重庆交通大学信息科学与工程学院

实　　验　　报　　告

班　　级：曙光 2101 班

姓　　名：李幸洋

学　　号：632107060506

实验项目名称：实验四　　MapReduce 编程

实验项目性质：设计性

实验所属课程：大数据平台架构

实验室(中心)：逸夫楼 407

指 导 教 师 ：何 伟

实验完成时间：2023 年 5 月 24 日

一、实验概述：

【实验目的】

1. 掌握 MapReduce 的基本原理与流程；
2. 掌握 MapReduce 的初级编程思想和方法；

【实验要求】

1. 保存程序，并自行存档；
2. 最终的程序都必须经过测试，验证是正确的；
3. 认真记录实验过程及结果，回答实验报告中的问题。

【实施环境】（使用的材料、设备、软件）

Linux 操作系统环境，VirtualBox 虚拟机，Java 开发环境，Hadoop。

二、实验内容

第 1 题 WordCount 练习

【实验内容】

按照教材或网站上的指导完成 WordCount 程序的编写，能够在集成开发环境中运行并打包成.Jar 文件在终端执行。

【实验过程】（步骤、记录、数据、程序等）

请提供相应 Java 代码及程序运行界面截图证明。

Java 代码：

```
1 个用法
public static class TokenizerMapper extends
        Mapper<Object, Text, Text, IntWritable> {
    1 个用法
    private static final IntWritable one = new IntWritable( value: 1);
    2 个用法
    private Text word = new Text();

    0 个用法
    public TokenizerMapper() {
    }

    public void map(Object key,
                    Text value,
                    Mapper<Object, Text, Text, IntWritable>.Context context
    ) throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
```

```
2 个用法
public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    2 个用法
    private IntWritable result = new IntWritable();

    0 个用法
    public IntSumReducer() {
    }

    public void reduce(
            Text key,
            Iterable<IntWritable> values,
            Reducer<Text, IntWritable, Text, IntWritable>.Context context
    )
            throws IOException, InterruptedException {
        int sum = 0;

        IntWritable val;
        for(Iterator i$ = values.iterator(); i$.hasNext(); sum += val.get()) {
            val = (IntWritable)i$.next();
        }

        this.result.set(sum);
        context.write(key, this.result);
    }
}
```

```java
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
    if(otherArgs.length < 2) {
        System.err.println("Usage: wordcount <in> [<in>...] <out>");
        System.exit( i: 2);
    }

    Job job = Job.getInstance(conf, jobName: "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(WordCount.TokenizerMapper.class);
    job.setCombinerClass(WordCount.IntSumReducer.class);
    job.setReducerClass(WordCount.IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    for(int i = 0; i < otherArgs.length - 1; ++i) {
        FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }

    FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion( verbose: true)?0:1);
}
```

```
2023-05-24 04:53:18,745 INFO mapreduce.Job:  map 100% reduce 100%
2023-05-24 04:53:18,748 INFO mapreduce.Job: Job job_local201470402_0001 completed successfully
2023-05-24 04:53:18,773 INFO mapreduce.Job: Counters: 35
       File System Counters
               FILE: Number of bytes read=666432453
               FILE: Number of bytes written=677254654
               FILE: Number of read operations=0
               FILE: Number of large read operations=0
               FILE: Number of write operations=0
               HDFS: Number of bytes read=267986
               HDFS: Number of bytes written=9825
               HDFS: Number of read operations=168
               HDFS: Number of large read operations=0
               HDFS: Number of write operations=13
       Map-Reduce Framework
               Map input records=760
               Map output records=3072
               Map output bytes=37956
               Map output materialized bytes=20815
               Input split bytes=1197
               Combine input records=3072
               Combine output records=1231
               Reduce input groups=595
               Reduce shuffle bytes=20815
               Reduce input records=1231
               Reduce output records=595
               Spilled Records=2462
               Shuffled Maps =10
               Failed Shuffles=0
               Merged Map outputs=10
               GC time elapsed (ms)=524
               Total committed heap usage (bytes)=2593329152
       Shuffle Errors
               BAD_ID=0
               CONNECTION=0
               IO_ERROR=0
               WRONG_LENGTH=0
               WRONG_MAP=0
               WRONG_REDUCE=0
       File Input Format Counters
```

