復旦大學

本科生课程报告

课程名称： 分布式系统　　**课程代码：** COMP130123.01

姓 名： 曾瑞莹　　**学　　号：** 16307130345

学 院： 计算机科学技术学院　　**专　　业：** 计算机科学与技术

分布式系统课程项目报告

——Wikipedia Index

16307130345 曾瑞莹

一、 项目简介

- 该项目分为两个部分。第一部分利用 MapReduce 框架构建出服务器上的 Wikipedia 的倒排索引，索引信息包括 TF 信息，DF 信息和 Position 信息；第二部分是针对语料样本，通过得到的索引文件，计算出每篇文档中每个单词的 TF-IDF，再实现可视化界面，可通过查找单词列出语料样本中含有该单词的所有文档 ID。(根据该单词在该文档中的 TF-IDF 的降序排列)

二、 项目实现环境

- Windows 10
- Ubuntu 16.04LTS
- IntelliJ IDEA
- Atom

三、 项目说明

- 本项目依赖于一些开源项目，包括 Hadoop File System，Maven，Bootstrap。
- 本项目第一部分通过 IntelliJ IDEA 构建一个 Maven 项目，用 Java 编写该项目的源代码，最后打包成 jar 包发送到服务器。并且在本地搭建了一个 Hadoop 的伪分布式环境以方便测试。
- 本项目第二部分使用 Atom 编辑器，先用 Python 对得到的索引文件进行处理，再用 Bootstrap 和 jQuery 实现异步的可视化界面。
- 在实现异步的可视化界面时，由于 Chrome 浏览器的安全策略决定了`file`协议访问的应用无法使用 XMLHttpRequest 对象，因此用浏览器打开该页面

前，需进行如下(https://www.cnblogs.com/micua/p/chrome-file-protocol-support-ajax.html)操作。

- 给定 PPT 上的一个单词的 TF 值为一篇文档中该单词出现的次数，但我认为这丢失了文档总数这一信息，因此以下前大部分内容中一个单词的 TF 值是指：一篇文档中该单词出现的次数/该文档单词总数。但最后一个版本的代码的 TF 值为一篇文档中该单词出现的次数。
- 一个单词的 DF 值表示该单词在语料中包含该单词的文档的文档总数。
- 一个单词的 Position 值表示一篇文档中该单词出现的段落位置总和。

四、 项目实现过程

1. 准备工作

- 从网上下载 Wikipedia 的部分 Xml 数据作为样本(约 13M)，观察后发现每一对<page></page>标签之间记录了一篇文档的相关信息，其中文档的 ID 为第一对<id></id>标签之间的内容，文档的内容为第一对<text></text>标签之间的内容。
- 熟悉 Maven 项目的框架，为项目实现添加依赖包。
- 参考 Github 上的资料，实现一个 XmlInputFormat 类继承于 TextInputFormat 类把 MapReduce 中原本基于行进行 map 的行为改变为基于<page></page>标签进行 map(XmlInputFormat.java)。

2. 第一阶段

- 此时对该项目第一部分的实现分成 4 个 Java 文件，第一个 Java 文件计算了每篇文章中每个单词的 TF 值(在最后提交的代码文件中该文件已被删除)，第二个 Java 文件计算了该语料中每个单词的 DF 值以及对应文档的 TF-IDF 值(即 DF_TFIDF.java)。MaxThreeKey.java 文件是在每篇文档中选出 TFIDF 值最大的前三个单词，原意是在前端界面中使用。

3. 第二阶段

- 此时由于老师说明无须记录每篇文档中每个单词的 TFIDF 值，因此就想用

Combiner 通过一次 MapReduce 计算出每个单词的 TF 和 DF 值(即 TF_DF.java)，于是又添加了一个 Combiner 类。用下载的小样本在本地伪分布式上测试可得出正确结果，但放到服务器上运行则会一直在 Combiner 处报错，错误类型为读入无效字符。同时，也用 Combiner 计算每个单词在每篇文档中的 Position 值，但此时 Combiner 的函数与 Reduce 的函数相同，在服务器上运行没有问题，并且测试到没有用 Combiner 也不会导致内存不足的问题。初步猜测 TF_DF.java 不可行是因为 Combiner 和 Reduce 的 key 和 value 要完全相同，而不能只是类型相同。

4. **第三阶段**

- 由于第二阶段碰到的 Combiner 问题，能力不足尚无法解决，最后只能分成两次 MapReduce 计算 TF(即 TF.java)和 DF(即 DF.java)的值并存到同一文件中，但此时记录三个信息则需要三次 MapReduce，并且 Position 信息与 TFDF 信息是分开存放的。同时老师要求 TF 值为一篇文档中该单词出现的次数，因此 TFij.java 则是计算出老师要求的 TF 值。

