



基于微软学术API和Hadoop的 统计论文作者频率应用

《云计算概论》期末项目文档

指导老师：吴维刚老师

学号：13349162

姓名：郑沛霖

目录

一、设计功能.....	3
二、代码编写.....	3
1.异步获取并处理数据——getPaper.js	3
1.1.获取微软学术搜索 API 数据	3
1.2.提取对应年份论文作者.....	4
2.MapReduce 统计作者发表——PaperAuthor.java	5
2.1 输入要统计的会议或期刊、年份区间	5
2.2 MapReduce 统计各个年份	5
三、运行过程及截图.....	5
1.编译 java 并打包为 jar	5
2.修改 js 文件指定会议或期刊及年份并获取 API 数据.....	6
3.添加 authors/到 hadoop 的输入为 authors.....	6
4. hadoop 中调用 PaperAuthor.jar.....	7
5.输入搜索的会议或期刊及区间，进行统计	7
6. hadoop cat 输出即可查看各个作者的发表频率统计	7
四、总结与心得.....	8

一、设计功能

这个项目是《云计算概论》的课程项目设计，我设计的功能为根据需要在 nodejs 输入期刊或会议名称及区间范围，获取微软学术搜索 API 的数据，并根据 java 控制台的输入，统计某个年份区间、某个会议或期刊的作者，从而统计出各个作者的发表频率。

二、代码编写

1.异步获取并处理数据——getPaper.js

1.1.获取微软学术搜索 API 数据

```
CorJ = "J";//会议为C，期刊为J
CJname = "tc";//会议或期刊的缩写
Ybegin = 2001;//开始年份
Yend = 2015;//结束年份

for (Y=Ybegin; Y<=Yend; Y++)
http.get("http://oxfordhk.azure-api.net/academic/v1.0/
evaluate?expr=AND(Composite("+CorJ+"."+CorJ+"N='"+
CJname+"'),Y="+Y+")&count=10000&attributes=Y,AA.
AuN&subscription-key=f7cc29509a8443c5b3a5e56b0e38b5a6
```

如图所示，通过在 nodejs 中设置会议或期刊的名字、年份等，调用微软学术搜索 API，在不同年份异步获取论文和数据；

（所使用 api 的 key 是参加比赛时留下的...）

1.2.提取对应年份论文作者

```
var shuju;
shuju=JSON.parse(data);
var nian;
nian = shuju.entities[0].Y.toString();

var data2 = "";
var str;
for (e in shuju.entities)
    if (shuju.entities[e].AA!=null)
        for (aa in shuju.entities[e].AA) {
            if (shuju.entities[e].AA[aa].AuN!=null)
                str = shuju.entities[e].AA[aa].AuN;
            if (str=="weigang wu")
                console.log("wow see "+CJname+nian);//
                快迟交了这一行就很尴尬了T-T
            str=str.replace(" ","-");
            data2 += str + " ";
        }

fs.writeFile(path.join("authors",CJname+nian), data2,
    function (err) {
        if (err) throw err;
        console.log(nian + ' saved'); //文件被保存
    });
```

在各个 get 异步完成时，通过对获得的数据进行 JSON 解析，并提取出对应的年份和作者，按年份存储到 authors 目录下。

这里没有做异常处理是因为相信我们输入的会议或期刊、年份，都是有论文的。。。

此处查看的文档为[微软认知服务文档](#)

2.MapReduce 统计作者发表——PaperAuthor.java

2.1 输入要统计的会议或期刊、年份区间

```
//输入会议名字
Scanner sc = new Scanner(System.in);

System.out.print("Enter Conference or Journal (etc:tpds,tc...) :");
String CJname = sc.nextLine();

System.out.print("Enter BeginningYear:");
int beginning = sc.nextInt();

System.out.print("Enter EndingYear:");
int ending = sc.nextInt();
```

读取控制台输入：会议或期刊、开始年份、结束年份

2.2 MapReduce 统计各个年份

```
//读取输入的会议或期刊的对应年份的作者记录
while (beginning!=ending+1) {
    String st = Integer.toString(beginning);
    FileInputFormat.addInputPath(job, new Path(inputDirName+"/"+CJname+st));
    beginning++;
}
```

此处使用的 MapReduce 是参考了 Lab-2C-InvertedIndex 的实验，在 MapReduce 的输入中，定义年份的输入范围，然后进行相应的映射和归约。

如图为定义年份的输入范围的代码部分。

三、运行过程及截图

1.编译 java 并打包为 jar

```
hadoop@ubuntu:/usr/local/hadoop/code$ javac -cp ../share/hadoop/common/hadoop-common-2.6.0.jar:../share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.6.0.jar:../share/hadoop/common/lib/commons-cli-1.2.jar PaperAuthor.java -d ./
Note: PaperAuthor.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
hadoop@ubuntu:/usr/local/hadoop/code$ jar -cvf PaperAuthor.jar *.class
added manifest
adding: PaperAuthor.class(in = 3189) (out= 1632)(deflated 48%)
adding: PaperAuthor$Elem.class(in = 1348) (out= 713)(deflated 47%)
adding: PaperAuthor$InvertedIndexCombiner.class(in = 1706) (out= 760)(deflated 55%)
adding: PaperAuthor$InvertedIndexMapper.class(in = 2120) (out= 919)(deflated 56%)
adding: PaperAuthor$InvertedIndexPartitioner.class(in = 839) (out= 461)(deflated 45%)
adding: PaperAuthor$InvertedIndexReducer.class(in = 2809) (out= 1228)(deflated 56%)
```

