实验五 MapReduce 基础入门实验报告

江昱峰 21009200038

2023 年 11 月 16 日

1 背景介绍

假如有一个需求，统计图书馆书籍数量。你数 1 号书架，我数 2 号书架..... 这就是 Map。我们人越多，数书就更快。接下来把所有人的统计数加在一起。这就是 Reduce。一个简单的 MapReduce 过程就被我们实现了。

MapReduce 的思想核心是分而治之，适用于大量复杂的任务处理场景（大规模数据处理场景）。Map 负责分，即把复杂的任务分解为若干个简单的任务来并行处理。可以进行拆分的前提是这些小任务可以并行计算，彼此间几乎没有依赖关系。Reduce 负责合，即对 Map 阶段的结果进行全局汇总。

2 实验目的

实践并掌握 MapReduce 基础入门，具体包括以下三部分内容：

- Map 端程序编写；
- Reduce 端程序编写；
- Driver 端程序编写。

3 实验知识

操作 HDFS 文件 API 概述：Hadoop 中关于文件操作类基本上全部是在 org.apache.hadoop.fs 包中，这些 API 能够支持的操作包含：打开文件，读写文件，删除文件等。

4 实验要求

完成 MapReduce 基础入门，具体包括以下三部分任务：

- Map 端程序编写；
- Reduce 端程序编写；
- Driver 端程序编写。

5 实验环境

本次实验实验环境为青椒课堂平台的 Linux（Centos 7.5）操作系统。

6 实验步骤与结果分析

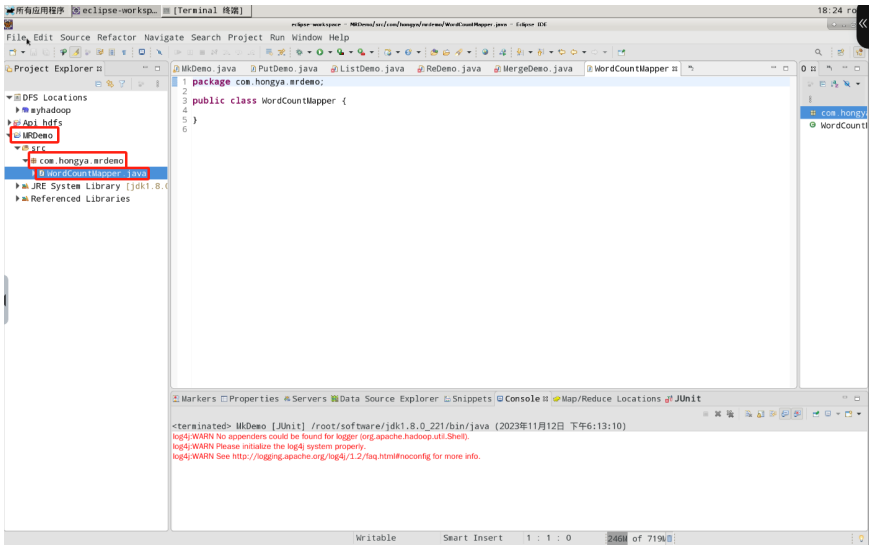
6.1 Map 端程序编写

6.1.1 任务 1：Map 端程序开发例一

