

# 实验二 倒排索引

171180554 夏宇    171860617 肖维城

## 一、实验设计思路

### Map函数：

**输入：**Key是当前文件分片的某一行数，LongWritable类型；Value是这一行对应内容。

**处理方式：**

对value中的每一个词，将该词和当前文件名合并一起作为输出的Key(Text类型)，并将输出Value设置为1(LongWritable类型)，表示该词在文件中出现一次。



可以看出上面的输出有两个问题：①可能将来自不同文档的同一个单词分配到不同的Reduce，②一个单词在同一个文档中出现了多次，会输出很多次，比如上面fish@doc1输出了两次，不方便reduce统计。

**为了解决问题：**

对问题①：自定义partition函数：

将map输出的key拆分，使用单词作为key进行划分；

对问题②：重写combine函数：

进行初步合并，将同一文档中的同一单词出现次数进行合并然后再包装输出。

### Reduce函数

**输入：**Key是某个文件中的某个词语（格式为：词语@文件名），Text类型；Value是该词语在该文件中出现的次数。

### 处理方式：

在Reduce处理之前，输入已经被很好的处理，Reduce函数的处理目的很清晰：

- 1.得到每个单词出现的总数以及出现的文件次数，求得平均值
- 2.按照规范输出。

所以在Reduce中，需要将某个单词在各个文件中出现的总数汇聚到一起，并且分别记录需要输出的内容，所以，设置记录输出的String类对象，将所需要输出的平均值以及每个文件中该词语出现情况的汇聚在一起，最终得到输出。

## 二、实验代码分析

### Map:

```
public class InvertedIndexMapper extends
Mapper<LongWritable,Text,Text,LongWritable> {
    @Override
    protected void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException
    {
        //取得输入文件的划分
        FileSplit fileSplit = (FileSplit)context.getInputSplit();
        String fileName=fileSplit.getPath().getName();
        //去掉文件名的后缀
        fileName = fileName.substring(0,fileName.lastIndexOf("."));
        fileName = fileName.substring(0,fileName.lastIndexOf("."));

        Text word = new Text();
        StringTokenizer itr = new StringTokenizer(value.toString());
        for(;itr.hasMoreTokens();){
            //word即为：词语@文件名
            word.set(itr.nextToken()+"@"+fileName);
            //Context：收集Mapper输出的<k,v>对。
            context.write(word,new LongWritable(1));
        }
    }
}
```

### Partitioner:

```
public class InvertedIndexPartitioner extends HashPartitioner<Text,
LongWritable> {
    @Override
    public int getPartition(Text key, LongWritable value, int numReduceTasks) {
        String term = key.toString().split("@")[0];
        //使用内置的Partition进行划分
        return super.getPartition(new Text(term),value,numReduceTasks);
    }
}
```

### Combiner:

```
public class InvertedIndexCombiner extends Reducer<Text,
LongWritable,Text,LongWritable>{
```

```

@Override
protected void reduce(Text key, Iterable<LongWritable> values, Context
context)
    throws IOException, InterruptedException
{
    long sum = 0;
    Iterator<LongWritable> it = values.iterator();
    //将一个文件内部相同的单词求和累积起来
    for(;it.hasNext();){
        sum+=it.next().get();
    }
    context.write(key,new LongWritable(sum));
}
}

```

## Reducer:

```

public class InvertedIndexReducer extends Reducer<Text, LongWritable,Text,Text>
{
    private String t_prev;//当前词语名
    private int showFile;//当前词语出现的文件计数
    private long showTime;//当前词语计数
    private Text postingList;
    private String outputList;
    @Override
    protected void setup(Context context)
        throws IOException, InterruptedException {
        t_prev = "";
        showFile =0;
        showTime =0;
        postingList = new Text();
    }

    @Override
    protected void reduce(Text key, Iterable<LongWritable> values, Context
context)
        throws IOException, InterruptedException {
        String keys[] = key.toString().split("@");
        String term = keys[0];
        //检查当前单词名与下一个读入的单词名，如果相同且非空，则进行输出操作，否则继续执行
        if(!term.equals(t_prev)) && t_prev!=""){
            // 计算平均值并配置输出
            double ave = (double)showTime/showFile;
            String tmp = String.format("%.2f", ave) + ",";
            outputList = tmp + outputList;
            postingList.set(outputList);
            //通过context输出结果
            context.write(new Text(t_prev),postingList);
            postingList = new Text();
            outputList = "";
            showTime =0;
            showFile =0;
        }
        long cnt = 0;
        Iterator<LongWritable> it = values.iterator();
        //累计得到一个文件中莫个词语的总值值
    }
}

