作业五

一、设计思路：

1.开发环境与工具：

> 基于vscode，安装Java开发插件和Maven插件，借助maven在vscode上搭建hadoop开发环境

2.编写MapReduce程序：

> - 读取输入文件A和B，将它们作为输入数据。
> - 编写Mapper阶段，Mapper的任务是将输入数据按照指数编号和成分股代码进行映射，然后输出（指数编号，成分股代码）作为键值对。
> - 编写Reducer阶段，Reducer的任务是接收Mapper输出的键值对，并在Reducer内部进行数据合并和去重操作。
> - 最后，Reducer将合并后的数据写入输出文件。

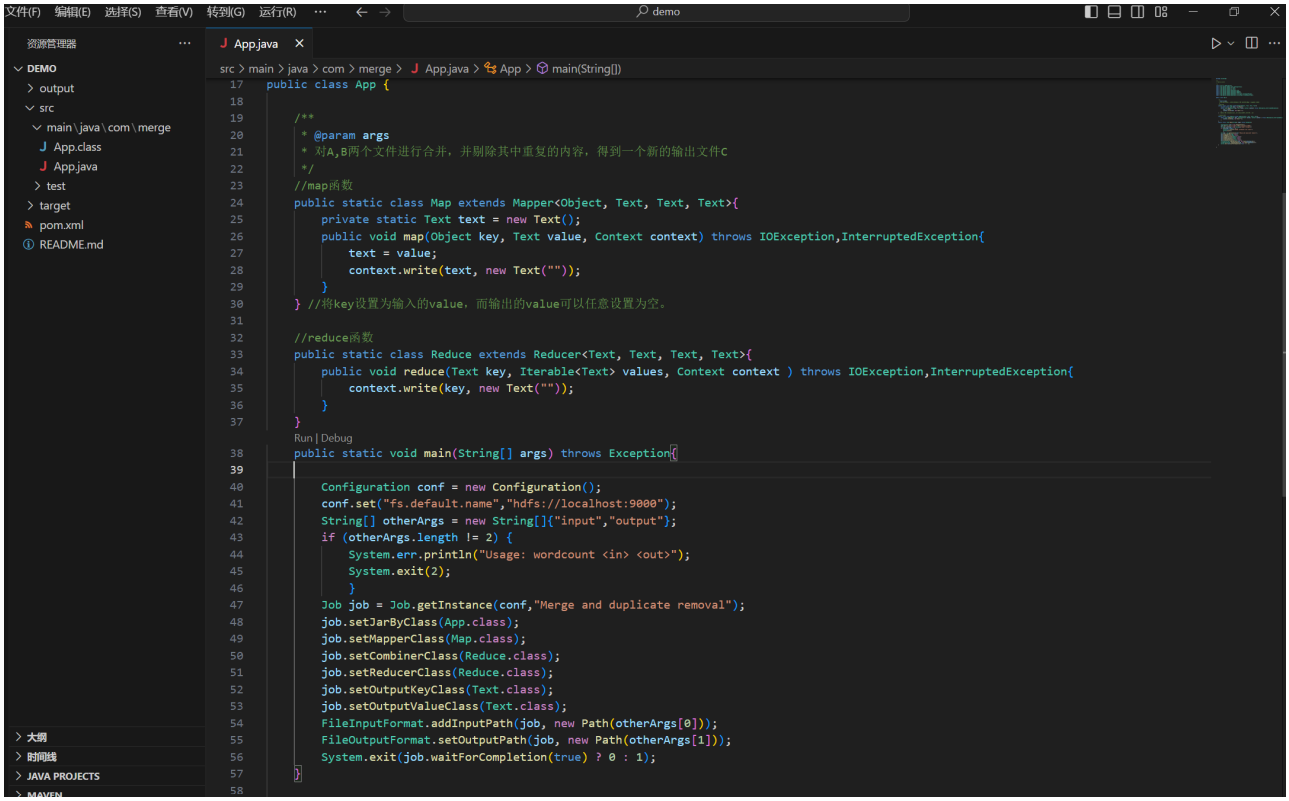
3.运行MapReduce程序： 使用Hadoop集群来运行MapReduce程序，指定输入文件A和B以及输出文件的路径。