本节任务当中，要实现 WordCount 的 Map 端程序编写，根据 Map 端编程步骤进行相关练习，理解 Map 端业务逻辑。

MapReduce 项目创建：

1. 项目名：MRDemo。
2. 包结构：com.hongya.mrdemo。
3. Map 端程序业务类：WordCountMapper。



需求介绍：按照给定的文本，内容如下所示，对其进行 WordCount 统计，完成 Map 端程序代码。

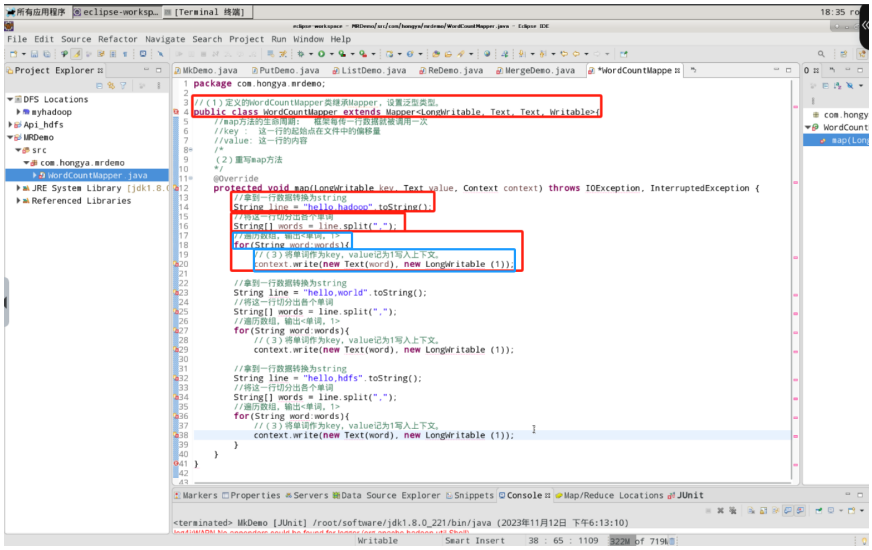
hello,hadoop

hello,word

hello,hdfs

Map 端编码步骤：

1. 定义的 WordCountMapper 类继承 Mapper，设置泛型类型。
2. 重写 map 方法：
   - 获取每一行数据并转换成 String。
   - 按逗号切分每行数据获取单词。
   - 遍历数组，获取每个单词。
3. 将单词作为 key，value 记为 1 写入上下文。



6.1.2 任务 2：Map 端程序开发例二

准备工作：在任务一包下继续创建类 Demo2_Map 进行本节 Map 端程序开发。

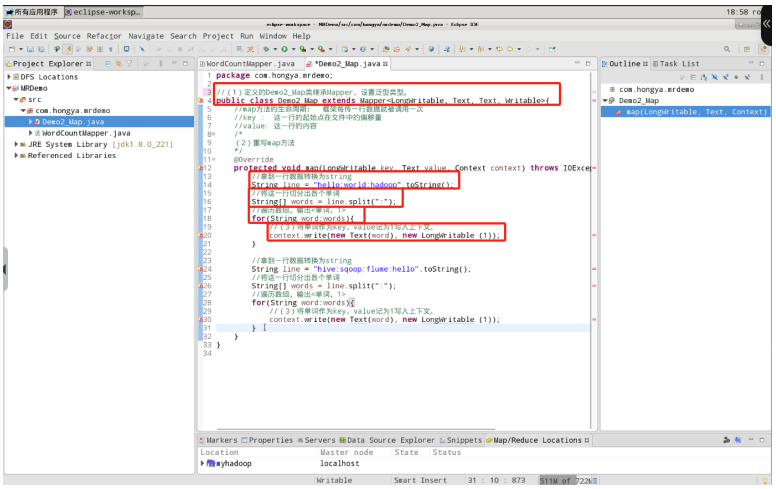
需求介绍：按照给定的文本，内容如下所示，对其进行 WordCount 统计，完成 Map 端程序代码。

hello:world:hadoop

hive:sqoop:lume:hello

Map 端编码步骤：

1. 定义的 Demo2_Map 类继承 Mapper，设置泛型类型。
2. 重写 map 方法：
   - 获取每一行数据并转换成 String。
   - 切分每行数据获取单词。
   - 遍历数组，获取每个单词。
3. 将单词作为 key，value 记为 1 写入上下文。

