

FBDP-作业5

邵一淼 191098180

实验环境

单机使用：Java8+IntelliJ IDEA 2018.3.6

集群使用：Java7+wsl2+docker

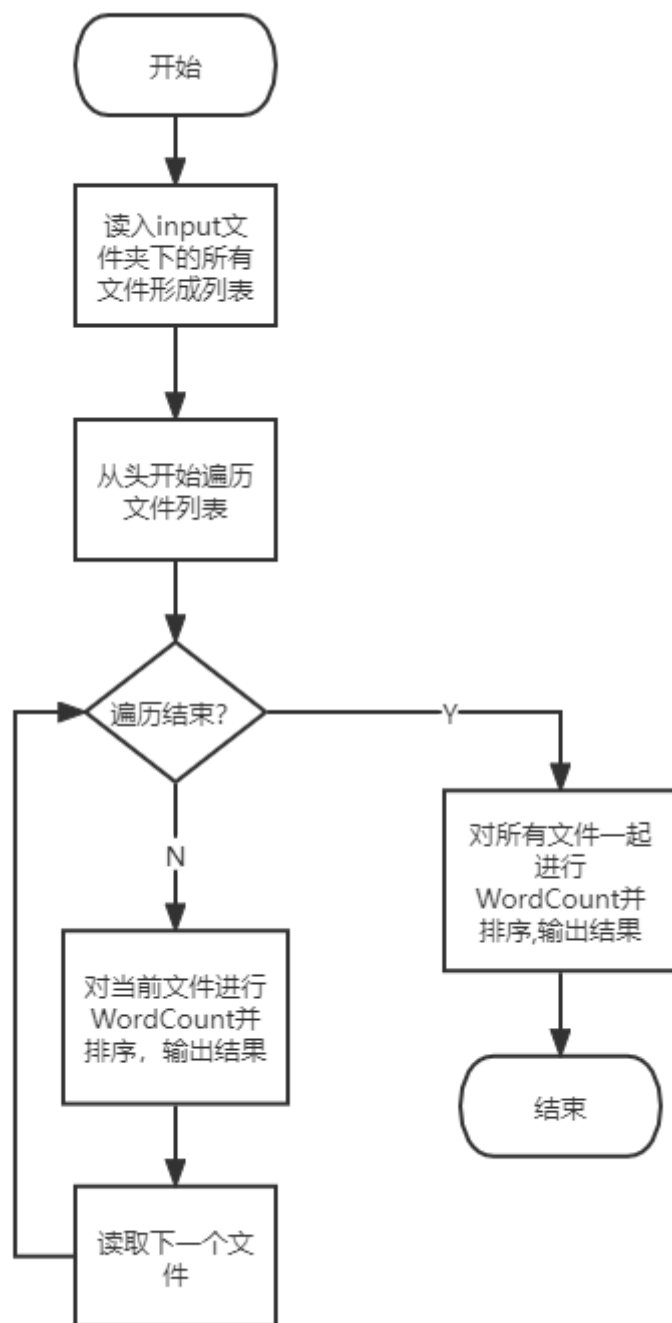
需求分析

在HDFS上加载莎士比亚文集的数据文件（shakespeare-txt.zip解压后目录下的所有文件），编写MapReduce程序进行词频统计，并按照单词出现次数从大到小排列，输出（1）每个作品的前100个高频单词；（2）所有作品的前100个高频单词，要求忽略大小写，忽略标点符号（punctuation.txt），忽略停用词（stop-word-list.txt），忽略数字，单词长度 ≥ 3 。输出格式为"<排名>: <单词>, <次数>", 输出根据作品名称不同分别写入不同的文件。