二、程序运行结果说明：

1. **在vscode上新建Maven项目，并在pom.xml文件中新增有关hadoop的相关依赖配置**

```xml
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>

  <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-common</artifactId>
      <version>3.3.6</version>
  </dependency>

  <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-hdfs</artifactId>
      <version>3.3.6</version>
  </dependency>

  <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-client</artifactId>
      <version>3.3.6</version>
  </dependency>

  <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-yarn-api</artifactId>
      <version>3.3.6</version>
  </dependency>
```

2. **编译源代码，实现MapReduce的合并去重思想**



3. **导出jar文件，将程序复制到本地Hadoop系统的执行目录，在伪分布式环境下进行测试**

```
# user @ ubuntu in /hadoop_installs/hadoop-3.3.6 [17:35:29] C:1
$ bin/hdfs dfs -mkdir /user/user/input

# user @ ubuntu in /hadoop_installs/hadoop-3.3.6 [17:35:38]
$ bin/hdfs dfs -put A.csv input

# user @ ubuntu in /hadoop_installs/hadoop-3.3.6 [17:35:56]
$ bin/hdfs dfs -put B.csv input

# user @ ubuntu in /hadoop_installs/hadoop-3.3.6 [17:36:03]
$ bin/hadoop jar example-hjy/demo/target/demo-1.0-SNAPSHOT.jar com.merge.App input output
2023-10-22 17:36:16,433 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2023-10-22 17:36:17,055 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2023-10-22 17:36:17,413 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool inte
rface and execute your application with ToolRunner to remedy this.
2023-10-22 17:36:17,430 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/user/.staging
/job_1697963916839_0006
2023-10-22 17:36:17,668 INFO input.FileInputFormat: Total input files to process : 2
2023-10-22 17:36:17,729 INFO mapreduce.JobSubmitter: number of splits:2
2023-10-22 17:36:17,761 INFO Configuration.deprecation: fs.default.name is deprecated. Instead, use fs.defaultFS
2023-10-22 17:36:17,870 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1697963916839_0006
2023-10-22 17:36:17,870 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-10-22 17:36:18,029 INFO conf.Configuration: resource-types.xml not found
2023-10-22 17:36:18,029 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-10-22 17:36:18,113 INFO impl.YarnClientImpl: Submitted application application_1697963916839_0006
2023-10-22 17:36:18,162 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1697963916839_0006/
2023-10-22 17:36:18,163 INFO mapreduce.Job: Running job: job_1697963916839_0006
2023-10-22 17:36:24,258 INFO mapreduce.Job: Job job_1697963916839_0006 running in uber mode : false
2023-10-22 17:36:24,259 INFO mapreduce.Job:  map 0% reduce 0%
2023-10-22 17:36:29,352 INFO mapreduce.Job:  map 50% reduce 0%
2023-10-22 17:36:33,385 INFO mapreduce.Job:  map 100% reduce 0%
2023-10-22 17:36:36,403 INFO mapreduce.Job:  map 100% reduce 100%
2023-10-22 17:36:37,413 INFO mapreduce.Job: Job job_1697963916839_0006 completed successfully
2023-10-22 17:36:37,502 INFO mapreduce.Job: Counters: 54
   File System Counters
        FILE: Number of bytes read=6773
        FILE: Number of bytes written=842766
        FILE: Number of read operations=0
```

