop$ ./sbin/start-dfs.sh #启动hadoop
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [UbuntuRita]
hadoop@UbuntuRita:/usr/local/hadoop$  ./bin/hdfs dfs -ls .
Found 5 items
drwxr-xr-x   - hadoop supergroup          0 2024-04-01 17:27 MovieUserRatingsInfo
drwxr-xr-x   - hadoop supergroup          0 2024-03-13 00:18 input
drwxr-xr-x   - hadoop supergroup          0 2024-03-13 23:16 test
drwxr-xr-x   - hadoop supergroup          0 2024-05-11 21:18 user
drwxr-xr-x   - hadoop supergroup          0 2024-03-31 20:27 week5
hadoop@UbuntuRita:/usr/local/hadoop$ ./bin/hdfs dfs -ls /user/hadoop
Found 5 items
drwxr-xr-x   - hadoop supergroup          0 2024-04-01 17:27 /user/hadoop/MovieUserRatingsInfo
drwxr-xr-x   - hadoop supergroup          0 2024-03-13 00:18 /user/hadoop/input
drwxr-xr-x   - hadoop supergroup          0 2024-03-13 23:16 /user/hadoop/test
drwxr-xr-x   - hadoop supergroup          0 2024-05-11 21:18 /user/hadoop/user
drwxr-xr-x   - hadoop supergroup          0 2024-03-31 20:27 /user/hadoop/week5
hadoop@UbuntuRita:/usr/local/hadoop$  ./bin/hdfs dfs -mkdir /spark
```

```
hadoop@UbuntuRita:/usr/local/hadoop$ ./bin/hdfs dfs -ls /spark
hadoop@UbuntuRita:/usr/local/hadoop$ ./bin/hdfs dfs -put /home/hadoop/datasets/export.csv /spark
2024-05-23 23:05:09,809 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTr
usted = false, remoteHostTrusted = false
hadoop@UbuntuRita:/usr/local/hadoop$ ./bin/hdfs dfs -ls /spark
Found 1 items
-rw-r--r--   1 hadoop supergroup     109949 2024-05-23 23:05 /spark/export.csv
hadoop@UbuntuRita:/usr/local/hadoop$
```

在 hdfs 中查看数据文件 export.csv 的内容：

```
hadoop@UbuntuRita:/usr/local/hadoop$ ./bin/hdfs dfs -cat /spark/export.csv
2024-05-23 23:07:12,258 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTr
usted = false, remoteHostTrusted = false
battery_level,c02_level,cca2,cca3,cn,device_id,device_name,humidity,ip,latitude,lcd,longitude,scal
e,temp,timestamp
8,868,US,USA,United States,1,meter-gauge-1xbYRYcj,51,68.161.225.1,38,green,-97,Celsius,34,14584440
54093
7,1473,NO,NOR,Norway,2,sensor-pad-2n2Pea,70,213.161.254.1,62.47,red,6.15,Celsius,11,1458444054119
2,1556,IT,ITA,Italy,3,device-mac-36TWSKiT,44,88.36.5.1,42.83,red,12.83,Celsius,19,1458444054120
6,1080,US,USA,United States,4,sensor-pad-4mzWkz,32,66.39.173.154,44.06,yellow,-121.32,Celsius,28,1
458444054121
4,931,PH,PHL,Philippines,5,therm-stick-5gimpUrBB,62,203.82.41.9,14.58,green,120.97,Celsius,25,1458
444054122
3,1210,US,USA,United States,6,sensor-pad-6al7RTAobR,51,204.116.105.67,35.93,yellow,-85.46,Celsius,
27,1458444054122
3,1129,CN,CHN,China,7,meter-gauge-7GeDoanM,26,220.173.179.1,22.82,yellow,108.32,Celsius,18,1458444
054123
0,1536,JP,JPN,Japan,8,sensor-pad-8xUD6pzsQI,35,210.173.177.1,35.69,red,139.69,Celsius,27,145844405
4123
3,807,JP,JPN,Japan,9,device-mac-9GcjZ2pw,85,118.23.68.227,35.69,green,139.69,Celsius,13,1458444054
124
7,1470,US,USA,United States,10,sensor-pad-10BsywSYUF,56,208.109.163.218,33.61,red,-111.89,Celsius,
26,1458444054125
3,1544,IT,ITA,Italy,11,meter-gauge-11dlMTZty,85,88.213.191.34,42.83,red,12.83,Celsius,16,145844405
4125
0,1260,US,USA,United States,12,sensor-pad-12Y2kIm0o,92,68.28.91.22,38,yellow,-97,Celsius,12,145844
4054126
6,1007,IN,IND,India,13,meter-gauge-13GrojanSGBz,92,59.144.114.250,28.6,yellow,77.2,Celsius,13,1458
444054127
```

三、Spark Shell 统计文件行数

1. 在 Spark Shell 中读取 HDFS 上的上述文件：

定义文件路径：

