

# MapReduce 实验三报告

账号: 2017st20

组长: 严德美 MF1733071

组员: 周天烁 MG1733099

李 论 MG1733027

裴俊宇 MG1633056

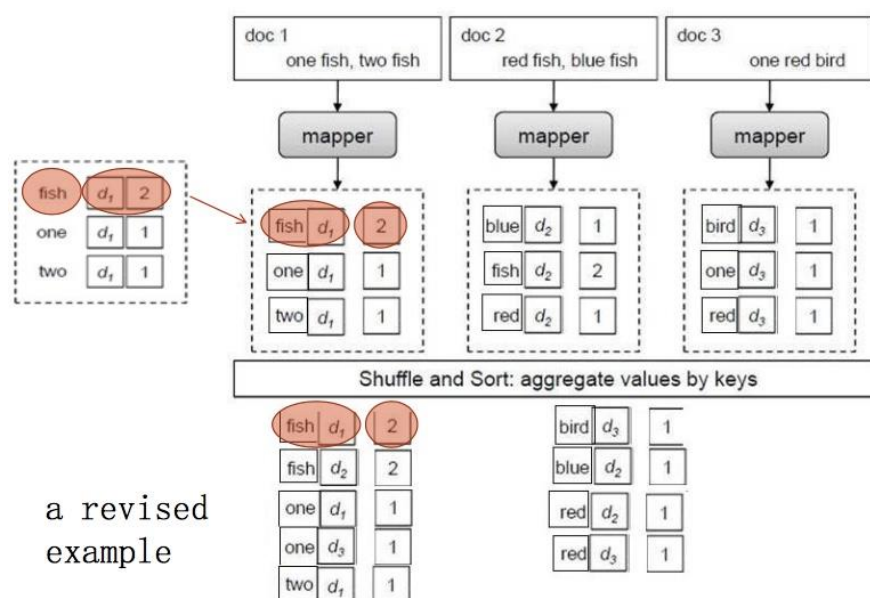
## 一、 实验内容

实现带词频属性的文档倒排算法。在统计词语的倒排索引时，除了要输出带词频属性的倒排索引，还需要计算每个词语的平均出现次数并输出。其中，每个词语的平均出现次数 = 词语在全部文档中出现的频数总和 / 包含该词语的文档数。对于每个词语，输出一个键值对，该键值对的格式如下：

词语\TAB 平均出现次数,小说 1:词频;小说 2:词频;...;小说 N:词频;

## 二、 设计思路

为了保证能够有足够的 memory 来存储每个 term 的所有 postings，我们采用了可扩展的带词频属性的文档倒排算法，将<term, docid>一起作为组合键 key 来进行统计：

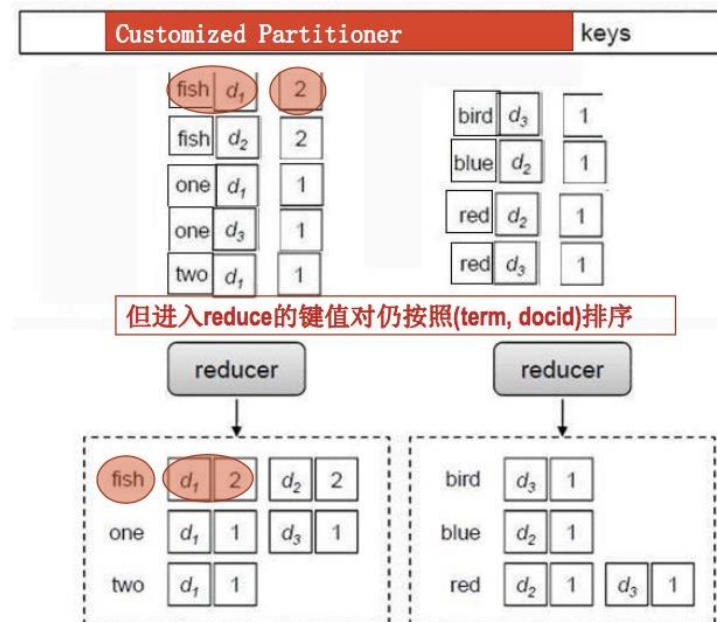


Map 阶段:

程序逐个读取文档中的词进行词频统计，得到一系列的键值对 Key-Value。其中，Key 为<term, docid>组成的 Text，term 和 docid 用逗号分隔开，Value 为前面统计出的在 docid 中 term 出现的频数 tf，类型为 IntWritable。