```

```
while(it.hasNext()){
    cnt+=it.next().get();
}
//累加多个文件中某个词语出现的次数以及出现该词语的文件总数
showTime += cnt;
showFile += 1;
//记录单个文件中某个词语的情况到输出序列中
String tmp = keys[1] + ":" + cnt + ",";
outputList =outputList +tmp;
t_prev = term;
}

@Override
protected void cleanup(Context context) throws IOException,
InterruptedException {
    double ave = (double)showTime/showFile;
    String tmp = String.format("%.2f", ave) + ",";
    outputList = tmp + outputList;
    postingList.set(outputList);
    context.write(new Text(t_prev),postingList);
}
}
```

## 三、输出结果文件

输出结果文件存放在大数据实验平台的HDFS系统中，其路径为 /user/2020st05/output。

节选部分结果如下：

```

估计    1.63.卧龙生30.素手劫:3;卧龙生39.铁苗神剑:1;古龙13.楚留香05蝙蝠传奇:1;古龙20.多情剑客无情剑:2;古龙53.七种武器03孔雀翎:1;古龙63.武林外史:1;李
凉04.本尊分身:7;李凉18.酒赌小浪子:2;李凉25.魔手邪怪:2;梁羽生01.白发魔女传:1;梁羽生03.冰河洗剑录:2;梁羽生15.慧剑心魔:1;梁羽生19.狂侠天娇魔女:1;梁羽生
20.联剑风云录:1;梁羽生30.散花女侠:1;梁羽生37.游剑江湖:1;金庸02雪山飞狐:1;金庸09书剑恩仇录:1;金庸12倚天屠龙记:1;
定计划    1.00.梁羽生18.绝塞传烽录:1;
定论    1.63.卧龙生02.春秋笔:2;卧龙生03.翠袖玉环:1;卧龙生06.飞铃:1;卧龙生09.风雨燕归来:1;卧龙生11.黑白双娇:4;卧龙生12.剑气洞彻九重天:1;卧龙生13.剑无
痕:1;卧龙生15.蜂雪玄箱:2;卧龙生16.金笔点龙记:1;卧龙生17.金凤剪:1;卧龙生18.金剑雕翎:2;卧龙生19.惊鸿一剑震江湖:1;卧龙生20.梦幻之刀:1;卧龙生21.妙绝天香
:1;卧龙生22.女捕头:1;卧龙生23.飘花令:1;卧龙生24.七绝剑:3;卧龙生25.七绝剑III还情剑:1;卧龙生29.双凤旗:5;卧龙生30.素手劫:1;卧龙生31.桃花血令:2;卧龙生32.
天鹤语:1;卧龙生33.天剑绝刀:1;卧龙生34.天龙甲:2;卧龙生36.天香飘:2;卧龙生38.铁剑玉佩:2;卧龙生39.铁苗神剑:2;卧龙生41.无形剑:1;卧龙生43.血剑丹心:2;卧龙
生44.烟锁江湖:4;卧龙生45.燕子传奇:2;卧龙生46.摇花放鹰传:1;卧龙生48.银月飞霜:2;卧龙生52.哀紫烟:2;古龙16.楚留香08午夜兰花:1;古龙18.大旗英雄传:1;古龙24
.护花铃:1;古龙32.剑玄录:1;古龙36.猎鹰.赌局:1;古龙58.情人箭:1;李凉01.暗器高手:1;李凉04.本尊分身:3;李凉08.公孙小刀:2;李凉09.红顶记:3;李凉17.惊神关小刀
:1;李凉22.矛盾天师:2;李凉25.魔手邪怪:1;李凉26.奇神杨小邪续集:2;李凉27.奇神杨小邪:4;李凉33.天开大帝:1;李凉34.天下第一当:1;李凉35.武林雄游记:1;李凉36.
