MapReduce 2.0程序设计（涉及多语言）-理论部分

讲师：董西成

博客：dongxicheng.org

微信号：hadoop-123

（二维码见右）

Open Passion Value

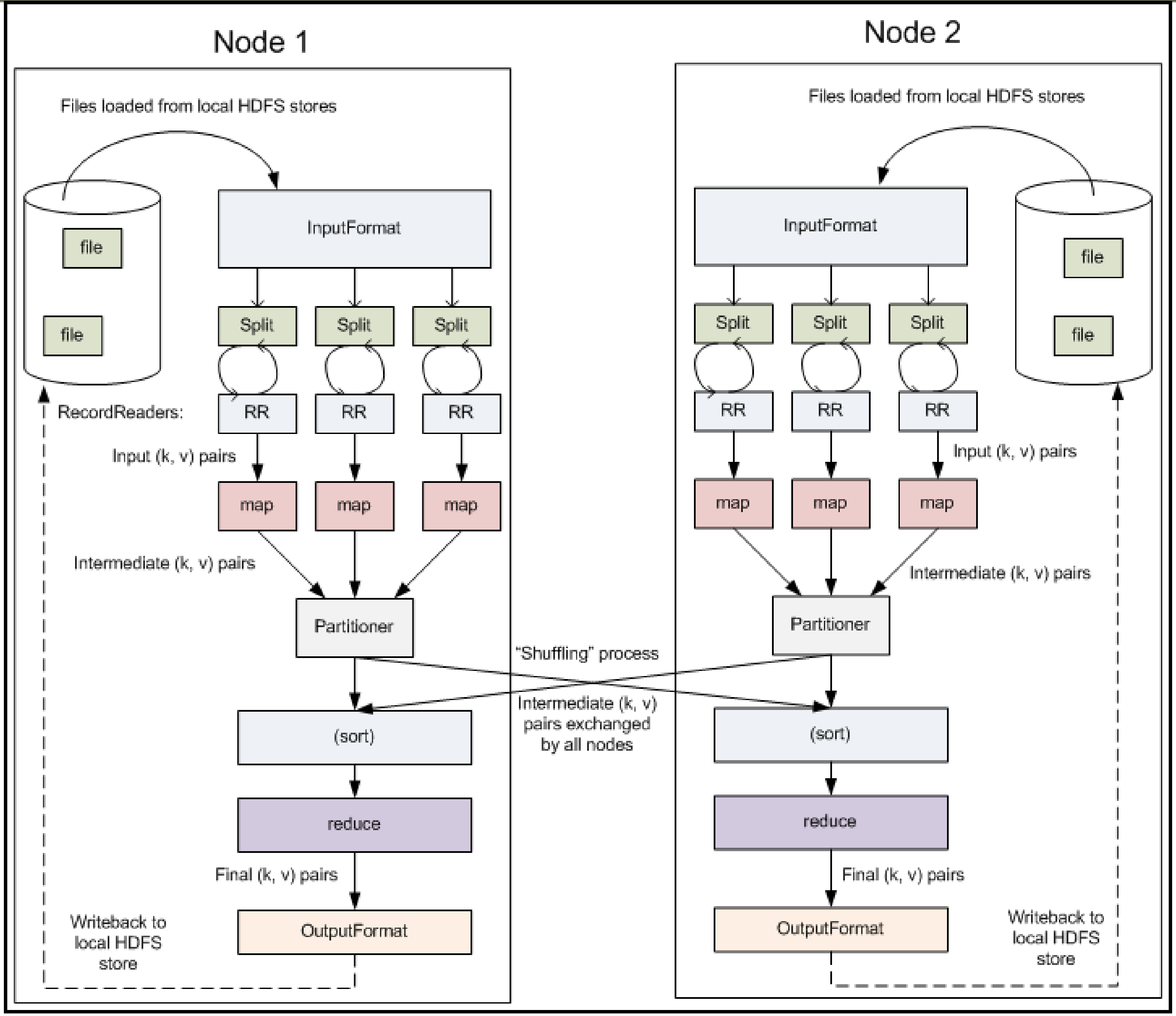
目录

1. MapReduce 2.0的编程模型
2. MapReduce 2.0编程接口介绍
3. Java编程
4. 多语言编程
5. 总结

MapReduce编程模型

- MapReduce将整个运行过程分为两个阶段：Map阶段和Reduce阶段
- Map阶段由一定数量的Map Task组成
  - ✓ 输入数据格式解析：InputFormat
  - ✓ 输入数据处理：Mapper
  - ✓ 数据分组：Partitioner
- Reduce阶段由一定数量的Reduce Task组成
  - ✓ 数据远程拷贝
  - ✓ 数据按照key排序
  - ✓ 数据处理：Reducer
  - ✓ 数据输出格式：OutputFormat

MapReduce编程模型—外部物理结构



MapReduce编程模型

- ➢ **Map阶段**
  - ✓**InputFormat（默认TextInputFormat）**
  - ✓**Mapper**
  - ✓**Combiner（local reducer）**
  - ✓**Partitioner**
- ➢ **Reduce阶段**
  - ✓**Reducer**
  - ✓**OutputFormat（默认TextOutputFormat）**

目录

1. MapReduce的编程模型
2. MapReduce编程接口介绍
3. Java编程
4. 多语言编程
5. 总结

MapReduce编程接口

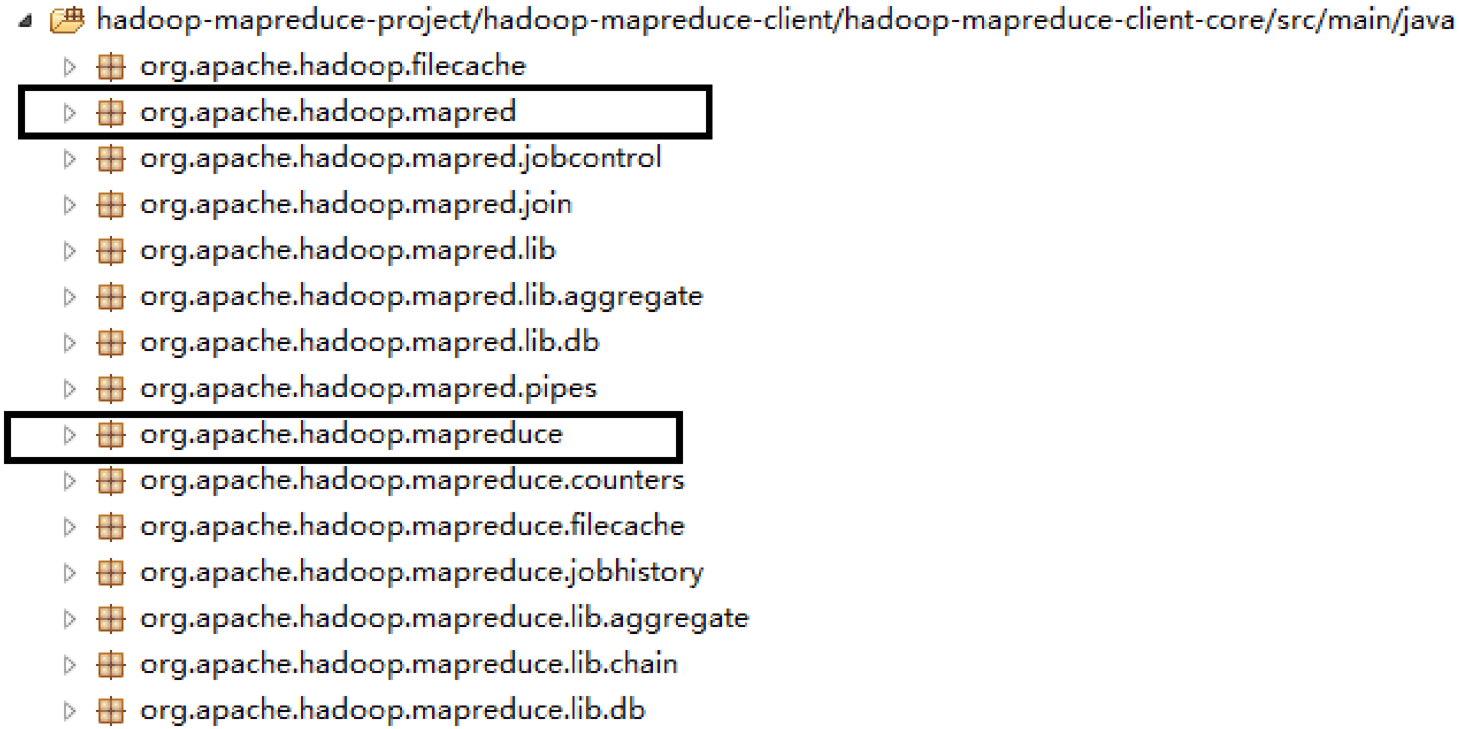
- Hadoop提供了三种编程方式：
  - ✓ Java（最原始的方式）
  - ✓ Hadoop Streaming（支持多语言）
  - ✓ Hadoop Pipes（支持C/C++）
- Java编程接口是所有编程方式的基础；
- 不同的编程接口只是暴露给用户的形式不同而已，内部执行引擎是一样的；
- 不同编程方式效率不同。

Java编程接口

- **Java编程接口组成；**
  - ✓ 旧**API**：所在**java**包：**org.apache.hadoop.mapred**
  - ✓ 新**API**：所在**java**包：**org.apache.hadoop.mapreduce**
- 新**API**具有更好的扩展性；
- 两种编程接口只是暴露给用户的形式不同而已，内部执行引擎是一样的；
- 旧**API**可以完全兼容**Hadoop 2.0**，但新**API**不行。

Java新旧API

- 从hadoop 1.0.0开始，所有发行版均包含新旧两类API；

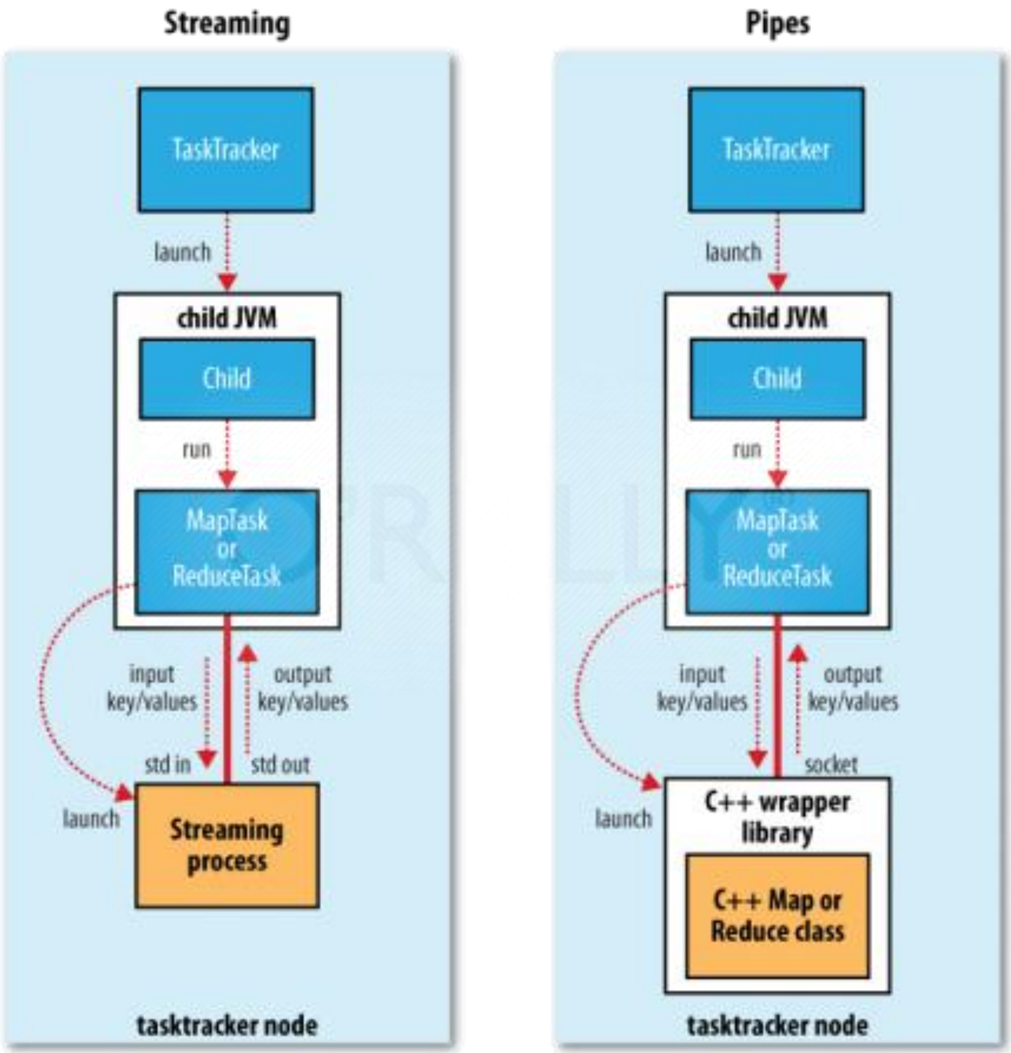


```
hadoop-mapreduce-project/hadoop-mapreduce-client/hadoop-mapreduce-client-core/src/main/java
    org.apache.hadoop.filecache
    org.apache.hadoop.mapred
    org.apache.hadoop.mapred.jobcontrol
    org.apache.hadoop.mapred.join
    org.apache.hadoop.mapred.lib
    org.apache.hadoop.mapred.lib.aggregate
    org.apache.hadoop.mapred.lib.db
    org.apache.hadoop.mapred.pipes
    org.apache.hadoop.mapreduce
    org.apache.hadoop.mapreduce.counters
    org.apache.hadoop.mapreduce.filecache
    org.apache.hadoop.mapreduce.jobhistory
    org.apache.hadoop.mapreduce.lib.aggregate
    org.apache.hadoop.mapreduce.lib.chain
    org.apache.hadoop.mapreduce.lib.db
```

Hadoop Streaming

- 与Linux管道机制一致
- 通过标准输入输出实现进程间通信
- 标准输入输出是任何语言都有的
- 几个举例：
  - ✓ cat 1.txt | grep “dong” | sort
  - ✓ cat 1.txt | python grep.py | java sort.jar

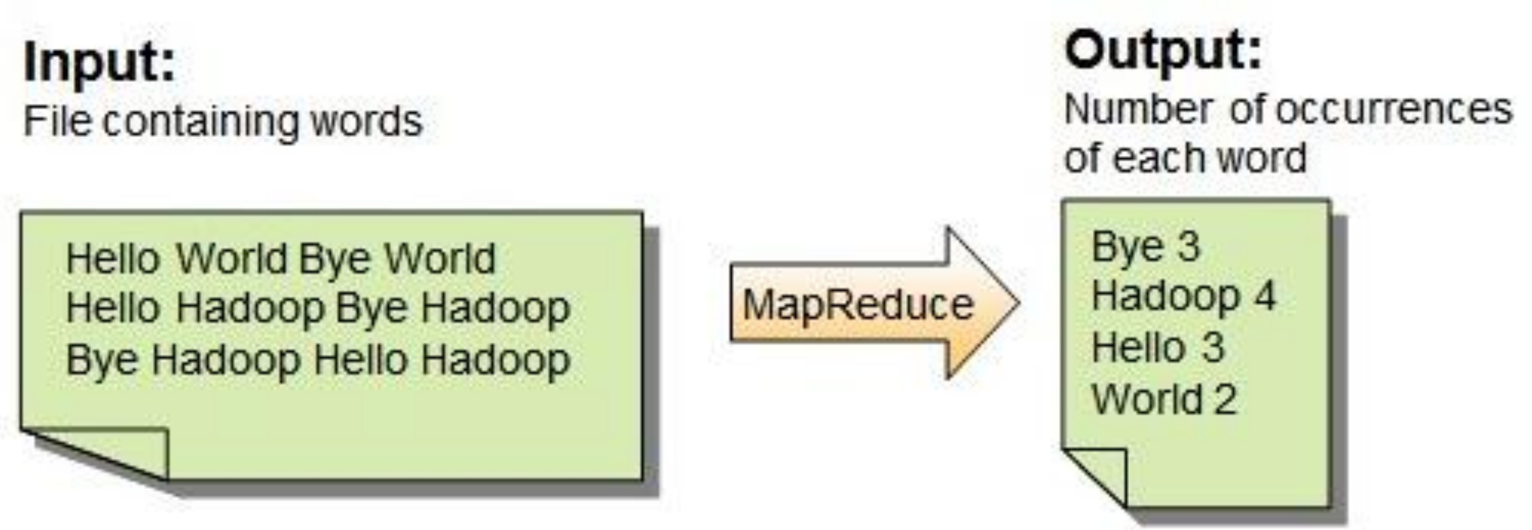
Hadoop Streaming/pipes



目录

1. MapReduce的编程模型
2. MapReduce编程接口介绍
3. Java编程
4. 多语言编程
5. 总结

实例1：WordCount问题



Input:
File containing words

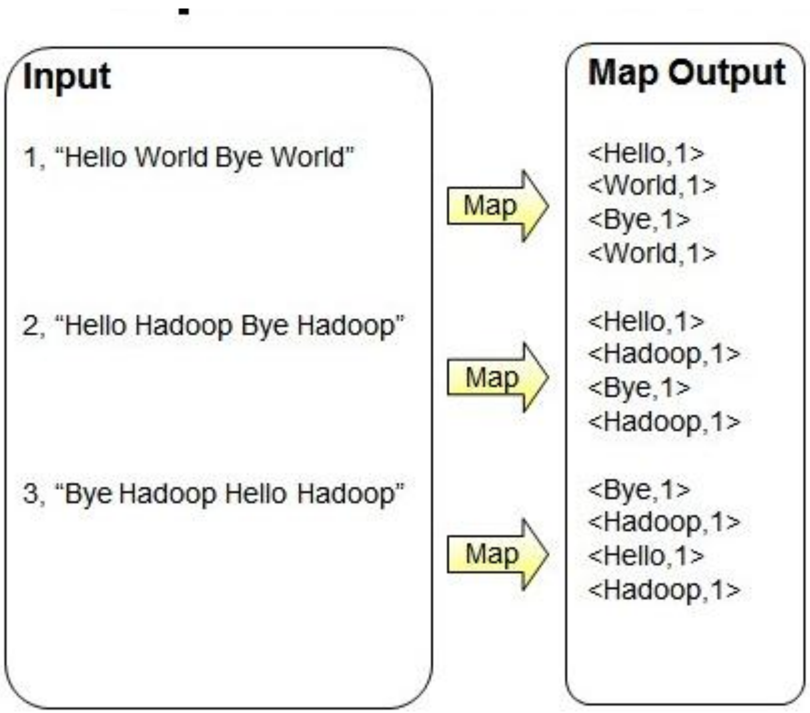
Hello World Bye World
Hello Hadoop Bye Hadoop
Bye Hadoop Hello Hadoop

