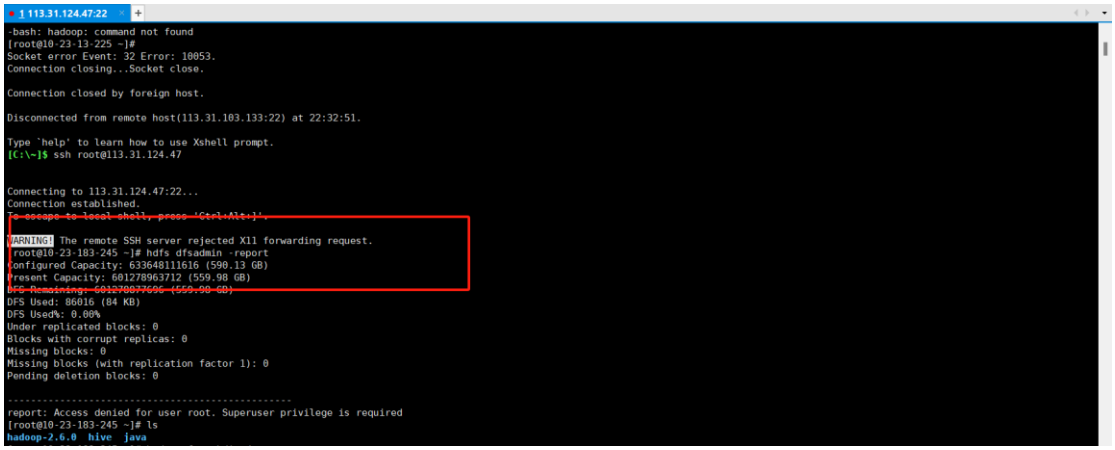
云计算实验三报告

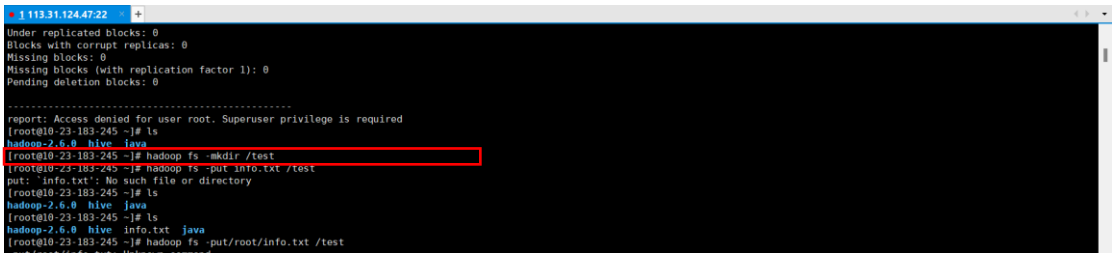
--10215501435 杨茜雅

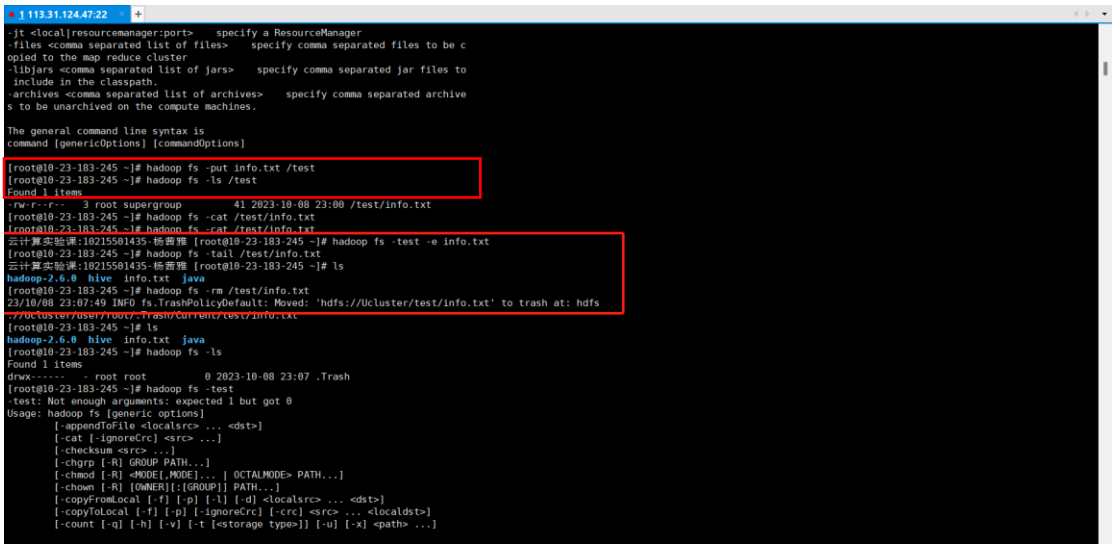
作业 1:



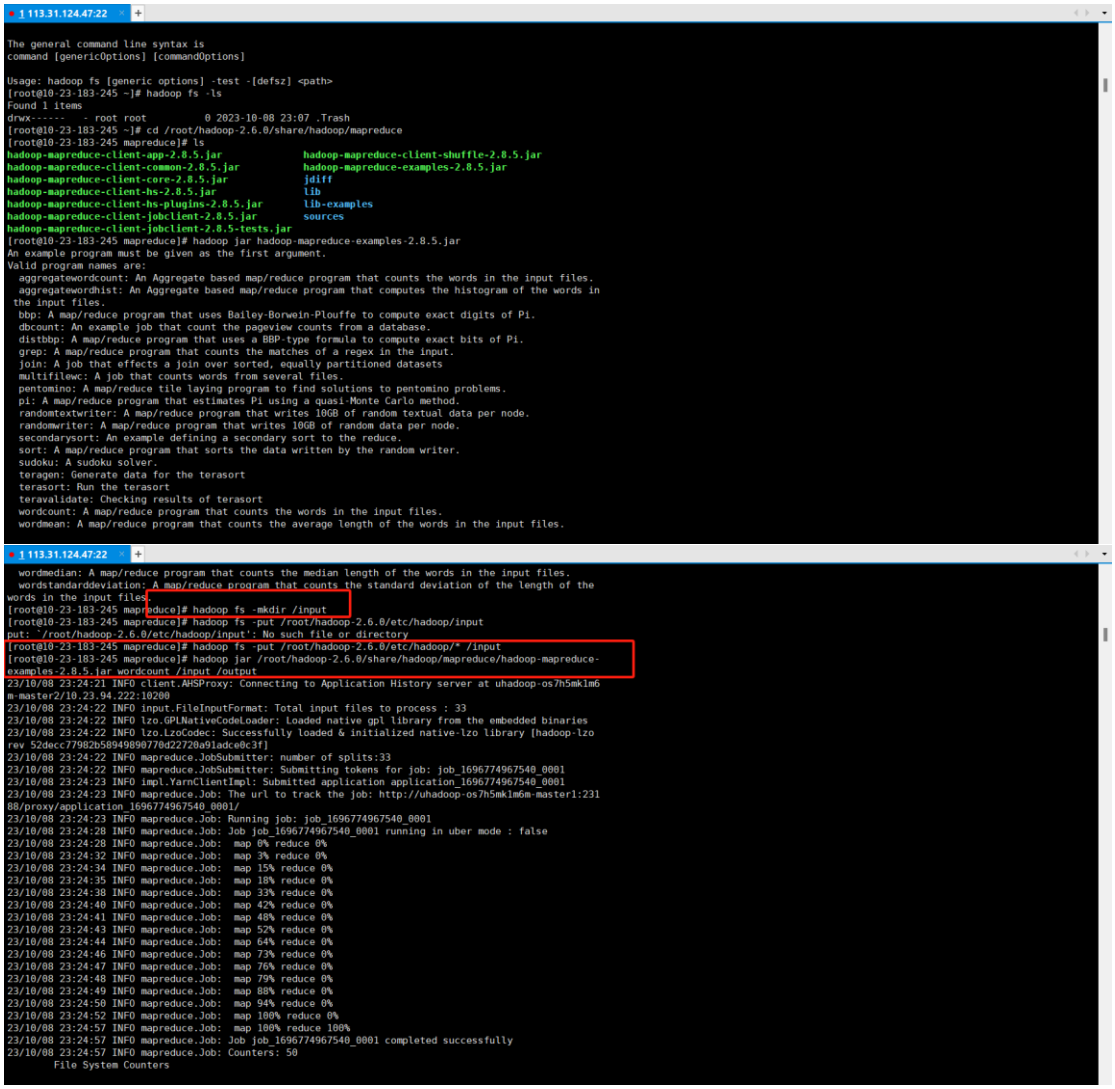
作业 2:

创建文件夹 test，上传 info.txt， 使用 mkdir 和 put，查看内容，删除

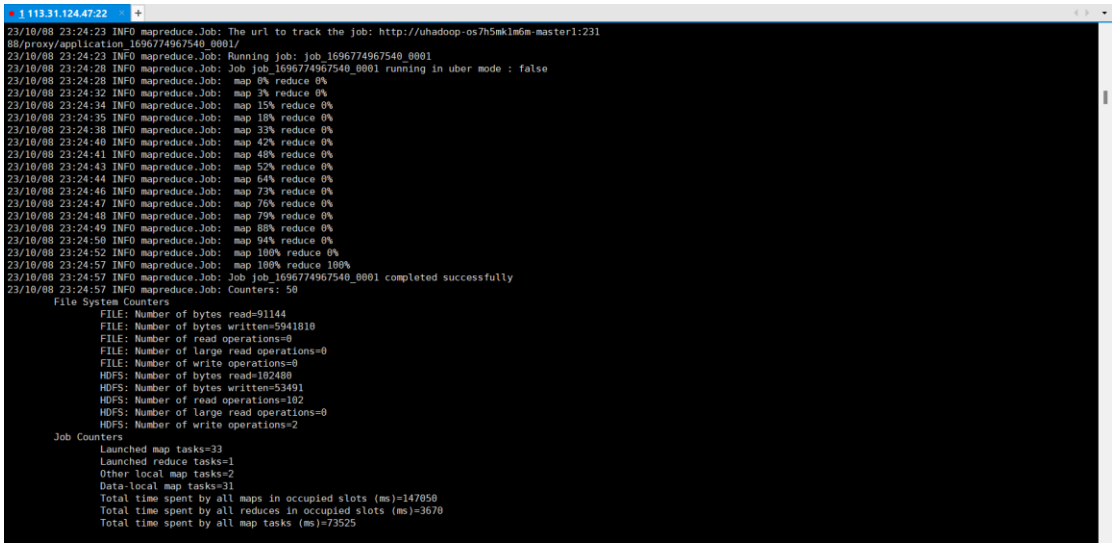


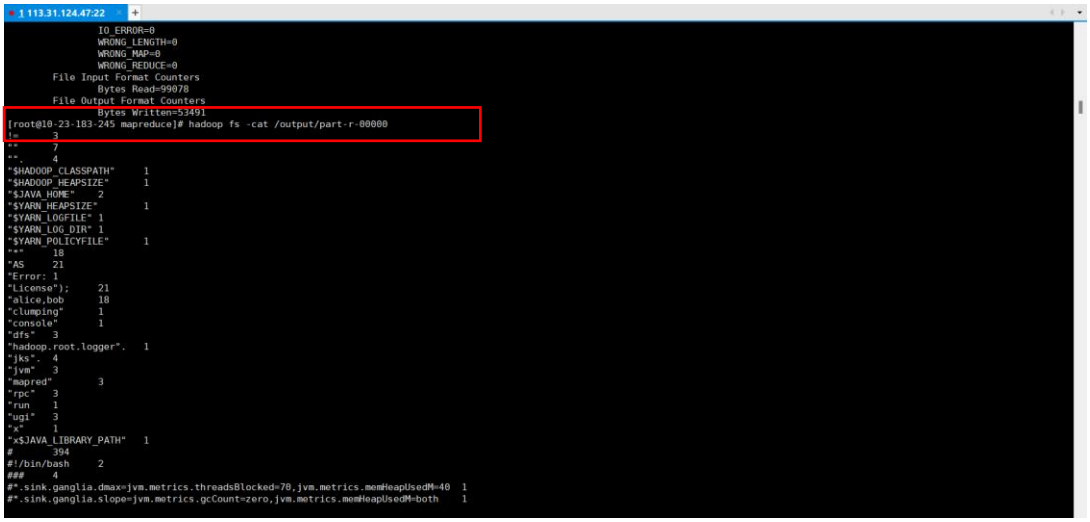


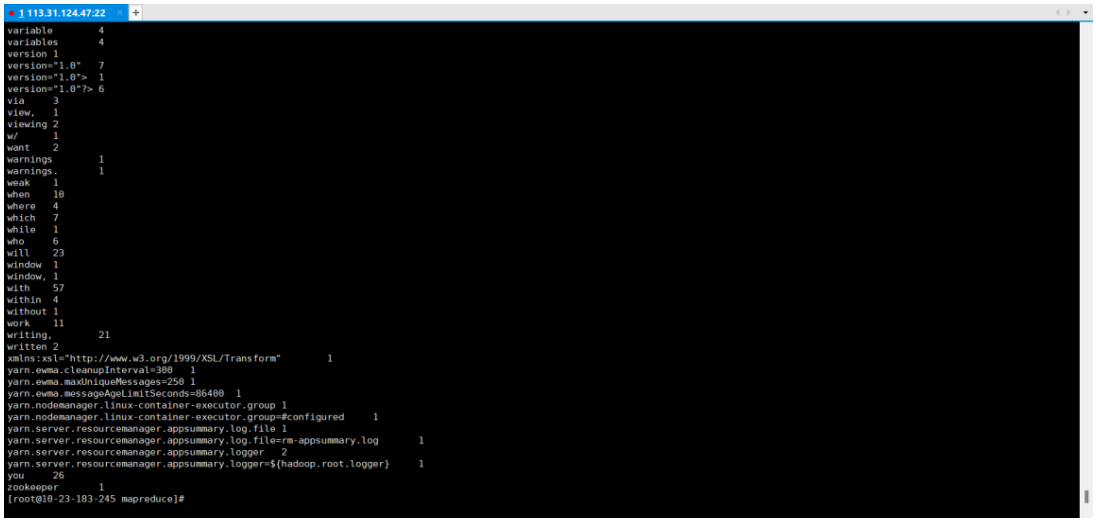
作业 3:



以下是部分输出截图，因为太长了



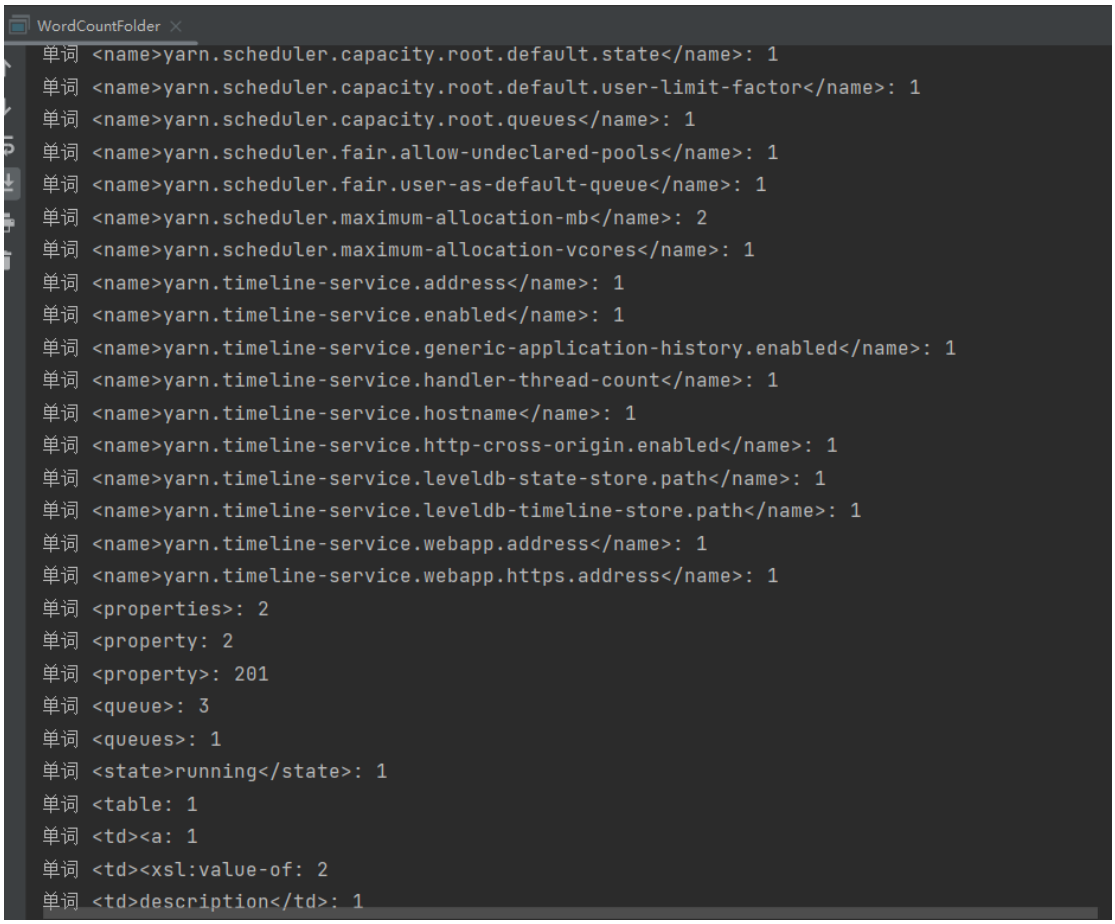


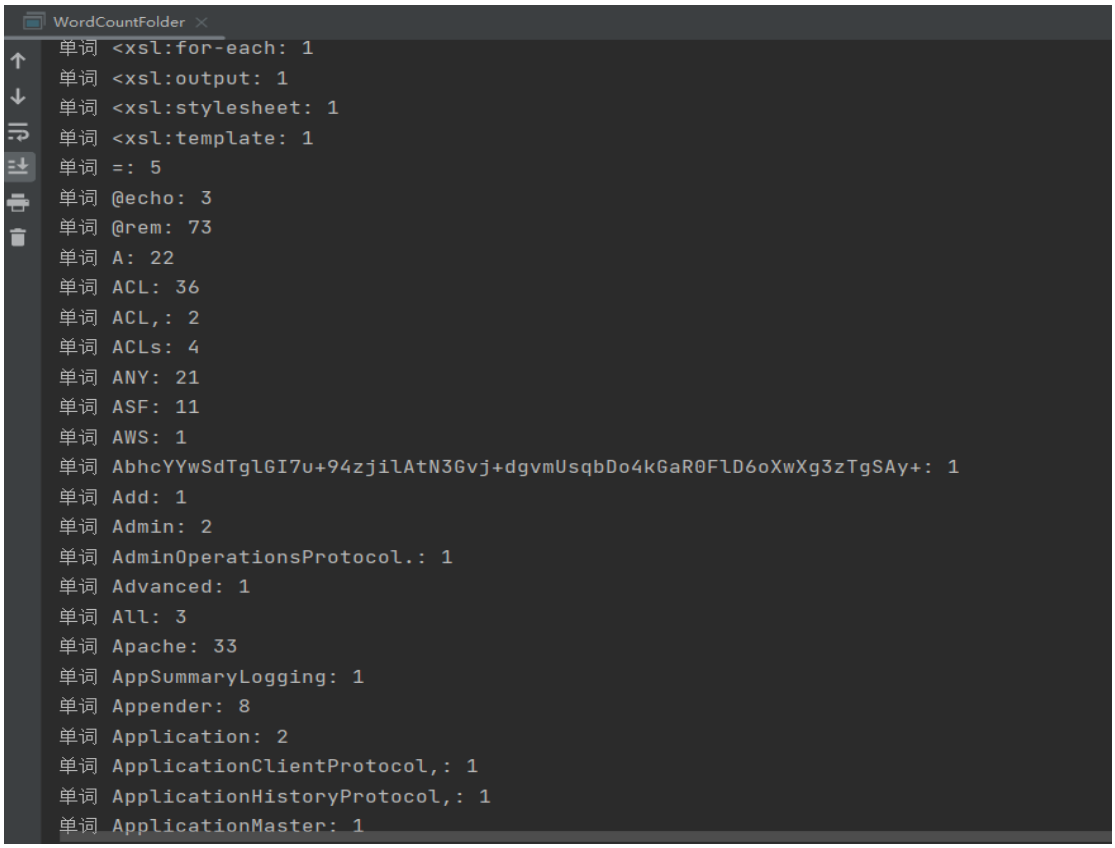


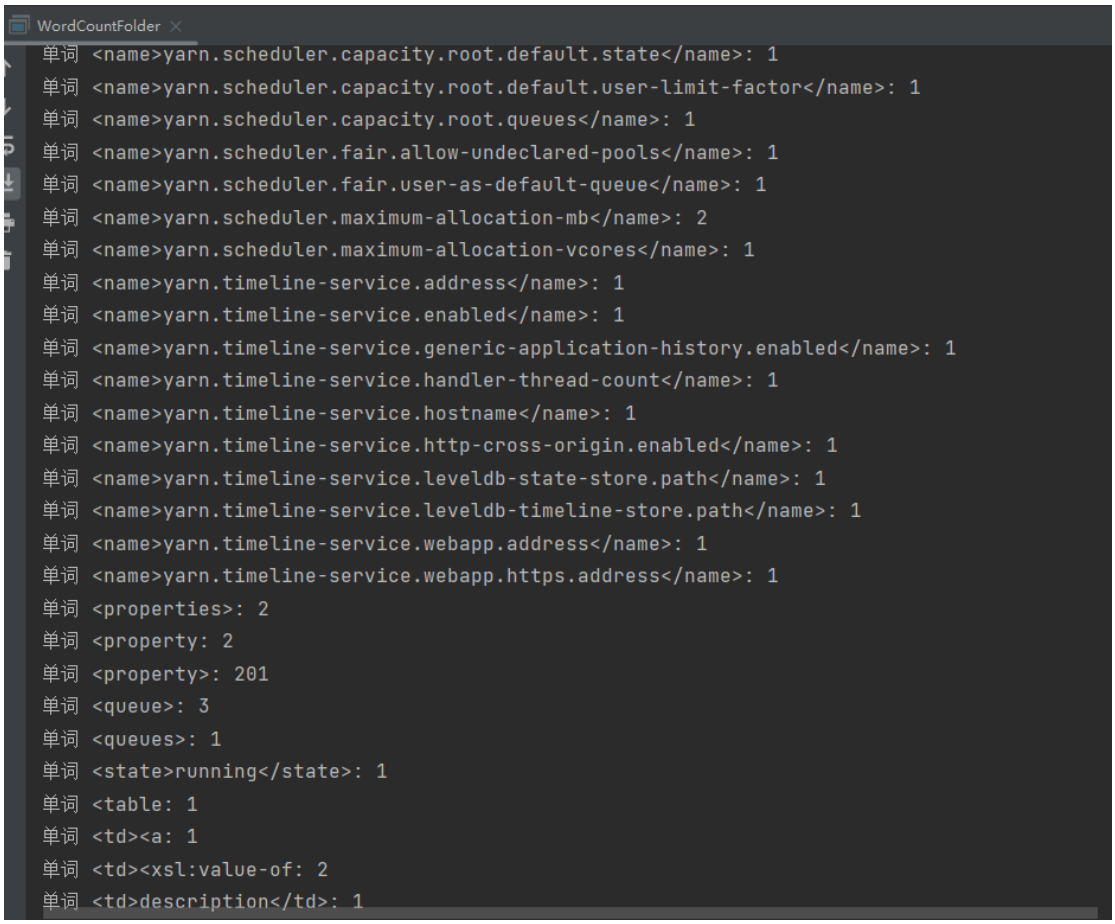
作业 4：实现单线程的 WordCount，记录运行时间

```java
package wc;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

public class WordCountFolder {
    public static void main(String[] args) {
        long start = System.currentTimeMillis();