```
scala> val filePath = "hdfs://localhost:9000/spark/export.csv"
filePath: String = hdfs://localhost:9000/spark/export.csv
```

使用 sc.textFile() 读取文件：

```
scala> val data = sc.textFile(filePath)
data: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/spark/export.csv MapPartitionsRDD[5
] at textFile at <console>:24
```

2. 统计出文件的行数：

```
scala> val lineCount = data.count()
lineCount: Long = 1001

scala> println(s"Total number of lines in the file: $lineCount")
Total number of lines in the file: 1001
```

四、基于 Java 语言的 Spark 应用程序统计文件行数

编写基于 Java 语言的 Spark 应用程序，读取 HDFS 中的上述文件，然后，统计出文件的行数

1. 安装 Maven

在 Maven 官网下载安装文件：

解压并修改权限：

```
hadoop@UbuntuRita:/usr/local/hadoop$ sudo tar -zxf ~/Downloads/apache-maven-3.9.6-bin.tar.gz -C /usr/l
ocal
[sudo] password for hadoop:
hadoop@UbuntuRita:/usr/local/hadoop$ cd /usr/local
```

```
hadoop@UbuntuRita:/usr/local$ ls
apache-maven-3.9.6  eclipse-installer  games   hbase  include  lib  n     share  src
bin                 etc                hadoop  hive   kafka    man  sbin  spark
hadoop@UbuntuRita:/usr/local$ sudo mv apache-maven-3.9.6/ ./maven
hadoop@UbuntuRita:/usr/local$ sudo chown -R hadoop ./maven
hadoop@UbuntuRita:/usr/local$
```

2. 创建 Java 应用程序

创建一个文件夹 sparkapp 作为应用程序根目录。使用 vim ./sparkapp/src/main/RowCount.java 在 ./sparkapp2/src/main/java 下建立一个名为 RowCount.java 的文件，并添加代码：

```
hadoop@UbuntuRita:/usr/local$ cd ~
hadoop@UbuntuRita:~$ mkdir -p ./sparkapp/src/main/java
hadoop@UbuntuRita:~$ vim ./sparkapp/src/main/RowCount.java
```

修改 RowCount.java：

```
hadoop@UbuntuRita: ~/sparkapp/src/main/java/com/example
```

```java
package com.example;

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.*;
import org.apache.spark.api.java.function.Function;

public class RowCount {
    public static void main(String[] args) {
        // 指定文件位于 HDFS 上的路径
        String hdfsFile = "hdfs://localhost:9000/spark/export.csv"; // 根据实际配置调整地址和端口

        // 配置 Spark
        SparkConf conf = new SparkConf().setMaster("local").setAppName("File Line Counter");

        // 初始化 JavaSparkContext
        JavaSparkContext sc = new JavaSparkContext(conf);

        // 从 HDFS 读取文件，创建一个 JavaRDD
        JavaRDD<String> fileData = sc.textFile(hdfsFile);

        // 计算文件的行数
        long lineCount = fileData.count();

        // 输出行数
        System.out.println("Total number of lines in the file: " + lineCount);

        // 关闭 SparkContext
        sc.close();
    }
}
```

3. 编译打包

通过 Maven 进行编译打包。

首先新建 pom.xml 文件，添加下述内容，以声明该独立应用程序的信息以及与 Spark 的依赖关系：

```
hadoop@UbuntuRita:~$ cd ~/sparkapp
hadoop@UbuntuRita:~/sparkapp$ vim pom.xml
```

```
hadoop@UbuntuRita: ~/sparkapp
```

```xml
<project>
    <groupId>cn.edu.xmu</groupId>
    <artifactId>simple-project</artifactId>
    <modelVersion>4.0.0</modelVersion>
    <name>Simple Project</name>
    <packaging>jar</packaging>
    <version>1.0</version>
    <repositories>
        <repository>
            <id>jboss</id>
            <name>JBoss Repository</name>
            <url>http://repository.jboss.com/maven2/</url>
        </repository>
    </repositories>
    <dependencies>
        <dependency> <!-- Spark dependency -->
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.11</artifactId>
            <version>2.4.0</version>
        </dependency>
    </dependencies>
</project>
```

检查整个应用程序的文件结构:

```
hadoop@UbuntuRita:~/sparkapp$ find .
.
./src
./src/main
./src/main/java
./src/main/RowCount.java
./pom.xml
```

将整个应用程序打包成 Jar 包：