MapReduce

Output:
Number of occurrences of each word

Bye 3
Hadoop 4
Hello 3
World 2

WordCount问题—map阶段



Input

1, “Hello World Bye World”

2, “Hello Hadoop Bye Hadoop”

3, “Bye Hadoop Hello Hadoop”

Map Output

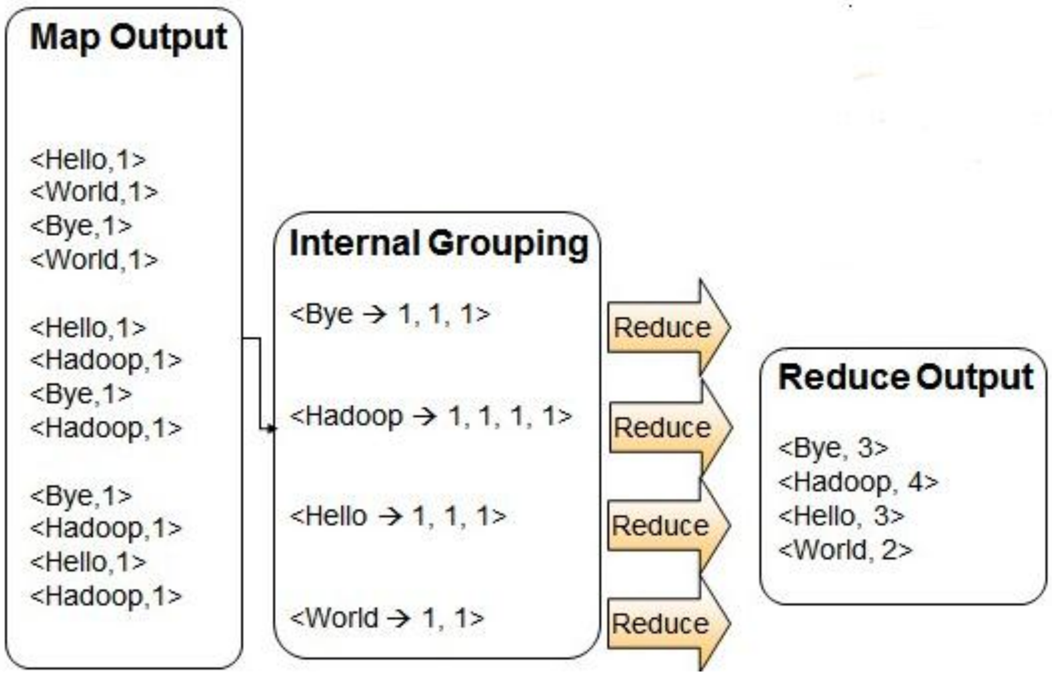
<Hello,1>
<World,1>
<Bye,1>
<World,1>

<Hello,1>
<Hadoop,1>
<Bye,1>
<Hadoop,1>

<Bye,1>
<Hadoop,1>
<Hello,1>
<Hadoop,1>

```
Map(K, V) {
  For each word w in V
    Collect(w, 1);
}
```

WordCount问题—reduce阶段



Map Output

<Hello,1>
<World,1>
<Bye,1>
<World,1>

<Hello,1>
<Hadoop,1>
<Bye,1>
<Hadoop,1>

<Bye,1>
<Hadoop,1>
<Hello,1>
<Hadoop,1>

Internal Grouping

<Bye → 1, 1, 1>

<Hadoop → 1, 1, 1, 1>

<Hello → 1, 1, 1>

<World → 1, 1>

Reduce

Reduce

Reduce

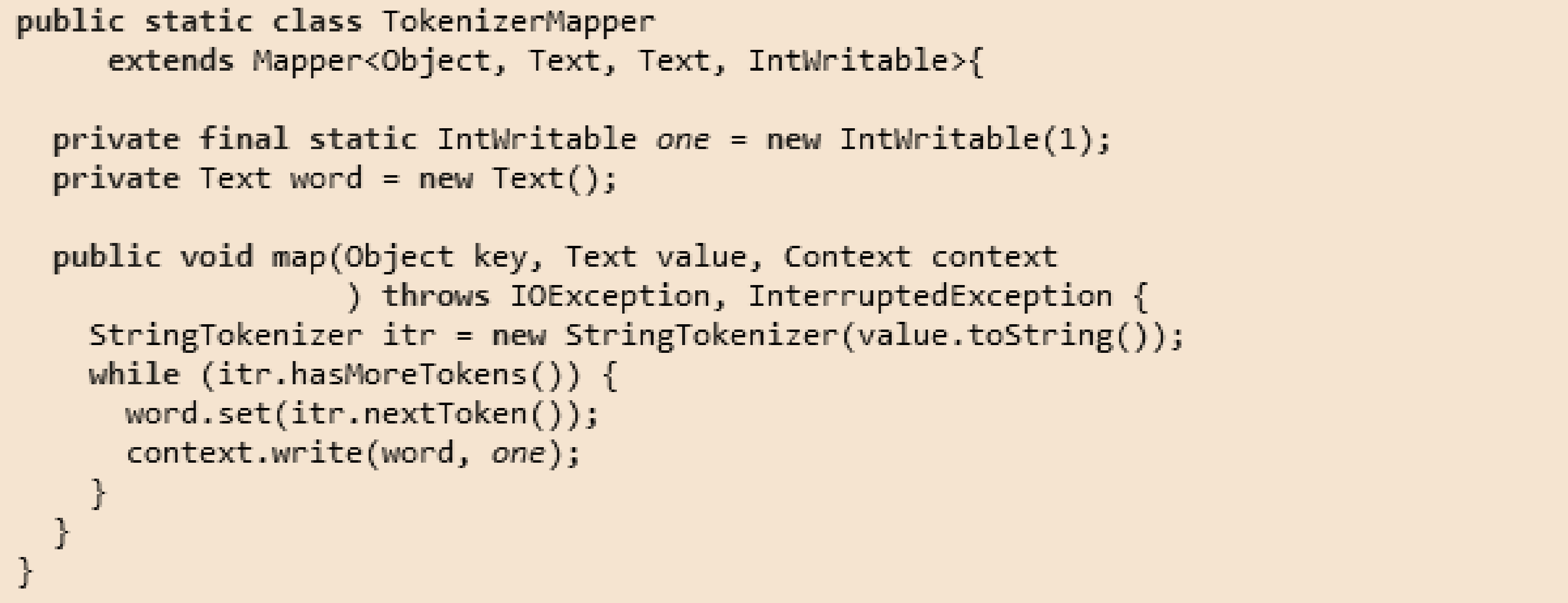
Reduce

Reduce Output

<Bye, 3>
<Hadoop, 4>
<Hello, 3>
<World, 2>

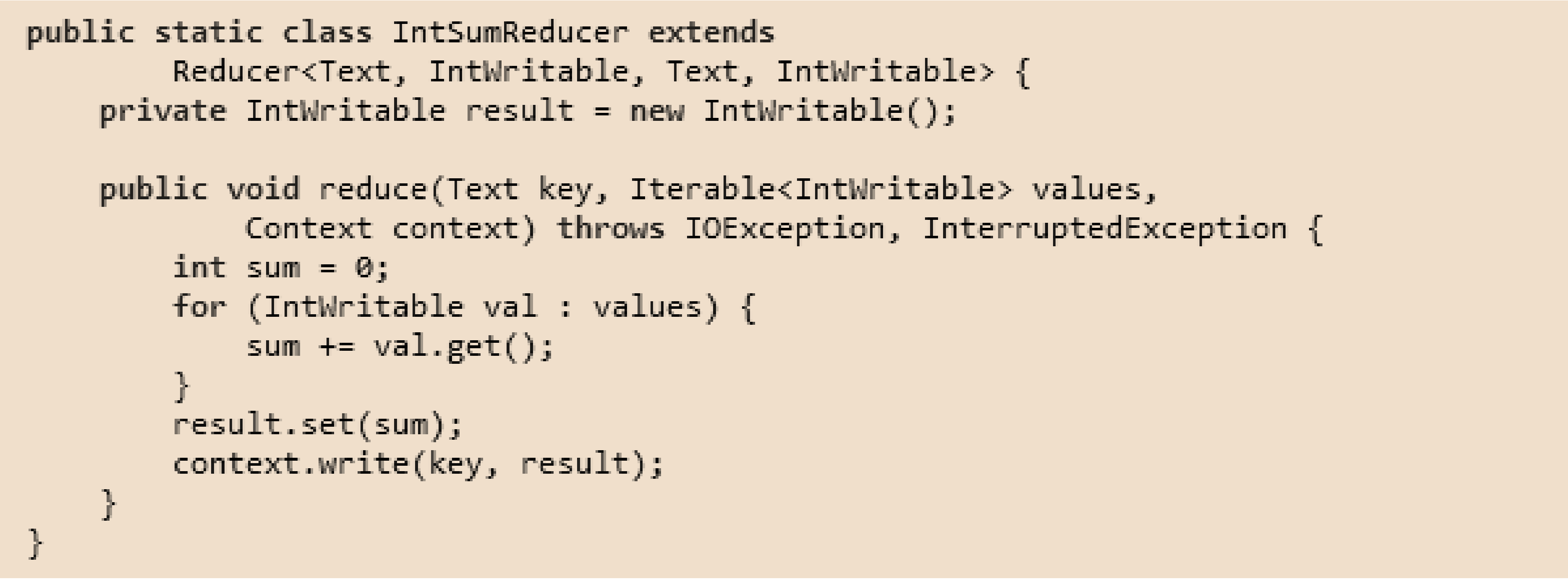
```
Reduce(K, V[ ]) {
  Int count = 0;
  For each v in V
    count += v;
  Collect(K, count);
}
```

WordCount问题—mapper设计与实现



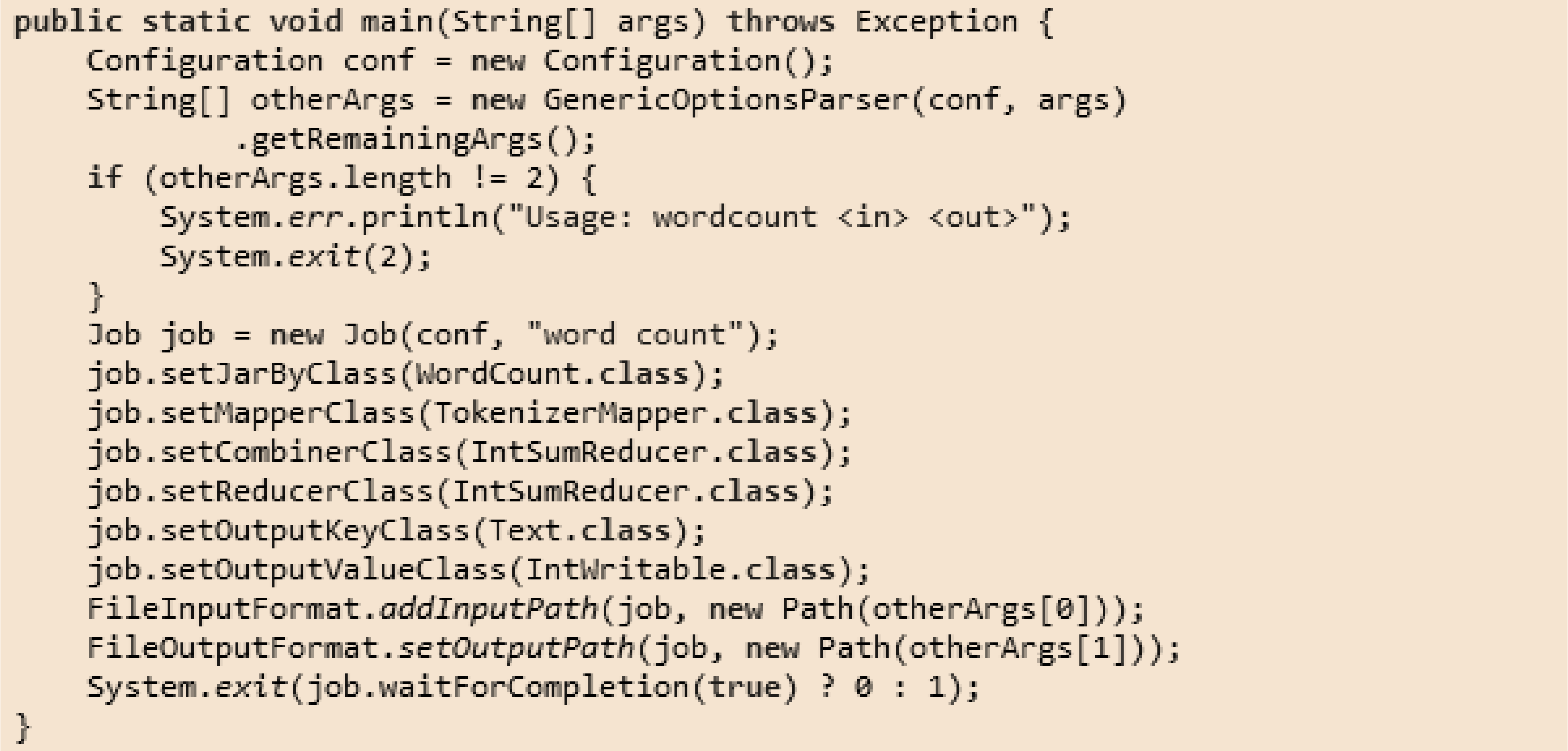
```
public static class TokenizerMapper
     extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context
                  ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}
```

WordCount问题—reducer设计与实现



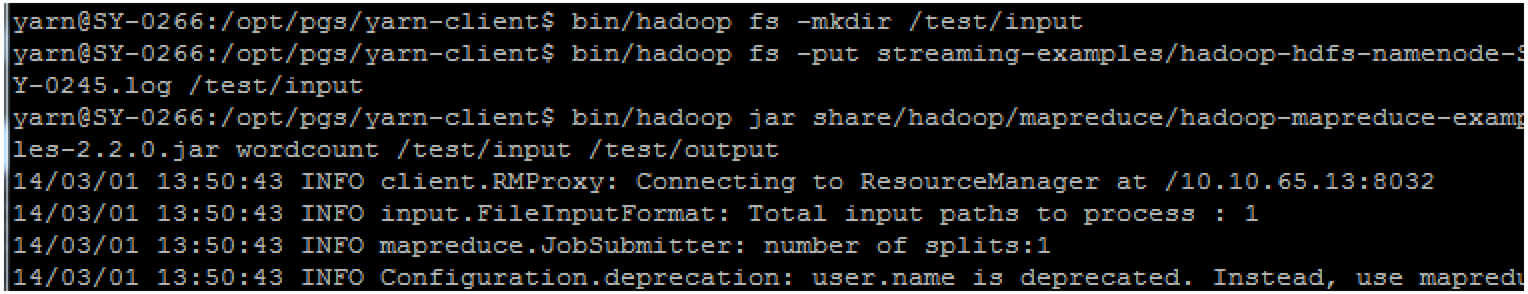
```
public static class IntSumReducer extends
        Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
            Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

WordCount问题—main函数设计与实现

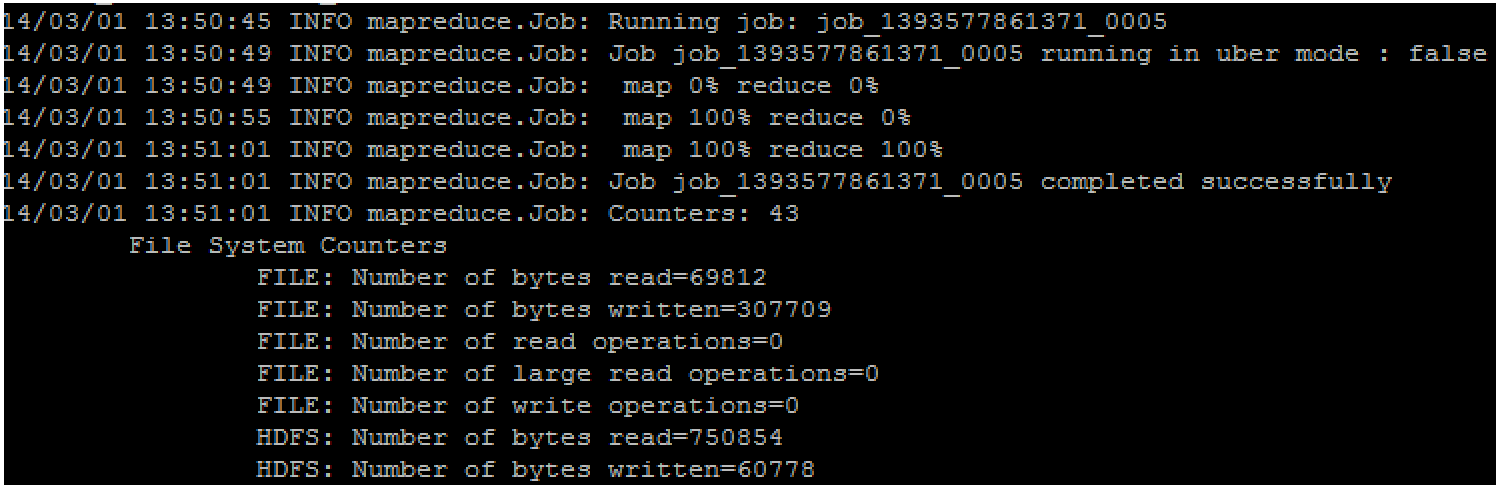


