**MapReduce 的原理及操作**

1. ****实验目标****

熟悉MapReduce的原理，熟练一些相关的基本操作。

1. ****实验内容****

* MapReduce 简介
* MapReduce 流程分析
* MapReduce 工作机制剖析

1. ****实验步骤及结果截图****

测试用例一

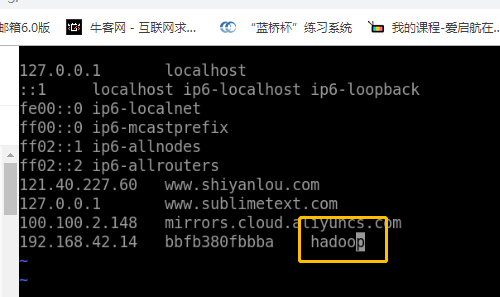
* 下载气象数据集部分数据，写一个 Map-Reduce 作业，求每年的最低温度。

1. 创建代码目录

（1）配置本机主机名为 hadoop，sudo 时需要输入 shiyanlou 用户的密码。将 hadoop 添加到最后一行的末尾。

sudo vim /etc/hosts # 将hadoop添加到最后一行的末尾

ping hadoop



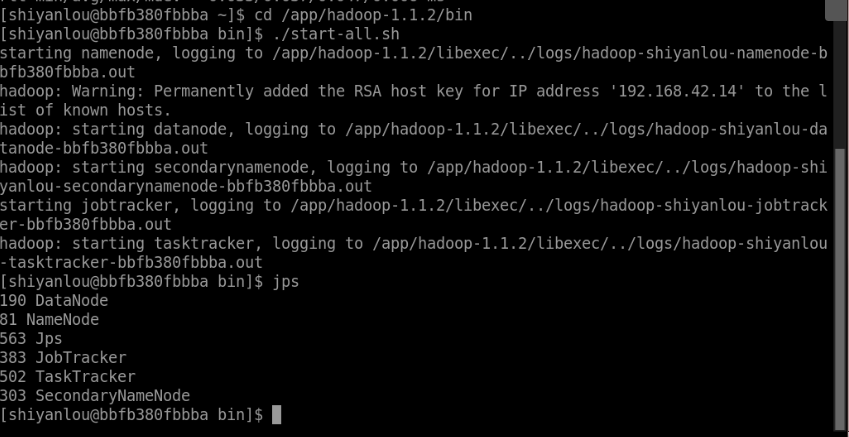
（2）启动 Hadoop

使用代码：

cd /app/hadoop-1.1.2/bin

./start-all.sh

jps # 查看启动的进程，确保 NameNode 和 DataNode 都有启动



1. 编写代码
2. 进入 /app/hadoop-1.1.2/myclass 目录，在该目录中建立 MinTemperature.java、MinTemperatureMapper.java 和 MinTemperatureReducer.java 代码文件，执行命令如下：