小鬼大赢家:1;李凉37.小鱼吃大鱼:1;李凉38.笑笑江湖:1;李凉39.新蜀山剑侠传:1;李凉41.杨小邪发威:2;梁羽生10.风云雷雨:1;梁羽生14.幻剑灵樨:3;梁羽生16.剑网尘
丝:1;梁羽生31.随笔集:1;毛不花:4;梁羽生37.游剑江湖:1;金庸07鹿鼎记:3;金庸09书剑恩仇录:1;金庸10神雕侠侣:1;金庸12倚天屠龙记:1;金庸13碧血剑:1;
定评    1.00.卧龙生51.玉手点将录:1;古龙26.浣花洗剑录:1;梁羽生27.萍踪侠影录:1;梁羽生32.随笔集:1;毛花六照:1;金庸08笑傲江湖:1;
定说    1.41.卧龙生04.地獄门:1;卧龙生06.飞铃:1;卧龙生11.黑白双娇:1;卧龙生12.剑气洞彻九重天:1;卧龙生21.妙绝天香:3;卧龙生36.天香飘:1;卧龙生38.铁剑玉佩
:3;卧龙生44.烟锁江湖:1;卧龙生54.指剑为媒:2;古龙01.白玉老虎:1;古龙24.护花铃:1;古龙32.剑玄录:2;古龙33.九月鹰飞:1;古龙47.怒剑狂花:1;李凉01.暗器高手:2;
李凉09.公孙小刀:1;李凉11.会醉才会赢:1;李凉16.江湖一品郎:1;李凉17.惊神关小刀:2;李凉21.六宝江湖行:1;李凉23.妙贼丁小勾:1;李凉24.妙贼丁小勾续集:1;李凉27
.奇神杨小邪:1;李凉30.淘气世家:4;李凉31.天才混流:1;李凉34.天下第一当:1;李凉35.武林雄游记:1;梁羽生06.大唐游侠传:2;梁羽生34.武当一剑:1;金庸04天龙八部:2;
金庸07鹿鼎记:2;金庸09书剑恩仇录:1;金庸12倚天屠龙记:1;金庸13碧血剑:1;
定词    1.00.李凉04.本尊分身:1;李凉06.超级邪侠:1;李凉38.笑笑江湖:1;
定调    1.00.李凉04.本尊分身:1;李凉06.超级邪侠:1;李凉38.笑笑江湖:1;
定韵    1.00.古龙63.武林外史:1;梁羽生24.鸣镭风云录:1;
定身法    1.36.卧龙生12.剑气洞彻九重天:1;卧龙生31.飘花血令:1;李凉14.江湖双响炮:1;李凉15.江湖一担皮:1;李凉40.新蜀山剑侠传:2;梁羽生03.冰河洗剑录:2;梁
羽生06.大唐游侠传:1;梁羽生07.弹指惊雷:2;梁羽生10.风云雷雨:1;梁羽生11.广陵剑:1;梁羽生12.瀚海雄风:1;梁羽生16.剑网尘丝:1;梁羽生18.绝塞传烽录:2;梁羽生19
.狂侠天娇魔女:1;梁羽生22.龙凤宝钗缘:1;梁羽生24.鸣镭风云录:2;梁羽生25.牧野流星:1;梁羽生27.萍踪侠影录:2;梁羽生29.塞外奇侠传:1;梁羽生34.武当一剑:2;梁羽
生36.侠骨丹心:2;梁羽生37.游剑江湖:1;
定地点    1.00.李凉04.本尊分身:1;李凉06.超级邪侠:1;李凉38.笑笑江湖:1;
它对    1.35.卧龙生04.地獄门:1;卧龙生06.飞铃:1;卧龙生11.黑白双娇:1;卧龙生34.天龙甲:1;卧龙生46.摇花放鹰传:1;卧龙生47.一代天骄:1;卧龙生48.银月飞霜:1;
古龙27.血鸢:1;古龙42.陆小凤05幽灵山庄:1;李凉05.超霸的男人:3;李凉11.会醉才会赢:2;李凉15.江湖一担皮:2;李凉35.武林雄游记:1;李凉41.杨小邪发威:1;梁羽生
03.冰河洗剑录:1;梁羽生06.大唐游侠传:2;梁羽生23.龙虎斗京华:1;梁羽生31.随笔集:1;毛不花:1;梁羽生32.随笔集:1;毛花六照:3;梁羽生35.武林天骄:1;
它将    1.32.古龙04.边城刀声:1;古龙14.楚留香06桃花传奇:1;古龙32.剑玄录:1;古龙35.绝代双骄:1;古龙36.猎鹰.赌局:1;古龙44.陆小凤07剑神一笑:1;古龙46.那一
剑的风情:1;李凉01.暗器高手:1;李凉04.本尊分身:3;李凉11.会醉才会赢:2;李凉13.江湖急数传:1;李凉19.酒狂任小赌:1;李凉20.狂侠天娇魔女:1;李凉23.妙贼丁小勾:1;