```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args)
            .getRemainingArgs();
    if (otherArgs.length != 2) {
        System.err.println("Usage: wordcount <in> <out>");
        System.exit(2);
    }
    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

WordCount问题—程序运行

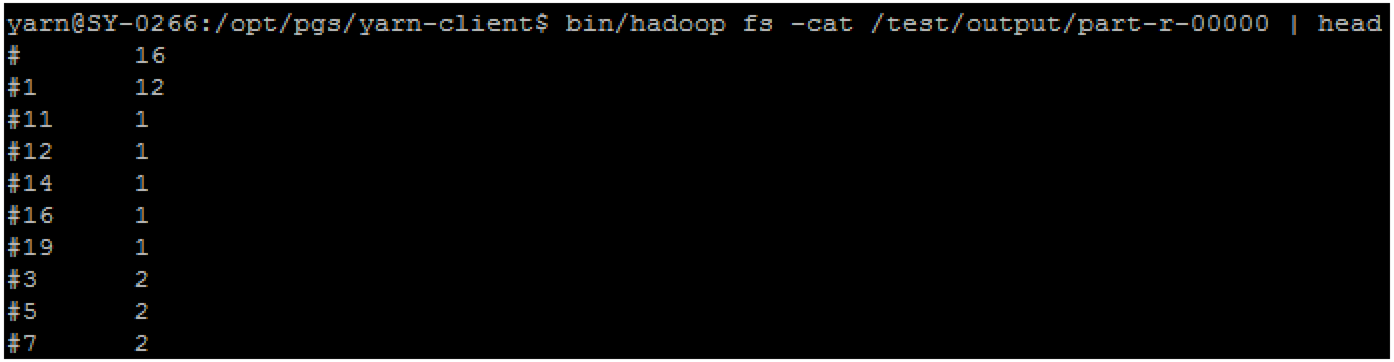


```
yarn@SY-0266:/opt/pgs/yarn-client$ bin/hadoop fs -mkdir /test/input
yarn@SY-0266:/opt/pgs/yarn-client$ bin/hadoop fs -put streaming-examples/hadoop-hdfs-namenode-S
Y-0245.log /test/input
yarn@SY-0266:/opt/pgs/yarn-client$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examp
les-2.2.0.jar wordcount /test/input /test/output
14/03/01 13:50:43 INFO client.RMProxy: Connecting to ResourceManager at /10.10.65.13:8032
14/03/01 13:50:43 INFO input.FileInputFormat: Total input paths to process : 1
14/03/01 13:50:43 INFO mapreduce.JobSubmitter: number of splits:1
14/03/01 13:50:43 INFO Configuration.deprecation: user.name is deprecated. Instead, use mapredu
```



```
14/03/01 13:50:45 INFO mapreduce.Job: Running job: job_1393577861371_0005
14/03/01 13:50:49 INFO mapreduce.Job: Job job_1393577861371_0005 running in uber mode : false
14/03/01 13:50:49 INFO mapreduce.Job:  map 0% reduce 0%
14/03/01 13:50:55 INFO mapreduce.Job:  map 100% reduce 0%
14/03/01 13:51:01 INFO mapreduce.Job:  map 100% reduce 100%
14/03/01 13:51:01 INFO mapreduce.Job: Job job_1393577861371_0005 completed successfully
14/03/01 13:51:01 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=69812
                FILE: Number of bytes written=307709
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=750854
                HDFS: Number of bytes written=60778
```

WordCount问题—程序运行



```
yarn@SY-0266:/opt/pgs/yarn-client$ bin/hadoop fs -cat /test/output/part-r-00000 | head
#       16
#1      12
#11     1
#12     1
#14     1
#16     1
#19     1
#3      2
#5      2
#7      2
```

Wordcount问题—输入数据格式解析

- 使用默认的TextInputFormat

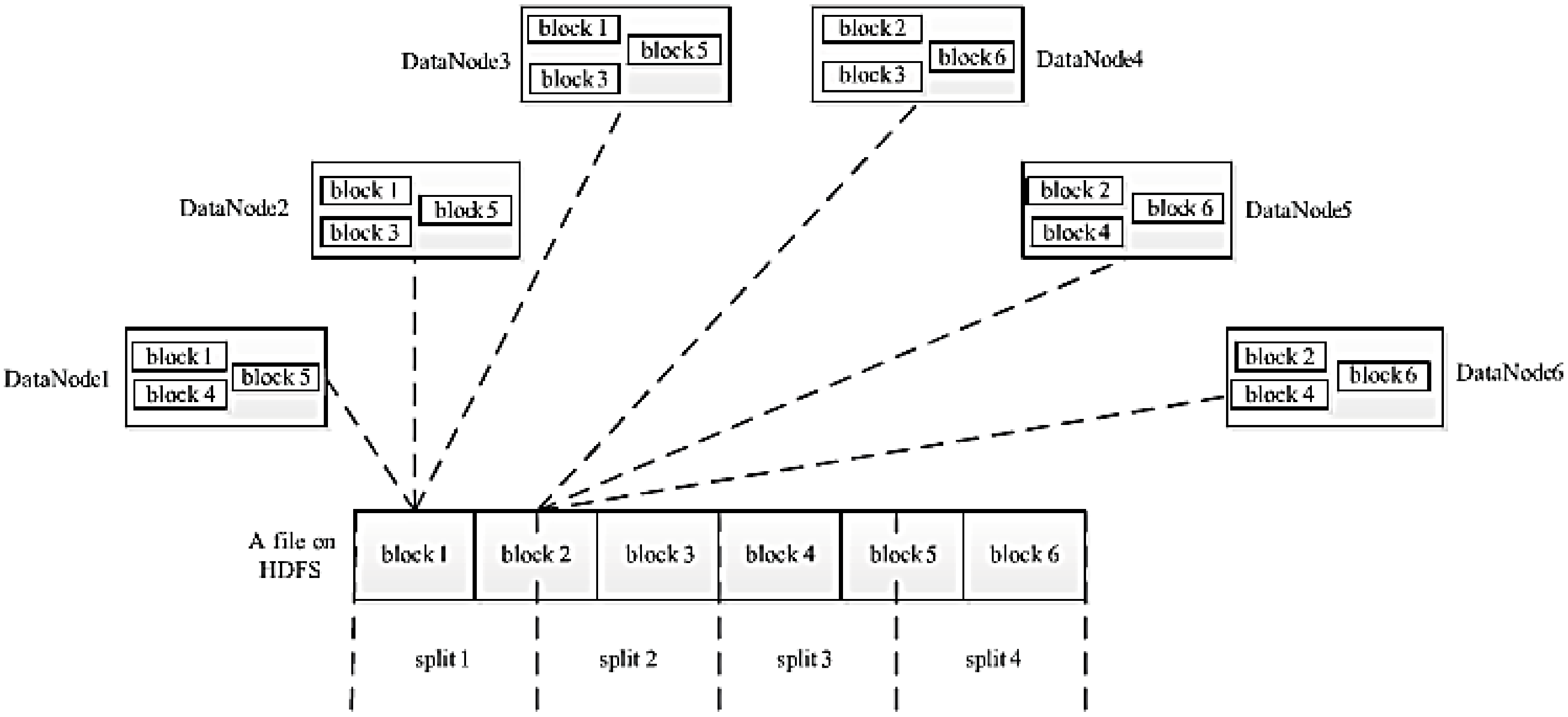
✓每个Map Task处理一个split;

✓一个split大小等于一个block;

✓如果最后一行数据被截断，则读取后一个block前半部分;

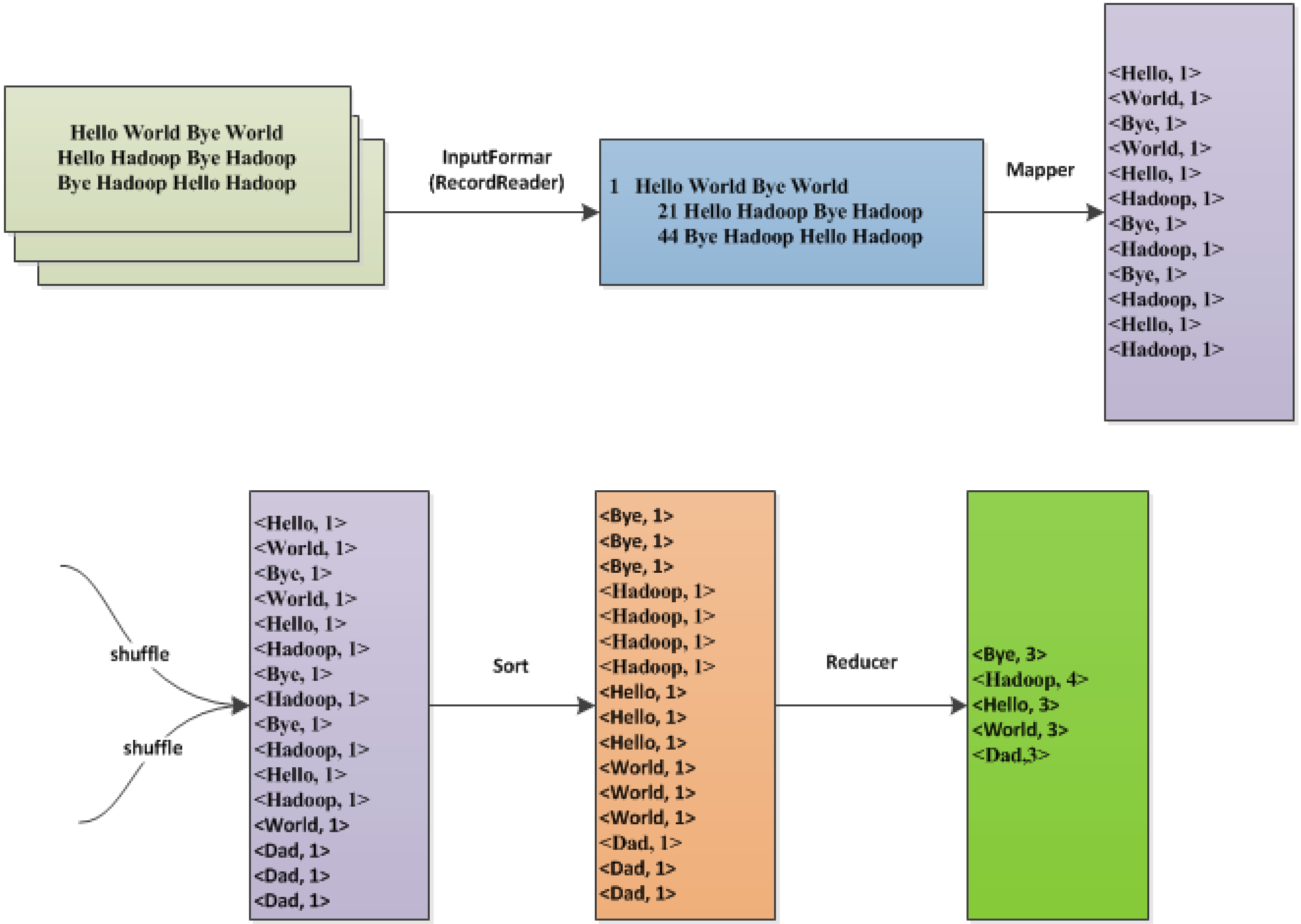
✓转换成key/value对，key是偏移量，value是行内容。

Wordcount问题—输入数据格式解析



split 与 block 的对应关系

Wordcount问题—数据流



Hello World Bye World
Hello Hadoop Bye Hadoop
Bye Hadoop Hello Hadoop

InputFormar
(RecordReader)

1 Hello World Bye World
21 Hello Hadoop Bye Hadoop
44 Bye Hadoop Hello Hadoop

Mapper

<Hello, 1>
<World, 1>
<Bye, 1>
<World, 1>
<Hello, 1>
<Hadoop, 1>
<Bye, 1>
<Hadoop, 1>
<Bye, 1>
<Hadoop, 1>
<Hello, 1>
<Hadoop, 1>

shuffle

shuffle

<Hello, 1>
<World, 1>
<Bye, 1>
<World, 1>
<Hello, 1>
<Hadoop, 1>
<Bye, 1>
<Hadoop, 1>
<Bye, 1>
<Hadoop, 1>
<Hello, 1>
<Hadoop, 1>
<World, 1>
<Dad, 1>
<Dad, 1>
<Dad, 1>

Sort

<Bye, 1>
<Bye, 1>
<Bye, 1>
<Hadoop, 1>
<Hadoop, 1>
<Hadoop, 1>
<Hadoop, 1>
<Hello, 1>
<Hello, 1>
<Hello, 1>
<World, 1>
<World, 1>
<World, 1>
<Dad, 1>
<Dad, 1>
<Dad, 1>

Reducer

<Bye, 3>
<Hadoop, 4>
<Hello, 3>
<World, 3>
<Dad,3>

InputFormat—输入数据解析

```
public interface InputFormat<K, V> {

  InputSplit[] getSplits(JobConf job, int numSplits) throws IOException;

  RecordReader<K, V> getRecordReader(InputSplit split,
                                                                   JobConf job,
                                                                  Reporter reporter) throws IOException;
}
```

- 默认为TextInputFormat，针对文本文件的；
- 用户可通过参数mapred.input.format.class设置InpuFormat实现

Mapper—map处理逻辑

org.apache.hadoop.mapred（旧API）：

```
public interface Mapper<K1, V1, K2, V2> extends JobConfigurable, Closeable {

  void map(K1 key, V1 value, OutputCollector<K2, V2> output, Reporter reporter)
  throws IOException;

}
```

- 新API位于org.apache.hadoop.mapreduce.Mapper中；
- 新API更加灵活。

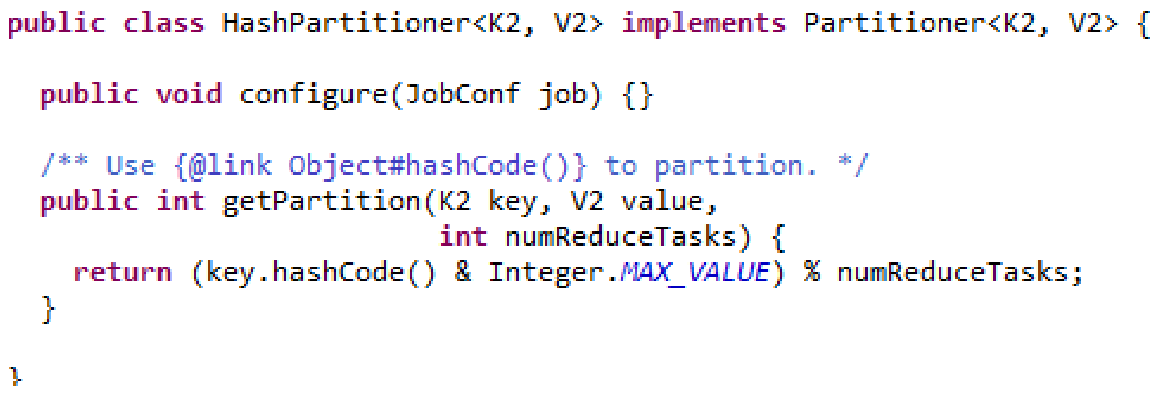
Partitioner—map输出结构分片

org.apache.hadoop.mapred（旧API）：

```
public interface Partitioner<K2, V2> extends JobConfigurable {

  int getPartition(K2 key, V2 value, int numPartitions);
}
```

org.apache.hadoop.mapreduce（新API）：

```
public abstract class Partitioner<KEY, VALUE> {

  public abstract int getPartition(KEY key, VALUE value, int numPartitions);
}
```



```
public class HashPartitioner<K2, V2> implements Partitioner<K2, V2> {

  public void configure(JobConf job) {}

  /** Use {@link Object#hashCode()} to partition. */
  public int getPartition(K2 key, V2 value,
                          int numReduceTasks) {
    return (key.hashCode() & Integer.MAX_VALUE) % numReduceTasks;
  }

}
```