下一步中，当对键值对进行 shuffle 处理以传送给合适的 Reducer 时，将按照新的键<term, docid>进行排序和选择 Reducer，这可能会导致同一个 term 的键值对被分区到不同的 Reducer 中。为了解决这个问题，需要定制 Partitioner。

其基本思想就是，把组合键<term, docid>临时拆开，让 partitioner 按照<term>而不是<term, docid>进行分区选择正确的 Reducer，这样可保证同一个 term 下的一组键值对一定被分区到同一个 Reducer 中。



### Reduce 阶段:

前面 Map 得到的键值对 (tuple<term, docid> tf) 作为 Reducer 的输入，将所有 term 相同的键值对合并起来，使得每个 term 对应着一个 postings list，其中每一个 posting 是 docid 和对应的词频 tf。所以，在 Map 阶段，Key 为一个词 term，Value 为这个 term 的 postings list。

最后，还需要计算每个词语的平均出现次数，用公式：平均出现次数 = 词语在全部文档中出现的频数总和 / 包含该词语的文档数，计算并输出。

## 三、 程序代码

### Map 代码:

```

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileSplit;
import java.io.IOException;
import java.util.StringTokenizer;

/**
 * Created by dmyan on 17-10-26.
 */
public class InvertedIndexMapper extends Mapper<LongWritable,Text,Text,IntWritable>{
    private Text word = new Text();
    private final IntWritable one = new IntWritable(1);
    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException,
    InterruptedException {

        FileSplit fileSplit = (FileSplit)context.getInputSplit();
        String fileName = fileSplit.getPath().getName();//获取文件名
        if(fileName.contains(".TXT.segmented"))
            fileName = fileName.replace(".TXT.segmented","");//去掉文件后缀.TXT.segmented
        if(fileName.contains(".txt.segmented"))
            fileName = fileName.replace(".txt.segmented","");//去掉文件后缀.txt.segmented
        StringTokenizer token = new StringTokenizer(value.toString());//字符拆解成单词
        while(token.hasMoreTokens()){
            word.set(token.nextToken()+","+fileName);//拼接字符和文件名<term,docid>
            context.write(word, one);
        }
    }
}

```

**Reduce 代码:**

```

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import java.io.IOException;
import java.util.*;

/**
 * Created by dmyan on 17-10-26.
 */
public class InvertedIndexReducer extends Reducer<Text, IntWritable, Text, Text> {
    private static List<String> postingsList = new ArrayList<String>();
    private static Text item = new Text(" ");
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values,
                          Context context) throws IOException, InterruptedException {
        Text wordpre;//当前词
        Text word = new Text();//保存文件名和对应的词频数
        int sum = 0;
        wordpre = new Text(key.toString().split(",")[0]);
        String fileName = key.toString().split(",")[1];
        for(IntWritable value : values){//求词频总数
            sum += value.get();
        }
        word.set(fileName+"."+sum);//文件名和总数
        if(!item.equals(wordpre)&&!item.equals(" ")){//遇到不同 key 要输出之前 key 的内容
            long frens = 0;//总词频数
            double fileCount = 0.0;//总文件数
            StringBuilder all = new StringBuilder();
            Iterator<String> iter = postingsList.iterator(); //迭代 postingsList 里面的内容
            String str ;
            if(iter.hasNext()){//遍历 postingsList 取出当前 key 对应的结果
                str = iter.next();
                all.append(str);
                fileCount++;
                frens += Long.parseLong(str.substring(str.indexOf(":")+1));
            }
            while(iter.hasNext()){
                str = iter.next();
                all.append(str);
                fileCount++;
                frens += Long.parseLong(str.substring(str.indexOf(":")+1));
                all.append(str);
            }
            if(frens>0){
                context.write(item,new Text(String.format("%.2f",frens/fileCount)+","+all.toString()))
            }
        }
    }
}

```

```

    }
    postingsList = new ArrayList<String>();//清空 postingsList
}
item.set(wordpre);//设置当前 term
postingsList.add(word.toString());//把当前文件名和词频数添加到 postingsList
}
@Override
public void cleanup(Context context) throws IOException, InterruptedException { //输出
    最后一次未得到输出的<t,p>
    long frens = 0;
    double fileCount = 0.0;
    StringBuilder all = new StringBuilder();
    Iterator<String> iter = postingsList.iterator();
    String str ;
    if(iter.hasNext()){
        str = iter.next();
        all.append(str);
        fileCount++;
        frens += Long.parseLong(str.substring(str.indexOf(":")+1));
    }
    while(iter.hasNext()){
        str = iter.next();
        all.append(";");
        fileCount++;
        frens += Long.parseLong(str.substring(str.indexOf(":")+1));
        all.append(str);
    }
    if(frens>0){
        context.write(item,new
Text(String.format("%.2f",frens/fileCount)+",""+all.toString()));
    }
}
}
}

