实验一: MapReduce 基本编程方法

胡嘉鑫　　102102145

2023 年 10 月 11 日

目录

1 实验目的

- 理解 MapReduce 工作流程;
- 掌握 MapReduce 基础编程方法.

2 实验平台

- OS: Linux
- Hadoop v3.1.3
- JDK v1.8

3 实验步骤

3.1 单词去重

3.1.1 Problem Description

描述: 将一个文件内的所有单词去重, 输出为去重后的单词.
Procedure:

1. 编写 MapReduce 代码;
2. 编译并打包项目;
3. 使用 hadoop jar 命令运行程序;
4. 到控制台查看输出文件结果.

Input:
one two three four five
one two three four
one two three
one two
hello world
hello China
hello fuzhou
hello hi
Expected output:
China
five
four
fuzhou
hello
hi

one

three

two

world

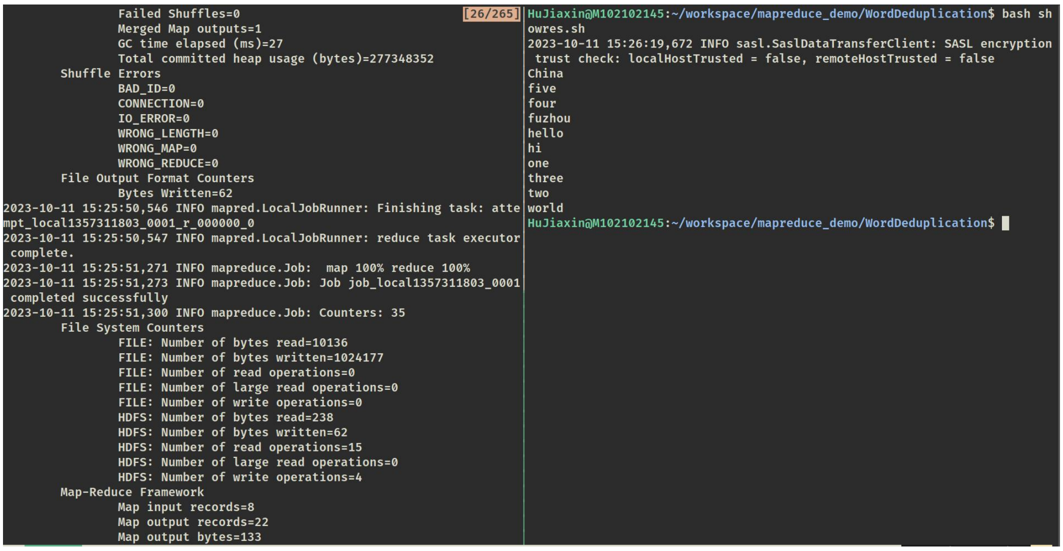
3.1.2 Code

```
package net.homework;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;

class AppMapper extends Mapper<LongWritable, Text, Text, Text> {
  Text k = new Text(); // out-key
  Text v = new Text(); // out-val

  @Override
  protected void map(LongWritable key, Text value, Context context)
  throws IOException, InterruptedException {
    String line = value.toString();
    String[] wordArr = line.split(" ");

    for (int i = 0; i < wordArr.length; i++) {
      k.set(wordArr[i]);
      v.set("");
```

```java
      context.write(k, v);
    }
  }
}

class AppReducer extends Reducer<Text, Text, Text, Text> {
  @Override
  protected void reduce(Text key, Iterable<Text> values, Context context)
  throws IOException, InterruptedException {
    context.write(key, new Text(""));
  }
}

public class WordDeduplication {
  public static void main(String[] args)
  throws IOException, ClassNotFoundException, InterruptedException {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf);

    job.setJarByClass(WordDeduplication.class);

    job.setMapperClass(AppMapper.class);
    job.setReducerClass(AppReducer.class);

    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(Text.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(Text.class);

    FileInputFormat.setInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    boolean result = job.waitForCompletion(true);
    System.exit(result ? 0 : 1);
  }
```

```
}
```