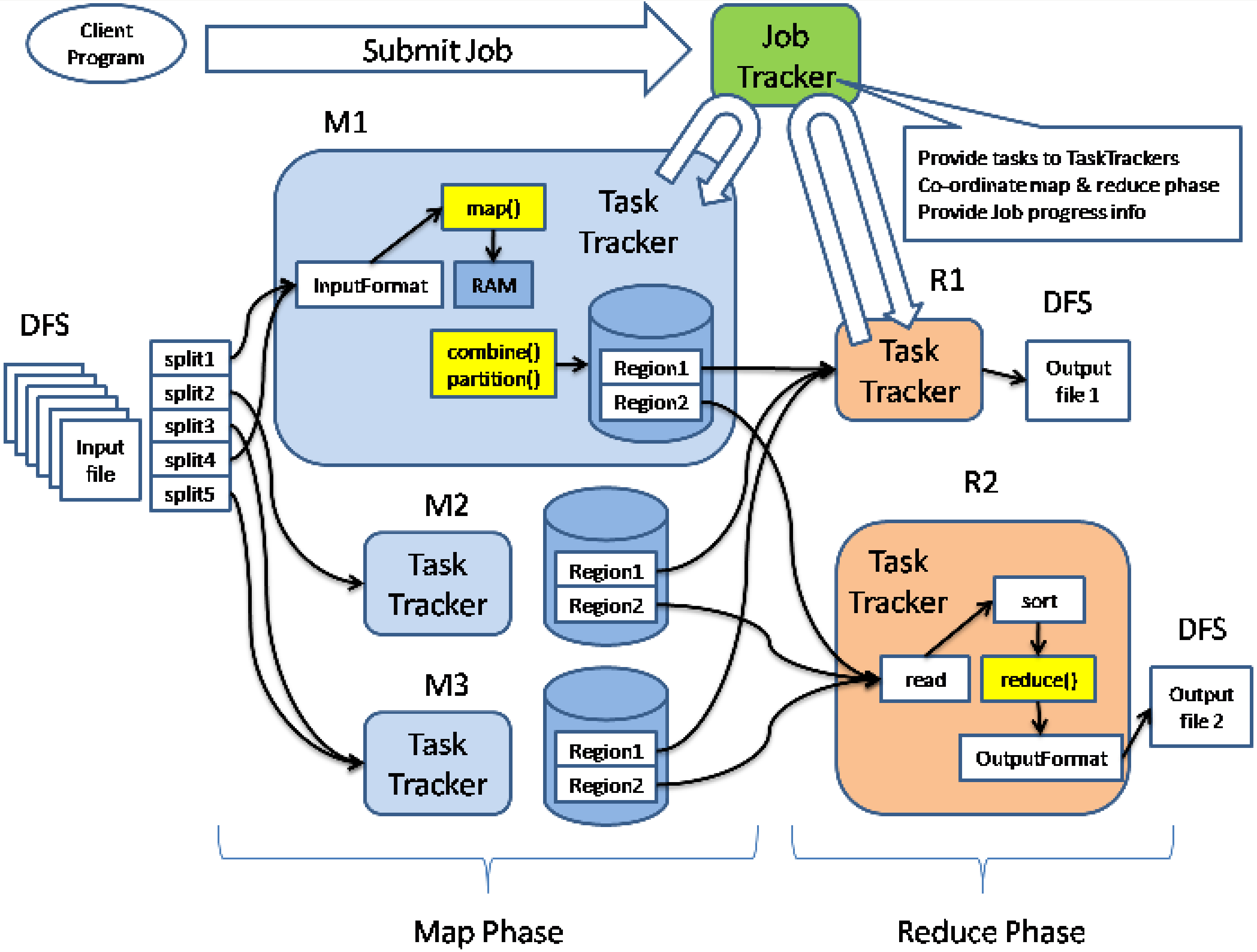
Reducer—reduce处理逻辑

org.apache.hadoop.mapred（旧API）：

```
public interface Reducer<K2, V2, K3, V3> extends JobConfigurable, Closeable {

  void reduce(K2 key, Iterator<V2> values,
              OutputCollector<K3, V3> output, Reporter reporter)
    throws IOException;
}
```

- **新API位于org.apache.hadoop.mapreduce.Reducer中；**
- **新API更加灵活。**

小结



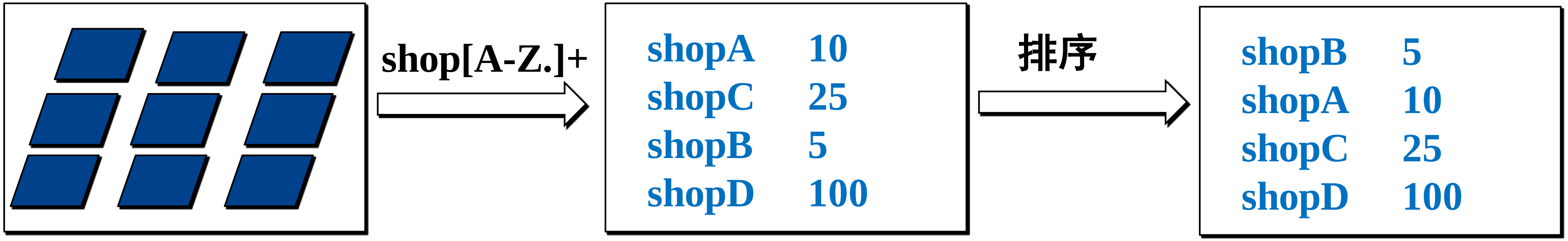
Provide tasks to TaskTrackers
Co-ordinate map & reduce phase
Provide Job progress info

实例2：Grep问题

一批TB或者PB量级的文档，需要完成以下功能：

- 搜索符合某种规则（正则表达式）的单词或者句子；
- 统计相应的单词或者句子的数目；
- 按照数目对其进行排序，并输出最终结果。

Grep问题解决思路



shop[A-Z.]+

shopA	10
shopC	25
shopB	5
shopD	100

排序

shopB	5
shopA	10
shopC	25
shopD	100

HDFS中的文档

频率统计结果

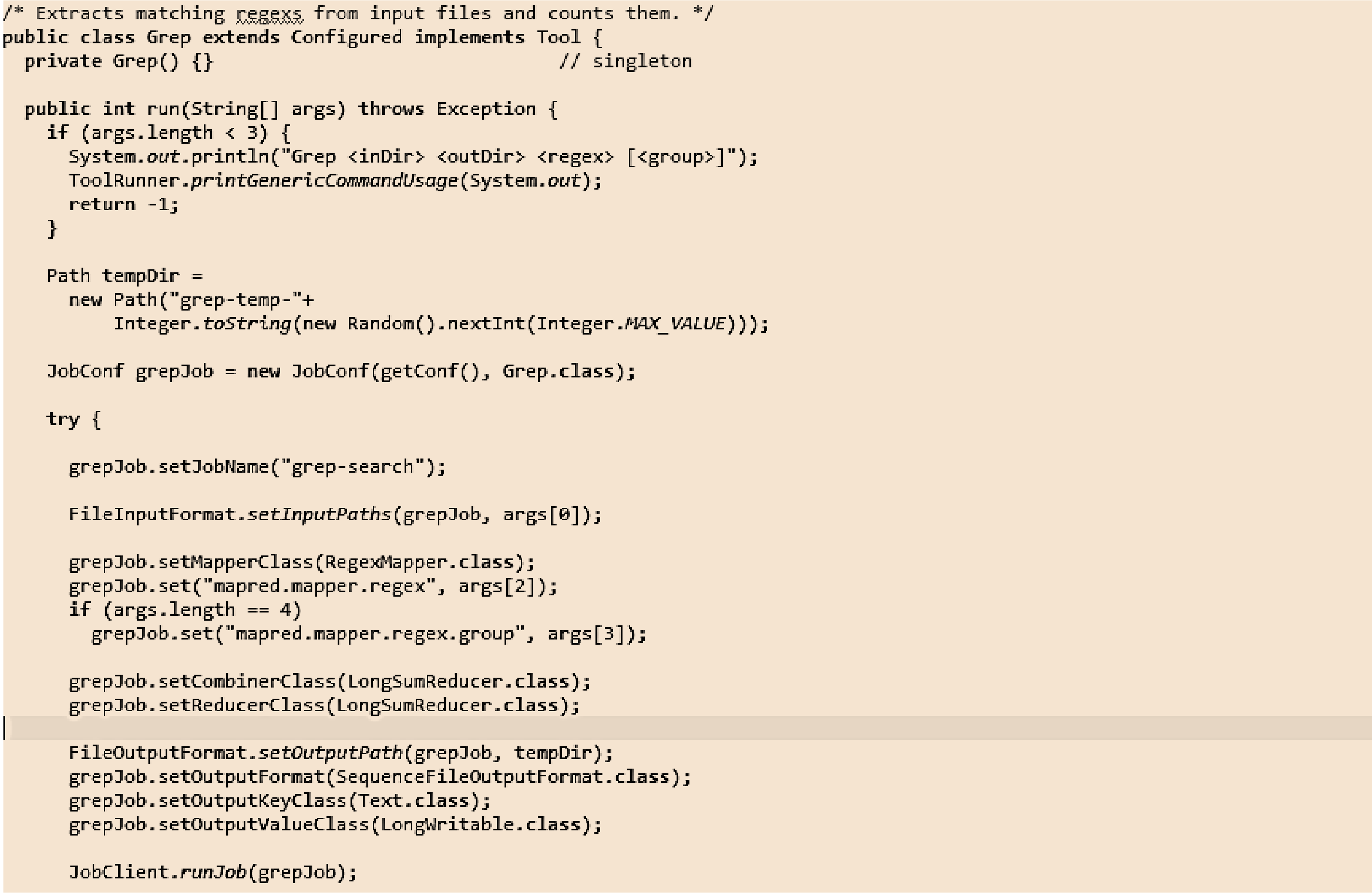
排序结果

Grep问题解决思路

分为两个作业：

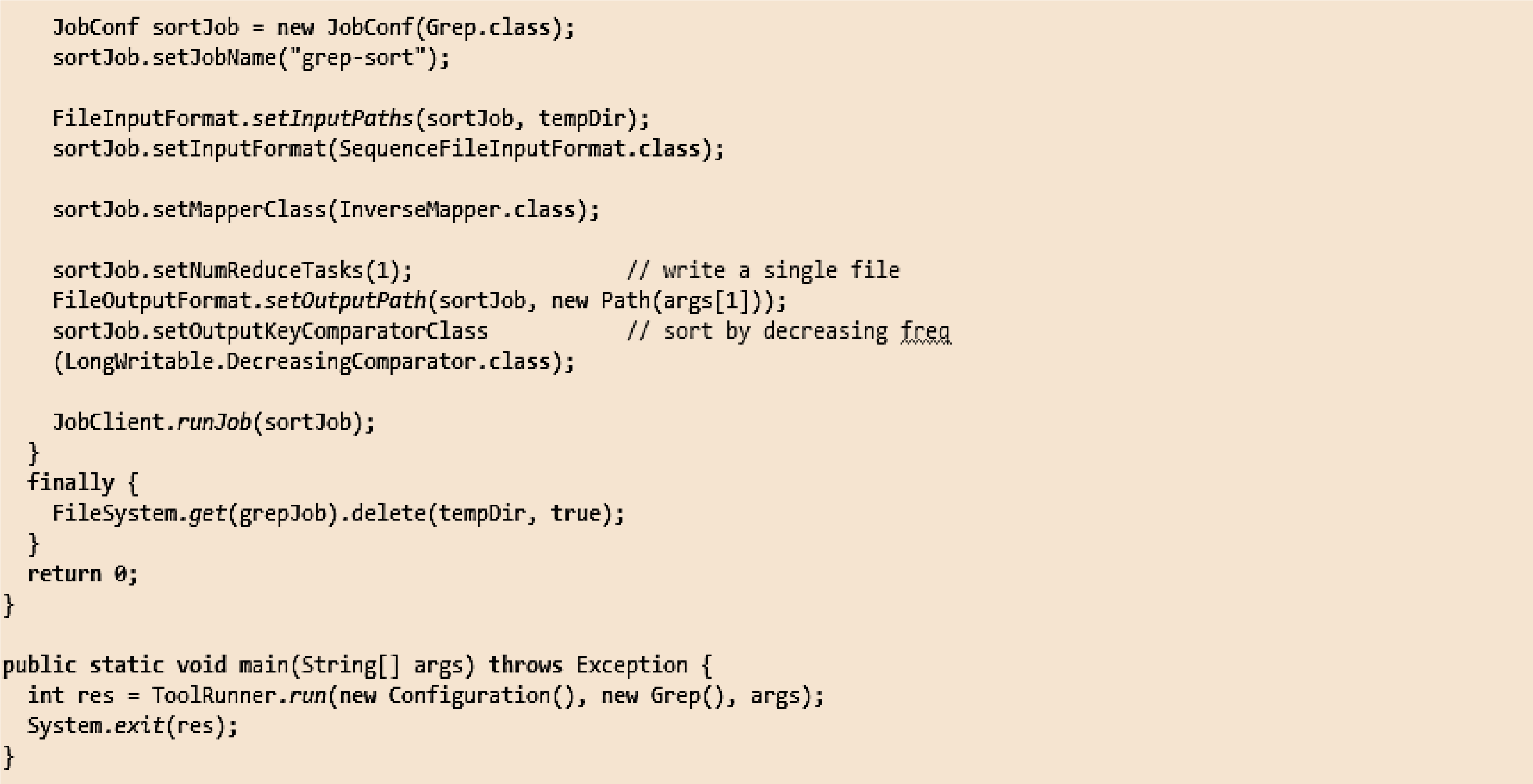
- ● 作业1：**WordCount**
  - ✓统计符合某种规则的单词数目
- ● 作业2：**Sort**
  - ✓按照单词数目进行全排序
  - ✓依赖于前一个作业的输出结果

Grep问题解决思路



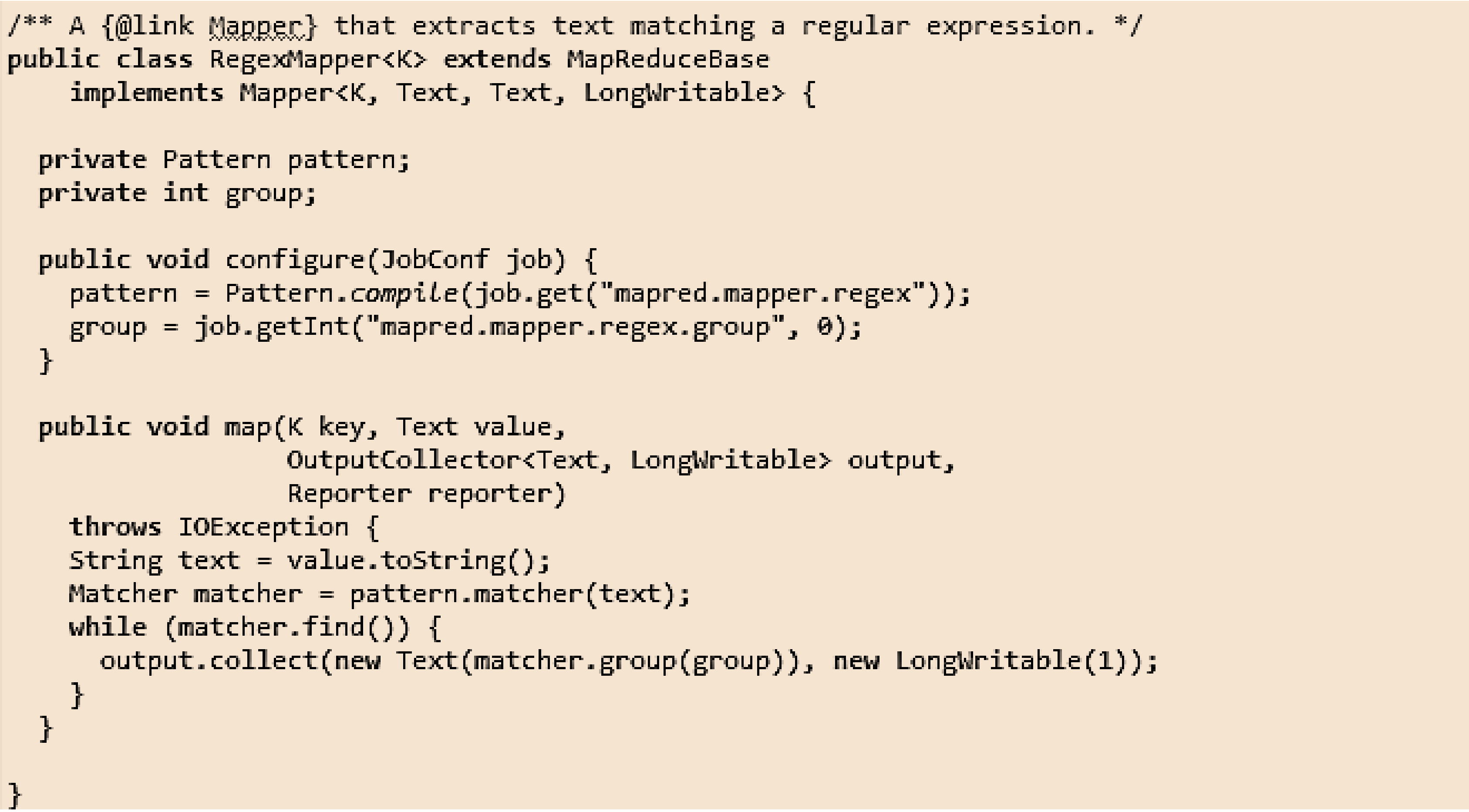
```
/* Extracts matching regexs from input files and counts them. */
public class Grep extends Configured implements Tool {
  private Grep() {}                                 // singleton

  public int run(String[] args) throws Exception {
    if (args.length < 3) {
      System.out.println("Grep <inDir> <outDir> <regex> [<group>]");
      ToolRunner.printGenericCommandUsage(System.out);
      return -1;
    }

    Path tempDir =
      new Path("grep-temp-"+
          Integer.toString(new Random().nextInt(Integer.MAX_VALUE)));

    JobConf grepJob = new JobConf(getConf(), Grep.class);

    try {

      grepJob.setJobName("grep-search");

      FileInputFormat.setInputPaths(grepJob, args[0]);

      grepJob.setMapperClass(RegexMapper.class);
      grepJob.set("mapred.mapper.regex", args[2]);
      if (args.length == 4)
        grepJob.set("mapred.mapper.regex.group", args[3]);

      grepJob.setCombinerClass(LongSumReducer.class);
      grepJob.setReducerClass(LongSumReducer.class);

      FileOutputFormat.setOutputPath(grepJob, tempDir);
      grepJob.setOutputFormat(SequenceFileOutputFormat.class);
      grepJob.setOutputKeyClass(Text.class);
      grepJob.setOutputValueClass(LongWritable.class);

      JobClient.runJob(grepJob);
```