        Map<String, Integer> wordMap = new TreeMap<>();
        File folder = new File( pathname: "E:\\桌面\\mapreduce\\hadoop"); // 设置文件夹路径
        File[] files = folder.listFiles();
        if (files != null) {
            for (File file : files) {
                if (file.isFile()) {
                    processFile(file, wordMap);
                }
            }
        }
        long end = System.currentTimeMillis();
        Set<String> words = wordMap.keySet();
        Iterator<String> iterator = words.iterator();
        while (iterator.hasNext()) {
            String word = iterator.next();
            Integer count = wordMap.get(word);
            System.out.println("单词 " + word + ": " + count);
        }

        System.out.println("程序总用时: " + (end - start) + " 毫秒");
    }
    private static void processFile(File file, Map<String, Integer> wordMap) {
        try (BufferedReader br = new BufferedReader(new FileReader(file))) {
            String tmpStr;
            while ((tmpStr = br.readLine()) != null) {
                String[] split = tmpStr.trim().split( regex: "\\s+");
                for (int i = 0; i < split.length; i++) {
                    String word = split[i];
                    wordMap.put(word, wordMap.getOrDefault(word, defaultValue: 0) + 1);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

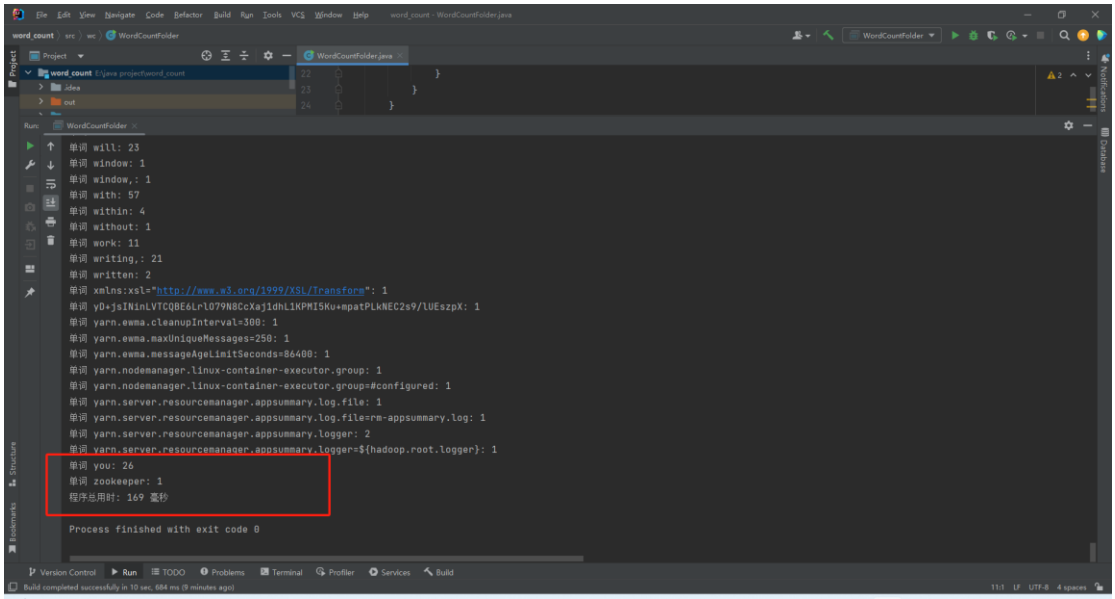
结果输出（截取部分，因为太长了）：



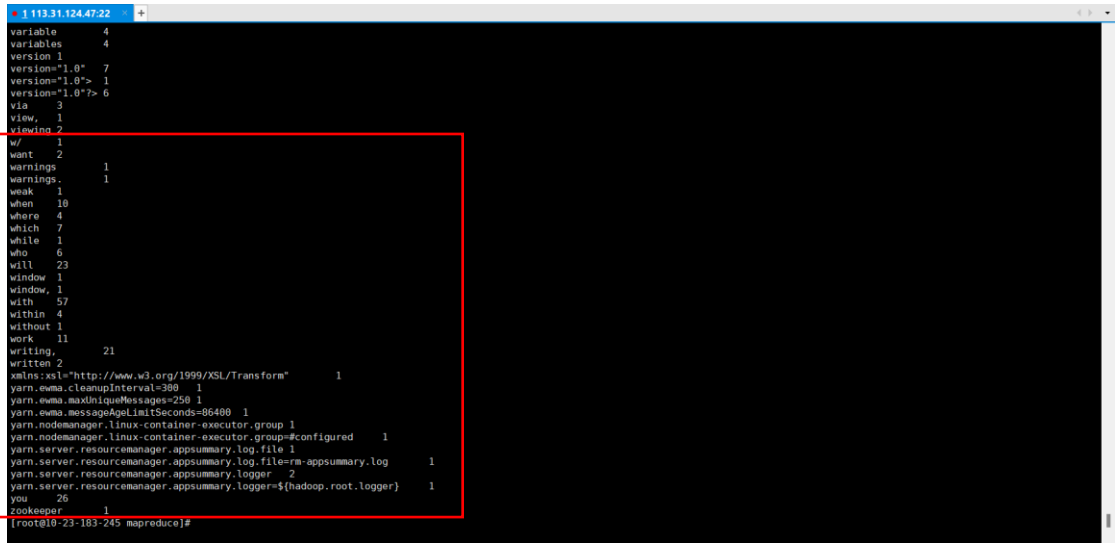




运行时间截图：



并且对比作业 3 中的某一张输出截图，我们可以发现两种实现方式的输出结果是一样的。


1 113.31.124.47:22
variable 4
variables 4
version 1
version="1.0" 7
version="1.0"> 1
version="1.0"?> 6
via 3
view, 1
viewing 2
w/ 1
want 2
warnings 1
warnings. 1
weak 1
when 10
where 4
which 7
while 1
who 6
will 23
window 1
window, 1
with 57
within 4
without 1
work 11
writing, 21
written 2
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" 1
yarn.ewma.cleanupInterval=300 1
yarn.ewma.maxUniqueMessages=250 1
yarn.ewma.messageAgeLimitSeconds=86400 1
yarn.nodemanager.linux-container-executor.group 1
yarn.nodemanager.linux-container-executor.group=#configured 1
yarn.server.resourcemanager.appsummary.log.file 1
yarn.server.resourcemanager.appsummary.log.file=rm-appsummary.log 1
yarn.server.resourcemanager.appsummary.logger 2
yarn.server.resourcemanager.appsummary.logger=${hadoop.root.logger} 1
you 26
zookeeper 1
[root@10-23-183-245 mapreduce]#