终端

输出结果：

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat output1/_SUCCESS
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat output1/part-r-00000
2023-05-24 04:55:54,695 INFO sasl.SaslDataTransferClient: SASL encryption trust chec
"*"     21
"AS     9
"License");     9
"alice,bob      21
"clumping"      1
(the    9
-->     17
-1      1
-1,     1
0.0     1
1-MAX_INT.      1
1.      1
1.0.    1
2.0     9
40      2
40+20=60        1
:       2
<!--    17
</configuration>        9
</description>  33
</name> 2
</property>     62
<?xml   8
<?xml-stylesheet        4
<configuration> 9
<description>   31
<description>ACL        25
<description>Abase      1
<description>Default    1
<name>default.key.acl.DECRYPT_EEK</name>        1
<name>default.key.acl.GENERATE_EEK</name>       1
<name>default.key.acl.MANAGEMENT</name> 1
<name>default.key.acl.READ</name>       1
<name>dfs.datanode.data.dir</name>      1
```

第 2 题. 数据去重

【实验内容】

对数据文件中的数据进行去重。数据文件中的每行都是一个数据。

输入如下所示：

1）file1：

2012-3-1 a
2012-3-2 b
2012-3-3 c
2012-3-4 d
2012-3-5 a
2012-3-6 b
2012-3-7 c
2012-3-3 c

2）file2：

2012-3-1 b
2012-3-2 a
2012-3-3 b
2012-3-4 d
2012-3-5 a
2012-3-6 c
2012-3-7 d
2012-3-3 c

输出如下所示：

2012-3-1 a
2012-3-1 b
2012-3-2 a
2012-3-2 b
2012-3-3 b
2012-3-3 c
2012-3-4 d
2012-3-5 a
2012-3-6 b
2012-3-6 c
2012-3-7 c
2012-3-7 d

【实验过程】（步骤、记录、数据、程序等）

请提供相应 Java 代码及程序运行界面截图证明。

Java 代码：

```
1 个用法
public static class ReadMapper extends Mapper<LongWritable, Text, Text, NullWritable> {
    2 个用法
    private static  Text line = new Text();

    0 个用法
    public ReadMapper() {
    }

    @Override
    protected void map(
            LongWritable key,
            Text value,
            Mapper<LongWritable, Text, Text, NullWritable>.Context context)
            throws IOException, InterruptedException {
        line = value;
        context.write(line, NullWritable.get());
    }
}

/**
 * reducer 会自动去除重复
 */
1 个用法
public static class WriteReducer extends
        Reducer<Text, NullWritable, Text, NullWritable> {
    0 个用法
    public WriteReducer() {

    }
    //重写reduce()方法
    @Override
    protected void reduce(
            Text key,
            Iterable<NullWritable> values,
            Reducer<Text, NullWritable, Text, NullWritable>.Context context)
            throws IOException, InterruptedException {

        context.write(key,NullWritable.get());
    }
}
```

```java
public static void main(String[] args) throws  Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
    if(otherArgs.length < 2) {
        System.err.println("Usage: clean repeat <in> [<in>...] <out>");
        System.exit( i: 2);
    }

    Job job = Job.getInstance(conf,  jobName: "clean repeat");
    job.setJarByClass(CleanRepeat.class);
    job.setMapperClass(CleanRepeat.ReadMapper.class);
    job.setReducerClass(CleanRepeat.WriteReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(NullWritable.class);

    for(int i = 0; i < otherArgs.length - 1; ++i) {
        FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }

    FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion( verbose: true)?0:1);
}
```

输出结果图：

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat /user/hadoop/output1/part-r-00000
2023-05-25 06:59:08,819 INFO sasl.SaslDataTransferClient: SASL encryption trust check: l
2012-3-1 a
2012-3-1 b
2012-3-2 a
2012-3-2 b
2012-3-3 b
2012-3-3 c
2012-3-4 d
2012-3-5 a
2012-3-6 b
2012-3-6 c
2012-3-7 c
2012-3-7 d
hadoop@ubuntu:/usr/local/hadoop$
```

运行结果图：

```
2023-05-24 05:30:05,131 INFO mapred.LocalJobRunner: Finishing task: attempt_local882077872_0001_r_000000_0
2023-05-24 05:30:05,131 INFO mapred.LocalJobRunner: reduce task executor complete.
2023-05-24 05:30:05,549 INFO mapreduce.Job:  map 100% reduce 100%
2023-05-24 05:30:05,550 INFO mapreduce.Job: Job job_local882077872_0001 completed successfully
2023-05-24 05:30:05,570 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=242314828
                FILE: Number of bytes written=246208680
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=968
                HDFS: Number of bytes written=132
                HDFS: Number of read operations=35
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=6
        Map-Reduce Framework
                Map input records=28
                Map output records=28
                Map output bytes=308
                Map output materialized bytes=382
                Input split bytes=333
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=382
                Reduce input records=28
                Reduce output records=12
                Spilled Records=56
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=154
                Total committed heap usage (bytes)=1033650176
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
```

第 3 题 数据排序

【实验内容】

对输入文件中数据进行排序。输入文件中的每行内容均为一个数字，即一个数据。要求在输出中每行有两个间隔的数字，其中，第一个代表原始数据在原始数据集中的位次，第二个代表原始数据。然后修改原来的程序，使其按照从大到小的顺序排序。

输入：

1）file1：

2
32
654
32
15
756
65223

2）file2：

5956
22
650
92

3）file3：

26
54
6

输出：

1 2
2 6
3 15
4 22
5 26
6 32
7 32
8 54
9 92
10 650
11 654
12 756
13 5956

14　65223

【实验过程】（步骤、记录、数据、程序等）

请提供相应 Java 代码及程序运行界面截图证明。

Java 代码：

```
1 个用法
public static class MergeMap extends
        Mapper<Object, Text, IntWritable, IntWritable> {
    2 个用法
    private static IntWritable data = new IntWritable();

    @Override
    protected void map(
            Object key,
            Text value,
            Mapper<Object, Text, IntWritable, IntWritable>.Context context)
            throws IOException, InterruptedException {
        String line = value.toString();
        data.set(Integer.parseInt(line));
        context.write(data, new IntWritable( value: 1));
    }
}
```

```
 * 按照写入个数排序，重定义排序规则完成倒序
 */
1 个用法
public static class MergeReduce
        extends Reducer<IntWritable, IntWritable, IntWritable, IntWritable> {
    3 个用法
    private static IntWritable line_num = new IntWritable( value: 1);


    public void reduce(
            IntWritable key,
            Iterable<IntWritable> values, Context context
    )
            throws IOException,InterruptedException{
        for(IntWritable val : values){
            context.write(line_num, key);
            line_num = new IntWritable( value: line_num.get() + 1);
        }
    }
}
```

```java
/**
 * 重写比较器实现倒序
 */
1 个用法
public static class IntWritableComparator extends WritableComparator {

    0 个用法
    public IntWritableComparator() {
        super(IntWritable.class, createInstances: true);
    }

    @Override
    public int compare(WritableComparable a, WritableComparable b) {
        //向下转型
        IntWritable ia = (IntWritable) a;
        IntWritable ib = (IntWritable) b;
        return ib.compareTo(ia);
    }
}
```

```java
public static void main(String[] args)
        throws IOException, InterruptedException, ClassNotFoundException {
    Configuration conf = new Configuration();
    String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
    if(otherArgs.length < 2) {
        System.err.println("Usage: merge sort <in> [<in>...] <out>");
        System.exit( i: 2);
    }

    Job job = Job.getInstance(conf, jobName: "merge sort");
    job.setJarByClass(MergeSort.class);
    job.setMapperClass(MergeSort.MergeMap.class);
    job.setReducerClass(MergeSort.MergeReduce.class);
    job.setOutputKeyClass(IntWritable.class);
    job.setOutputValueClass(IntWritable.class);
    job.setSortComparatorClass(MergeSort.IntWritableComparator.class);

    for(int i = 0; i < otherArgs.length - 1; ++i) {
        FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }

    FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion( verbose: true)?0:1);
}
```