李凉28.忍者龟:1;李凉36.小鬼大赢家:1;李凉39.新蜀山剑侠传:1;李凉41.杨小邪发威:1;梁羽生27.萍踪侠影录:1;
它就    1.63.卧龙生02.春秋笔:1;卧龙生04.地獄门:1;卧龙生05.花逐月:1;卧龙生06.飞铃:1;卧龙生11.黑白双娇:4;卧龙生17.金凤剪:1;卧龙生21.妙绝天香:1;卧龙
生22.女捕头:4;卧龙生40.无名箫:1;卧龙生44.烟锁江湖:1;卧龙生46.摇花放鹰传:2;卧龙生50.玉钗盟:3;古龙05.边城浪子:1;古龙09.楚留香04血海飘香:1;古龙12.楚留香
04归人还魂(鬼恋传奇):1;古龙13.楚留香05蝙蝠传奇:2;古龙17.大地飞鹰:2;古龙20.多情剑客无情剑:1;古龙25.欢乐英雄:1;古龙26.浣花洗剑录:1;古龙27.血鸢:1;
古龙42.陆小凤05幽灵山庄:1;古龙45.名剑风流:1;古龙46.那一剑的风情:1;古龙47.怒剑狂花:2;古龙51.七种武器01长生剑:1;古龙56.七种武器06离别钩:1;古龙66.火并
萧十一郎:1;古龙67.英雄无泪:1;古龙69.圆月弯刀:2;李凉01.暗器高手:1;李凉06.超级邪侠:2;李凉08.公孙小刀:4;李凉11.会醉才会赢:2;李凉15.江湖一担皮:3;李凉16.
江湖一品郎:4;李凉19.酒狂任小赌:1;李凉22.矛盾天师:1;李凉25.魔手邪怪:1;李凉26.奇神杨小邪续集:5;李凉27.奇神杨小邪:2;李凉29.神偷小千:1;李凉30.淘气世家:5
李凉31.天才混流:1;李凉34.天下第一当:2;李凉36.小鬼大赢家:3;李凉38.笑笑江湖:1;李凉41.杨小邪发威:1;梁羽生02.冰川天女传:3;梁羽生06.大唐游侠传:1;梁羽生0
9.风雷震九州:1;梁羽生12.瀚海雄风:1;梁羽生17.江湖三女侠:2;梁羽生19.狂侠天娇魔女:1;梁羽生21.梁羽生传奇:2;梁羽生22.龙凤宝钗缘:1;梁羽生23.龙虎斗京华:1;
梁羽生26.女帝奇英传:1;梁羽生27.萍踪侠影录:1;梁羽生37.游剑江湖:1;金庸10神雕侠侣:1;金庸13碧血剑:1;
它想    1.08.卧龙生03.翠袖玉环:1;卧龙生06.飞铃:1;卧龙生10.黑白双:1;卧龙生19.惊鸿一剑震江湖:1;古龙10.楚留香02大沙漠:1;古龙58.情人箭:1;李凉03.百败小赢
家:1;李凉04.本尊分身:1;李凉27.奇神杨小邪:1;李凉30.淘气世家:1;李凉34.天下第一当:2;李凉38.笑笑江湖:1;
它意    1.50.卧龙生20.梦幻之刀:2;卧龙生51.玉手点将录:1;
它所    1.22.卧龙生04.地獄门:3;卧龙生31.桃花血令:1;卧龙生39.铁苗神剑:1;卧龙生44.烟锁江湖:1;古龙04.边城刀声:1;古龙17.大地飞鹰:1;古龙26.浣花洗剑录:1;
古龙46.那一剑的风情:1;古龙47.怒剑狂花:1;古龙51.七种武器01长生剑:1;古龙55.七种武器05霸王枪:1;古龙68.游侠录:1;李凉03.百败小赢家:2;李凉05.超霸的男人:1;
李凉11.会醉才会赢:1;梁羽生27.萍踪侠影录:1;李凉28.忍者龟:1;李凉31.天才混流:3;李凉33.天开大帝:2;李凉36.小鬼大赢家:1;李凉39.新蜀山剑侠传:1;梁羽生02.
冰川天女传:1;梁羽生03.冰河洗剑录:1;梁羽生09.风雷震九州:1;梁羽生13.还剑奇情录:1;梁羽生36.侠骨丹心:1;梁羽生38.云海玉弓缘:1;
它把    1.11.卧龙生07.飞燕惊龙:1;卧龙生42.新仙鹤神针:1;卧龙生44.烟锁江湖:1;李凉04.本尊分身:1;李凉05.超霸的男人:2;李凉13.江湖急救站:1;李凉24.妙贼丁小
勾续集:1;李凉26.奇神杨小邪续集:1;李凉27.奇神杨小邪:1;
它方    1.00.李凉04.本尊分身:1;李凉17.惊神关小刀:1;
```