```

## 四、 运行结果

输出结果（HDFS 上的 hdfsOutput 目录下）

```
2017st20@master01:~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
hadoop@dmmyan-Ubuntu:~$ ssh 2017st20@114.212.190.91  
2017st20@114.212.190.91's password:  
Last login: Sun Oct 29 12:26:04 2017 from 114.212.84.48  
[2017st20@master01 ~]$ hadoop fs -ls hdfsOutput  
17/10/29 15:26:35 WARN util.NativeCodeLoader: Unable to load native-hadoop libra  
ry for your platform... using builtin-java classes where applicable  
Found 2 items  
-rw-r--r-- 3 2017st20 hadoop_user 0 2017-10-28 17:26 hdfsOutput/_SUCC  
ESS  
-rw-r--r-- 3 2017st20 hadoop_user 109270109 2017-10-28 17:26 hdfsOutput/part-  
r-00000  
[2017st20@master01 ~]$
```

查看运行结果:

`hadoop fs -cat hdfsOutput/part-r-00000`

```
2017st20@master01:~  
鱼肚白 2.00,卧龙生23.飘花令:2;卧龙生31.桃花血令:2;卧龙生43.血剑丹心:4;古龙02.白  
玉雕龙:1;古龙12.楚留香04借尸还魂(鬼恋传奇):1;古龙30.剑客行:1;古龙36.猎鹰·赌局:2;  
古龙46.那一剑的风情:1;古龙47.怒剑狂花:1;古龙48.飘香剑雨:1;古龙58.情人箭:1;李凉02  
.霸枪艳血:1;李凉37.小鱼吃大鱼:2;梁羽生07.弹指惊雷:3;梁羽生09.风雷震九洲:4;梁羽生  
10.风云雷电:2;梁羽生12.瀚海雄风:2;梁羽生14.幻剑灵旗:1;梁羽生15.慧剑心魔:2;梁羽生  
16.剑网尘丝:1;梁羽生19.狂侠天娇魔女:4;梁羽生24.鸣镝风云录:5;梁羽生30.散花女侠:1;  
梁羽生37.游剑江湖:4;金庸07鹿鼎记:1  
鱼脍 2.00,古龙12.楚留香04借尸还魂(鬼恋传奇):2  
鱼腥 1.00,卧龙生11.黑白双娇:1;卧龙生12.剑气洞彻九重天:1;卧龙生41.无形剑:1;古  
龙15.楚留香07新月传奇:1;古龙26.浣花洗剑录:1;古龙54.七种武器04多情环:1;古龙61.失  
魂引:1;古龙64.湘妃剑:1;李凉15.江湖一担皮:1;金庸12倚天屠龙记:1  
鱼腥味 1.20,卧龙生01.镖旗:1;古龙15.楚留香07新月传奇:2;古龙46.那一剑的风情:1;李  
凉04.本尊分身:1;金庸12倚天屠龙记:1  
鱼腰 1.00,金庸08笑傲江湖:1  
鱼舟 2.20,卧龙生13.剑无痕:3;卧龙生18.金剑雕翎:2;卧龙生29.双凤旗:3;卧龙生37.天  
涯情侣:2;卧龙生39.铁苗神剑:1  
鱼船 1.57,卧龙生20.梦幻之刀:1;卧龙生37.天涯情侣:1;李凉09.红顶记:4;李凉16.江湖  
一品郎:1;李凉34.天下第一当:2;梁羽生19.狂侠天娇魔女:1;梁羽生30.散花女侠:1  
鱼花 1.00,李凉29.神偷小千:1  
鱼苗 1.00,李凉19.酒狂任小赌:1;梁羽生04.冰魄寒光剑:1  
鱼滕 3.00,卧龙生38.铁剑玉佩:3  
鱼虫 1.33,卧龙生11.黑白双娇:1;李凉40.新蜀山剑侠传:2;金庸12倚天屠龙记:1  
鱼虾 2.06,卧龙生04.地狱门:1;卧龙生07.飞燕惊龙:1;卧龙生11.黑白双娇:5;卧龙生13.  
剑无痕:1;卧龙生16.金笔点龙记:1;卧龙生24.七绝剑:2;卧龙生29.双凤旗:3;卧龙生35.天马
```

