

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： CS1804交换**

**学 号： X2020I1007**

**姓 名： 刘日星**

**指导教师： 杨驰**

**报告日期： 2021.12.03**

**计算机科学与技术学院**

**目录**

[实验一 wordCount算法及其实现 1](#_Toc89439814)

[**1.1实验目的** 1](#_Toc89439815)

[**1.2 实验内容** 1](#_Toc89439816)

[**1.3 实验过程** 1](#_Toc89439817)

[1.3.1 编程思路 1](#_Toc89439818)

[1.3.2 遇到的问题及解决方式 1](#_Toc89439819)

[1.3.3 实验测试与结果分析 1](#_Toc89439820)

[**1.4 实验总结** 1](#_Toc89439821)

# 实验一 wordCount算法及其实现

## **1.1实验目的**

1、理解map-reduce算法思想与流程；

2、应用map-reduce思想解决wordCount问题；

3、（可选）掌握并应用combine与shuffle过程。

## **1.2 实验内容**

提供9个预处理过的源文件（source01-09）模拟9个分布式节点，每个源文件中包含一百万个由英文、数字和字符（不包括逗号）构成的单词，单词由逗号与换行符分割。

要求应用map-reduce思想，模拟9个map节点与3个reduce节点实现wordCount功能，输出对应的map文件和最终的reduce结果文件。由于源文件较大，要求使用多线程来模拟分布式节点。

学有余力的同学可以在map-reduce的基础上添加combine与shuffle过程，并可以计算线程运行时间来考察这些过程对算法整体的影响。

提示：实现shuffle过程时应保证每个reduce节点的工作量尽量相当，来减少整体运行时间。

## **1.3 实验过程**

### 1.3.1 编程思路

利用python内置库threading模块里的Thread类实现多线程编程操作，也就是使用多线程操作来模拟分布式节点的操作。对每个源文件里的内容进行格式化处理和输出，按照Map-Reduce思想模拟map节点与reduce节点，分别输出到不同目标文件内，实现wordCount的功能。其中，每一条线程的运行都会采用time.clock函数对每次运行所花费的时间计时。

Map功能：

通过调用threading内置库里的Thread类创建与源文件数量相等的线程，每条线程对应一个source文件，并且每条线程的运行操作均相同，用于运行线程的函数也相同。在map函数里，打开每一个source文件进行逐行读取文件内的内容，然后使用strip()函数去掉每行末尾的换行符，并且使用split()函数以逗号隔开每一个单词。最后将每个单词以 word，1 的格式逐个换行写入对应的map目标文件内。

Shuffle功能：

使用与map文件数量相等的线程数分别将map文件内的内容（单词）按首字母分类，不同首字母的单词分别写入不同的shuffle目标文件中，写入模式为a（写入文件，若文件不存在则会先创建再写入，不会覆盖原文件，而是在文件末尾追加），每次写入都是在文件末端写入而不是重新覆盖原文件，此操作方便给Reduce部分进行统计单词数并且去重。

Reduce功能：

读取每一个shuffle文件内的单词，并且用python的字典形式将每个单词读取出来存入字段，并且判断其是否在字典里重复出现，每当重复出现1次，则计数加1，如果没有，则将该单词存入字典中，并且将其value值置为1。然后将单词按首字母排序以及 “word num” 的格式写入reduce目标文件内。