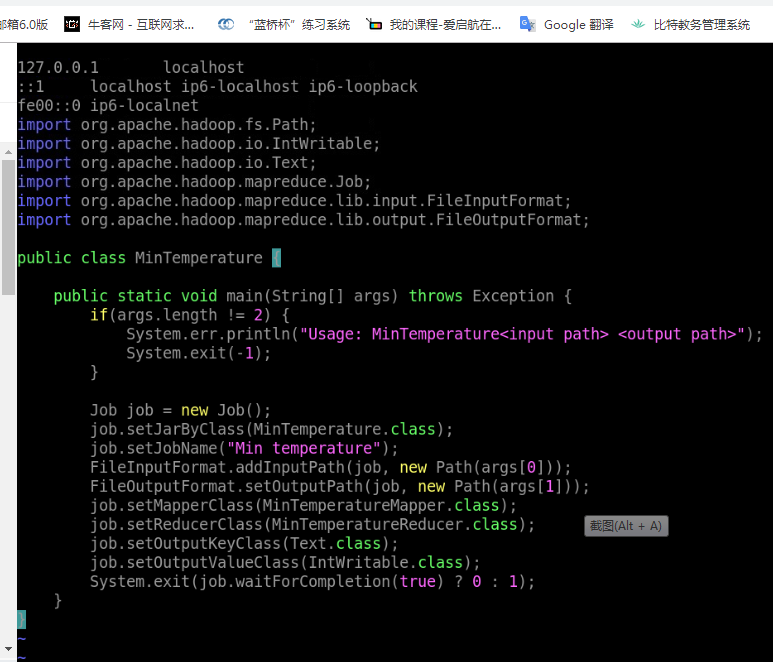
cd /app/hadoop-1.1.2/myclass/

vi MinTemperature.java

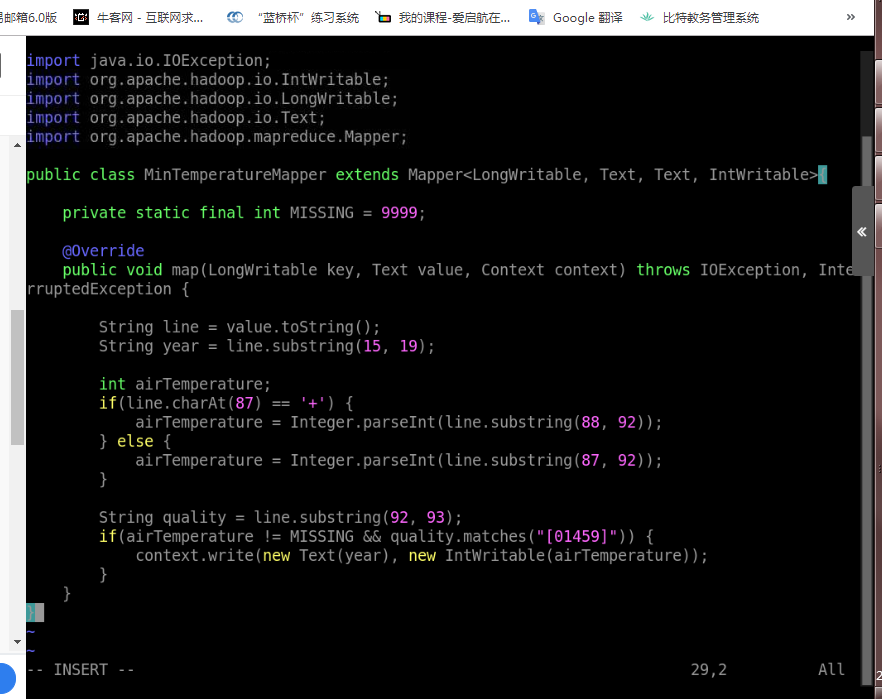
vi MinTemperatureMapper.java

vi MinTemperatureReducer.java

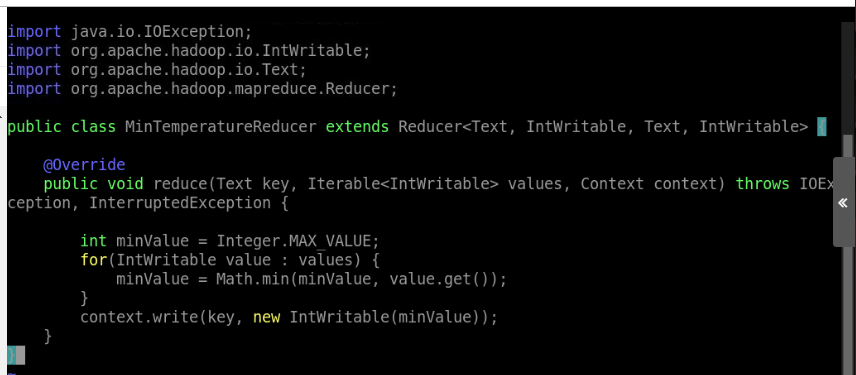
MinTemperature.java：



MinTemperatureMapper.java：

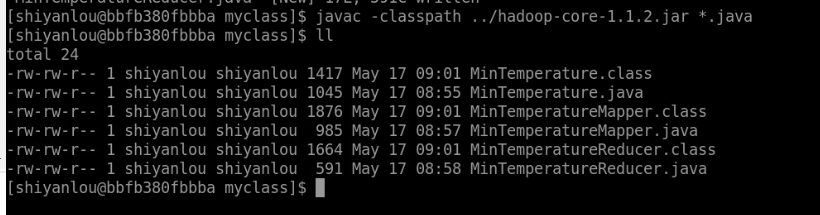


MinTemperatureReducer.java:



1. 编译代码
2. 在 /app/hadoop-1.1.2/myclass 目录中，使用如下命令对 java 代码进行编译，为保证编译成功，加入 classpath 变量，引入 hadoop-core-1.1.2.jar包：

javac -classpath ../hadoop-core-1.1.2.jar \*.java

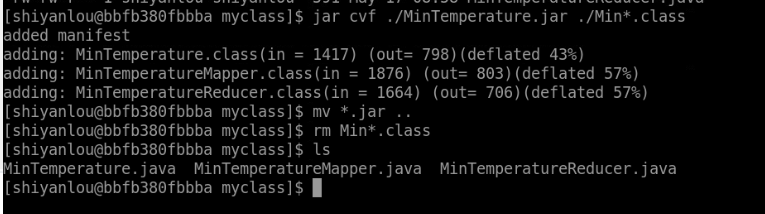


1. 打包编译文件
2. 把编译好的 class 文件打包，否则在执行过程会发生错误。把打好的包移动到上级目录并删除编译好的 class 文件：

jar cvf ./MinTemperature.jar ./Min\*.class

mv \*.jar ..

rm Min\*.class

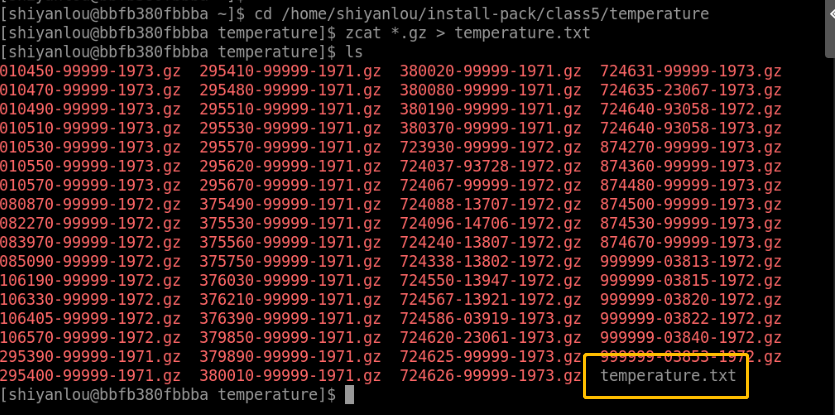


1. 解压气象数据并上传到 HDFS 中
2. 气象数据在 /home/shiyanlou/install-pack/class5/temperature 子目录下。把 NCDC 气象数据解压，并使用 zcat 命令把这些数据文件解压并合并到一个 temperature.txt 文件中。

cd /home/shiyanlou/install-pack/class5/temperature

zcat \*.gz > temperature.txt

tail temperature.txt

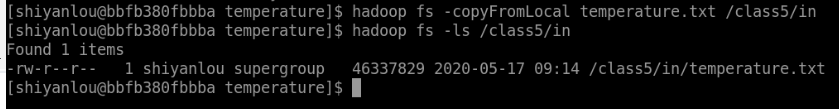


1. 合并后把这个文件上传到 HDFS 文件系统的 /class5/in 目录中：

hadoop fs -mkdir -p /class5/in

hadoop fs -copyFromLocal temperature.txt /class5/in

hadoop fs -ls /class5/in



1. 运行程序
2. 以 jar 的方式启动 MapReduce 任务，执行输出目录为 /class5/out：

cd /app/hadoop-1.1.2

hadoop jar MinTemperature.jar MinTemperature /class5/in/temperature.txt /class5/out

1. 查看结果
2. 执行成功后，查看 /class5/out 目录中是否存在运行结果，使用 cat 查看结果（温度需要除以 10）：

hadoop fs -ls /class5/out

hadoop fs -cat /class5/out/part-r-00000

测试用例二

* 如果求温度的平均值，能使用 combiner 吗？有没有变通的方法？

1. 编写代码

（1）进入 /app/hadoop-1.1.2/myclass 目录，在该目录中建立 AvgTemperature.java、AvgTemperatureMapper.java、AvgTemperatureCombiner.java 和AvgTemperatureReducer.java 代码文件，代码内容已在前面列出，执行命令如下：