为了获得每个作品的前个高频词和所有作品的前个高频词，需要做两类WordCount，第一类是对每个作品分别统计一次，第二类是对全部作品一起做一次。

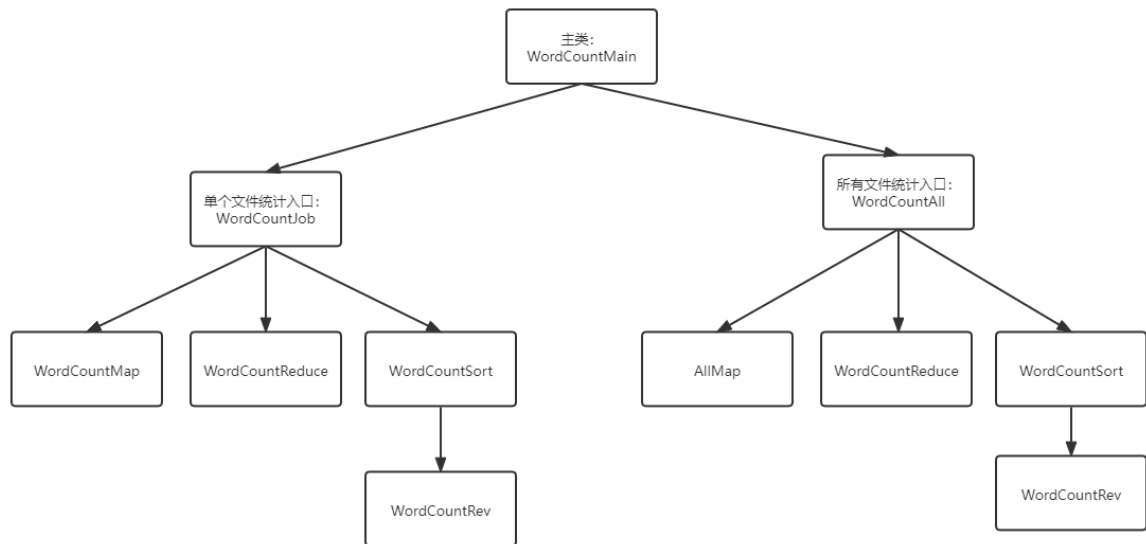
特殊要求：忽略大小写，忽略标点符号

应对方法：在map时，读入停用词和标点文件，构建停用词、标点集合，在将单词写入文件时，与停用词、标点集合相匹配，然后剔除无关单词。



设计思路

下图为类关系，箭头代表调用



类名	主要实现的功能
WordCountMain	主类，主要用于读入input，并对每个文件调用WordCountJob.job方法，对所有文件使用WordCountAll.job方法。
WordCountJob	该类对输入文件进行一个词频统计的job,和一个排序的sortjob，job1中设置mapper类为WordCountMap,reducer类为WordCountReduce
WordCountMap	实现map类
WordCountReduce	实现reduce类
WordCountSort	实现取前100个从大到小
WordCountRev	实现sort
WordCountAll	实现对所有文件的WordCount

以下尝试用伪代码展示部分算法过程，伪代码运用不熟练，参考了一些资料，若有使用错误请批评指正。

```

class WordCountMap
  procedure map(docid n, doc d)
    for all term t in doc d
      EMIT(term t, <n, 1>)

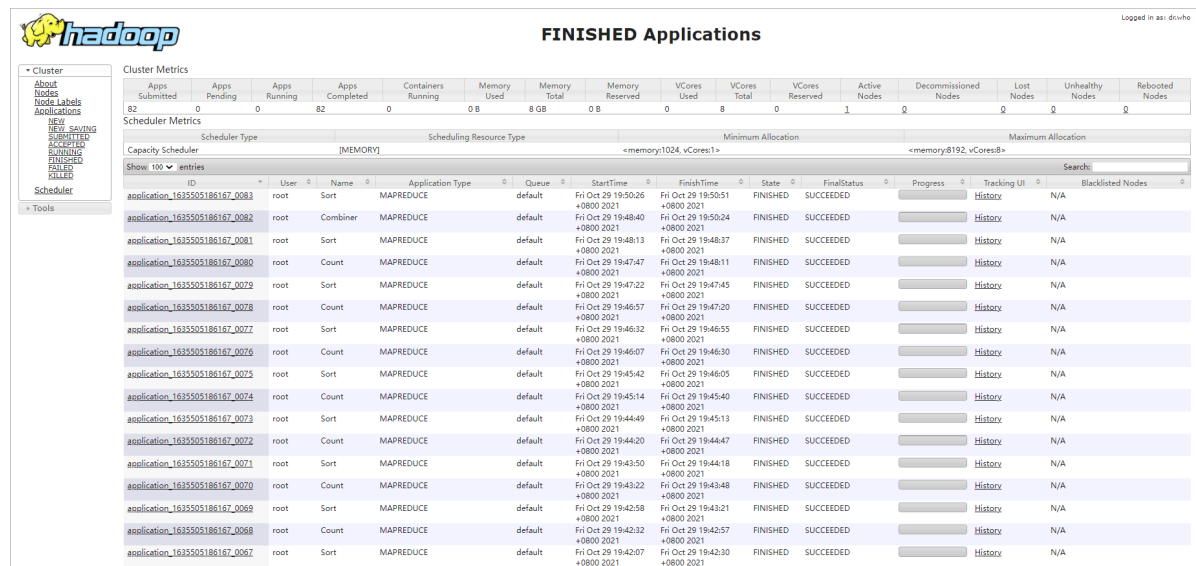
class WordCountReducer
  method INITIALIZE
    t(pre) <-- null
  procedure reducer(term t, postings[<docid n1, tf1>, <docid n2, tf2>....])
    P <-- new ASSOCIATIVE_SORTED_MAP
    if t(pre) != t AND t(pre) != null
      EMIT(t, P)
      P.RESET
    for all posting <n, tf> in postings[....]
      P{n, tf} = P{n, tf++};
  method CLOSE
  
```

实验结果

实验代码及结果已上传[Fairy-Miaomiao/WordCount: Use MapReduce to do WordCount.](https://github.com/Fairy-Miaomiao/WordCount: Use MapReduce to do WordCount.)
(github.com)

实验结果可在output文件夹中查看，所有文件的前100个高频词可在output/allresult文件夹中查看。单个文件的前100个高频词可在output/top100result中寻找与文件同名txt查看。

以下展示网站截图：



The screenshot shows the Hadoop YARN web interface with the title "FINISHED Applications". It displays a table of application metrics and a detailed list of finished applications.