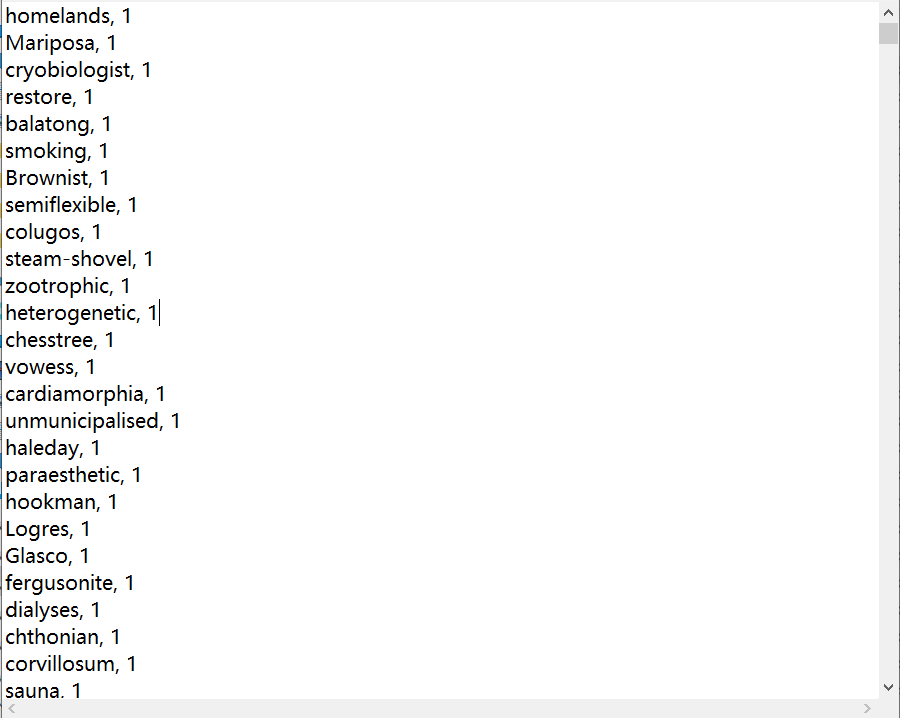
### 1.3.2 遇到的问题及解决方式

刚开始实验的时候，发现不太会使用多线程操作，也就是Python的内置库Threading模块，以及模块里的Thread类函数。所以就进行了一段在线学习，在网络上寻找各种关于Threading内置库的使用方法和教程进行学习和研究。以及一开始不太了解Map-Reduce的意思，也就是map的功能和reduce的功能，经过翻阅课上老师讲的内容以及网络上搜索，渐渐地明白了map是映射，reduce是归约，二者结合起来实现大数据并行处理和分布式文件系统管理。