查看“风雪”的输出结果:

`hadoop fs -cat hdfsOutput/part-r-00000|grep "^风雪[:space:]]" --color=auto`



```
2017st20@master01:~  
e:]]" --color=auto  
17/10/29 12:30:43 WARN util.NativeCodeLoader: Unable to load native-hadoop libra  
ry for your platform... using builtin-java classes where applicable  
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.天才混混: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  
[2017st20@master01 ~]$
```

查看“江湖”的输出结果:

hadoop fs -cat hdfsOutput/part-r-000000|grep "^江湖[[:space:]]" --color=auto

```
2017st20@master01:~  
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.情剑无刃: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3  
3.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:1  
36;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3
```

集群 AllApplication (<http://114.212.190.91:8088/>) 页面: 2017st20

All Applications

Cluster: 114.212.190.91:8088/cluster

Cluster Metrics

Apps Submitted: 142

Apps Pending: 0

Apps Running: 0

Apps Completed: 142

Containers Running: 0

Memory Used: 0 B

Memory Total: 456 GB

Memory Reserved: 0 B

VCores Used: 0

VCores Total: 126

VCores Reserved: 0

Active Nodes: 13

Decommissioned Nodes: 0

Lost Nodes: 0

Unhealthy Nodes: 0

Rebooted Nodes: 0

User Metrics for dr.who

Apps Submitted: 0

Apps Pending: 0

Apps Running: 0

Apps Completed: 0

Containers Running: 0

Containers Pending: 0

Containers Reserved: 0

Memory Used: 0 B

Memory Pending: 0 B

Memory Reserved: 0 B

VCores Used: 0

VCores Pending: 0

VCores Reserved: 0

Scheduler Metrics

Scheduler Type: Fair Scheduler

Scheduling Resource Type: [MEMORY, CPU]

Minimum Allocation: <memory:4096, vCores:1>

Maximum Allocation: <memory:10240, vCores:10>

Show 20 entries

Search: 2017st20

| ID                             | User     | Name         | Application Type | Queue         | StartTime                      | FinishTime                     | State    | FinalStatus | Progress | Tracking UI |
|--------------------------------|----------|--------------|------------------|---------------|--------------------------------|--------------------------------|----------|-------------|----------|-------------|
| application_1508726229114_0148 | 2017st20 | invert index | MAPREDUCE        | root.2017st20 | Sat Oct 28 17:25:13 +0800 2017 | Sat Oct 28 17:26:36 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |
| application_1508726229114_0147 | 2017st20 | invert index | MAPREDUCE        | root.2017st20 | Sat Oct 28 17:22:07 +0800 2017 | Sat Oct 28 17:23:09 +0800 2017 | FINISHED | FAILED      |          | History     |
| application_1508726229114_0146 | 2017st20 | invert index | MAPREDUCE        | root.2017st20 | Sat Oct 28 17:09:25 +0800 2017 | Sat Oct 28 17:10:48 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |
| application_1508726229114_0138 | 2017st20 | invert index | MAPREDUCE        | root.2017st20 | Fri Oct 27 21:20:46 +0800 2017 | Fri Oct 27 21:21:59 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |
| application_1508726229114_0137 | 2017st20 | invert index | MAPREDUCE        | root.2017st20 | Fri Oct 27 20:56:33 +0800 2017 | Fri Oct 27 20:57:44 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |
| application_1508726229114_0136 | 2017st20 | invert index | MAPREDUCE        | root.2017st20 | Fri Oct 27 20:32:24 +0800 2017 | Fri Oct 27 20:33:41 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |
| application_1508726229114_0008 | 2017st20 | word count   | MAPREDUCE        | root.2017st20 | Tue Oct 24 10:24:49 +0800 2017 | Tue Oct 24 10:25:07 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |
| application_1508726229114_0007 | 2017st20 | word count   | MAPREDUCE        | root.2017st20 | Tue Oct 24 10:20:47 +0800 2017 | Tue Oct 24 10:21:50 +0800 2017 | FINISHED | FAILED      |          | History     |
| application_1508726229114_0006 | 2017st20 | word count   | MAPREDUCE        | root.2017st20 | Tue Oct 24 10:12:31 +0800 2017 | Tue Oct 24 10:12:59 +0800 2017 | FINISHED | SUCCEEDED   |          | History     |