## 四、“江湖”及“风雪”输出结果

## 江湖

江湖 116.06,卧龙生01.镖旗:275;卧龙生02.春秋笔:329;卧龙生03.翠袖玉环:402;卧龙生04.地狱门:105;卧龙生05.飞花逐月:298;卧龙生06.飞铃:244;卧龙生07.飞燕惊龙:269;卧龙生08.风尘侠隐:228;卧龙生09.风雨燕归来:198;卧龙生10.黑白剑:223;卧龙生11.黑白双娇:117;卧龙生12.剑气洞彻九重天:299;卧龙生13.剑无痕:261;卧龙生14.剑仙列传:25;卧龙生15.绛雪玄霜:274;卧龙生16.金笔点龙记:318;卧龙生17.金凤剪:326;卧龙生18.金剑雕翎:397;卧龙生19.惊鸿一剑震江湖:346;卧龙生20.梦幻之刀:106;卧龙生21.妙绝天香:28;卧龙生22.女捕头:317;卧龙生23.飘花令:263;卧龙生24.七绝剑:130;卧龙生25.七绝剑II还情剑:144;卧龙生26.情剑无刀:7;卧龙生27.琼楼十二曲:181;卧龙生28.神州豪侠:261;卧龙生29.双凤旗:261;卧龙生30.素手劫:374;卧龙生31.桃花血令:131;卧龙生32.天鹤谱:106;卧龙生33.天剑绝刀:321;卧龙生34.天龙甲:293;卧龙生35.天马霜衣:278;卧龙生36.天香飘:246;卧龙生37.天涯情侣:113;卧龙生38.铁剑玉佩:71;卧龙生39.铁苗神剑:358;卧龙生40.无名箫:136;卧龙生41.无形剑:256;卧龙生42.新仙鹤神针:188;卧龙生43.血剑丹心:350;卧龙生44.烟锁江湖:238;卧龙生45.燕子传奇:132;卧龙生46.摇花放鹰传:446;卧龙生47.一代天骄:353;卧龙生48.银月飞霜:132;卧龙生49.幽灵四艳:221;卧龙生50.玉钗盟:320;卧龙生51.玉手点将录:150;卧龙生52.袁紫烟:30;卧龙生53.岳小钗:308;卧龙生54.指剑为媒:57;古龙01.白玉老虎:111;古龙02.白玉雕龙:24;古龙03.碧血洗银枪:81;古龙04.边城刀声:50;古龙05.边城浪子:62;古龙06.彩环曲:76;古龙07.残金缺玉:82;古龙08.苍穹神剑:131;古龙09.楚留香01血海飘香:47;古龙10.楚留香02大沙漠:23;古龙11.楚留香03画眉鸟:96;古龙12.楚留香04借尸还魂(鬼恋传奇):41;古龙13.楚留香05蝙蝠传奇:80;古龙14.楚留香06桃花传奇:8;古龙15.楚留香07新月传奇:20;古龙16.楚留香08午夜兰花:68;古龙17.大地飞鹰:56;古龙18.大旗英雄传:151;古龙19.大人物:37;古龙20.多情剑客无情剑:150;古龙21.飞刀,又见飞刀:30;古龙22.风铃中的刀声:76;古龙23.孤星传:175;古龙24.护花铃:193;古龙25.欢乐英雄:64;古龙26.浣花洗剑录:285;古龙27.血鸢:11;古龙28.剑·花·烟雨·江南:18;古龙29.剑毒梅香:110;古龙30.剑客行:187;古龙31.剑气书香:27;古龙32.剑玄录:160;古龙33.九月鹰飞:50;古龙34.绝不低头:3;古龙35.绝代双骄:249;古龙36.猎鹰·赌局:107;古龙37.流星·蝴蝶·剑:34;古龙38.陆小凤01陆小凤前传(金鹏王朝):15;古龙39.陆小凤02绣花大盗:23;古龙40.陆小凤03决战前后:26;古龙41.陆小凤04银钩赌坊:10;古龙42.陆小凤05幽灵山庄:40;古龙43.陆小凤06凤舞九天:42;古龙44.陆小凤07剑神一笑:47;古龙45.名剑风流:201;古龙46.那一剑的风情:32;古龙47.