3.1.3 Result



```
                Failed Shuffles=0                                     [26/265]
                Merged Map outputs=1
                GC time elapsed (ms)=27
                Total committed heap usage (bytes)=277348352
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=62
2023-10-11 15:25:50,546 INFO mapred.LocalJobRunner: Finishing task: atte
mpt_local1357311803_0001_r_000000_0
2023-10-11 15:25:50,547 INFO mapred.LocalJobRunner: reduce task executor
 complete.
2023-10-11 15:25:51,271 INFO mapreduce.Job:  map 100% reduce 100%
2023-10-11 15:25:51,273 INFO mapreduce.Job: Job job_local1357311803_0001
 completed successfully
2023-10-11 15:25:51,300 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=10136
                FILE: Number of bytes written=1024177
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=238
                HDFS: Number of bytes written=62
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=8
                Map output records=22
                Map output bytes=133
```

```
HuJiaxin@M102102145:~/workspace/mapreduce_demo/WordDeduplication$ bash sh
owres.sh
2023-10-11 15:26:19,672 INFO sasl.SaslDataTransferClient: SASL encryption
 trust check: localHostTrusted = false, remoteHostTrusted = false
China
five
four
fuzhou
hello
hi
one
three
two
world
HuJiaxin@M102102145:~/workspace/mapreduce_demo/WordDeduplication$ 
```

图 1: 单词去重运行过程及其结果

3.2 计算股票的资本损益

3.2.1 Problem Description

描述: 统计买卖的每个股票收益. (将每个股票的名称作为 key 值, 当操作为 Buy 时, value 记为负的价格, 当操作为 Sell 时, value 记为正的价格, 以这个 key 和 value 作为 map 阶段输出, reduce 阶段的输入)
Procedure:

1. 编写 MapReduce 代码;
2. 编译并打包项目;
3. 使用 hadoop jar 命令运行程序;
4. 到控制台查看输出文件结果.

Input:
Leetcode Buy 1000
Corona Buy 10

Leetcode Sell 9000

Handbags Buy 30000

Corona Sell 1010

Corona Buy 1000

Corona Sell 500

Corona Buy 1000

Handbags Sell 7000

Corona Sell 10000

Expected output:

Corona Masks 9500

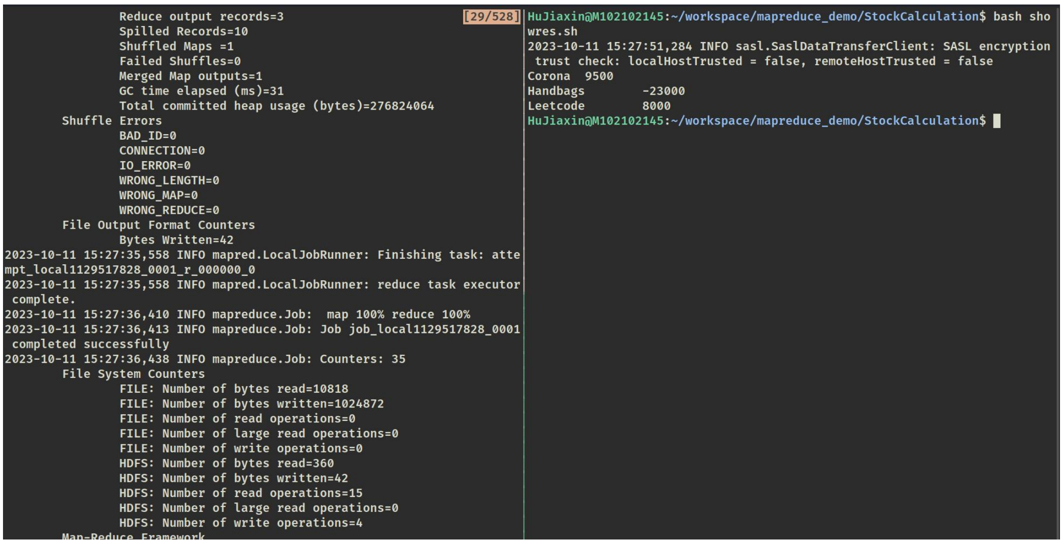
Handbags -23000

Leetcode 8000

3.2.2 Code

```java
package net.homework;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;

class AppMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
  Text k = new Text(); // out-key
  IntWritable v = new IntWritable(); // out-val
  @Override
  protected void map(LongWritable key, Text value, Context context)
  throws IOException, InterruptedException {
    String line = value.toString();
    String[] arr = line.split(" ");
    String stockName = arr[0],
           status = arr[1],
           numStr = arr[2];

    int num = Integer.parseInt(numStr);
    if (status.equals("Buy")) {
      num = -num;
    }
    k.set(stockName);
    v.set(num);
    context.write(k, v);
  }
}

class AppReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
  @Override
  protected void reduce(Text key, Iterable<IntWritable> values, Context context)
  throws IOException, InterruptedException {
    int sum = 0;

    for (IntWritable item : values) {
      sum += item.get();
    }

    context.write(key, new IntWritable(sum));
  }
}

public class StockCalculation {
  public static void main(String[] args)
  throws IOException, ClassNotFoundException, InterruptedException {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf);

    job.setJarByClass(StockCalculation.class);

    job.setMapperClass(AppMapper.class);
    job.setReducerClass(AppReducer.class);

    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(IntWritable.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    FileInputFormat.setInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    boolean result = job.waitForCompletion(true);
    System.exit(result ? 0 : 1);
  }
}
```

3.2.3 Result



```
                Reduce output records=3                                      [29/528]
                Spilled Records=10
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=31
                Total committed heap usage (bytes)=276824064
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=42
2023-10-11 15:27:35,558 INFO mapred.LocalJobRunner: Finishing task: atte
mpt_local1129517828_0001_r_000000_0
2023-10-11 15:27:35,558 INFO mapred.LocalJobRunner: reduce task executor
 complete.
2023-10-11 15:27:36,410 INFO mapreduce.Job:  map 100% reduce 100%
2023-10-11 15:27:36,413 INFO mapreduce.Job: Job job_local1129517828_0001
 completed successfully
2023-10-11 15:27:36,438 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=10818
                FILE: Number of bytes written=1024872
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=360
                HDFS: Number of bytes written=42
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
```

```
HuJiaxin@M102102145:~/workspace/mapreduce_demo/StockCalculation$ bash sho
wres.sh
2023-10-11 15:27:51,284 INFO sasl.SaslDataTransferClient: SASL encryption
 trust check: localHostTrusted = false, remoteHostTrusted = false
Corona  9500
Handbags        -23000
Leetcode        8000
HuJiaxin@M102102145:~/workspace/mapreduce_demo/StockCalculation$ 
```

图 2: 计算股票损益运行过程及其结果

3.3 求互相关注的用户

3.3.1 Problem Description

Procedure:

1. 编写 MapReduce 代码;
2. 编译并打包项目;
3. 使用 hadoop jar 命令运行程序;
4. 到控制台查看输出文件结果.

Input:
A<B,C,D,F,E,O
B<A,C,E,K
C<F,A,D,I
D<A,E,F,L
E<B,C,D,M,L
F<A,B,C,D,E,O,M
G<A,C,D,E,F

H<A,C,D,E,O

I<A,O

J<B,O

K<A,C,D

L<D,E,F

M<E,F,G

O<A,H,I,J,K

如第一行表示用户 B,C,D,F,E,O 关注了 A, 现要求找出互相关注的所有用户对, 输出不能重复 (输出了 A<->B 就不能输出 B<->A).

Expected output:

A<->B

A<->C

A<->D

A<->F

A<->O

B<->E

C<->F

D<->E

D<->F

D<->L

E<->L

E<->M

F<->M

H<->O

I<->O

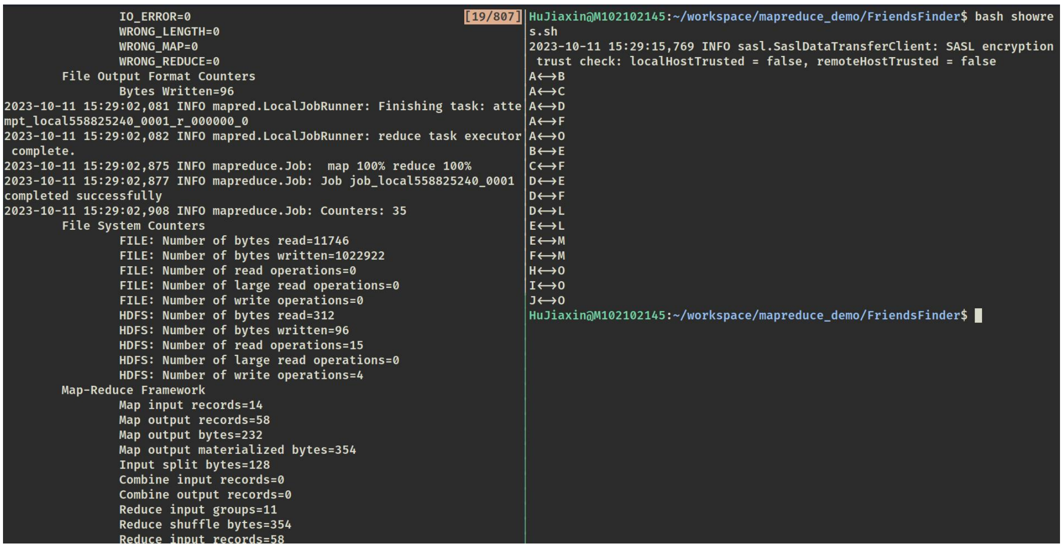
J<->O

3.3.2 Code

```
package net.homework;

import java.io.IOException;
import java.util.HashSet;
import java.util.ArrayList;
import java.util.Collections;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;

class AppMapper extends Mapper<LongWritable, Text, Text, Text> {
  Text k = new Text(); // out-key

  @Override
  protected void map(LongWritable key, Text value, Context context)
  throws IOException, InterruptedException {
    String line = value.toString();
    String[] arr = line.split("<");
    String user = arr[0],
           fans = arr[1];
    String[] fansArr = fans.split(",");

    k.set(user);

    for (int i = 0; i < fansArr.length; i++) {
      if (user.compareTo(fansArr[i]) <= 0) {
        context.write(k, new Text(fansArr[i]));
      } else {
        context.write(new Text(fansArr[i]), k);
      }
    }
  }
}
```

```java
class AppReducer extends Reducer<Text, Text, Text, Text> {
  @Override
  protected void reduce(Text key, Iterable<Text> values, Context context)
  throws IOException, InterruptedException {
    HashSet<String> set = new HashSet<String>();
    ArrayList<String> fansList = new ArrayList<String>();
    for (Text item : values) {
      String fan = item.toString();
      if (set.contains(fan)) {
        fansList.add(fan);
      } else {
        set.add(fan);
      }
    }
    Collections.sort(fansList);
    for (String item : fansList) {
        context.write(key, new Text(item));
    }
  }
}

public class FriendsFinder {
  public static void main(String[] args)
  throws IOException, ClassNotFoundException, InterruptedException {
    Configuration conf = new Configuration();

    conf.set("mapred.textoutputformat.ignoreseparator", "true");
    conf.set("mapred.textoutputformat.separator", "<->");

    Job job = Job.getInstance(conf);

    job.setJarByClass(FriendsFinder.class);

    job.setMapperClass(AppMapper.class);
    job.setReducerClass(AppReducer.class);
    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(Text.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(Text.class);

    FileInputFormat.setInputPaths(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    boolean result = job.waitForCompletion(true);
    System.exit(result ? 0 : 1);
  }
}
```

3.3.3 Result



```
                IO_ERROR=0                                              [19/807]
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=96
2023-10-11 15:29:02,081 INFO mapred.LocalJobRunner: Finishing task: atte
mpt_local558825240_0001_r_000000_0
2023-10-11 15:29:02,082 INFO mapred.LocalJobRunner: reduce task executor
 complete.
2023-10-11 15:29:02,875 INFO mapreduce.Job:  map 100% reduce 100%
2023-10-11 15:29:02,877 INFO mapreduce.Job: Job job_local558825240_0001
completed successfully
2023-10-11 15:29:02,908 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=11746
                FILE: Number of bytes written=1022922
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=312
                HDFS: Number of bytes written=96
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=14
                Map output records=58
                Map output bytes=232
                Map output materialized bytes=354
                Input split bytes=128
                Combine input records=0
                Combine output records=0
                Reduce input groups=11
                Reduce shuffle bytes=354
                Reduce input records=58
```

```
HuJiaxin@M102102145:~/workspace/mapreduce_demo/FriendsFinder$ bash showre
s.sh
2023-10-11 15:29:15,769 INFO sasl.SaslDataTransferClient: SASL encryption
 trust check: localHostTrusted = false, remoteHostTrusted = false
A⟷B
A⟷C
A⟷D
A⟷F
A⟷O
B⟷E
C⟷F
D⟷E
D⟷F
D⟷L
E⟷L
E⟷M
F⟷M
H⟷O
I⟷O
J⟷O
HuJiaxin@M102102145:~/workspace/mapreduce_demo/FriendsFinder$ █
```

图 3: 求相互关注的用户运行过程及其结果

4 出现的问题及其解决方案

没有问题.