5. **第四阶段**

- 由于时间问题可视化界面只在本地实现，并且语料为样本语料。对第三阶段得到的 tfdf 信息，用 Python 做预处理得到每个单词在每篇文档的 TFIDF 值，并且存为 json 文件。用 Bootstrap 和 jQuery 实现界面与一点逻辑(比较简陋)。

6. **第五阶段**

- 由于老师要求 TF、DF 和 Position 要在同一个索引文件中，因此又重新写了一个版本，此时只需要两次 MapReduce 过程就可以记录三个信息，并且三个信息存放在同一个索引文件中，此时分为两个步骤。
- 第一步骤 TF_Position.java，该文件计算了每个单词在每篇文档中的 TF 值和 Position 值。Map 函数按<page></page>标签读取维基百科语料，

输入为<默认 key 值，每篇文档信息>，通过对每篇文档信息进行处理可得到每篇文档的 ID 以及内容，此时 Map 函数的输出为<单词+文档 ID，单词段落>。Reduce 函数的输入与 Map 函数的输出相同，处理后，Reduce 函数的输出为<单词+文档 ID，TF 值+Position 值>，将该中间结果输出。

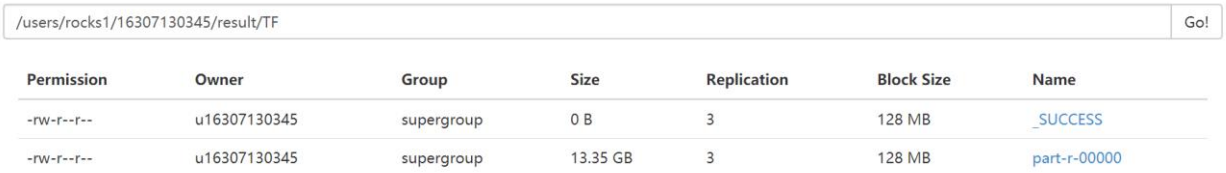
- 第二步骤 TF_DF_Position.java，该文件计算了每个单词的 DF 值并且将三个信息同时输出。Map 函数按行读取第一步骤的中间结果，输入为<默认 key 值，每个单词的 TF 和 Position 值>，通过处理可将 Map 输出变为<单词，文档 ID+TF+Position>。Reduce 函数的输入与 Map 函数的输出相同，处理后，Reduce 函数的输出为<单词，DF+文档 ID+TF+Position>。
- 在该倒排索引文件中，一个单词理应只出现一次，即应该对第二步骤结果进行压缩。于是实现了 MyWritable 类接口为 Writable 类 (MyWritable.java)。将同一个单词的所有这三个信息通过文档 ID 区别开进行存储。存储方式如下图：


```
{"aamnihol":     ["1#2087235#1#30"]},
{"aamnloled":    ["1#317702#1#51"]},
{"aamnlrmauon":  ["1#2066033#1#68"]},
{"aamnmc":       ["1#311661#1#27"]},
{"aamnmmwr":     ["1#320137#1#60"]},
{"aamnnemgm":    ["1#544663#1#26"]},
{"aamnoa":       ["1#526371#1#45"]},
{"aamnrca":      ["1#531772#1#31"]},
{"aamnroa":      ["1#531604#1#31"]},
{"aamntoi":      ["1#2097490#1#43"]},
{"aamnvyo":      ["1#2087240#1#91"]},
{"aamnzr":       ["1#552823#1#7"]},
{"aamo":         ["4#553112#1#69","4#358365#1#14","4#1943665#1#30","4#553546#1#60"]},
{"aamoa":        ["2#538317#1#40","2#2123039#1#25"]},
{"aamoas":       ["1#563313#1#45"]},
{"aamodt":       ["23#1647421#2#14-19","23#1002598#1#41","23#1058273#1#17","23#1331948#1#192
"23#1331949#1#24","23#1647343#2#29-21","23#1647413#1#30","23#1647420#1#18","23#1647423#3#5-
7","23#1647425#1#54","23#1647426#1#40","23#1647429#2#37-7","23#1647431#2#28-43","23#1647432
#6-40","23#1647433#1#11","23#1649094#1#9","23#1649102#1#65","23#1649114#1#64","23#1649120#1
04","23#233735#1#161","23#234311#1#167","23#415772#4#300-299-503","23#1647412#6#31-55-28-35
9"]},
{"aamol":        ["1#1061535#1#46"]},
{"aamom":        ["1#325601#1#23"]},
{"aamomns":      ["1#540616#1#26"]},
{"aamomr":       ["1#312543#1#3"]},
{"aamomttnt":    ["1#2335247#1#5"]},
{"aamon":        ["1#406090#1#2"]},
{"aamona":       ["1#1263269#1#38"]},
{"aamonasaria":  ["1#2062147#1#39"]},
{"aamonfiai":    ["1#1127030#1#28"]},
{"aamong":       ["1#1643918#1#1043"]},
{"aamonition":   ["1#1263232#1#97"]},
{"aamonium":     ["1#2100802#1#20"]},
```