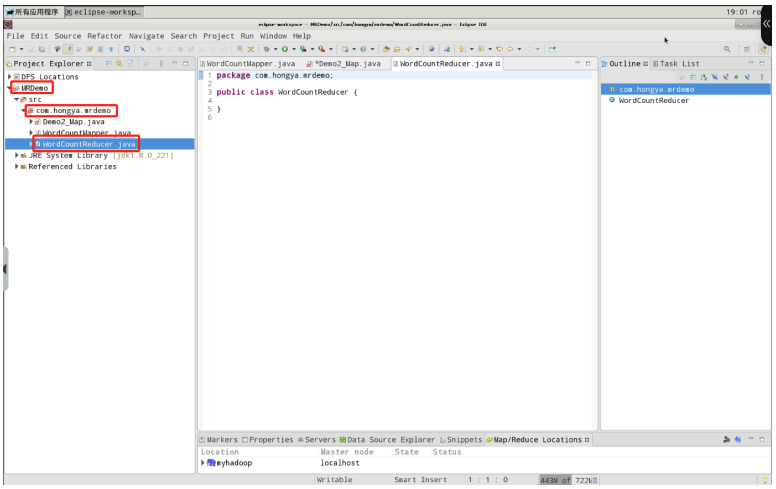


6.2 Reduce 端程序编写

6.2.1 Reduce 端程序编写例一

MapReduce 项目创建：

1. 项目名：MRDemo。
2. 包结构：com.hongya.mrdemo。
3. Reduce 端程序业务类：WordCountReducer。

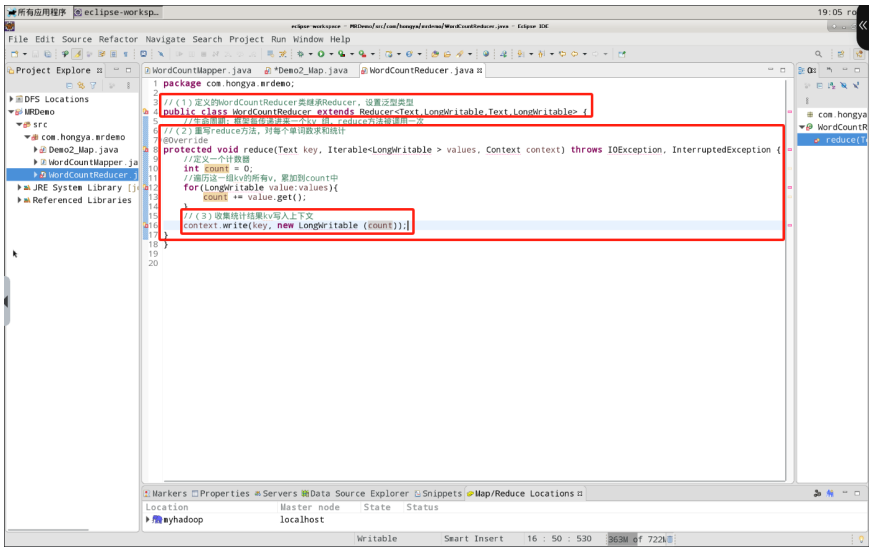


Reduce 端编程步骤：

1. 定义的 WordCountReducer 类继承 Reducer，设置泛型类型。

2. 重写 reduce 方法，对每个单词数求和统计。

3. 收集统计结果 kv 写入上下文。



6.2.2 任务 2：Reduce 端程序编写例二

准备工作：在任务一包下继续创建类 Demo2_Reduce 进行本节 Reduce 端程序开发。

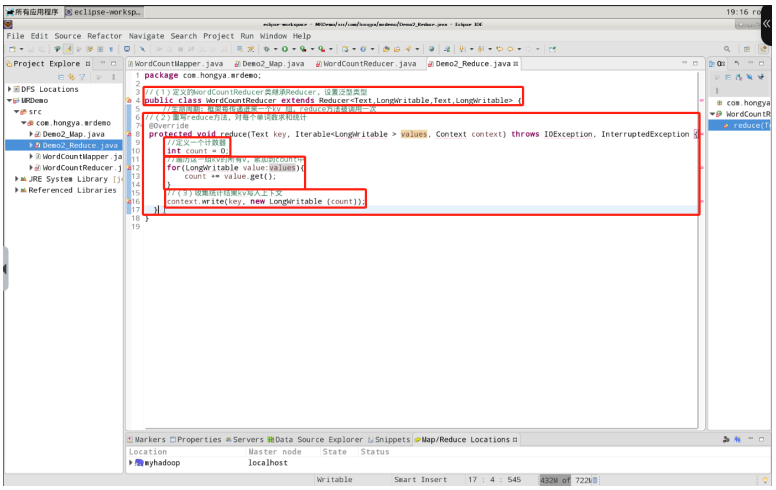
Reduce 端编码步骤：

1. 定义的 Demo2_Reduce 类继承 Reducer，设置泛型类型。

2. 重写 reduce 方法，对每个单词数求和统计。

