

**<<Hadoop-MapReduce**

**测试报告>>**



小组成员：

陈昆度

杨云龙

李嘉艺

马宇晴

北京航空航天大学

2016-05

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V1.0 | 2016/5/26 | 陈昆度 杨云龙 李嘉艺 马宇晴 | 陈昆度 |  |
| V1.01 | 2016/6/25 | 陈昆度 杨云龙 李嘉艺 马宇晴 | 陈昆度 | 修改模糊用语，增加总结 |

目录

[1. 测试概述 4](#_Toc454541624)

[1.1 标识 4](#_Toc454541625)

[1.2 文档说明 4](#_Toc454541626)

[1.3 编写目的 4](#_Toc454541627)

[1.4 参考资料 4](#_Toc454541628)

[2. 测试配置 4](#_Toc454541629)

[2.1 测试软硬件环境配置 4](#_Toc454541630)

[2.2 测试人员分配 5](#_Toc454541631)

[3. 测试结果汇总 5](#_Toc454541632)

[3.1基础模块测试 5](#_Toc454541633)

[3.1.1 linux部署测试用例 5](#_Toc454541634)

[3.1.2 Hadoop部署测试用例 6](#_Toc454541635)

[3.1.3 Maven部署测试用例 7](#_Toc454541636)

[3.2实现模块测试 8](#_Toc454541637)

[3.2.1 Python爬虫测试 8](#_Toc454541638)

[3.2.2预处理数据测试 8](#_Toc454541639)

[3.2.3函数执行测试 9](#_Toc454541640)

[3.2.4输出结果测试 10](#_Toc454541641)

[3.2.5 Inputsplit测试 11](#_Toc454541642)

[3.3非功能需求模块测试 12](#_Toc454541643)

[4. 测试问题汇总 12](#_Toc454541644)

[4.1测试问题 12](#_Toc454541645)

[4.2解决方案 13](#_Toc454541646)

[4.3测试总结 13](#_Toc454541647)

[5. 附录：测试程序 14](#_Toc454541648)

# 测试概述

## 标识

Hadoop版本号：Hadoop\_2.6.0

## 文档说明

本文档用于针对本组实验项目进行的相关软件测试进行报告，内容主要包括

1. 软件测试用例步骤以及测试步骤
2. 测试过程中出现的问题
3. 对测试用例说明分析
4. 对于测试出现的问题，解决的方案。

## 编写目的

本文档是Hadoop-mapreduce测试文档。目的在于总结测试阶段的测试情况以及分析测试结果，测试系统是否符合需求文档，是否已经达到预期的功能目标和非功能目标，并对测试质量进行分析评估。

## 参考资料

1.《hadoop需求规格说明书》

2.《hadoop测试需求说明书 v1.02》

3.Tom White. Hadoop权威指南中文版[M].北京：清华大学出版社，2009.166-181.

4.张良均. Hadoop大数据分析与挖掘实战[M].北京：机械工业出版社,2016.42-56.

5.翟周伟.Hadoop核心技术[M]. 北京：机械工业出版社,2015.143-156.

# 测试配置

## 测试软硬件环境配置

|  |  |
| --- | --- |
| 需求名称 | 详细要求 |
| 操作系统 | Linux（Ubuntu 14.04桌面版） |
| 运行平台 | Hadoop2.6.0平台，java1.7.0 |
| CPU | 四核英特尔处理器 |
| 内存 | 16G |
| 硬盘 | 500G |
| 计算机 | 两台台式机，且都配置好Hadoop |

## 测试人员分配

软件测试人员配置如表-2所示。

表2 测试人员配置表

|  |  |  |
| --- | --- | --- |
| 成员 | 主要职责 | 测试用例编号 |
| 马宇晴 | 负责环境的配置 |  |
| 陈昆度 | 负责审核及校对结果 |  |
| 李嘉艺 | 测试1-3 | TC001 TC002 TC003 |
| 杨云龙 | 测试3-9 | TC004 - TC009 |

# 测试结果汇总

## 3.1基础模块测试

### 3.1.1 linux部署测试用例

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | linux部署测试 | | | | **用例标识** | TC-001 | |
| **测试追踪** | | | 测试规格说明书2.1 | | | | | | |
| **用例说明** | | 测试Linux环境搭建是否成功 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 使用VMware打开安装好的linux虚拟机，进入Linux环境 | | | 成功进入Ubuntu14.04 | | 期望结果与实际结果一致 | | |  |
| 2 | 通过命令行命令测试Linux基础环境是否配置完毕，如JVM配置 | | | Linux基础环境配置完毕，java 1.7.0配置完成 | | 期望结果与实际结果一致 | | |  |
| **前提和约束** | | | 无 | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 李嘉艺 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