Grep问题解决思路



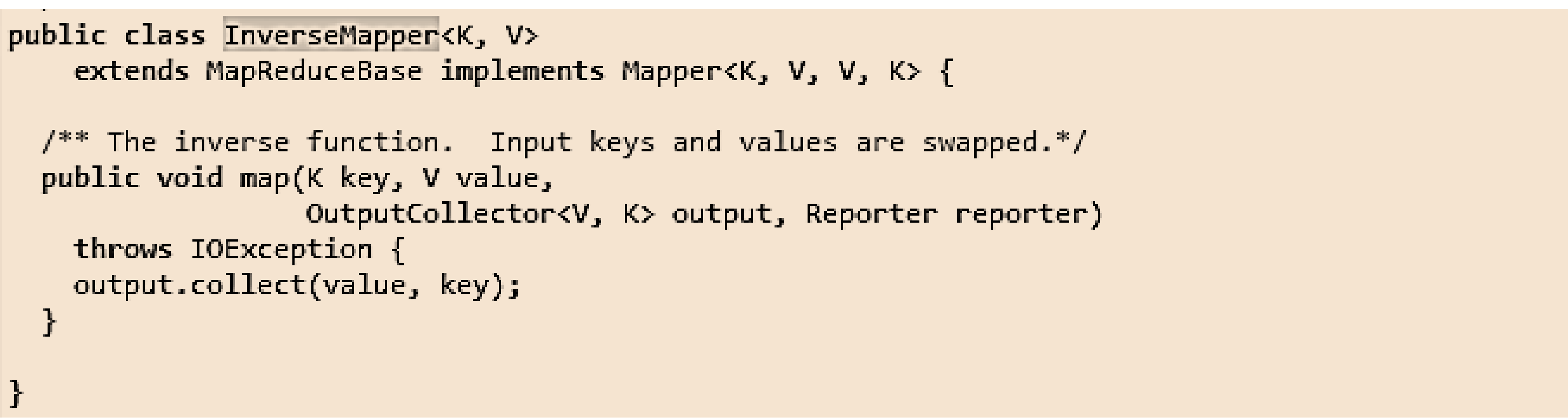
```
    JobConf sortJob = new JobConf(Grep.class);
    sortJob.setJobName("grep-sort");

    FileInputFormat.setInputPaths(sortJob, tempDir);
    sortJob.setInputFormat(SequenceFileInputFormat.class);

    sortJob.setMapperClass(InverseMapper.class);

    sortJob.setNumReduceTasks(1);                 // write a single file
    FileOutputFormat.setOutputPath(sortJob, new Path(args[1]));
    sortJob.setOutputKeyComparatorClass           // sort by decreasing freq
    (LongWritable.DecreasingComparator.class);

    JobClient.runJob(sortJob);
  }
  finally {
    FileSystem.get(grepJob).delete(tempDir, true);
  }
  return 0;
}

public static void main(String[] args) throws Exception {
  int res = ToolRunner.run(new Configuration(), new Grep(), args);
  System.exit(res);
}
```

Grep问题解决思路



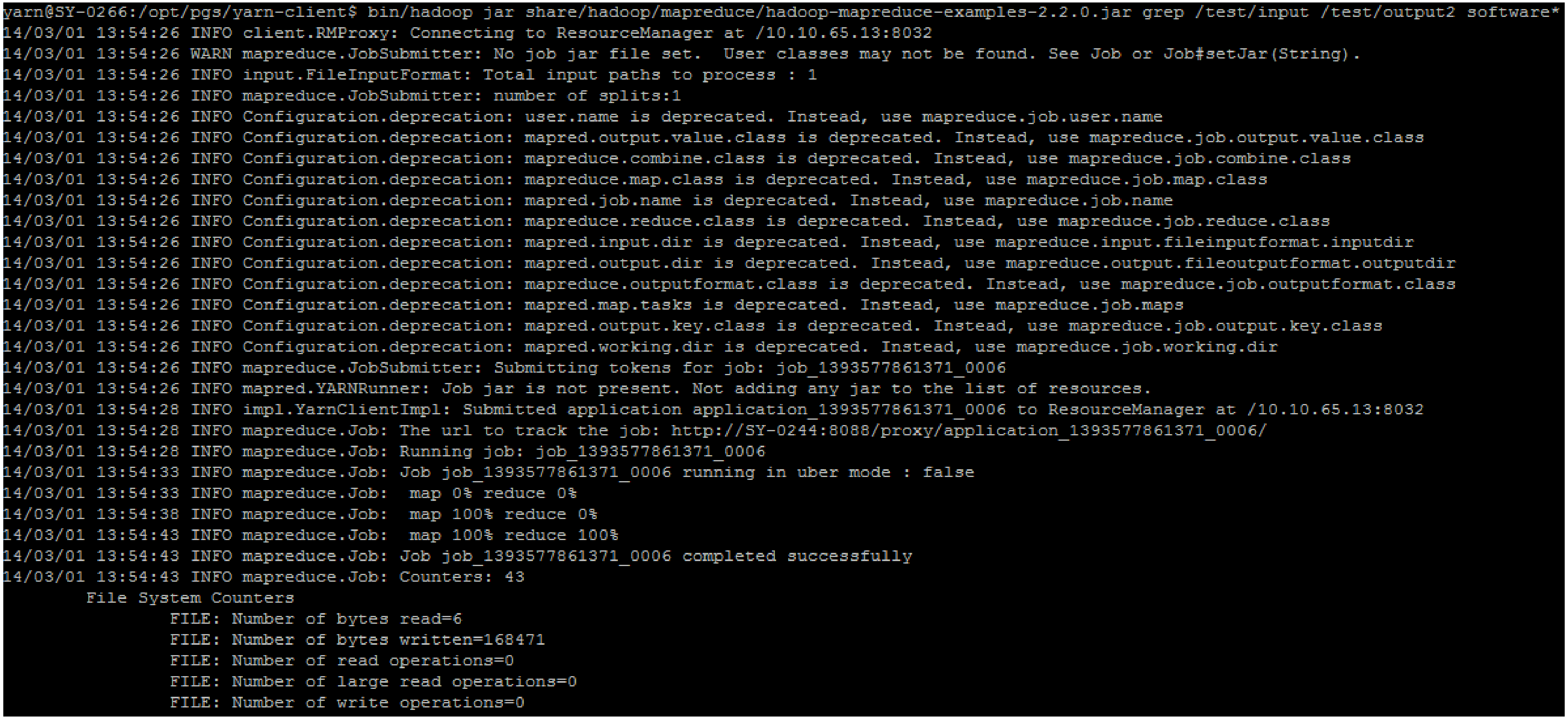
```java
/** A {@link Mapper} that extracts text matching a regular expression. */
public class RegexMapper<K> extends MapReduceBase
    implements Mapper<K, Text, Text, LongWritable> {

  private Pattern pattern;
  private int group;

  public void configure(JobConf job) {
    pattern = Pattern.compile(job.get("mapred.mapper.regex"));
    group = job.getInt("mapred.mapper.regex.group", 0);
  }

  public void map(K key, Text value,
                  OutputCollector<Text, LongWritable> output,
                  Reporter reporter)
    throws IOException {
    String text = value.toString();
    Matcher matcher = pattern.matcher(text);
    while (matcher.find()) {
      output.collect(new Text(matcher.group(group)), new LongWritable(1));
    }
  }

}
```

Grep问题解决思路



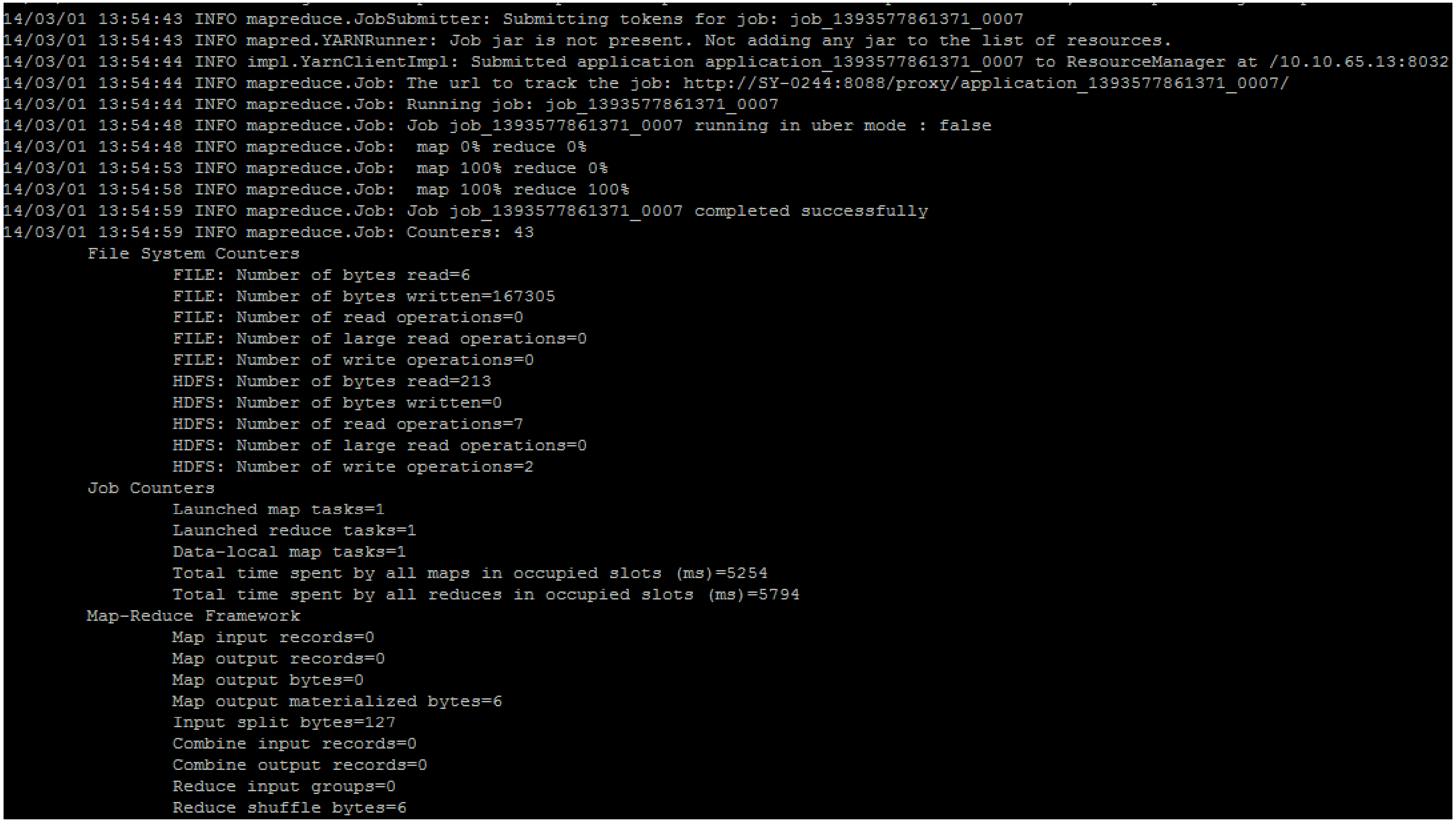
```
public class InverseMapper<K, V>
    extends MapReduceBase implements Mapper<K, V, V, K> {

  /** The inverse function.  Input keys and values are swapped.*/
  public void map(K key, V value,
                  OutputCollector<V, K> output, Reporter reporter)
    throws IOException {
    output.collect(value, key);
  }

}
```

Grep程序运行



```
yarn@SY-0266:/opt/pgs/yarn-client$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.2.0.jar grep /test/input /test/output2 software*
14/03/01 13:54:26 INFO client.RMProxy: Connecting to ResourceManager at /10.10.65.13:8032
14/03/01 13:54:26 WARN mapreduce.JobSubmitter: No job jar file set.  User classes may not be found. See Job or Job#setJar(String).
14/03/01 13:54:26 INFO input.FileInputFormat: Total input paths to process : 1
14/03/01 13:54:26 INFO mapreduce.JobSubmitter: number of splits:1
14/03/01 13:54:26 INFO Configuration.deprecation: user.name is deprecated. Instead, use mapreduce.job.user.name
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.output.value.class is deprecated. Instead, use mapreduce.job.output.value.class
14/03/01 13:54:26 INFO Configuration.deprecation: mapreduce.combine.class is deprecated. Instead, use mapreduce.job.combine.class
14/03/01 13:54:26 INFO Configuration.deprecation: mapreduce.map.class is deprecated. Instead, use mapreduce.job.map.class
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.job.name is deprecated. Instead, use mapreduce.job.name
14/03/01 13:54:26 INFO Configuration.deprecation: mapreduce.reduce.class is deprecated. Instead, use mapreduce.job.reduce.class
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputformat.inputdir
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.outputdir
14/03/01 13:54:26 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated. Instead, use mapreduce.job.outputformat.class
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.output.key.class is deprecated. Instead, use mapreduce.job.output.key.class
14/03/01 13:54:26 INFO Configuration.deprecation: mapred.working.dir is deprecated. Instead, use mapreduce.job.working.dir
14/03/01 13:54:26 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1393577861371_0006
14/03/01 13:54:26 INFO mapred.YARNRunner: Job jar is not present. Not adding any jar to the list of resources.
14/03/01 13:54:28 INFO impl.YarnClientImpl: Submitted application application_1393577861371_0006 to ResourceManager at /10.10.65.13:8032
14/03/01 13:54:28 INFO mapreduce.Job: The url to track the job: http://SY-0244:8088/proxy/application_1393577861371_0006/
14/03/01 13:54:28 INFO mapreduce.Job: Running job: job_1393577861371_0006
14/03/01 13:54:33 INFO mapreduce.Job: Job job_1393577861371_0006 running in uber mode : false
14/03/01 13:54:33 INFO mapreduce.Job:  map 0% reduce 0%
14/03/01 13:54:38 INFO mapreduce.Job:  map 100% reduce 0%
14/03/01 13:54:43 INFO mapreduce.Job:  map 100% reduce 100%
14/03/01 13:54:43 INFO mapreduce.Job: Job job_1393577861371_0006 completed successfully
14/03/01 13:54:43 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=6
                FILE: Number of bytes written=168471
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
```

Grep程序运行



```
14/03/01 13:54:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1393577861371_0007
14/03/01 13:54:43 INFO mapred.YARNRunner: Job jar is not present. Not adding any jar to the list of resources.
14/03/01 13:54:44 INFO impl.YarnClientImpl: Submitted application application_1393577861371_0007 to ResourceManager at /10.10.65.13:8032
14/03/01 13:54:44 INFO mapreduce.Job: The url to track the job: http://SY-0244:8088/proxy/application_1393577861371_0007/
14/03/01 13:54:44 INFO mapreduce.Job: Running job: job_1393577861371_0007
14/03/01 13:54:48 INFO mapreduce.Job: Job job_1393577861371_0007 running in uber mode : false
14/03/01 13:54:48 INFO mapreduce.Job:  map 0% reduce 0%
14/03/01 13:54:53 INFO mapreduce.Job:  map 100% reduce 0%
14/03/01 13:54:58 INFO mapreduce.Job:  map 100% reduce 100%
14/03/01 13:54:59 INFO mapreduce.Job: Job job_1393577861371_0007 completed successfully
14/03/01 13:54:59 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=6
                FILE: Number of bytes written=167305
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=213
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=7
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=5254
                Total time spent by all reduces in occupied slots (ms)=5794
        Map-Reduce Framework
                Map input records=0
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
                Input split bytes=127
                Combine input records=0
                Combine output records=0
                Reduce input groups=0
                Reduce shuffle bytes=6
```