五、 项目运行方式(使用的维基百科语料均使用其他同学上传到 hdfs 的现成语料)

1. TF.java:

- hadoop jar WikiIndex.jar TF {InputPath1} {OutputPath1}

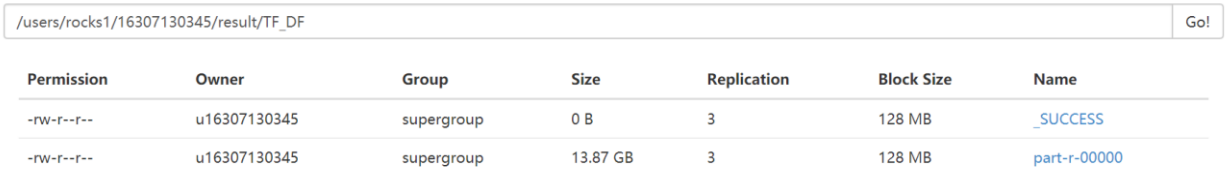


/users/rocks1/16307130345/result/TF Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	u16307130345	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	u16307130345	supergroup	13.35 GB	3	128 MB	part-r-00000

2. DF.java:

- Hadoop jar WikiIndex.jar DF {OutputPath1} {OutputPath2}

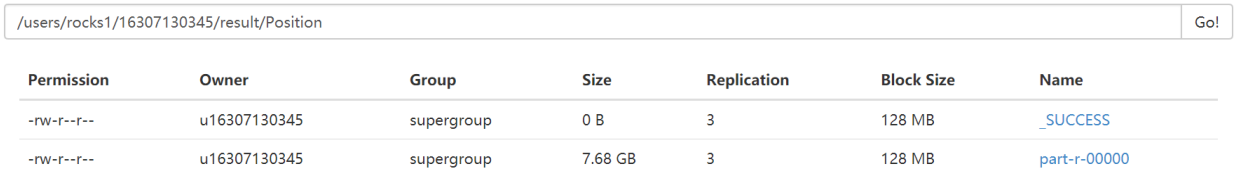


/users/rocks1/16307130345/result/TF_DF Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	u16307130345	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	u16307130345	supergroup	13.87 GB	3	128 MB	part-r-00000

3. Position.java:

- Hadoop jar WikiIndex.jar Position {InputPath1} {OutputPath3}

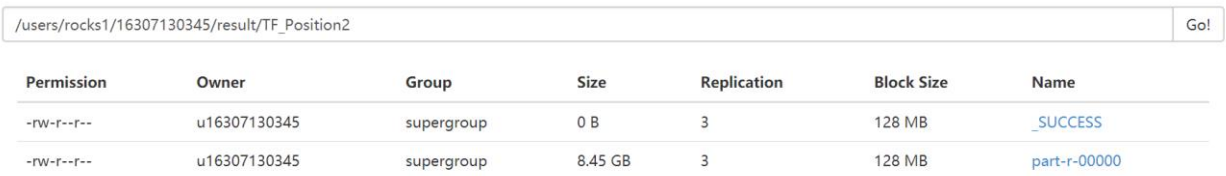


/users/rocks1/16307130345/result/Position Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	u16307130345	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	u16307130345	supergroup	7.68 GB	3	128 MB	part-r-00000

4. TF_Position.java:

- Hadoop jar WikiIndex.jar TF_Position {InputPath1} {OutputPath4}



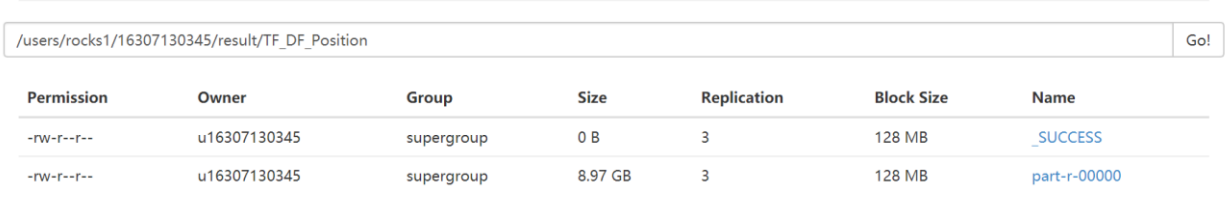
/users/rocks1/16307130345/result/TF_Position2 Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	u16307130345	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	u16307130345	supergroup	8.45 GB	3	128 MB	part-r-00000

5. TF_DF_Position.java:

- hadoop jar WikiIndex.jar TF_DF_Position {OutputPath4}

  {OutputPath5}



Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	u16307130345	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	u16307130345	supergroup	8.97 GB	3	128 MB	part-r-00000

6. 可视化界面：