   - 定义计数器 count。
   - 遍历出 value 并累加到 count 中。

3. 收集统计结果 kv 写入上下文。

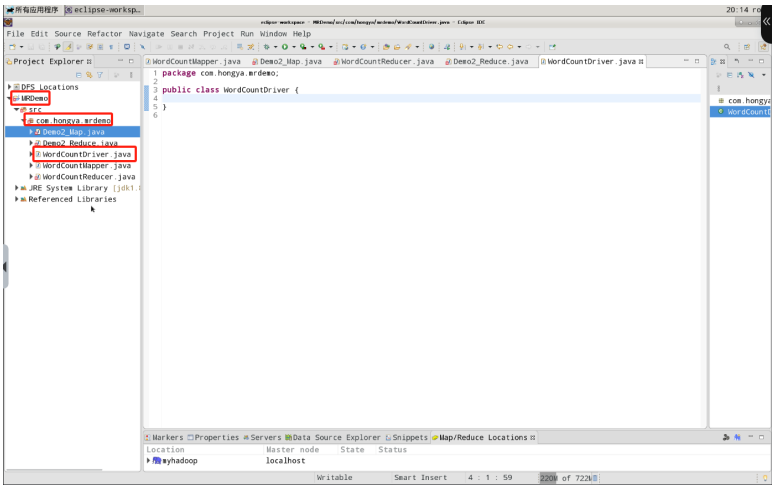


6.3 Driver 端程序编写

6.3.1 Driver 端程序编写例一

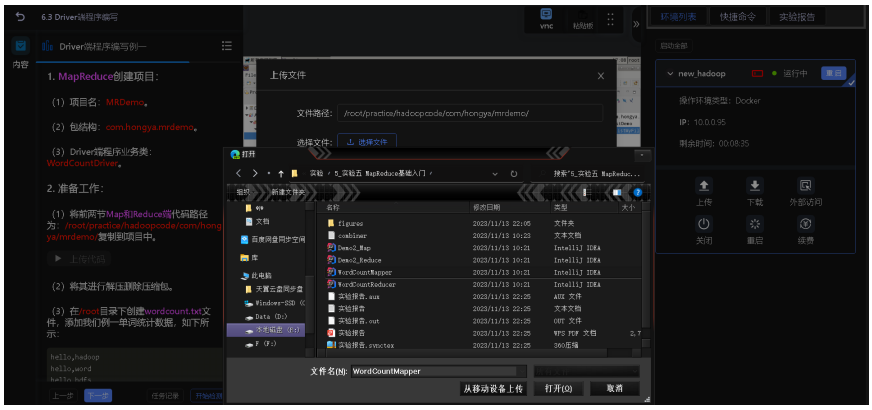
MapReduce 创建项目：

1. 项目名：MRDemo。
2. 包结构：com.hongya.mrdemo。
3. Driver 端程序业务类：WordCountDriver。



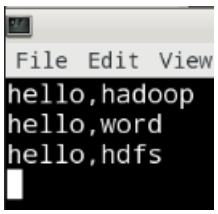
准备工作：

1. 点击按钮上传前两节 Map 和 Reduce 端代码，上传路径为:/root/eclipse-workspace/MRDemo/src/com/hongya/。



2. 将其进行解压删除压缩包。

3. 在/root 目录下创建 wordcount.txt 文件，添加我们例一单词统计数据，如下所示：
   hello,hadoop
   hello,word
   hello,hdfs