怒剑狂花:102;古龙48.飘香剑雨:67;古龙49.七杀手:20;古龙50.七星龙王:55;古龙51.七种武器01长生剑:23;古龙52.七种武器02碧玉刀:22;古龙53.七种武器03孔雀翎:10;古龙54.七种武器04多情环:8;古龙55.七种武器05霸王枪:54;古龙56.七种武器06离别钩:39;古龙57.七种武器07拳头:16;古龙58.情人箭:197;古龙59.三少爷的剑:86;古龙60.神君别传:12;古龙61.失魂引:96;古龙62.天涯·明月·刀:69;古龙63.武林外史:179;古龙64.湘妃剑:169;古龙65.萧十一郎:56;古龙66.火并萧十一郎:89;古龙67.英雄无泪:54;古龙68.游侠录:103;古龙69.圆月弯刀:155;古龙70.月异星邪:102;李凉01.暗器高手:53;李凉02.霸枪艳血:34;李凉03.百败小赢家:79;李凉04.本尊分身:97;李凉05.超霸的男人:7;李凉06.超级邪侠:209;李凉07.赌棍小狂侠:31;李凉08.公孙小刀:69;李凉09.红顶记:48;李凉10.滑头傻小子:43;李凉11.会醉才会赢:189;李凉12.活宝小淘气:121;李凉13.江湖急救站:118;李凉14.江湖双响炮:291;李凉15.江湖一担皮:197;李凉16.江湖一品郎:186;李凉17.惊神关小刀:38;李凉18.酒赌小浪子:17;李凉19.酒狂任小赌:167;李凉20.狂侠南宫鹰:15;李凉21.六宝江湖行:72;李凉22.矛盾天师:59;李凉23.妙贼丁小勾:16;李凉24.妙贼丁小勾续集:13;李凉25.魔手邪怪:6;李凉26.奇神扬小邪续集:98;李凉27.奇神杨小邪:102;李凉28.忍者龟:77;李凉29.神偷小干:17;李凉30.淘气世家:81;李凉31.天才混混:89;李凉32.天才混混外集:27;李凉33.天齐大帝:62;李凉34.天下第一当:52;李凉35.武林嘻游记:180;李凉36.小鬼大赢家:29;李凉37.小鱼吃大鱼:42;李凉38.笑笑江湖:69;李凉39.新蜀山剑侠传:20;李凉40.新蜀山剑侠传续:1;李凉41.杨小邪发威:34;梁羽生01.白发魔女传:63;梁羽生02.冰川天女传:65;梁羽生03.冰河洗剑录:65;梁羽生04.冰魄寒光剑:4;梁羽生05.草莽龙蛇传:143;梁羽生06.大唐游侠传:97;梁羽生07.弹指惊雷:77;梁羽生08.飞凤潜龙:4;梁羽生09.风雷震九州:135;梁羽生10.风云雷电:139;梁羽生11.广陵剑:149;梁羽生12.瀚海雄风:69;梁羽生13.还剑奇情录:8;梁羽生14.幻剑灵旗:27;梁羽生15.慧剑心魔:102;梁羽生16.剑网尘丝:127;梁羽生17.江湖三女侠:252;梁羽生18.绝塞传烽录:40;梁羽生19.狂侠天娇魔女:180;梁羽生20.联剑风云录:126;梁羽生21.梁羽生传奇:74;梁羽生22.龙凤宝钗缘:109;梁羽生23.龙虎斗京华:182;梁羽生24.鸣镝风云录:134;梁羽生25.牧野流星:35;梁羽生26.女帝奇英传:49;梁羽生27.萍踪侠影录:40;梁羽生28.七剑下天山:77;梁羽生29.塞外奇侠传:2;梁羽生30.散花女侠:48;梁羽生31.随笔集:笔不花:4;梁羽生32.随笔集:笔花六照:15;梁羽生34.武当一剑:94;梁羽生35.武林天骄:45;梁羽生36.侠骨丹心:103;梁羽生37.游剑江湖:275;梁羽生38.云海玉弓缘:81;金庸01.飞狐外传:67;金庸02.雪山飞狐:29;金庸03.连城诀:35;金庸04.天龙八部:126;金庸05