输出结果图：

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat /user/hadoop/output2/part-r-00000
2023-05-25 06:58:59,271 INFO sasl.SaslDataTransferClient: SASL encryption trust check: l
1       65223
2       5956
3
4
5       650
6       92
7       54
8       32
9       32
10      26
11      22
12      15
13      6
14      2
```

运行结果图：

```
2023-05-24 05:30:05,131 INFO mapred.LocalJobRunner: Finishing task: attempt_local882077872_0001_r_000000_0
2023-05-24 05:30:05,131 INFO mapred.LocalJobRunner: reduce task executor complete.
2023-05-24 05:30:05,549 INFO mapreduce.Job:  map 100% reduce 100%
2023-05-24 05:30:05,550 INFO mapreduce.Job: Job job_local882077872_0001 completed successfully
2023-05-24 05:30:05,570 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=242314828
                FILE: Number of bytes written=246208680
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=968
                HDFS: Number of bytes written=132
                HDFS: Number of read operations=35
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=6
        Map-Reduce Framework
                Map input records=28
                Map output records=28
                Map output bytes=308
                Map output materialized bytes=382
                Input split bytes=333
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=382
                Reduce input records=28
                Reduce output records=12
                Spilled Records=56
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=154
                Total committed heap usage (bytes)=1033650176
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
```

第 4 题 多表关联

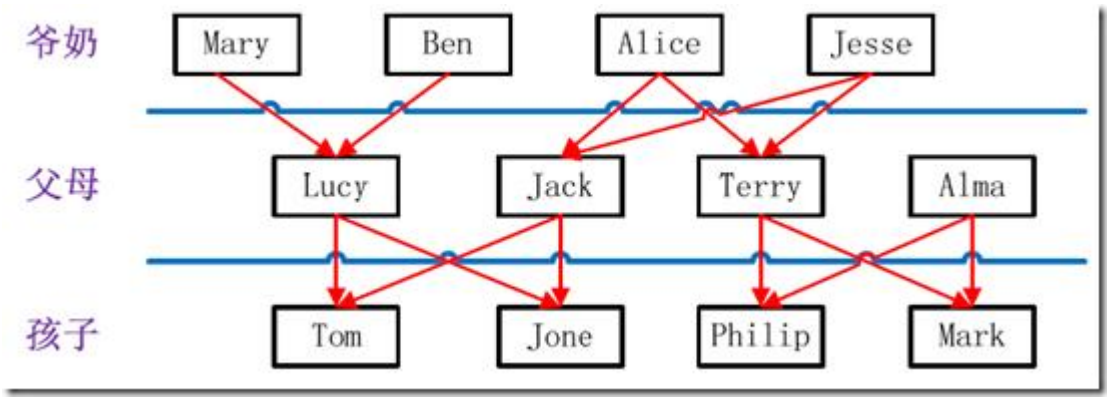
【实验内容】

文档中给出 **child-parent**（孩子-父母）表，要求输出 **grandchild-grandparent**（孙子-爷奶）表。样例输入如下所示。

file：

child	parent
Tom	Lucy
Tom	Jack
Jone	Lucy
Jone	Jack
Lucy	Mary
Lucy	Ben
Jack	Alice
Jack	Jesse
Terry	Alice
Terry	Jesse
Philip	Terry
Philip	Alma
Mark	Terry
Mark	Alma

家族**树状**关系谱：



样例输出如下所示。

file：

grandchild	grandparent
Tom	Alice
Tom	Jesse
Jone	Alice
Jone	Jesse
Tom	Mary
Tom	Ben
Jone	Mary

Jone Ben
Philip Alice
Philip Jesse
Mark Alice
Mark Jesse

【实验过程】（步骤、记录、数据、程序等）

请提供相应 Java 代码及程序运行界面截图证明。

Java 代码：