### Hadoop部署测试用例

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | Hadoop部署测试 | | | | **用例标识** | TC-002 | |
| **测试追踪** | | | 测试规格说明书2.2 | | | | | | |
| **用例说明** | | 测试Hadoop环境搭建是否成功 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 进入Hadoop安装目录，执行./sbin/start-dfs.sh | | | Hadoop成功运行 | | 期望结果与实际结果一致 | | |  |
| 2 | 进入Hadoop安装目录，执行./bin/hdfs dfs –mkdir input | | | Hadoop成功在分布式文件系统Hdfs新建目录input | | 期望结果与实际结果一致 | | |  |
| 3 | 进入Hadoop安装目录，执行./bin/hdfs dfs –put file input | | | Hadoop成功在分布式文件系统Hdfs上传文件input | | 期望结果与实际结果一致 | | |  |
| 4 | 在配置好的Eclipse中新建Map-reudce模块测试程序test.java | | | 程序成功运行，输出文件计数结果 | | 期望结果与实际结果一致 | | |  |
| **前提和约束** | | | 成功配置好Linux环境以及Hadoop环境 | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 李嘉艺 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

### Maven部署测试用例

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | Maven部署测试 | | | | **用例标识** | TC-003 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 测试Maven环境搭建是否成功 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 在命令行输入mvn -v | | | 成功输出安装Maven版本3.3.9 | | 期望结果与实际结果一致 | | |  |
| 2 | 打开hadoop2.6.0源码，输入mvn package -Pdist –DskipTests -Dtar | | | 成功编译完成hadoop源码，输出Success结果 | | 期望结果与实际结果一致 | | |  |
| **前提和约束** | | | 成功配置Linux环境、maven环境、安装好编译hadoop源码的依赖 | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 李嘉艺 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

## 3.2实现模块测试

### 3.2.1 Python爬虫测试

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | Python爬虫测试 | | | | **用例标识** | TC-004 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 测试爬取数据是否正确。 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 用爬虫程序爬取对应url的网站数据 | | | 程序运行正常，不会出现超时或者网址错误的情况。 | | 期望结果与实际结果一致 | | |  |
| 2 | 查看从开始爬取，到爬取结束程序的运行是否正常 | | | 抽取的数据正常，按照预期的格式进行了整理。 | | 期望结果与实际结果一致 | | |  |
| **前提和约束** | | | 拥有可以获取的网站信息 | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 杨云龙 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

### 3.2.2预处理数据测试

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | 预处理数据测试 | | | | **用例标识** | TC-005 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 检验数据是否完全经过格式化处理 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 将预处理后的数据输出到txt文件 | | | 正常的提取了数据，并且保存到了文件中 | | 期望结果与实际结果一致 | | |  |
| 2 | 打开txt文件，人工查看预处理后的数据是否按要求格式化 | | | 完全按照要求进行了格式化 | | 期望结果与实际结果一致 | | |  |
| 3 | 编写简单测试程序，产看预处理后的数据是否按照要求格式化 | | | 没有发现任何错误 | | 期望结果与实际结果一致 | | |  |
| **前提和约束** | | | 拥有原始数据文件 | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 杨云龙 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

### 3.2.3函数执行测试

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | 函数执行测试 | | | | **用例标识** | TC-006 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 1. 测试Map-Reduce函数运行是否正确  2. 测试性价比计算和排序是否正确 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 在Hadoop环境下运行编写的Map-Reduce程序 | | | 能够正常运行，没有出现异常数据或者报错。 | | 期望结果与实际结果一致 | | |  |
| 2 | 在输出性价比计算的结果 | | | 能够正常输出到txt文件中 | | 期望结果与实际结果一致 | | |  |
| 3 | 输出按照性价比进行排序的结果，检查结果的正确性 | | | 没有发现任何错误 | | 期望结果与实际结果一致 | | |  |
| **前提和约束** | | | 拥有预处理后的数据文件 | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 杨云龙 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

### 3.2.4输出结果测试

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | 输出结果测试 | | | | **用例标识** | TC-007 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 检查输出结果的正确性 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 将Map-Reduce函数计算和排序的结果输出到txt文件 | | | 能够正常输出，没有数据不同的情况 | | 期望结果与实际结果一致 | | |  |
| 2 | 从排序结果中抽取排名靠前的几组计算机的性价比，和爬虫程序得到原始数据的价格进行比价，检查价格是否合理 | | | 没有不合理的排序 | | 期望结果与实际结果一致 | | |  |
| 3 | 更换价格区间，运行Map-Reduce函数，重新得到计算和排序的结果，输出到txt文件，重复第二步。 | | | 没有发现任何错误 | | 在5000-7000价格段出现了一个异常数据 | | |  |
| **前提和约束** | | | 拥有预处理后的数据文件 | | | | | | |
| **结果评价标准** | | | 测试不通过 | | | | | | |
| **设计人员** | | | 杨云龙 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