射雕英雄传:57;金庸06白马啸西风:5;金庸07鹿鼎记:75;金庸08笑傲江湖:199;金庸09书剑恩仇录:64;金庸10神雕侠侣:81;金庸11侠客行:61;金庸12倚天屠龙记:145;金庸13碧血剑:65;金庸14鸳鸯刀:35;

风雪

风雪 4.53,卧龙生01.镖旗:3;卧龙生07.飞燕惊龙:16;卧龙生08.风尘侠隐:1;卧龙生09.风雨燕归来:1;卧龙生12.剑气洞彻九重天:36;卧龙生15.绛雪玄霜:4;卧龙生18.金剑雕翎:9;卧龙生19.惊鸿一剑震江湖:2;卧龙生22.女捕头:4;卧龙生25.七绝剑II还情剑:2;卧龙生27.琼楼十二曲:1;卧龙生31.桃花血令:2;卧龙生36.天香飘:2;卧龙生38.铁剑玉佩:7;卧龙生39.铁苗神剑:7;卧龙生42.新仙鹤神针:15;卧龙生47.一代天骄:3;卧龙生49.幽灵四艳:1;卧龙生52.袁紫烟:6;古龙07.残金缺玉:7;古龙13.楚留香05蝙蝠传奇:1;古龙14.楚留香06桃花传奇:1;古龙20.多情剑客无情剑:4;古龙23.孤星传:2;古龙24.护花铃:1;古龙25.欢乐英雄:3;古龙29.剑毒梅香:2;古龙31.剑气书香:3;古龙42.陆小凤05幽灵山庄:1;古龙46.那一剑的风情:3;古龙47.怒剑狂花:2;古龙56.七种武器06离别钩:1;古龙61.失魂引:14;古龙63.武林外史:24;古龙64.湘妃剑:5;古龙67.英雄无泪:8;李凉01.暗器高手:1;李凉02.霸枪艳血:5;李凉03.百败小赢家:1;李凉04.本尊分身:6;李凉09.红顶记:1;李凉11.会醉才会赢:1;李凉14.江湖双响炮:4;李凉16.江湖一品郎:1;李凉19.酒狂任小赌:32;李凉31.天才混泥:4;李凉34.天下第一当:1;李凉36.小鬼大赢家:1;李凉39.新蜀山剑侠传:1;李凉40.新蜀山剑侠传续:12;梁羽生01.白发魔女传:4;梁羽生02.冰川天女传:5;梁羽生03.冰河洗剑录:1;梁羽生07.弹指惊雷:1;梁羽生09.风雷震九州:1;梁羽生10.风云雷电:3;梁羽生12.瀚海雄风:1;梁羽生14.幻剑灵旗:1;梁羽生20.联剑风云录:1;梁羽生21.梁羽生传奇:1;梁羽生22.龙凤宝钗缘:5;梁羽生25.牧野流星:5;梁羽生26.女帝奇英传:3;梁羽生27.萍踪侠影录:3;梁羽生30.散花女侠:1;梁羽生37.游剑江湖:1;梁羽生38.云海玉弓缘:1;金庸01飞狐外传:3;金庸03连城诀:1;金庸04天龙八部:1;金庸05射雕英雄传:7;金庸06白马啸西风:11;金庸07鹿鼎记:1;金庸10神雕侠侣:2;金庸12倚天屠龙记:1;


五、WebUI报告

登录大数据实验平台并得到如下结果：

运行记录：

2020.04.21 21:21:08 CST	2020.04.21 21:21:12 CST	2020.04.21 21:23:29 CST	job_1572597966684_3201	Inverted Index	2020st05	root.team05	SUCCEEDED	218	218	1	1