2.修改 js 文件指定会议或期刊及年份并获取 API 数据

这里以 **tpds** 为例。

```
CorJ = "J";//会议为C，期刊为J
CJname = "tpds";//会议或期刊的缩写
Ybegin = 2001;//开始年份
Yend = 2015;//结束年份
```

```
hadoop@ubuntu:/usr/local/hadoop/code$ sudo nodejs getPaper.js
2002 saved
2001 saved
2003 saved
2004 saved
2006 saved
wow see tpds2009
2007 saved
2008 saved
2009 saved
2010 saved
2012 saved
wow see tpds2013
2013 saved
2011 saved
2014 saved
wow see tpds2015
2015 saved
2005 saved
```

(在已编译打包且数据已获得的时候，1-2 步可跳过)

3.添加 authors/到 hadoop 的输入为 authors

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -put code/authors/ authors
```

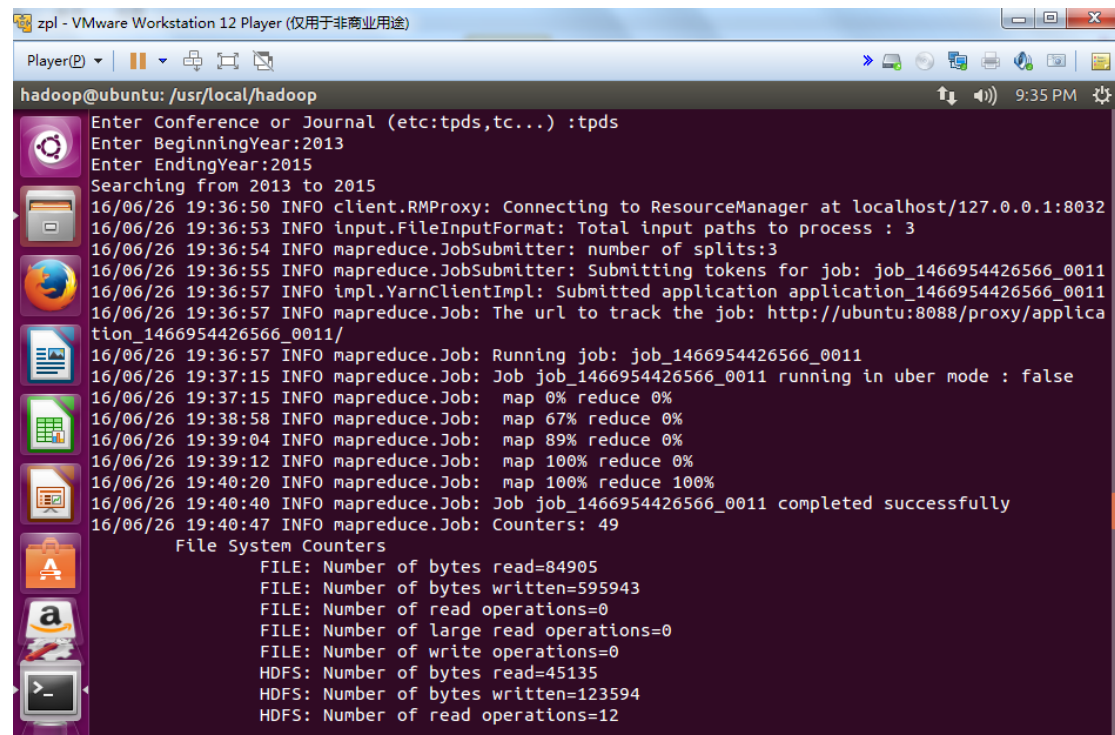
4. hadoop 中调用 PaperAuthor.jar

如：bin/hadoop jar code/PaperAuthor.jar PaperAuthor authors output10

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop jar code/PaperAuthor.jar PaperAuthor authors output10
```