Showing 1 to 9 of 9 entries (filtered from 142 total entries)

First Previous 1 Next Last

## WebUI 页面（http://114.212.190.91:19888/jobhistory）：2017st20

|                         |                         |                         |  |              |          |               |           |            |                |               |                   |
|-------------------------|-------------------------|-------------------------|--|--------------|----------|---------------|-----------|------------|----------------|---------------|-------------------|
| 2017.10.28 17:25:13 CST | 2017.10.28 17:25:19 CST | 2017.10.28 17:26:35 CST | <a href="#">job_1508726229114_0148</a> | invert index | 2017st20 | root.2017st20 | SUCCEEDED | 218        | 218            | 1             | 1                 |
| 2017.10.28 17:22:07 CST | 2017.10.28 17:22:07 CST | 2017.10.28 17:23:03 CST | <a href="#">job_1508726229114_0147</a> | invert index | 2017st20 | root.2017st20 | FAILED    | 29         | 29             | 0             | 0                 |
| 2017.10.28 17:09:25 CST | 2017.10.28 17:09:36 CST | 2017.10.28 17:10:48 CST | <a href="#">job_1508726229114_0146</a> | invert index | 2017st20 | root.2017st20 | SUCCEEDED | 218        | 218            | 1             | 1                 |
| 2017.10.27 21:20:46 CST | 2017.10.27 21:20:52 CST | 2017.10.27 21:21:59 CST | <a href="#">job_1508726229114_0138</a> | invert index | 2017st20 | root.2017st20 | SUCCEEDED | 218        | 218            | 1             | 1                 |
| 2017.10.27 20:56:33 CST | 2017.10.27 20:56:18 CST | 2017.10.27 20:57:23 CST | <a href="#">job_1508726229114_0137</a> | invert index | 2017st20 | root.2017st20 | SUCCEEDED | 218        | 218            | 1             | 1                 |
| Submit Time             | Start Time              | Finish Time             | Job ID                                 | Name         | User     | Queue         | State     | Maps Total | Maps Completed | Reduces Total | Reduces Completed |

## Job 详细页面：

|   |  |  |  |  |  |  |  |  |  |
|---|--|--|--|--|--|--|--|--|--|
| MapReduce Job job_1508726229114_0148  |  |  |  |  |  |  |  |  |  |
| <div> <div>  hadoop </div> <div> MapReduce Job job_1508726229114_0148 <div> <div> Application Job Overview Counters Configuration Map tasks Reduce tasks Tools </div> <div> <div> Job Name: invert index<br/>User Name: 2017st20<br/>Queue: root.2017st20<br/>State: SUCCEEDED<br/>Uberized: false<br/>Submitted: Sat Oct 28 17:25:13 CST 2017<br/>Started: Sat Oct 28 17:25:19 CST 2017<br/>Finished: Sat Oct 28 17:26:35 CST 2017<br/>Elapsed: 1mins, 16sec<br/>Diagnostics:<br/>Average Map Time: 5sec<br/>Average Shuffle Time: 24sec<br/>Average Merge Time: 16sec<br/>Average Reduce Time: 22sec </div> </div> </div> </div> </div> |  |  |  |  |  |  |  |  |  |
| <div> <div> <div> ApplicationMaster Attempt Number Start Time Node Logs </div> <div> 1 Sat Oct 28 17:25:16 CST 2017 slave007.8042 logs </div> </div> </div>   |  |  |  |  |  |  |  |  |  |
| <div> <div> <div> Task Type Total Complete </div> <div> Map 218 218 </div> <div> Reduce 1 1 </div> </div> <div> <div> Attempt Type Failed Killed Successful </div> <div> Maps 0 1 218 </div> <div> Reduces 0 0 1 </div> </div> </div>   |  |  |  |  |  |  |  |  |  |

## 五、 实验感想

这次实验是实现带词频属性的文档倒排算法，在实验过程中， 仔细体会了



MapReduce 并行编程模型，hadoop 系统架构和 MapReduce 程序执行过程，学会了使用编写 Mapper, Partitioner 和 Reducer，学会了使用集群进行 MapReduce 任务的执行，使用 WebU 查看任务的执行状态，查看任务执行历史，更加进一步的加深了对 MapReduce 框架的理解，对以后进一步学习大有益处。

## 六、 不足和改进的地方

无法对每次词语的平均出现次数进行全局排序，达不到现在主流的搜索引擎的搜索要求，希望对以后课程的学习，掌握和理解能够实现这一功能。