```java
/**
 * 以父母一代为基准，上下找到爷孙关系
 */
1个用法
public static class FamilyMapper extends Mapper<Object, Text, Text, Text> {
    @Override
    protected void map(
            Object key,
            Text value,
            Mapper<Object, Text, Text, Text>.Context context
    )
            throws IOException, InterruptedException {

        String line = value.toString();
        String[] person = line.split( s: " ");

        context.write(new Text(person[0]), new Text( string: person[1]+"-2"));
        context.write(new Text(person[1]), new Text( string: person[0]+"-1"));

    }
}
```

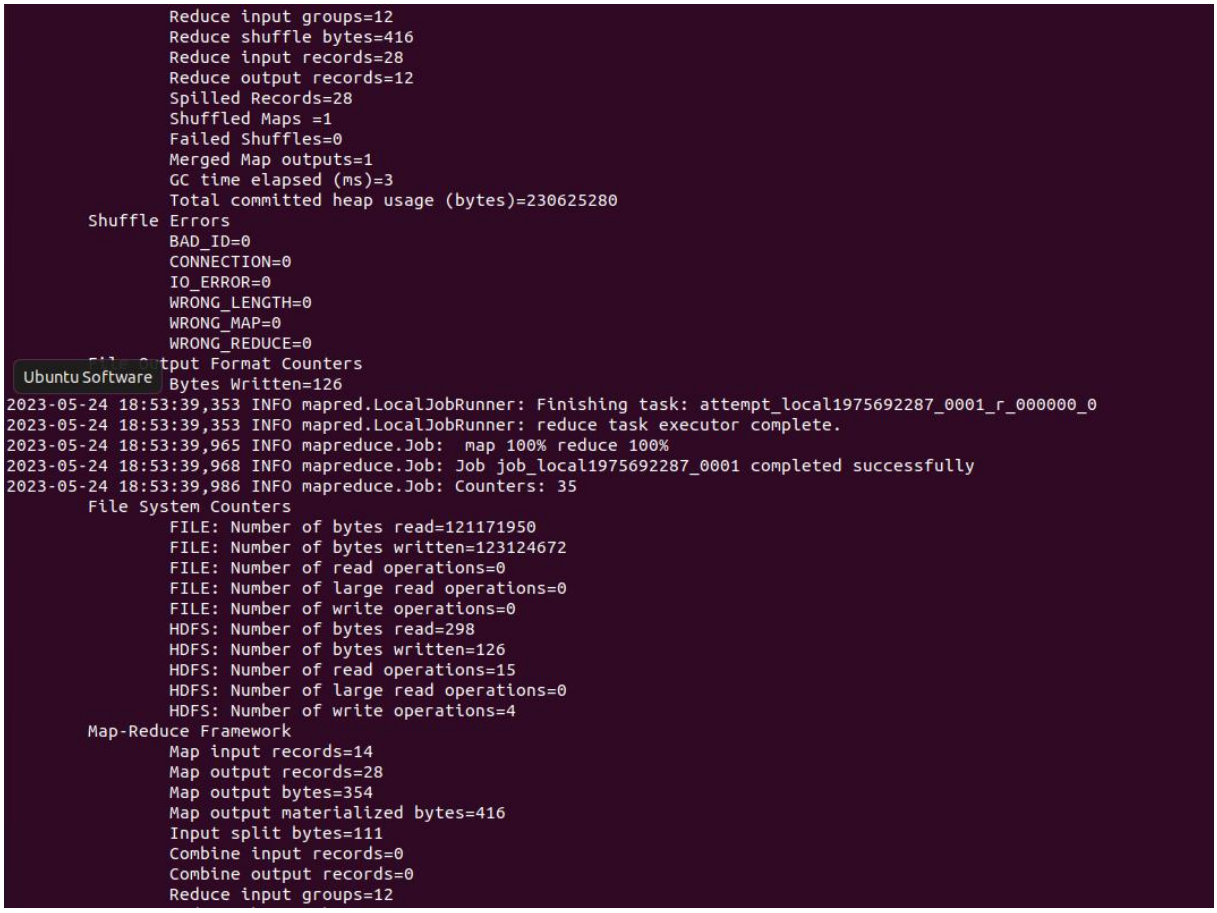
```java
public static class FamilyReducer extends Reducer<Text, Text, Text, Text> {
    @Override
    protected void reduce(
            Text key,
            Iterable<Text> values,
            Reducer<Text, Text, Text, Text>.Context context
    )
            throws IOException, InterruptedException {
        List<String> upForParent = new ArrayList<>();
        List<String> downForChild = new ArrayList<>();