### 3.2.5 Inputsplit测试

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | Inputsplit测试 | | | | **用例标识** | TC-008 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 检查改进的函数是否可以正常使用 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 将源代码文件打包 | | | 能够将源文件代码在没有错误的情况下进行打包 | | 期望结果与实际结果一致 | | |  |
| 2 | 使用maven对源代码文件进行编译。 | | | 可以成功编译 | | 编译时出现错误，出现同名变量i | | |  |
| 3 | 运行编译好的环境 | | | 没有发现任何错误 | |  | | |  |
| **前提和约束** | | | 拥有Linux环境和Maven | | | | | | |
| **结果评价标准** | | | 测试不通过 | | | | | | |
| **设计人员** | | | 陈昆度 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

## 3.3非功能需求模块测试

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **用例名称** | | | 性能测试 | | | | **用例标识** | TC-009 | |
| **测试追踪** | | | 测试规格说明书2.3 | | | | | | |
| **用例说明** | | 检查改进后的Map-Reduce运行同样数据，速度是否变快 | | | | | | | |
| **操作过程** | | | | | | | | | |
| **序号** | **输入及操作说明** | | | **期望的测试结果** | | **评价标准** | | | **备注** |
| 1 | 未改进前，运行程序，检测程序的运行时间 | | | 能够正常运行 | | 期望结果与实际结果一致 | | |  |
| 2 | 改进后，在同样的数据集下，运行程序，检测程序的运行时间 | | | 速度相比以前更快了，之前的速度是28s，改进之后的速度是17s。 | | 期望结果与实际结果一致 | | |  |
| 3 | 运行编译好的环境 | | | 没有发现任何错误 | |  | | |  |
| **前提和约束** | | | 拥有Linux环境和Maven | | | | | | |
| **结果评价标准** | | | 测试通过 | | | | | | |
| **设计人员** | | | 陈昆度 | | **执行人员** | |  | | |
| **检测人** | | |  | | **执行日期** | |  | | |

# 测试问题汇总

## 4.1测试问题

主要的测试问题出现有两个。

1. 出现了异常的数据

异常数据的位置于测试用例3.2.4输出结果测试的第三步。

1. 在编译的时候变量出现了错误。

编译错误位于3.2.5 Inputsplit测试第二步

## 4.2解决方案

1.异常数据的原因是因为原始数据的格式与其他数据之间发生了冲突，其中标签“联想拯救者 i7”一项的格式与其他标签的格式有所不同，硬盘的描述没有区分机械和SSD。我们对原始数据做出了修正，成功的解决了这个问题。

2.对变量的名称进行了修改，其中因为变量i之前在源代码中已经定义并且使用过了，我们把变量名换成k1，成功解决了这个问题。

## 4.3测试总结

总的来说测试结果比较理想，其中关键性的功能都如期运行。其中出现的两个问题，包括一个数据错误和一个编译错误，都迅速找到了原因并且成功解决。总而言之两个错误在软件制作阶段时就有出现过，因此我们能够很快的找到发生的原因。

总的测试时间大概为4个小时，其中maven编译和数据核对花费了较长的时间，其他测试内容都很快得到了结果。

我们的测试用例涵盖了所有的需求用例，因此测试得到的结果能够反映出我们的软件是可靠而且快速的。

# 附录：测试程序

测试程序：test.java

package org.apache.hadoop.examples;

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

public static class TokenizerMapper

extends Mapper<Object, Text, Text, IntWritable>{

private final static IntWritable one = new IntWritable(1);

private Text word = new Text();

public void map(Object key, Text value, Context context

) throws IOException, InterruptedException {

StringTokenizer itr = new StringTokenizer(value.toString());

while (itr.hasMoreTokens()) {

word.set(itr.nextToken());

context.write(word, one);

}

}

}

public static class IntSumReducer

extends Reducer<Text,IntWritable,Text,IntWritable> {

private IntWritable result = new IntWritable();

public void reduce(Text key, Iterable<IntWritable> values,

Context context

) throws IOException, InterruptedException {

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

result.set(sum);

context.write(key, result);

}

}

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();

if (otherArgs.length != 2) {

System.err.println("Usage: wordcount <in> <out>");

System.exit(2);

}

Job job = new Job(conf, "word count");

job.setJarByClass(WordCount.class);

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(otherArgs[0]));

FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}