三、运行成功的WEB页面截图：

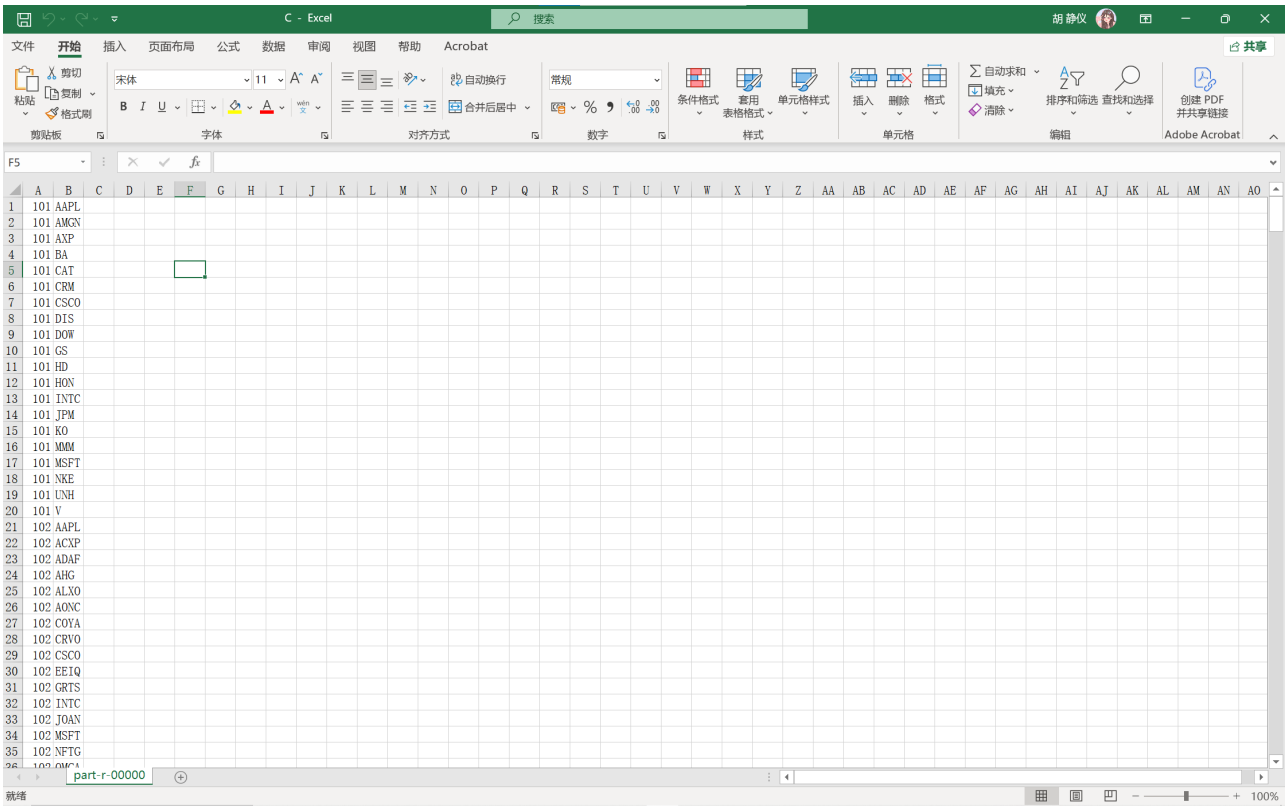
1. **part-r-00000和_SUCCESS截图**

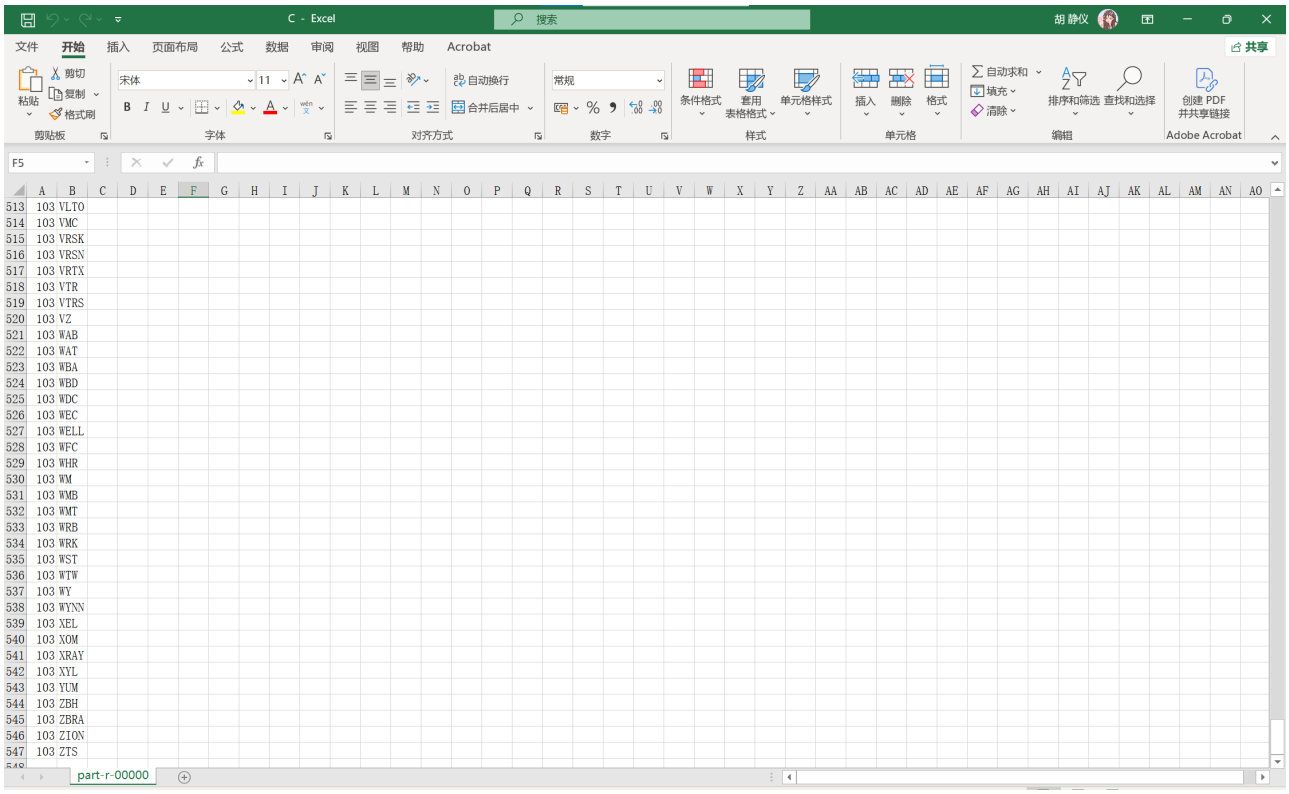
part-r-00000 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
101,AAPL
101,AMGN
101,AXP
101,BA
101,CAT
101,CRM
101,CSCO
101,DIS
101,DOW
101,GS
101,HD
101,HON
101,INTC
101,JPM
101,KO
101,MMM
101,MSFT
101,NKE
101,UNH
101,V
102,AAPL
102,ACXP
102,ADAF
102,AHG
102,ALXO
102,AONC
102,COYA
102,CRVO
102,CSCO
102,EEIQ
102,GRTS
102,INTC
102,JOAN
102,MSFT
102,NFTG
102,OMGA
102,ORGS
102,PRZO
102,SBFM
102,TSBX
```

2. **C.xlsx截图**





四、存在的不足和可能的改进之处

1. Reducer效率： 在Reducer阶段，如果数据的去重和合并操作比较复杂，可能会导致Reducer的执行时间较长。这可能会降低任务的整体性能; 可以优化Reducer中的去重和合并算法以提高效率。

2. 数据传输： 如果Reducer需要处理大量的数据，可能会导致大量的数据传输，从Mapper到Reducer节点。这可能会占用集群的网络带宽，降低性能。可以考虑使用Combiner来减少Mapper和Reducer之间的数据传输。

3. 数据分布不均匀： 如果数据分布不均匀，可能会导致某些Reducer的工作负载比其他Reducer更重，从而影响性能。在这种情况下，可以尝试使用自定义分区器来更均匀地分配数据

4. 自动调整： 使用Hadoop的资源管理器（如YARN）来动态分配资源，以适应不同规模的任务。这可以提高可扩展性。

5. 分布式缓存： 如果你需要额外的数据来进行去重和合并操作，可以考虑使用分布式缓存来减少数据传输。