

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级： CS1908**

**学 号： U201915160**

**姓 名： 陈千鹤**

**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](#_Toc89456270)

[**1.1实验目的** 1](#_Toc89456271)

[**1.2 实验内容** 1](#_Toc89456272)

[**1.3 实验过程** 1](#_Toc89456273)

[1.3.1 编程思路 1](#_Toc89456274)

[1.3.2 遇到的问题及解决方式 2](#_Toc89456275)

[1.3.3 实验测试与结果分析 2](#_Toc89456276)

[**1.4 实验总结** 3](#_Toc89456277)

# 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

1. 首先实现Mapper函数：读取文件，并处理每一行数据，将单词拆分出来，并存储为(word, 1)的形式

2. 然后实现Reducer函数：读取(word, k)形式的数据，并将相同词频合并，形成(word, sum)形式数据输出

3. 实现combine函数：通过mapper输入的结果，将相同词的词频加和，输出(word, k)形式数据

4. 实现shuffle过程：将mapper输出结果根据word进行排序，并且在输入reducer函数时等分数据，使得每个reduce节点的工作量相当

5. 实现多线程：Mapper函数用9个线程并行读取文件，Reducer函数用3个现成处理数据加和

6. 汇总输出：利用Reducer函数输出的结果，汇总每个单词的总词频并输出

### 1.3.2 遇到的问题及解决方式

1. 不会多线程。

解决方案：通过网络搜索了解Python多线程的实现方式，学习并使用多线程

### 1.3.3 实验测试与结果分析

1. Mapper函数部分输出结果如图1所示。

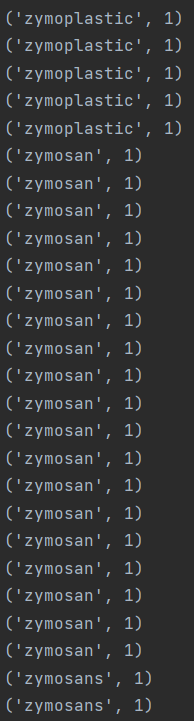


图1 Mapper函数部分输出结果

2. Reducer函数输出部分结果如图2所示。

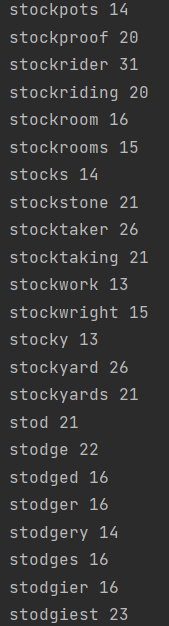


图2 Reducer函数部分输出结果

## **1.4 实验总结**

1. 理解了map-reduce算法思想以及实现流程

2. 学习了多线程实现方法。

3. 学习了combine与shuffle过程的思想，并予以实现

4. 亲自实现了map-reduce算法

5. 学会了应用map-reduce思想解决wordCount问题