Driver 端代码逻辑：

1. 在 main 方法中创建 job 对象。

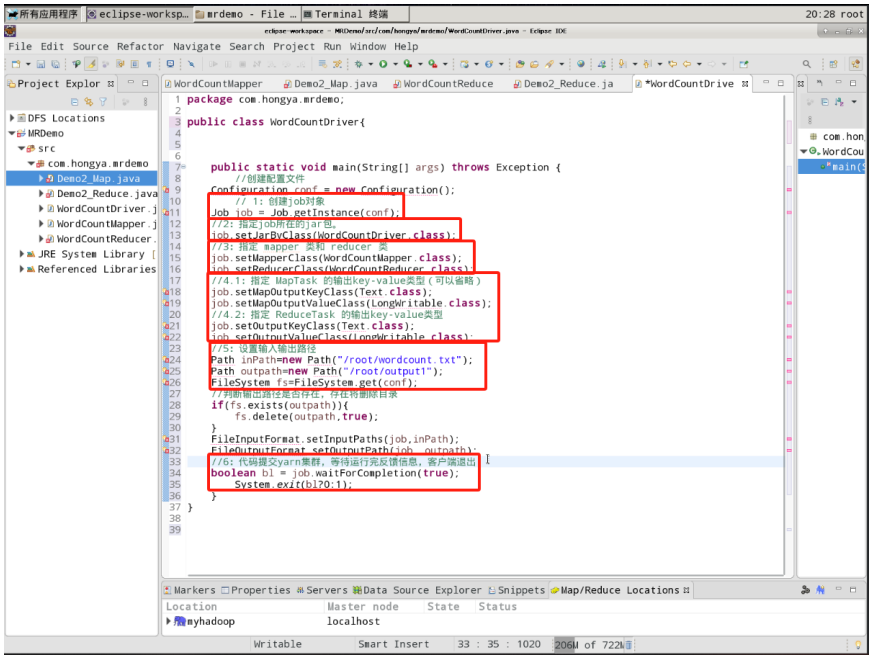
2. 指定 job 所在的 jar 包。

3. 指定 Mapper 类为例一的 WordCountMaaper，Reducer 类为例一的 WordCountReducer。

4. 指定 MapTask 和 ReduceTask 的输出 key-value 类型。若两者输出类型一致，MapTask 输出类型可省略。

5. 设置输入路径为/root/wordcount.txt，输出路径为/root/output1。

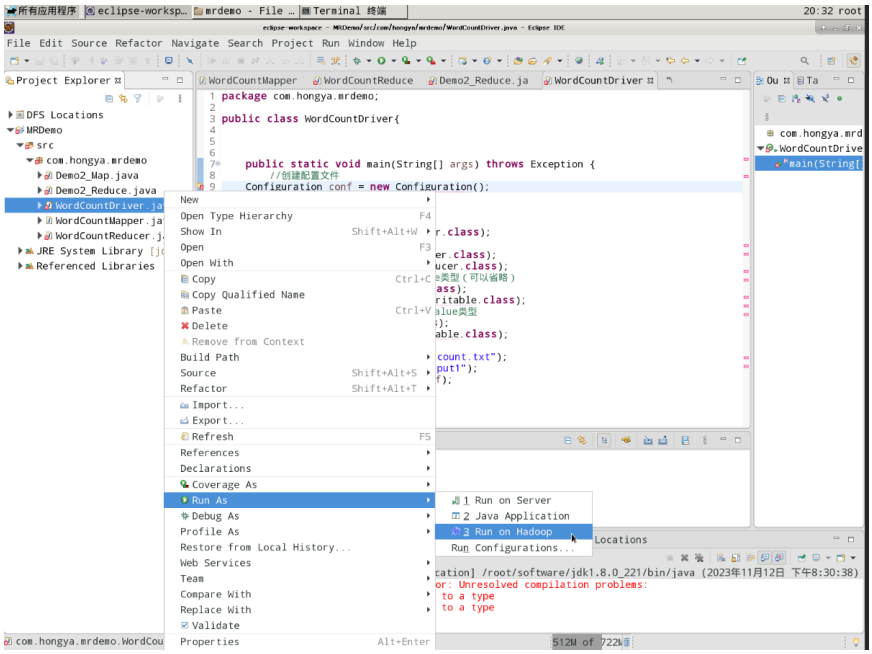
6. 将 job 提交给 yarn 集群。



至此，整个例一单词统计涉及到的程序已经编写完毕，接下来将运行程序查看处理结果。

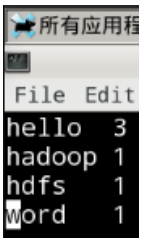
运行程序并查看结果：

1. 右击 WordCountDriver 类，选择 Run as->Run on Hadoop。



2. 等待程序执行。

3. 查看输出结果。会在指定的输出路径下生成 part-r-00000 文件，此文件为输出数据所在文件。



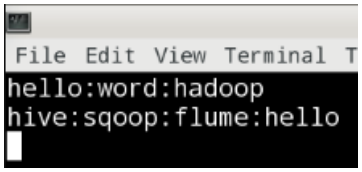
6.3.2 任务 2：Driver 端程序编写例二

准备工作：

1. 在任务一包结构下继续创建例二的 Driver 端程序类：Demo2_Driver。

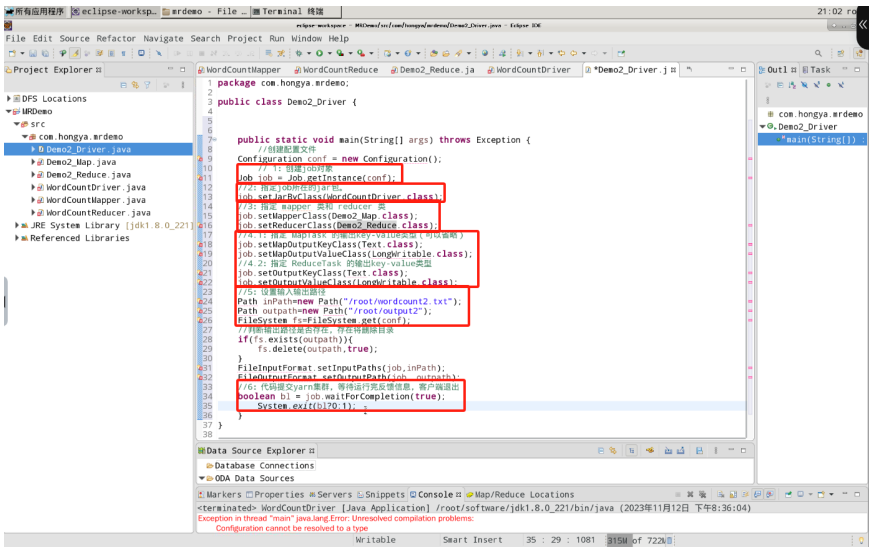