目录

1. MapReduce的编程模型
2. MapReduce编程接口介绍
3. Java编程
4. 多语言编程
5. 总结

多语言程序设计思路

- 以标准输入流作为输入；

✓C++：cin

✓C：scanf

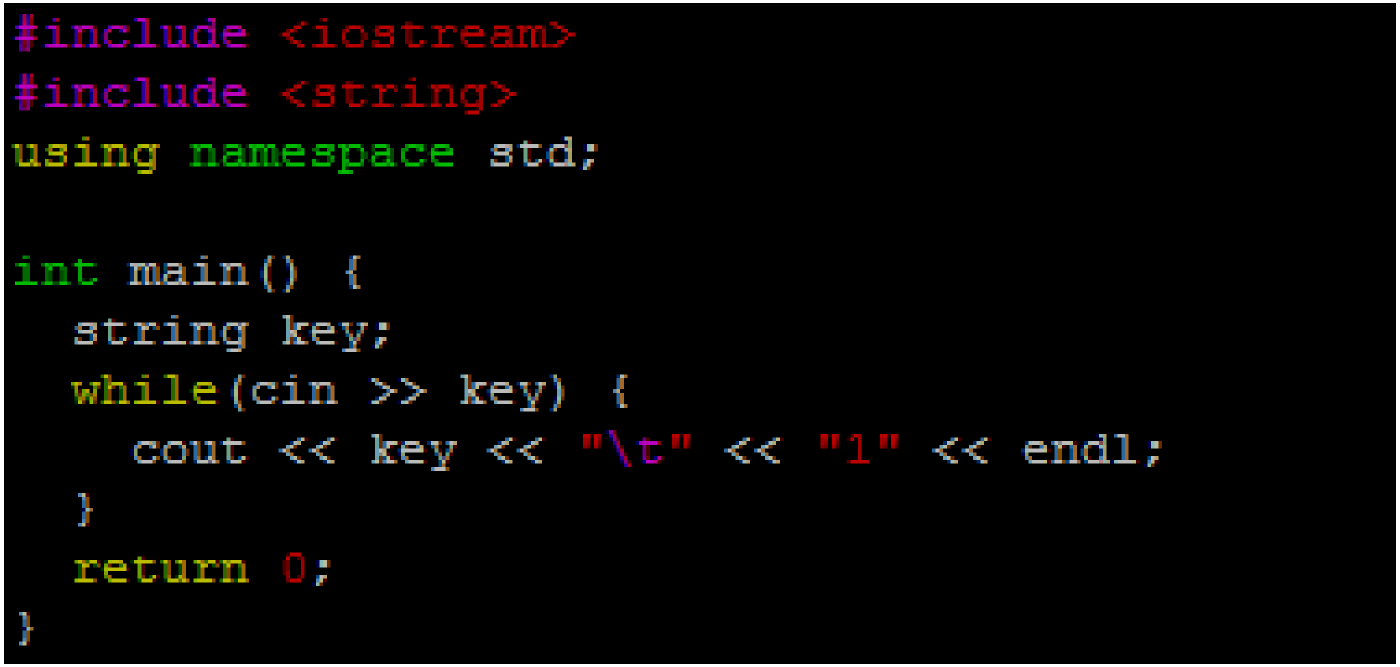
- 以标准输出流作为输出；

✓C++：cout

✓C：printf

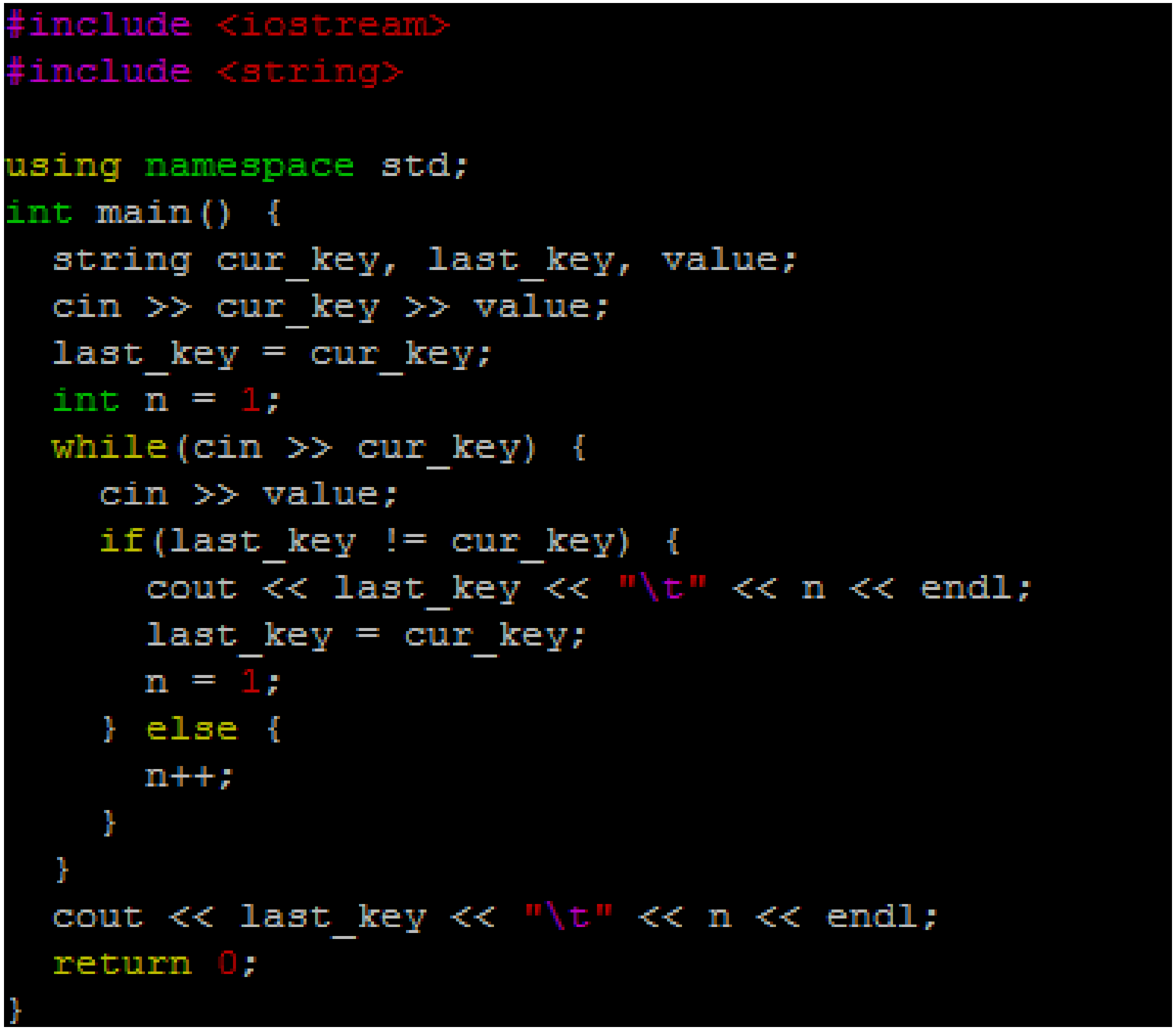
- 可实现Mapper和Reducer，其他组件（InputFormat、Partitioner等需要用Java语言实现）

实例1：用C++实现Wordcount（Mapper实现）



```
#include <iostream>
#include <string>
using namespace std;

int main() {
  string key;
  while(cin >> key) {
    cout << key << "\t" << "1" << endl;
  }
  return 0;
}
```

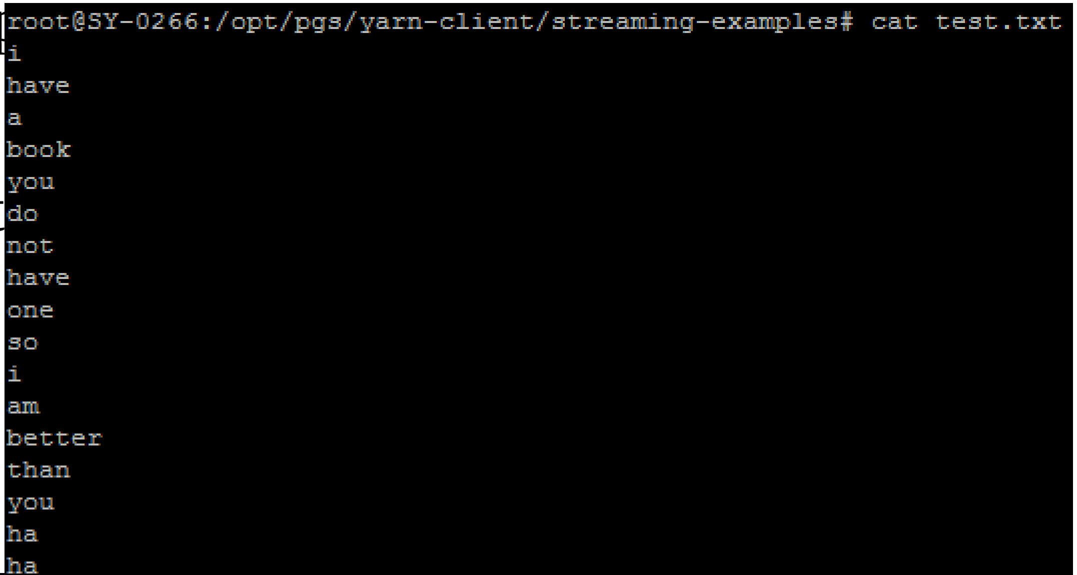
实例1：用C++实现Wordcount（Reducer实现）



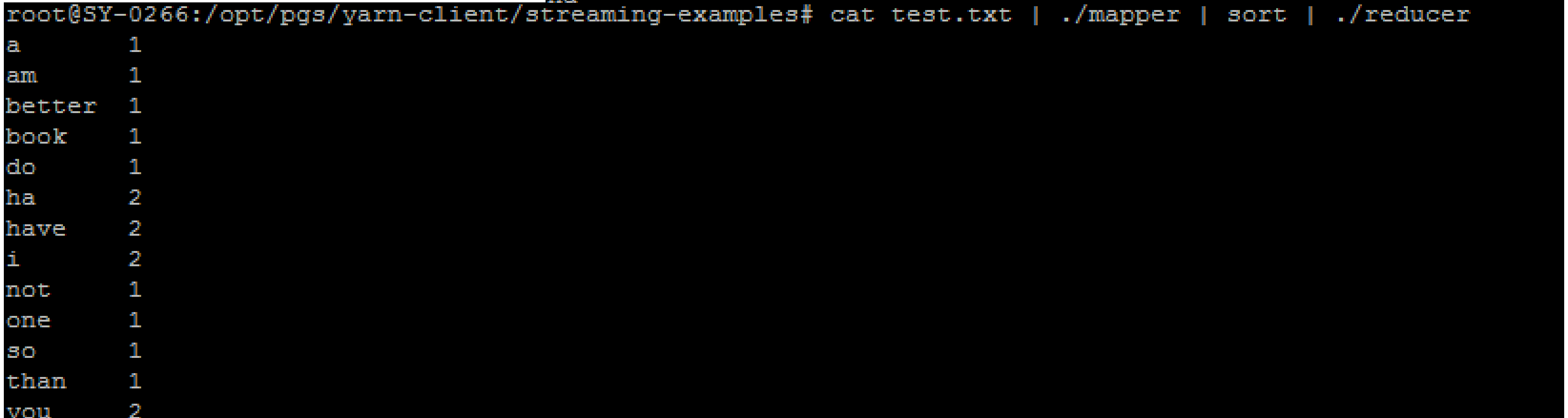
```
#include <iostream>
#include <string>

using namespace std;
int main() {
  string cur_key, last_key, value;
  cin >> cur_key >> value;
  last_key = cur_key;
  int n = 1;
  while(cin >> cur_key) {
    cin >> value;
    if(last_key != cur_key) {
      cout << last_key << "\t" << n << endl;
      last_key = cur_key;
      n = 1;
    } else {
      n++;
    }
  }
  cout << last_key << "\t" << n << endl;
  return 0;
}
```

实例1：用C++实现Wordcount（编译运行）

- 编译程序，生成可执行文件；
  - ✓ g++ -o mapper mapper.cpp
  - ✓ g++ -o reducer reducer.cpp
- 测试程序；
  - ✓ cat test.txt | ./mapper | sort



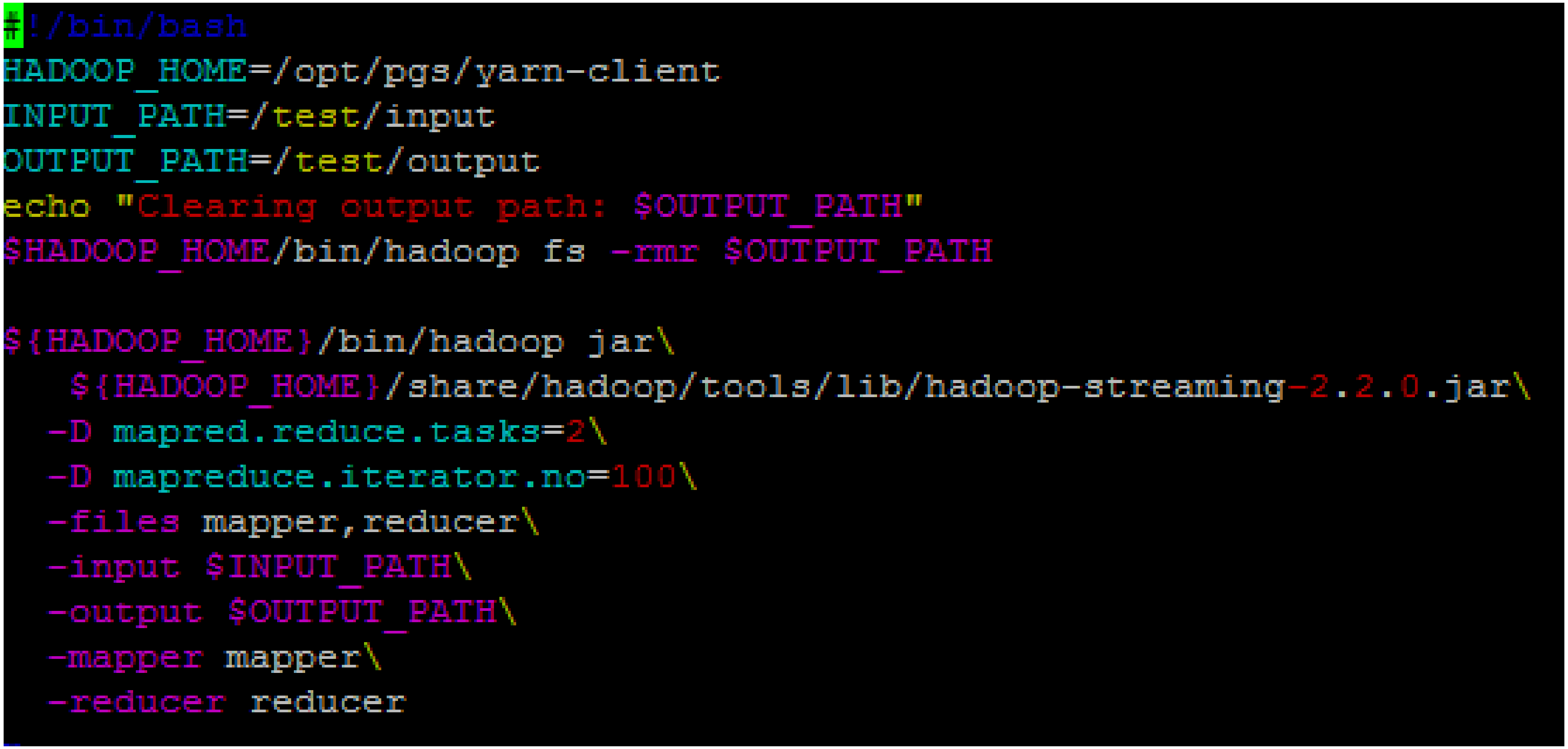
```
root@SY-0266:/opt/pgs/yarn-client/streaming-examples# cat test.txt
i
have
a
book
you
do
not
have
one
so
i
am
better
than
you
ha
ha
```



```
root@SY-0266:/opt/pgs/yarn-client/streaming-examples# cat test.txt | ./mapper | sort | ./reducer
a       1
am      1
better  1
book    1
do      1
ha      2
have    2
i       2
not     1
one     1
so      1
than    1
you     2
```