cd /app/hadoop-1.1.2/myclass/

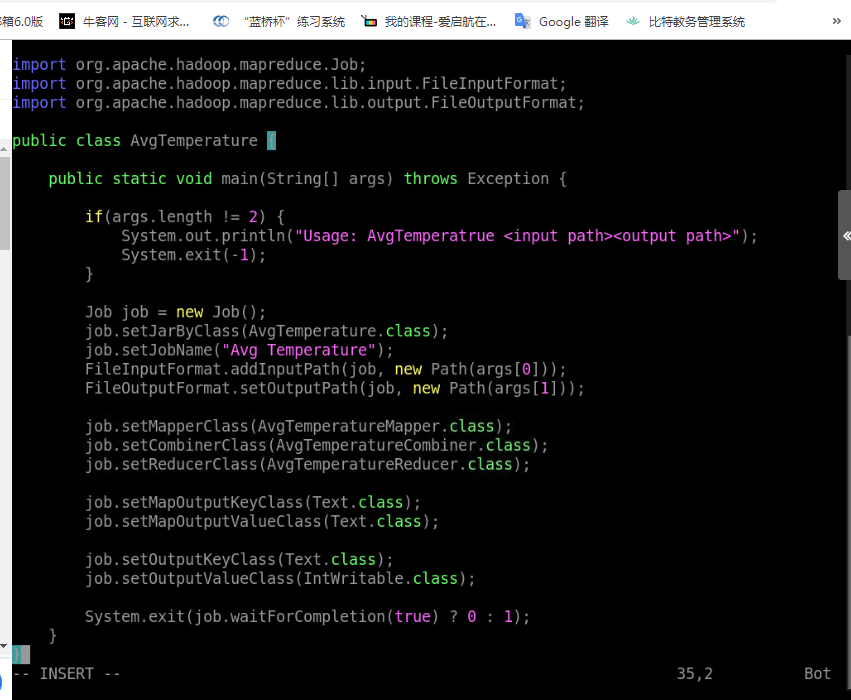
vi AvgTemperature.java

vi AvgTemperatureMapper.java

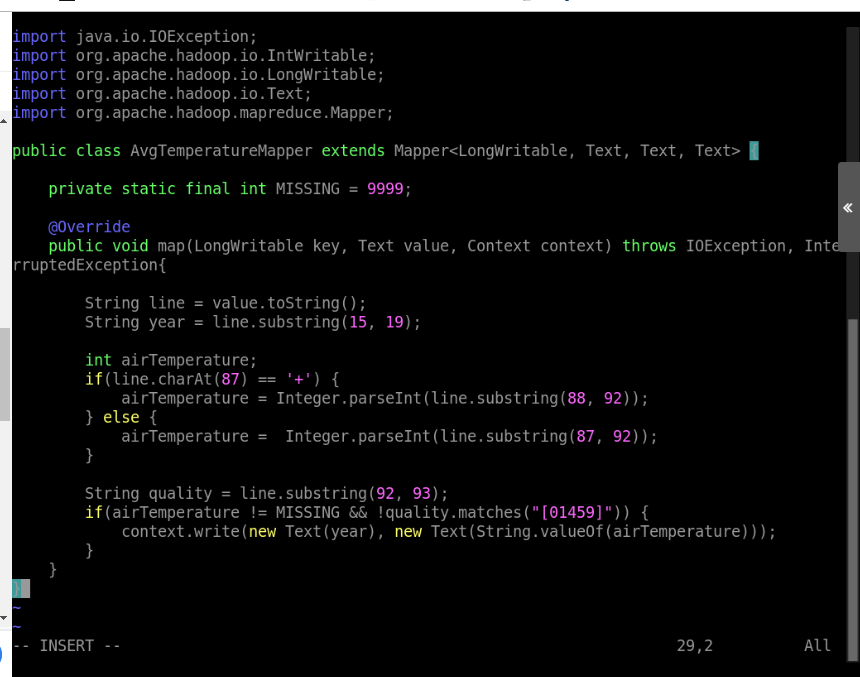
vi AvgTemperatureCombiner.java

vi AvgTemperatureReducer.java

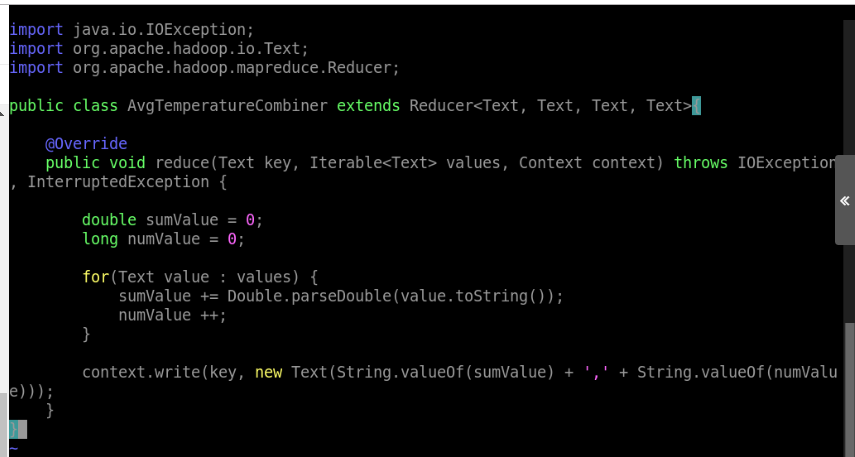
AvgTemperature.java：



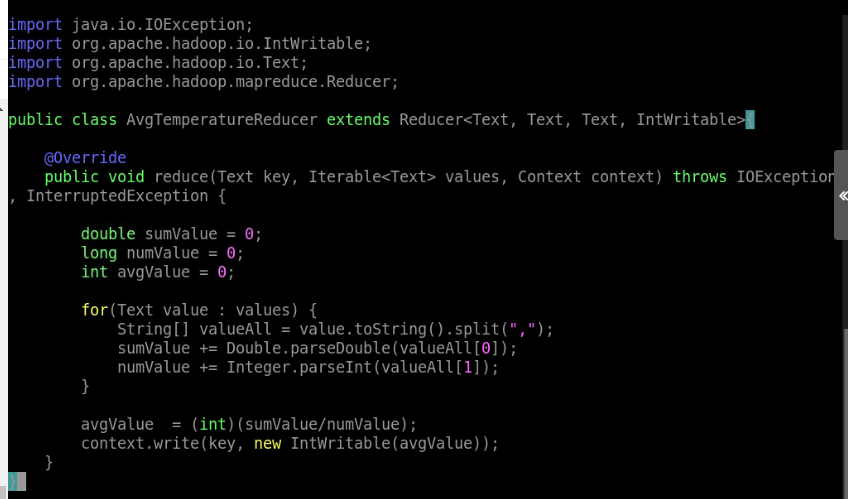
AvgTemperatureMapper.java：



AvgTemperatureCombiner.java：