        for (Text val : values) {
            String str = val.toString();
            if (str.endsWith("-1")) {
                downForChild.add(str.substring( i: 0,  i1: str.length() - 2));
            } else if (str.endsWith("-2")) {
                upForParent.add(str.substring( i: 0,  i1: str.length() - 2));
            }
        }

        for(String child: downForChild) {
            for(String parent: upForParent) {
                context.write(new Text(child), new Text(parent));
            }
        }

    }
}
```

```java
public static void main(String[] args)
        throws IOException, InterruptedException, ClassNotFoundException {
    Configuration conf = new Configuration();
    String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
    if(otherArgs.length < 2) {
        System.err.println("Usage: find family <in> [<in>...] <out>");
        System.exit( i: 2);
    }

    Job job = Job.getInstance(conf,  jobName: "find family");
    job.setJarByClass(FindFamily.class);
    job.setMapperClass(FindFamily.FamilyMapper.class);
    job.setReducerClass(FindFamily.FamilyReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(Text.class);

    for(int i = 0; i < otherArgs.length - 1; ++i) {
        FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }

    FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion( verbose: true)?0:1);
}
```

输出结果图：

```
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -cat /user/hadoop/output3/
2023-05-24 18:54:02,559 INFO sasl.SaslDataTransferClient: SASL encryption
Jone    Alice
Jone    Jesse
Tom     Alice
Tom     Jesse
Jone    Mary
Jone    Ben
Tom     Mary
Tom     Ben
Mark    Alice
Mark    Jesse
Philip  Alice
Philip  Jesse
```

运行结果图：


Reduce input groups=12
Reduce shuffle bytes=416
Reduce input records=28
Reduce output records=12
Spilled Records=28
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=3
Total committed heap usage (bytes)=230625280
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
tput Format Counters
Bytes Written=126
2023-05-24 18:53:39,353 INFO mapred.LocalJobRunner: Finishing task: attempt_local1975692287_0001_r_000000_0
2023-05-24 18:53:39,353 INFO mapred.LocalJobRunner: reduce task executor complete.
2023-05-24 18:53:39,965 INFO mapreduce.Job: map 100% reduce 100%
2023-05-24 18:53:39,968 INFO mapreduce.Job: Job job_local1975692287_0001 completed successfully
2023-05-24 18:53:39,986 INFO mapreduce.Job: Counters: 35
File System Counters
FILE: Number of bytes read=121171950
FILE: Number of bytes written=123124672
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=298
HDFS: Number of bytes written=126
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=28
Map output bytes=354
Map output materialized bytes=416
Input split bytes=111
Combine input records=0
Combine output records=0
Reduce input groups=12
Ubuntu Software