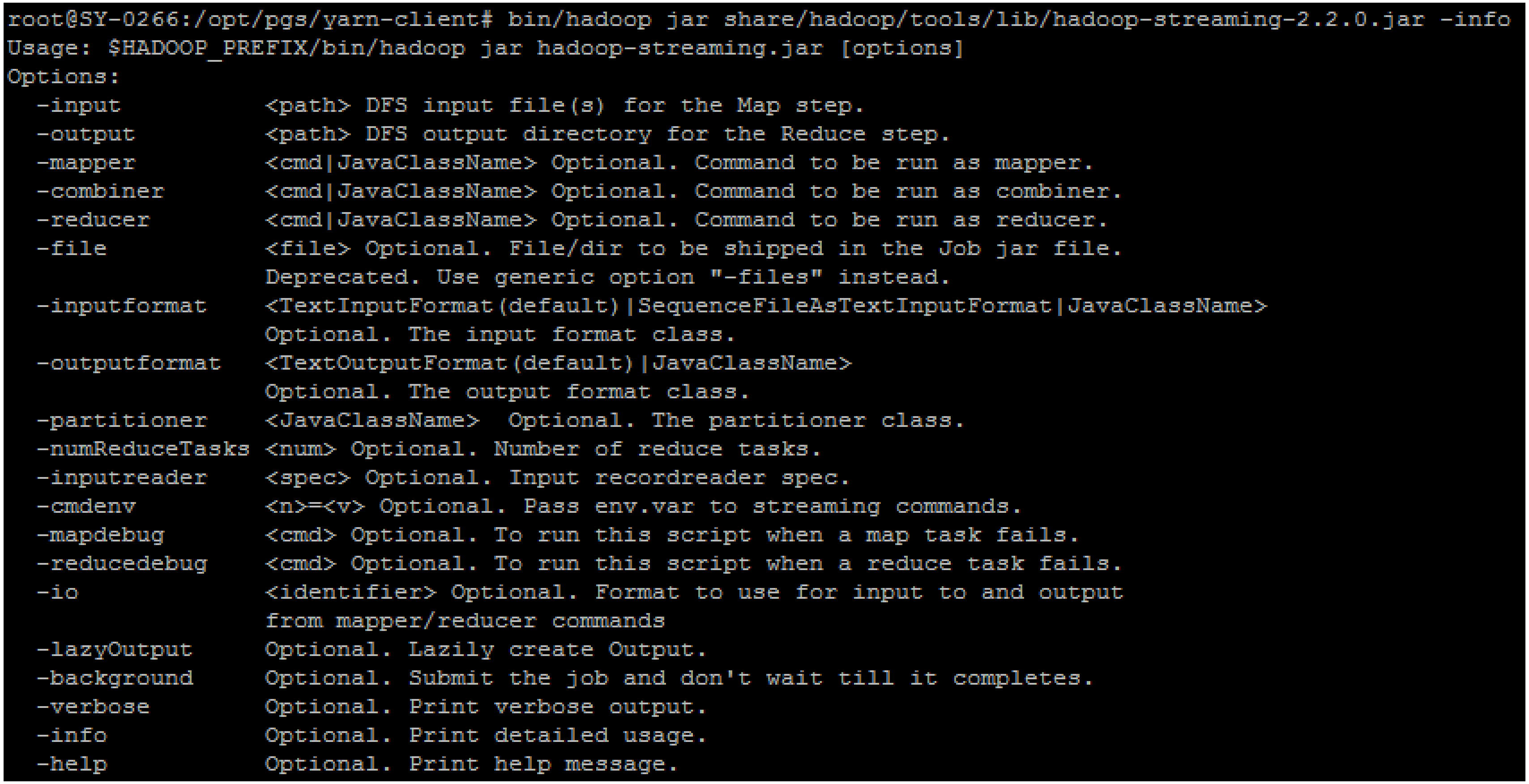
实例1：用C++实现Wordcount（编译运行）

- **Hadoop上运行Wordcount程序**

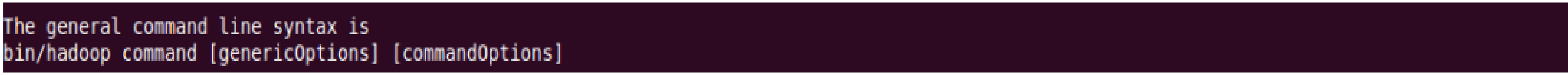


```
#!/bin/bash
HADOOP_HOME=/opt/pgs/yarn-client
INPUT_PATH=/test/input
OUTPUT_PATH=/test/output
echo "Clearing output path: $OUTPUT_PATH"
$HADOOP_HOME/bin/hadoop fs -rmr $OUTPUT_PATH

${HADOOP_HOME}/bin/hadoop jar\
   ${HADOOP_HOME}/share/hadoop/tools/lib/hadoop-streaming-2.2.0.jar\
  -D mapred.reduce.tasks=2\
  -D mapreduce.iterator.no=100\
  -files mapper,reducer\
  -input $INPUT_PATH\
  -output $OUTPUT_PATH\
  -mapper mapper\
  -reducer reducer
```

Streaming程序运行方式



```
root@SY-0266:/opt/pgs/yarn-client# bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-2.2.0.jar -info
Usage: $HADOOP_PREFIX/bin/hadoop jar hadoop-streaming.jar [options]
Options:
  -input          <path> DFS input file(s) for the Map step.
  -output         <path> DFS output directory for the Reduce step.
  -mapper         <cmd|JavaClassName> Optional. Command to be run as mapper.
  -combiner       <cmd|JavaClassName> Optional. Command to be run as combiner.
  -reducer        <cmd|JavaClassName> Optional. Command to be run as reducer.
  -file           <file> Optional. File/dir to be shipped in the Job jar file.
                  Deprecated. Use generic option "-files" instead.
  -inputformat    <TextInputFormat(default)|SequenceFileAsTextInputFormat|JavaClassName>
                  Optional. The input format class.
  -outputformat   <TextOutputFormat(default)|JavaClassName>
                  Optional. The output format class.
  -partitioner    <JavaClassName>  Optional. The partitioner class.
  -numReduceTasks <num> Optional. Number of reduce tasks.
  -inputreader    <spec> Optional. Input recordreader spec.
  -cmdenv         <n>=<v> Optional. Pass env.var to streaming commands.
  -mapdebug       <cmd> Optional. To run this script when a map task fails.
  -reducedebug    <cmd> Optional. To run this script when a reduce task fails.
  -io             <identifier> Optional. Format to use for input to and output
                  from mapper/reducer commands
  -lazyOutput     Optional. Lazily create Output.
  -background     Optional. Submit the job and don't wait till it completes.
  -verbose        Optional. Print verbose output.
  -info           Optional. Print detailed usage.
  -help           Optional. Print help message.
```



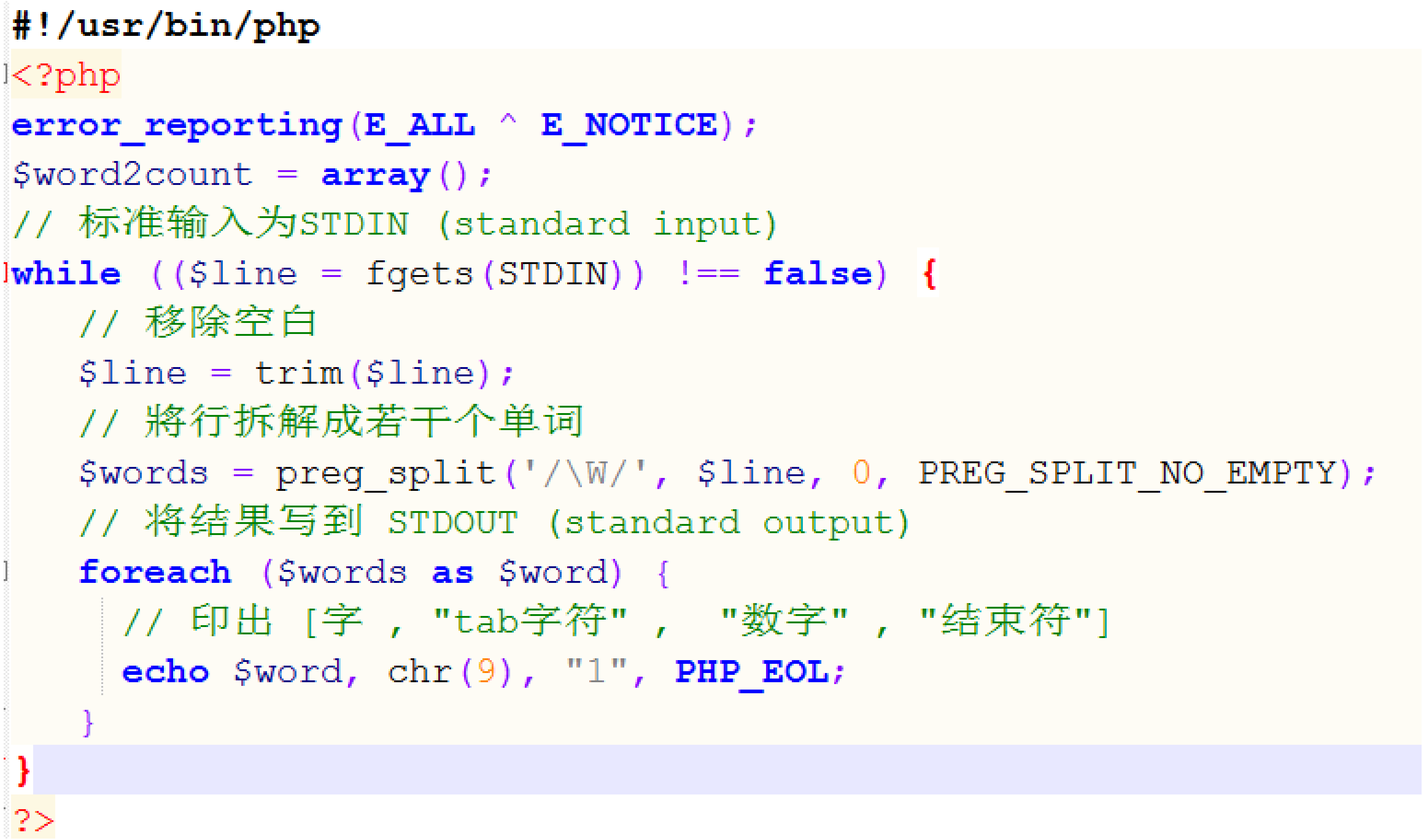
```
The general command line syntax is
bin/hadoop command [genericOptions] [commandOptions]
```

Streaming程序运行方式说明

- 区分通用参数和命令行参数，通用参数应放在命令行参数前面，否则不起作用。通用参数有7个：
  - ✓ -conf -D -fs -jt -files - libjars -archives
- “-file”或者“-files”参数，设置要分发到各个节点上的文件，对于mapper和reducer文件，必须要用或者“-files”或“-file”指定。
  - ✓ -files mapper,reducer
  - ✓ -file mapper -file reducer
- 每次运行程序前，清空输出目录

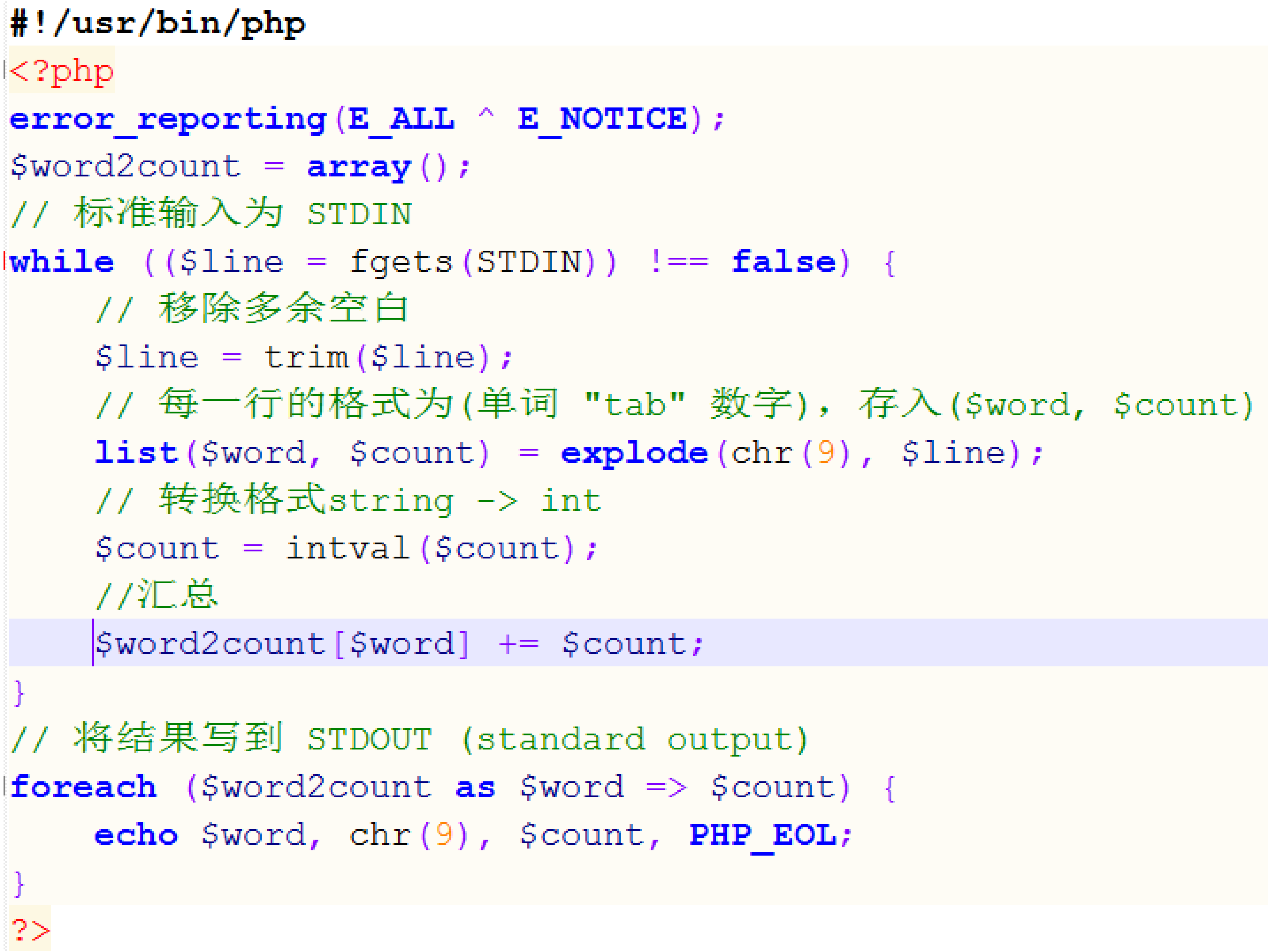
```
bin/hadoop fs -rmr /test/output
```

实例2：用PHP实现Wordcount（Mapper实现）



```php
#!/usr/bin/php
<?php
error_reporting(E_ALL ^ E_NOTICE);
$word2count = array();
// 标准输入为STDIN (standard input)
while (($line = fgets(STDIN)) !== false) {
    // 移除空白
    $line = trim($line);
    // 将行拆解成若干个单词
    $words = preg_split('/\W/', $line, 0, PREG_SPLIT_NO_EMPTY);
    // 将结果写到 STDOUT (standard output)
    foreach ($words as $word) {
      // 印出 [字 , "tab字符" ,  "数字" , "结束符"]
      echo $word, chr(9), "1", PHP_EOL;
    }
}
?>
```

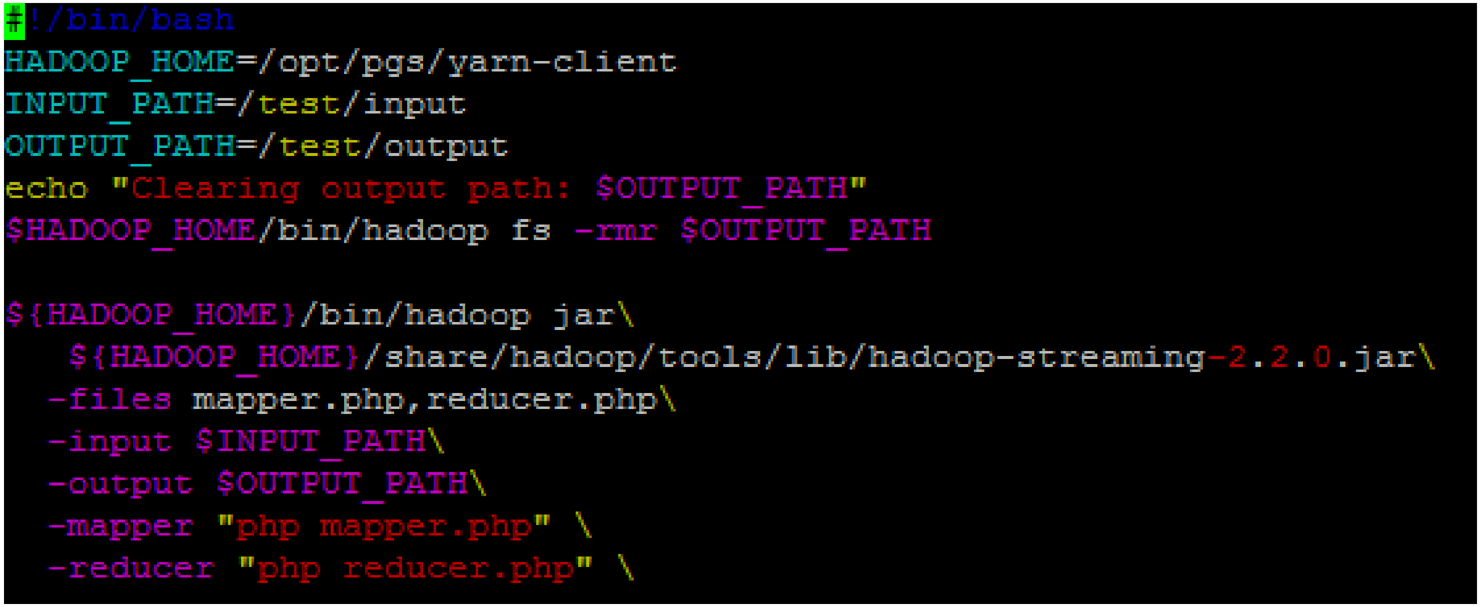
实例2：用PHP实现Wordcount （Reducer实现）



```php
#!/usr/bin/php
<?php
error_reporting(E_ALL ^ E_NOTICE);
$word2count = array();
// 标准输入为 STDIN
while (($line = fgets(STDIN)) !== false) {
    // 移除多余空白
    $line = trim($line);
    // 每一行的格式为(单词 "tab" 数字)，存入($word, $count)
    list($word, $count) = explode(chr(9), $line);
    // 转换格式string -> int
    $count = intval($count);
    //汇总
    $word2count[$word] += $count;
}
// 将结果写到 STDOUT (standard output)
foreach ($word2count as $word => $count) {
    echo $word, chr(9), $count, PHP_EOL;
}
?>
```