Cluster Metrics															
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	V-Cores Used	V-Cores Total	V-Cores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
82	0	0	82	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

Scheduler Metrics		Scheduling Resource Type		Minimum Allocation		Maximum Allocation	
Capacity Scheduler	Scheduler Type	[MEMORY]		<memory1024, vCores1>		<memory8192, vCores8>	

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
application_1635505186167_0083	root	Sort	MAPREDUCE	default	Fri Oct 29 19:50:26 +0800 2021	Fri Oct 29 19:50:51 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0082	root	Combiner	MAPREDUCE	default	Fri Oct 29 19:48:40 +0800 2021	Fri Oct 29 19:50:24 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0081	root	Sort	MAPREDUCE	default	Fri Oct 29 19:48:13 +0800 2021	Fri Oct 29 19:48:37 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0080	root	Count	MAPREDUCE	default	Fri Oct 29 19:47:47 +0800 2021	Fri Oct 29 19:48:11 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0079	root	Sort	MAPREDUCE	default	Fri Oct 29 19:47:22 +0800 2021	Fri Oct 29 19:47:45 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0078	root	Count	MAPREDUCE	default	Fri Oct 29 19:46:57 +0800 2021	Fri Oct 29 19:47:20 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0077	root	Sort	MAPREDUCE	default	Fri Oct 29 19:46:32 +0800 2021	Fri Oct 29 19:46:55 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0076	root	Count	MAPREDUCE	default	Fri Oct 29 19:46:07 +0800 2021	Fri Oct 29 19:46:30 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0075	root	Sort	MAPREDUCE	default	Fri Oct 29 19:45:42 +0800 2021	Fri Oct 29 19:46:05 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0074	root	Count	MAPREDUCE	default	Fri Oct 29 19:45:14 +0800 2021	Fri Oct 29 19:45:40 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0073	root	Sort	MAPREDUCE	default	Fri Oct 29 19:44:49 +0800 2021	Fri Oct 29 19:45:13 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0072	root	Count	MAPREDUCE	default	Fri Oct 29 19:44:20 +0800 2021	Fri Oct 29 19:44:47 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0071	root	Sort	MAPREDUCE	default	Fri Oct 29 19:43:50 +0800 2021	Fri Oct 29 19:44:18 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0070	root	Count	MAPREDUCE	default	Fri Oct 29 19:43:22 +0800 2021	Fri Oct 29 19:43:48 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0069	root	Sort	MAPREDUCE	default	Fri Oct 29 19:42:58 +0800 2021	Fri Oct 29 19:43:21 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0068	root	Count	MAPREDUCE	default	Fri Oct 29 19:42:32 +0800 2021	Fri Oct 29 19:42:57 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A
application_1635505186167_0067	root	Sort	MAPREDUCE	default	Fri Oct 29 19:42:07 +0800 2021	Fri Oct 29 19:42:30 +0800 2021	FINISHED	SUCCEEDED	<div></div>	History	N/A

实验改进

1、单个文件 vs 多个文件

一开始的设计思路为对读入的每个文件做一次完整的WordCount，然后重新读入input文件夹，做一次所有的WordCount，后来发现在做所有文件词频统计时，重新读入input文件是不必要的，为了节省时间可以使用单个文件WordCount的结果。

2、扩展input文件格式

当前对于input文件夹的使用主要是，读取input文件夹下的所有文件名并形成list，这要求input文件夹下至少全是txt格式，为了提高适用范围，优化扩展性，可以使用递归的方法读取input每个文件夹下的txt，使input文件夹中可以文件、文件夹并存，而不是单一文件。

3、绝对路径 vs 相对路径

此问题仅在windows单机编写代码时适用。

刚开始考虑到代码在不同机器上的可使用性，所有路径表示都使用了相对路径，一般情况下不会报错，只有当进行排序job的时候，不知为何当前位置跑到了一个子目录下，有点乱套，为避免这个问题，在单机上实验的时候我将其改为了绝对路径。

同时考虑到代码的适用范围，可改进的地方为：将所有路径先在主类的main方法中申明，然后使用传参的方式传到各个类方法中。即使有绝对路径需要修改也可以一目了然。若是所有都是用相对路径不报错则更佳。

参考文献：

[Hadoop学习笔记—12.MapReduce中的常见算法 - EdisonZhou - 博客园\(cnblogs.com\)](#)

[一些算法的MapReduce实现——倒排索引实现 iTer的专栏-CSDN博客](#)

[What does "emit" mean in general computer science terms? - Stack Overflow](#)

[emit - “emit”在一般计算机科学术语中是什么意思? - IT工具网\(coder.work\)](#)