----------------------------	----------------------------	----------------------------	------------------------	----------------	----------	-------------	-----------	-----	-----	---	---

具体运行结果：



### MapReduce Job job\_1572597966684\_3201

Logged in as: dr:who

Application

Job

Overview

Counters

Configuration

Map Tasks

Reduce Tasks

Tools

Job Name: Inverted Index

User Name: 2020st05

Queue: root.team05

State: SUCCEEDED

Uberized: false

Submitted: Tue Apr 21 21:21:08 CST 2020

Started: Tue Apr 21 21:21:12 CST 2020

Finished: Tue Apr 21 21:23:29 CST 2020

Elapsed: 2mins, 16sec

Diagnostics:

Average Map Time: 3sec

Average Shuffle Time: 1mins, 40sec

Average Merge Time: 2sec

Average Reduce Time: 9sec

ApplicationMaster	Attempt Number	Start Time	Node	Logs
1		Tue Apr 21 21:21:09 CST 2020	slave011-8042	logs

Task Type	218	Total	218	Complete
Map	1		1	
Reduce				
Attempt Type	Failed	Killed	Successful	
Maps	1	0	218	
Reduces	0	0	1	

六、实验分工说明

夏宇同学负责求平均数功能的编写，Map、Reduce功能的编写、修改，编译jar包并在实验平台上测试并完成实验报告。

肖维城同学负责项目构建、整体代码编写，包括Map、Reduce、Partion、Combine等功能。

七、总结分析

本次实验实现了课堂上的倒排索引算法，是在WordCount程序后编写的第二个MapReduce程序，所需要处理的问题较WordCount更为复杂一些，但是总体还是比较简单。通过编写程序，最主要的收获还是对MapReduce程序的构建有了更加深入的了解。

在对Map、Reduce、Partition、Combine等功能的应用中，了解到如何避免错误、提高处理效率，以及对之前理论课上hadoop整体的工作框架以及工作流程了解更深。而且在编程过程中，更加深刻地了解到MapReduce这种不太需要关注底层细节的编程方式的便捷性。