```
hadoop@UbuntuRita:~/sparkapp$ /usr/local/maven/bin/mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< cn.edu.xmu:simple-project >----------------------
[INFO] Building Simple Project 1.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resource
s-plugin/3.3.1/maven-resources-plugin-3.3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources
-plugin/3.3.1/maven-resources-plugin-3.3.1.pom (8.2 kB at 5.3 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resource
s-plugin/3.3.1/maven-resources-plugin-3.3.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources
-plugin/3.3.1/maven-resources-plugin-3.3.1.jar (31 kB at 61 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler
```

打包成功：

```
[INFO] Building jar: /home/hadoop/sparkapp/target/simple-project-1.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  25:06 min
[INFO] Finished at: 2024-05-24T00:53:31+08:00
[INFO] ------------------------------------------------------------------------
hadoop@UbuntuRita:~/sparkapp$
```

4. 运行程序

将生成的 jar 包通过 spark-submit 提交到 Spark 中运行：

```
hadoop@UbuntuRita:~/sparkapp$ /usr/local/spark/bin/spark-submit --class com.example.RowCount --master loc
al ~/sparkapp/target/simple-project-1.0.jar
24/05/24 01:20:51 WARN Utils: Your hostname, UbuntuRita resolves to a loopback address: 127.0.1.1; using
10.0.2.15 instead (on interface enp0s3)
24/05/24 01:20:51 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
24/05/24 01:20:51 INFO SparkContext: Running Spark version 3.5.1
24/05/24 01:20:51 INFO SparkContext: OS info Linux, 6.5.0-28-generic, amd64
24/05/24 01:20:51 INFO SparkContext: Java version 1.8.0_162
24/05/24 01:20:51 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
24/05/24 01:20:52 INFO ResourceUtils: ==============================================================
24/05/24 01:20:52 INFO ResourceUtils: No custom resources configured for spark.driver.
24/05/24 01:20:52 INFO ResourceUtils: ==============================================================
24/05/24 01:20:52 INFO SparkContext: Submitted application: File Line Counter
24/05/24 01:20:52 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores ->
 name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vendor: , of
fHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1
.0)
24/05/24 01:20:52 INFO ResourceProfile: Limiting resource is cpu
24/05/24 01:20:52 INFO ResourceProfileManager: Added ResourceProfile id: 0
24/05/24 01:20:52 INFO SecurityManager: Changing view acls to: hadoop
24/05/24 01:20:52 INFO SecurityManager: Changing modify acls to: hadoop
24/05/24 01:20:52 INFO SecurityManager: Changing view acls groups to:
24/05/24 01:20:52 INFO SecurityManager: Changing modify acls groups to:
24/05/24 01:20:52 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users
 with view permissions: hadoop; groups with view permissions: EMPTY; users with modify permissions: hadoo
p; groups with modify permissions: EMPTY
24/05/24 01:20:52 INFO Utils: Successfully started service 'sparkDriver' on port 46641.
24/05/24 01:20:52 INFO SparkEnv: Registering MapOutputTracker
24/05/24 01:20:52 INFO SparkEnv: Registering BlockManagerMaster
24/05/24 01:20:52 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper f
or getting topology information
24/05/24 01:20:52 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
24/05/24 01:20:52 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/05/24 01:20:52 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-a7695747-6a5e-42c6-8960
-6eb206745735
24/05/24 01:20:52 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
24/05/24 01:20:52 INFO SparkEnv: Registering OutputCommitCoordinator
24/05/24 01:20:52 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
24/05/24 01:20:52 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
24/05/24 01:20:52 INFO Utils: Successfully started service 'SparkUI' on port 4041.
24/05/24 01:20:53 INFO SparkContext: Added JAR file:/home/hadoop/sparkapp/target/simple-project-1.0.jar a
t spark://10.0.2.15:46641/jars/simple-project-1.0.jar with timestamp 1716484851847
```

通过 grep 查看所需的输出结果：

```
hadoop@UbuntuRita:~/sparkapp$ /usr/local/spark/bin/spark-submit --class com.example.RowCount --master loc
al ~/sparkapp/target/simple-project-1.0.jar 2>&1 | grep "Total number of lines in the file:"
Total number of lines in the file: 1001
hadoop@UbuntuRita:~/sparkapp$
```

出现的问题：

无

解决方案（列出遇到的问题和解决办法，列出没有解决的问题）：

实验过程中参考以下资料：

1. https://dblab.xmu.edu.cn/blog/2501/
2. HDFS 编程实践（Hadoop3.1.3）_厦大数据库实验室博客 (xmu.edu.cn)

注：报告篇幅可根据实际题目情况进行调整。