2. 在/root 目录下创建 wordcount2.txt 文件，添加我们例一单词统计数据，如下所示：

hello:word:hadoop
hive:sqoop:flume:hello



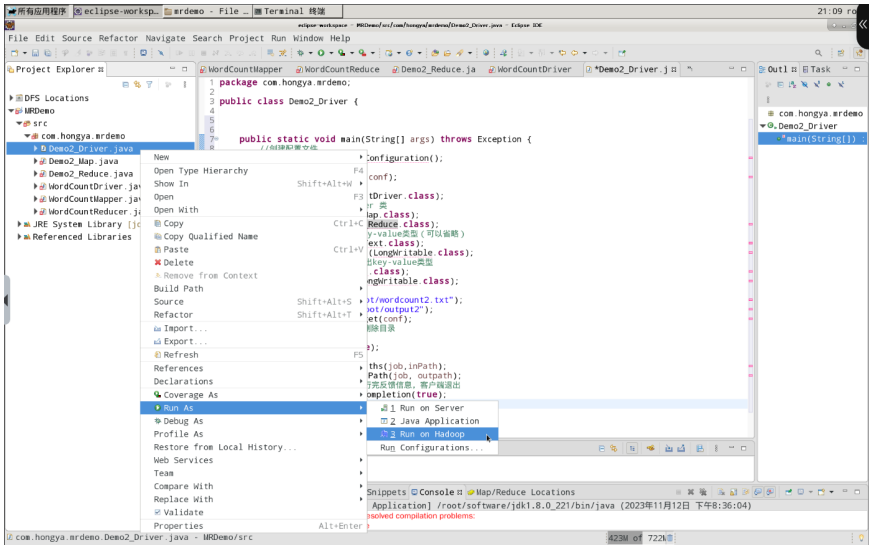
Driver 端代码逻辑：

1. 在 main 方法中创建 job 对象。
2. 指定 job 所在的 jar 包。
3. 指定 Mapper 类为例一的 Demo2_Map,Reducer 类为例一的 Demo2_Reduce。
4. 指定 MapTask 和 ReduceTask 的输出 key-value 类型。若两者输出类型一致，MapTask 输出类型可省略。
5. 设置输入路径为/root/wordcount2.txt，输出路径为/root/output2。
6. 将 job 提交给 yarn 集群。



运行程序并查看结果：

1. 右击 WordCountDriver 类，选择 Run as->Run as Map。



2. 等待程序执行。

3. 查看输出结果。程序执行成功，会在指定的输出路径下生成 part-r-00000 文件，此文件为输出数据所在文件。

7 结果分析

MapReduce 整体的作用类似于数据结构中的哈希表，可以分别对不同对象进行计数操作。

8 困难解决

本次实验较为简单，没有遇到困难。

9 心得体会

做完本次实验，除了掌握了实验目的部分中所有内容的收获之外，我还有以下几点心得体会：

- 对比分析了 MapReduce 和哈希表的异同点；
- 掌握了 Driver 端的功能及其对于 Map、Reduce 两端的作用。