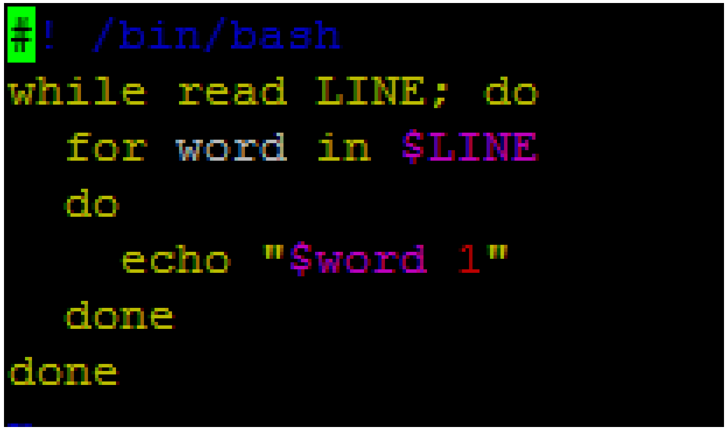
实例2：用PHP实现Wordcount（测试运行）

- ●测试mapper和reducer
  - ✓ cat test.txt| php mapper.php | sort | php reducer.php
- ●在Hadoop上运行



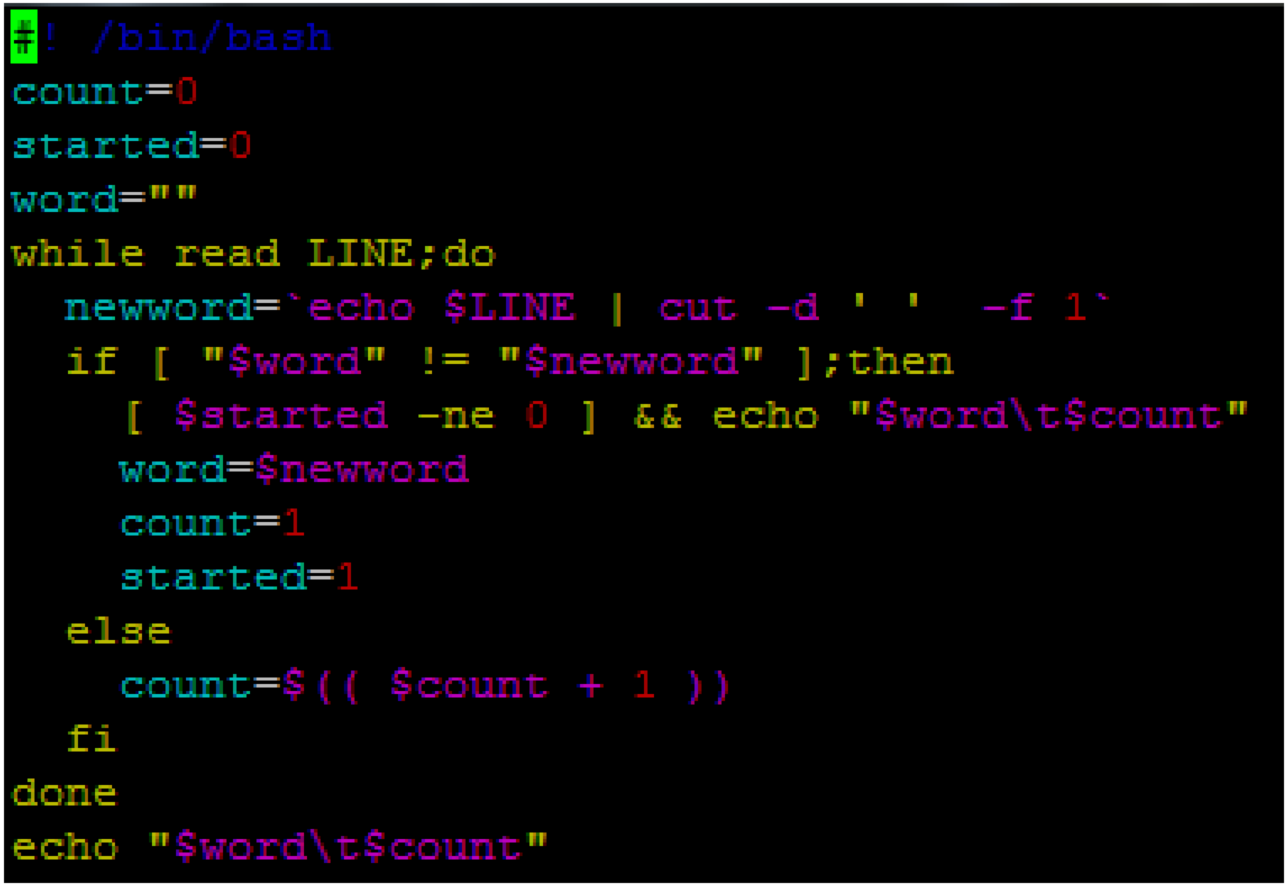
```
#!/bin/bash
HADOOP_HOME=/opt/pgs/yarn-client
INPUT_PATH=/test/input
OUTPUT_PATH=/test/output
echo "Clearing output path: $OUTPUT_PATH"
$HADOOP_HOME/bin/hadoop fs -rmr $OUTPUT_PATH

${HADOOP_HOME}/bin/hadoop jar\
   ${HADOOP_HOME}/share/hadoop/tools/lib/hadoop-streaming-2.2.0.jar\
  -files mapper.php,reducer.php\
  -input $INPUT_PATH\
  -output $OUTPUT_PATH\
  -mapper "php mapper.php" \
  -reducer "php reducer.php" \
```

实例3：用Shell脚本语言实现Wordcount（Mapper实现）



```
#! /bin/bash
while read LINE; do
  for word in $LINE
  do
    echo "$word 1"
  done
done
```

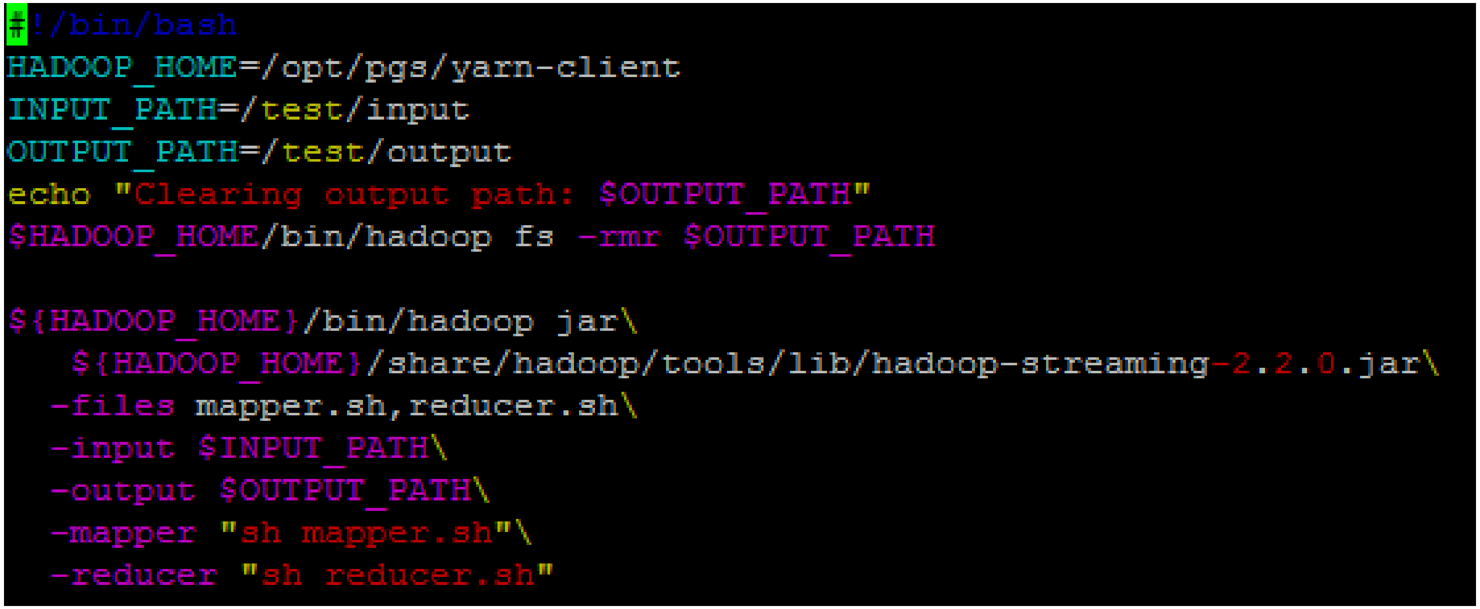
实例3：用Shell脚本语言实现Wordcount（Reducer实现）



```
#! /bin/bash
count=0
started=0
word=""
while read LINE;do
  newword=`echo $LINE | cut -d ' '  -f 1`
  if [ "$word" != "$newword" ];then
    [ $started -ne 0 ] && echo "$word\t$count"
    word=$newword
    count=1
    started=1
  else
    count=$(( $count + 1 ))
  fi
done
echo "$word\t$count"
```

实例3：用Shell脚本语言实现Wordcount（测试运行）

- **测试mapper和reducer**
  - ✓ **cat test.txt| sh mapper.sh | sort | sh reducer.sh**
- **在Hadoop上运行**



```
#!/bin/bash
HADOOP_HOME=/opt/pgs/yarn-client
INPUT_PATH=/test/input
OUTPUT_PATH=/test/output
echo "Clearing output path: $OUTPUT_PATH"
$HADOOP_HOME/bin/hadoop fs -rmr $OUTPUT_PATH

${HADOOP_HOME}/bin/hadoop jar\
   ${HADOOP_HOME}/share/hadoop/tools/lib/hadoop-streaming-2.2.0.jar\
  -files mapper.sh,reducer.sh\
  -input $INPUT_PATH\
  -output $OUTPUT_PATH\
  -mapper "sh mapper.sh"\
  -reducer "sh reducer.sh"
```

Hadoop Streaming高级编程

- 定义Hadoop Counter

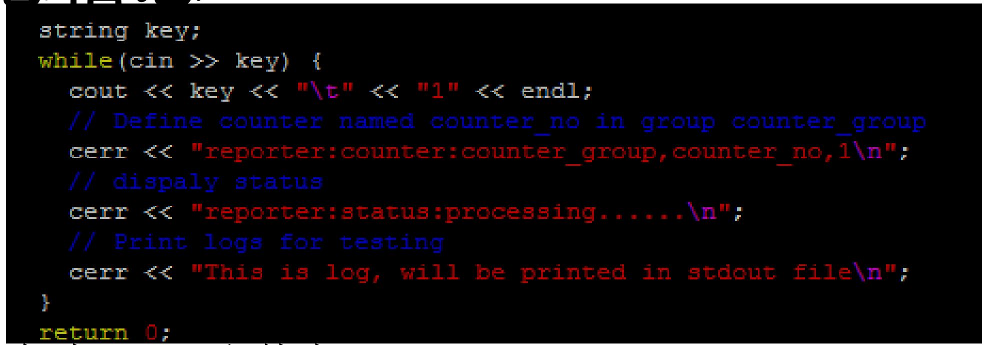
✓ 使用标准错误输出，格式为

reporter:counter:<group>,<counter>,<amount>

- 在运行过程中展示状态信息

✓ 使用标准错误输出，格式为

reporter:status:<message>



```
string key;
while(cin >> key) {
  cout << key << "\t" << "1" << endl;
  // Define counter named counter_no in group counter_group
  cerr << "reporter:counter:counter_group,counter_no,1\n";
  // dispaly status
  cerr << "reporter:status:processing......\n";
  // Print logs for testing
  cerr << "This is log, will be printed in stdout file\n";
}
return 0;
```

- 打印调试信息

✓ 使用标准错误输出，调试信息将被保存stderr文件中

- 获取conf配置信息，比如reduce task个数等

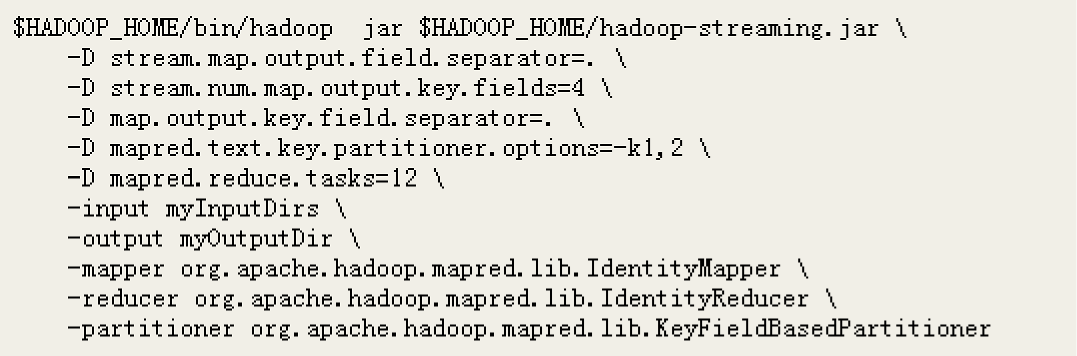
✓ 在环境变量中获取，所有的 “.” 被替换成了 “_”

int main(int argc, char *argv[], char *env[]) {

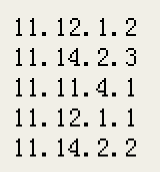
比如：**“mapreduce.job.reduces”**被换成了**"mapreduce_job_reduces"**

Hadoop Streaming高级编程

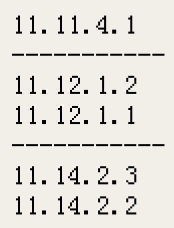
- 对多字段文本数据的支持



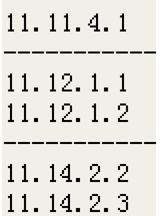
```
$HADOOP_HOME/bin/hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
    -D stream.map.output.field.separator=. \
    -D stream.num.map.output.key.fields=4 \
    -D map.output.key.field.separator=. \
    -D mapred.text.key.partitioner.options=-k1,2 \
    -D mapred.reduce.tasks=12 \
    -input myInputDirs \
    -output myOutputDir \
    -mapper org.apache.hadoop.mapred.lib.IdentityMapper \
    -reducer org.apache.hadoop.mapred.lib.IdentityReducer \
    -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner
```



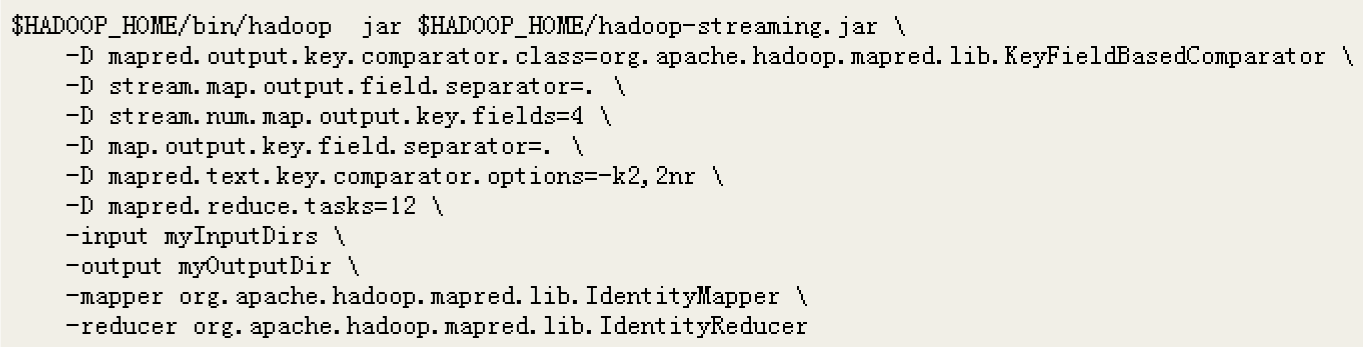
```
11.12.1.2
11.14.2.3
11.11.4.1
11.12.1.1
11.14.2.2
```



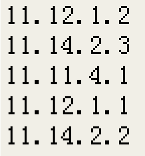
```
11.11.4.1
-----------
11.12.1.2
11.12.1.1
-----------
11.14.2.3
11.14.2.2
```



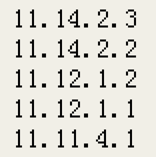
```
11.11.4.1
-----------
11.12.1.1
11.12.1.2
-----------
11.14.2.2
11.14.2.3
```



```
$HADOOP_HOME/bin/hadoop  jar $HADOOP_HOME/hadoop-streaming.jar \
    -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyFieldBasedComparator \
    -D stream.map.output.field.separator=. \
    -D stream.num.map.output.key.fields=4 \
    -D map.output.key.field.separator=. \
    -D mapred.text.key.comparator.options=-k2,2nr \
    -D mapred.reduce.tasks=12 \
    -input myInputDirs \
    -output myOutputDir \
    -mapper org.apache.hadoop.mapred.lib.IdentityMapper \
    -reducer org.apache.hadoop.mapred.lib.IdentityReducer
```



```
11.12.1.2
11.14.2.3
11.11.4.1
11.12.1.1
11.14.2.2
```



```
11.14.2.3
11.14.2.2
11.12.1.2
11.12.1.1
11.11.4.1
```

Java与Streaming编程方式比较

- **Java编程**
  - ✓Hadoop最原始开发语言；
  - ✓支持所有功能，是其他编程方式的基础。
- **Streaming编程**
  - ✓仅用于开发Mapper和Reducer，其他组件需采用Java实现；
  - ✓天生支持文本格式，但二进制格式支持较弱；
  - ✓通常用于简单的文本数据处理，加快开发效率。

总结

- **MapReduce的编程模型**
- **MapReduce编程接口介绍**
- **Java编程**
- **多语言编程**