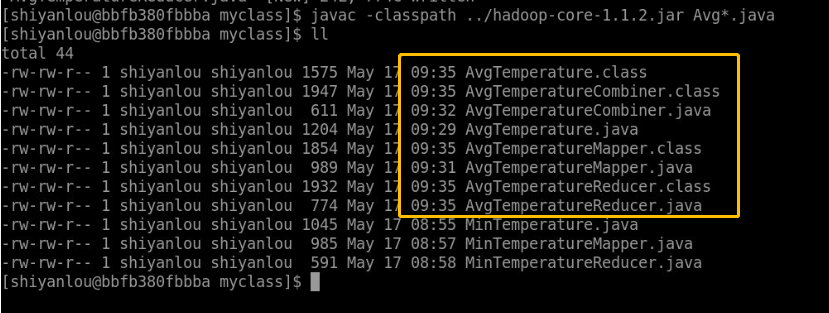


AvgTemperatureReducer.java:



1. 编译代码
2. 在 /app/hadoop-1.1.2/myclass 目录中，使用如下命令对 java 代码进行编译，为保证编译成功，加入 classpath 变量，引入 hadoop-core-1.1.2.jar包：

javac -classpath ../hadoop-core-1.1.2.jar Avg\*.java



1. 打包编译文件
2. 把编译好的 class 文件打包，否则在执行过程中会发生错误。把打好的包移动到上级目录并删除编译好的 class 文件：

jar cvf ./AvgTemperature.jar ./Avg\*.class

ls

mv \*.jar ..