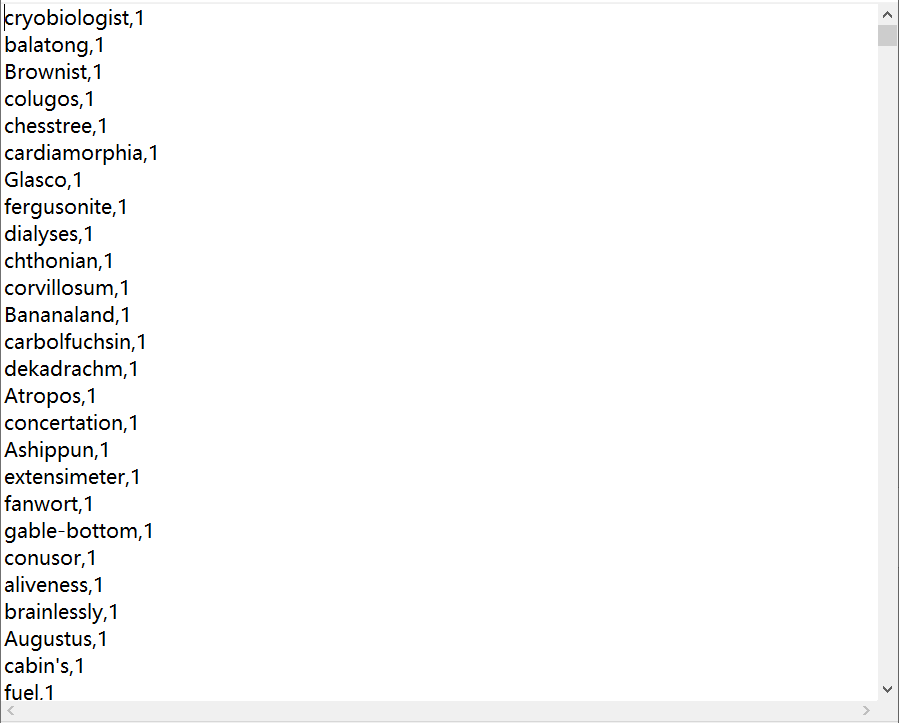
### 1.3.3 实验测试与结果分析

Test：

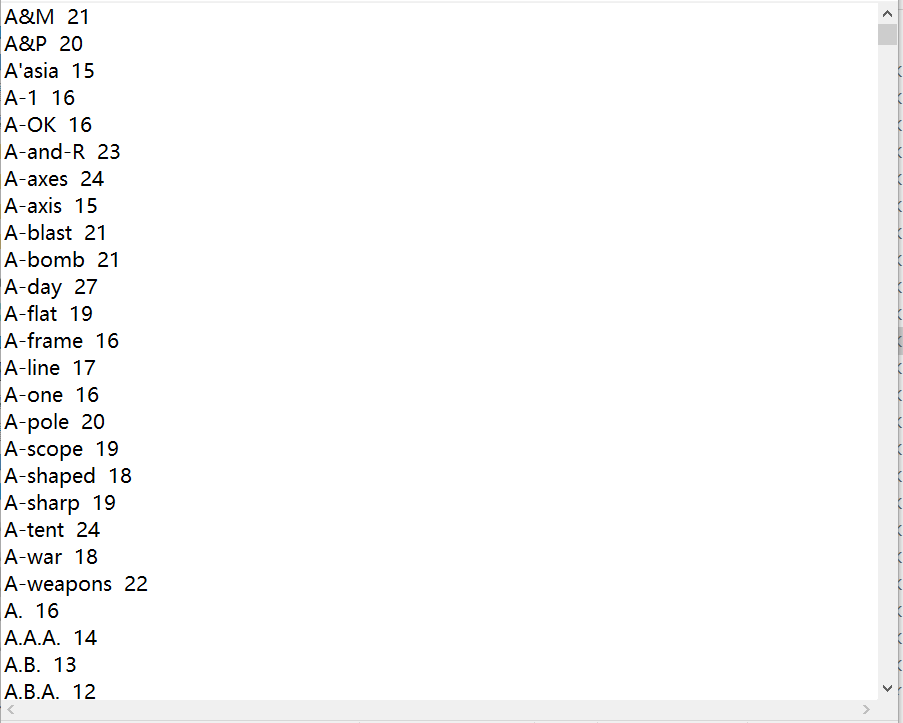
Map文件：以<word, 1>的格式将source文件里的所有单词写入map目标文件中，输出结果如下图所示：



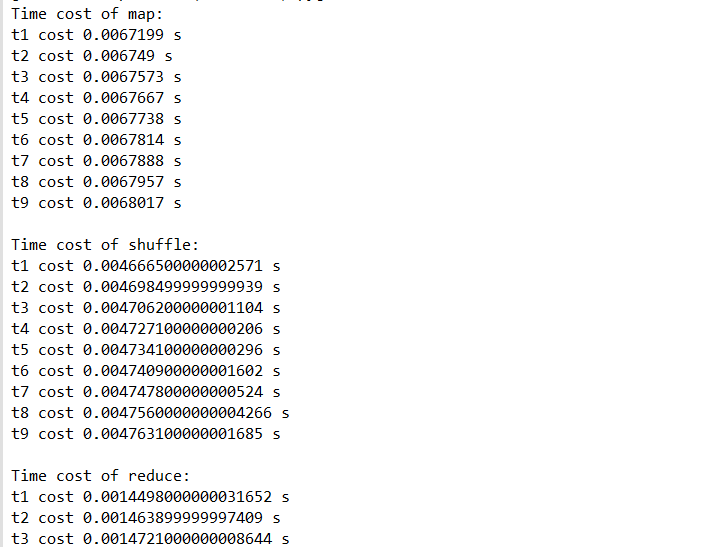
Shuffle文件：对每个map文件内的单词进行单词首字母的分类，最后分类成3个shuffle目标文件，文件内归类结果如下图所示：



Reduce文件：将shuffle文件内所有单词进行提取和统计，以<word, count>格式化输出每个单词在文件内重复出现的次数，并按单词首字母顺序写入reduce目标文件内，输出结果如下：



每一部分功能运行的时间计算如下图：



## **1.4 实验总结**

通过大数据分析的第一次实验“实现Map-Reduce”的过程，让我对分布式系统的运作以及大数据的分析和处理有了更深层次的理解和认知。在完成实验的过程中，对map映射功能和reduce归类功能（wordCount功能）的实现原理有了更深的了解。同时还学会了利用Python内的Threading内置库进行多线程操作，从而提升了整个程序的运行效率。同时从此次实验加深了对Hadoop的了解，其最底部是HDFS，它存储了Hadoop集群中所有存储节点上的文件。而HDFS的上层则是MapReduce，也就是本次实验模拟实现的目标。HDFS和MapReduce是分布式文件系统的核心处理过程，所以此次实验让我学习到了MapReduce的工作原理。