5.输入搜索的会议或期刊及区间，进行统计

这里以 **tpds** , 2013~2015 为例。



```
hadoop@ubuntu:/usr/local/hadoop
Enter Conference or Journal (etc:tpds,tc...) :tpds
Enter BeginningYear:2013
Enter EndingYear:2015
Searching from 2013 to 2015
16/06/26 19:36:50 INFO client.RMPProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
16/06/26 19:36:53 INFO input.FileInputFormat: Total input paths to process : 3
16/06/26 19:36:54 INFO mapreduce.JobSubmitter: number of splits:3
16/06/26 19:36:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1466954426566_0011
16/06/26 19:36:57 INFO impl.YarnClientImpl: Submitted application application_1466954426566_0011
16/06/26 19:36:57 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1466954426566_0011/
16/06/26 19:36:57 INFO mapreduce.Job: Running job: job_1466954426566_0011
16/06/26 19:37:15 INFO mapreduce.Job: Job job_1466954426566_0011 running in uber mode : false
16/06/26 19:37:15 INFO mapreduce.Job: map 0% reduce 0%
16/06/26 19:38:58 INFO mapreduce.Job: map 67% reduce 0%
16/06/26 19:39:04 INFO mapreduce.Job: map 89% reduce 0%
16/06/26 19:39:12 INFO mapreduce.Job: map 100% reduce 0%
16/06/26 19:40:20 INFO mapreduce.Job: map 100% reduce 100%
16/06/26 19:40:40 INFO mapreduce.Job: Job job_1466954426566_0011 completed successfully
16/06/26 19:40:47 INFO mapreduce.Job: Counters: 49
File System Counters
  FILE: Number of bytes read=84905
  FILE: Number of bytes written=595943
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=45135
  HDFS: Number of bytes written=123594
  HDFS: Number of read operations=12
```

6. hadoop cat 输出即可查看各个作者的发表频率统计

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -cat output10/*
```

输出如下图：

```
zpl - VMware Workstation 12 Player (仅用于非商业用途)
Player(P) | [Icons]
hadoop@ubuntu: /usr/local/hadoop
[Icons] weichen-liu papers count: 1 : (tpds2013, 1)
weichung-hsu papers count: 1 : (tpds2014, 1)
weidong-bao papers count: 1 : (tpds2015, 1)
weifa-liang papers count: 2 : (tpds2013, 1) (tpds2015, 2)
weigang-wu papers count: 2 : (tpds2013, 1) (tpds2015, 1)
weihang-wang papers count: 1 : (tpds2015, 1)
weiho-chung papers count: 1 : (tpds2014, 1)
weihua-zhuang papers count: 2 : (tpds2013, 1) (tpds2014, 1)
weihuang-fu papers count: 1 : (tpds2013, 1)
weijia-jia papers count: 2 : (tpds2013, 1) (tpds2014, 1)
weijia-song papers count: 1 : (tpds2013, 1)
weijie-wu papers count: 1 : (tpds2014, 1)
weijun-guo papers count: 1 : (tpds2014, 1)
weikuan-yu papers count: 1 : (tpds2014, 1)
weili-han papers count: 1 : (tpds2014, 1)
weili-wu papers count: 1 : (tpds2013, 1)
weilian-xue papers count: 1 : (tpds2014, 1)
weimin-zheng papers count: 2 : (tpds2014, 2) (tpds2015, 2)
weiming-zheng papers count: 1 : (tpds2015, 1)
weiping-zhu papers count: 2 : (tpds2014, 1) (tpds2015, 1)
weirong-jiang papers count: 1 : (tpds2014, 1)
weisheng-si papers count: 1 : (tpds2015, 1)
weishih-yang papers count: 1 : (tpds2013, 1)
weishinn-ku papers count: 1 : (tpds2015, 1)
weisong-shi papers count: 2 : (tpds2013, 1) (tpds2015, 1)
weiwei-fang papers count: 1 : (tpds2015, 1)
weiwei-sun papers count: 1 : (tpds2014, 1)
weixi-gu papers count: 1 : (tpds2014, 1)
weixian-liao papers count: 1 : (tpds2015, 1)
```

四、总结与心得

1.这是《云计算概论》的课程项目，在这一项目中，我初次接触 Hadoop\MapReduce 编程等，确实非常有难度，不过好在各种文档都非常齐全，总算完成。

2.在虚拟机的环境配置中，**由于 S3 外网不可用，且我的电脑配置不佳，无法同时跑多台虚拟机，诚实的说最后是配置了单虚拟机的 hadoop 集群。** Master 和 Slaves 都是本机，**但是如果需要扩展也可以修改配置即可运行分布式程序。**

3.在获取 API 数据的选择时 ,使用的 API-key 是参加微软编程之美复赛时留下的**商业版 key** ,有着较好的反应速度 ,最后不用 java 获取而用的是 nodejs ,是因为 nodejs 的**异步单线程比较适合我这种配置比较低的环境**。

4.获取 API 数据后 ,异步处理为对应年份的数据时 ,没有做异常处理是因为相信输入的会议或期刊及年份都是有效的 ,且没有时间做异常处理。。。

5.不足之处是只写了统计某个会议\期刊在某些年份的统计情况 ,以后还可以考虑更改输入 ,即可实现对某个领域的一批会议或期刊的统计。

6.我在所给的词频统计样例的基础上 ,添加了获取数据、提取作者、统计论文作者 ,修改输出统计的功能 , **进一步熟悉了 java\nodejs\hadoop\MapReduce 编程 ,也算是做了比较具有实用意义的统计年份区间的论文作者的功能** ,收获颇丰。

五、参考文档

1.[课程主页的各个文档](#)

2.[微软认知服务文档](#)

3.[NodeJS API](#)

4.[Hadoop 文档](#)

5.....