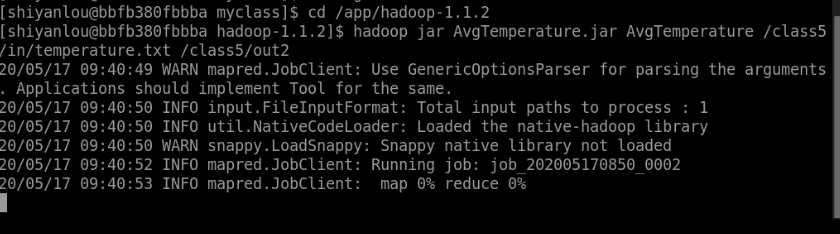
rm Avg\*.class

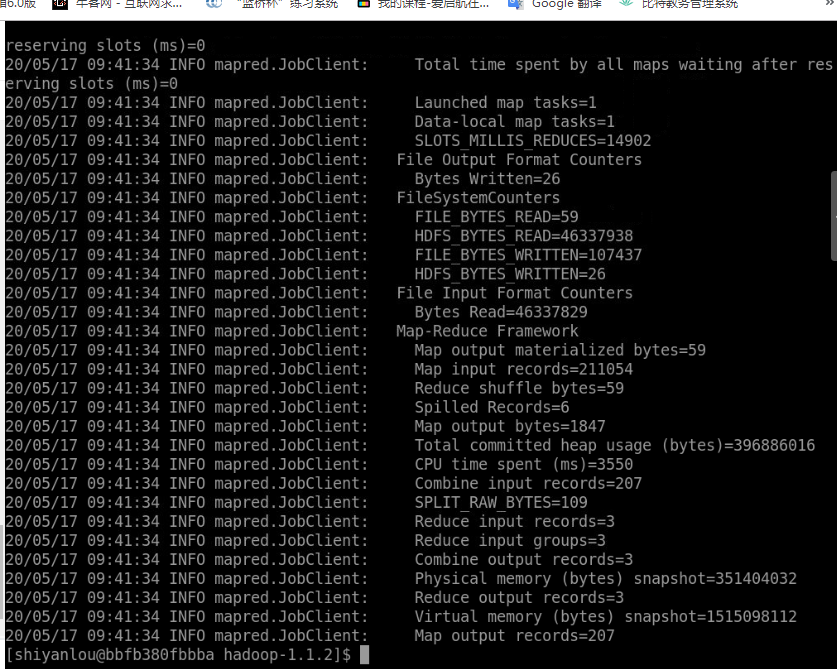


1. 运行程序
2. 以 jar 的方式启动 MapReduce 任务，执行输出目录为 /class5/out2：

cd /app/hadoop-1.1.2

hadoop jar AvgTemperature.jar AvgTemperature /class5/in/temperature.txt /class5/out2

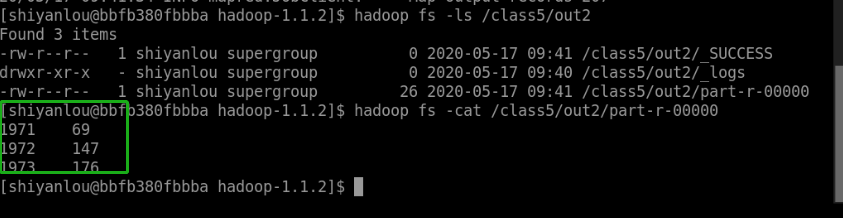




1. 查看结果
2. 执行成功后，查看 /class5/out2 目录中是否存在运行结果，使用 cat 查看结果（温度需要除以 10）：

hadoop fs -ls /class5/out2

hadoop fs -cat /class5/out2/part-r-00000



1. ****实验分析****

所给的两个测试用例，一个是 写一个Map-Reduce 作业，求每年的最低温度；另一个是如果求温度的平均值，问能否使用 combiner 。不能直接使用，因为求平均值和前面求最值存在差异，各局部最值的最值还是等于整体的最值的，但是对于平均值而言，各局部平均值的平均值将不再是整体的平均值了，所以不能直接用 combiner。可以通过变通的办法使用 combiner 来计算平均值，即在 combiner 的键值对中不直接存储最后的平均值，而是存储所有值的和个数，最后在 reducer 输出时再用和除以个数得到平均值。

1. ****心得体会****

通过本次实验，充分的了解了MapReduce 原理，以及相关的一系列操作，分别MapReduce 简介、MapReduce 流程分析、MapReduce 工作机制剖析三个方面领会了MapReduce的工作原理；同时还通过编写两个求温度的测试用例，熟悉了MapReduce相关的操作，增加了学习的兴趣和热情。