

**<<Hadoop-MapReduce**

**测试需求说明书>>**



小组成员：

陈昆度

杨云龙

李嘉艺

马宇晴

北京航空航天大学

2016-06

**版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 | 变更内容 |
| 1.0 | 2016/05/09 | 杨云龙 | 陈昆度，李嘉艺，马宇晴 |  |
| 1.01 | 2016/05/16 | 陈昆度 | 马宇晴，李嘉艺，杨云龙 | 增加数据词典，参考文献，RUCM的更正还有测试方法的详细化，同时加入工时的估算。 |
| 1.02 | 2016/05/25 | 陈昆度 | 马宇晴，李嘉艺，杨云龙 | RUCM补充和修正。 |
| 1.03 | 2016/06/25 | 陈昆度 | 马宇晴，李嘉艺，杨云龙 | 修改RUCM，增加总结报告 |

目录

[1 前言 5](#_Toc454540903)

[1.1 目的 5](#_Toc454540904)

[1.2 项目背景 5](#_Toc454540905)

[1.3 文档概述 6](#_Toc454540906)

[1.4 测试用例与需求用例参 照表 6](#_Toc454540907)

[1.5术语和缩略词 7](#_Toc454540908)

[1.6软件测试环境 7](#_Toc454540909)

[1.7参考文件 7](#_Toc454540910)

[2基本需求模块 8](#_Toc454540911)

[2.1操作系统测试 8](#_Toc454540912)

[2.1.1测试策略描述 8](#_Toc454540913)

[2.1.2测试用例 8](#_Toc454540914)

[2.1.3测试用例所需工时 9](#_Toc454540915)

[2.2 环境测试 10](#_Toc454540916)

[2.2.1测试策略描述 10](#_Toc454540917)

[2.2.2测试用例 10](#_Toc454540918)

[2.2.3测试用例所需工时 11](#_Toc454540919)

**[2.3 编译测试](#_Toc454540920)** [11](#_Toc454540920)

**[2.3.1测试策略描述](#_Toc454540921)** [11](#_Toc454540921)

**[3.3.2测试用例](#_Toc454540922)** [11](#_Toc454540922)

**[2.3.3测试用例所需工时](#_Toc454540923)** [12](#_Toc454540923)

[3实现需求模块 12](#_Toc454540924)

[3.1 Python爬虫测试 12](#_Toc454540925)

[3.1.1测试策略描述 12](#_Toc454540926)

[3.1.2测试用例 13](#_Toc454540927)

[3.1.3测试用例所需工时 14](#_Toc454540928)

[3.2数据预处理测试 14](#_Toc454540929)

[3.2.1测试策略描述 14](#_Toc454540930)

[3.2.2测试用例 14](#_Toc454540931)

[3.2.3测试用例所需工时 15](#_Toc454540932)

[3.3函数执行测试 15](#_Toc454540933)

[3.3.1测试策略描述 15](#_Toc454540934)

[3.3.2测试用例 15](#_Toc454540935)

[3.3.3测试用例所需工时 16](#_Toc454540936)

[3.4输出结果测试 17](#_Toc454540937)

[3.4.1测试策略描述 17](#_Toc454540938)

[3.4.2测试用例 17](#_Toc454540939)

[3.4.3测试用例所需工时 18](#_Toc454540940)

[3.5 Inputsplit测试 18](#_Toc454540941)

[3.5.1测试策略描述 18](#_Toc454540942)

[3.5.2测试用例 18](#_Toc454540943)

[3.5.3测试用例所需工时 19](#_Toc454540944)

[4非功能需求模块 20](#_Toc454540945)

[4.1运行速度测试 20](#_Toc454540946)

[4.1.1测试策略描述 20](#_Toc454540947)

[4.1.2测试用例 20](#_Toc454540948)

[4.1.3测试用例所需工时 21](#_Toc454540949)

[5.测试方案自我评价 21](#_Toc454540950)

# 前言

## 目的

本测试文档目的在于详细的介绍我们的用例以及测试方法。主要的测试内容，就是在Hadoop-MapReduce框架的下，编写的处理计算机数据的程序的准确性和处理数据的速度。我们的测试过程将主要针对两个大方面，排序和性能改进。

其中，功能实现部分包括InputSplit模块，数据抓取模块，Map-Reduce模块，价格排序模块和输出模块。而非功能需要则是对运行速度的要求。

我们预计通过测试达到对功能实现，工作速度，代码可用性的检查，根据检查结果我们将对这些模块进行改进。

在测试结束之后，我们将会得到以下可交付资料。

1. 针对不同用例测试得到的数据表格。
2. 每个用例模块测试成功与否的报告。
3. 测试不通过的模块的改进结果，以及重新测试的报告。
4. 总结报告。

## 项目背景

MapReduce是一种编程模型，用于大规模数据集（大于1TB）的并行运算。"Map（映射）"和"Reduce（归约）"，是它们的主要思想，都是从函数式编程语言里借来的，还有从矢量编程语言里借来的特性。它极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上。当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（归约）函数，用来保证所有映射的键值对中的每一个共享相同的键组。

Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

## 文档概述

本次测试文档主要内容包括：

1. 完成功能
2. 测试用例
3. 交付内容

## 测试用例与需求用例参 照表

本次测试严格遵照《需求规格说明书》以及项目实现内容来进行设计与实现，测试用例与需求用例的对照表如下：

|  |  |  |
| --- | --- | --- |
| 模块 | 需求用例 | 测试用例 |
| 基本需求 | Hadoop环境部署 | 操作系统测试 |
| Hadoop环境部署/JVM和Maven部署 | 环境测试 |
| JVM和Maven部署 | 编译测试 |
| 实现需求 | Python爬取数据 | Python爬虫测试 |
| 数据预处理 | 预处理数据测试 |
| Map-Reduce函数 | 函数执行测试 |
| 输出结果 | 输出结果测试 |
| 重载InputSplit | Inputsplit测试 |
| 非功能需求 | 运行速度 | 运行速度测试 |

## 1.5术语和缩略词

|  |  |
| --- | --- |
| **缩写、术语及符号** | **解 释** |
| Hadoop | 分布式系统基础架构 |
| MapReduce | 并行计算模型 |
| YARN | 新的 Hadoop 资源管理器 |
| UCM | Use Case Model用例建模 |
| RUCM | 限制性用例模型 |
| Map | 对数据进行分布式的搜索操作 |
| Reduce | 对Map的结果进行汇总和处理 |
| Job | Map和Reduce过程依赖的函数，也可以成为Mapper或者Reducer |
| InputSplite | Hadoop源码中的一个模块，负责将输入的文件分成若干个不同的小块 |
| Maven | 一个基于Linux的集成开发环境 |

## 1.6软件测试环境

|  |  |
| --- | --- |
| 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 |

## 1.7参考文件

1.《hadoop需求规格说明书》

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

3.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.

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

5. Jonathan R. Owens, Jon Lentz, Brian Femiano.Hadoop实战手册[M].北京：人民邮电出版社，2014.87-92.

# 2基本需求模块

## 2.1操作系统测试

## 2.1.1测试策略描述

用例的目的在于测试Linux以及其语言环境是否正确部署。一般来说，正常情况下，系统能够启动并且通过命令找到安装的环境包。我们在这里使用的Linux环境是ubantu-14.0.1

### 2.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 操作系统测试 | |
| Brief Description | 检查Linux以及JVM环境是否正确安装 | |
| Precondition | VMware安装成功，有Linux的安装包 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 运行windows环境下的linux虚拟机 |
| 2 | VAILDATE THAT尝试进入Linux的主界面 |
| 3 | 使用shell命令对Java环境的安装进行测试 VAILDATE THAT 测试Java的版本信息 |
| Postcondition  (Test) | Linux以及Java环境安装成功 |
| Specific Alternative Flows  (Test) | RFS basic flow 3 | |
| 1 | 使用shell时发现有功能安装失败（如JVM）； |
| 2 | ABORT |
| Postcondition  (Test) | Java环境安装失败 |
| Specific Alternative Flows  (Test) | RFS 2 | |
| 1 | 发现无法进入Linux系统 |
| 2 | ABORT |
| Postcondition  (Test) | Linux系统安装失败 |

### 2.1.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | 操作系统测试 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 运行windows环境下的linux虚拟机 | 0.1 |
| 2 | 尝试进入Linux的主界面 | 0.1 |
| 3 | 使用shell命令对Java环境的安装进行测试 | 0.2 |
| 总工时 | 0.4 | | |

## 2.2 环境测试

## 2.2.1测试策略描述

用例的目的是测试Hadoop环境是否搭建成功，环境配置分为配置成功和配置失败两种情况。正常情况下，Hadoop应该能够在分布式计算的情况下成功运行HDFS和MapReduce；任何一者不能够执行则为失败。

### 2.2.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Hdfs和Mapreduce测试 | |
| Brief Description | 测试Hadoop环境是否正常安装 | |
| Precondition | Linux以及Java环境配置完成 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 获取Hadoop的安装包 |
| 2 | 在指定的数台虚拟机中运行Hadoop集群 |
| 3 | 修改Hadoop中的配置文件以符合测试需求（如host）VAILDATE THAT 进行运行测试 |
| 4 | VAILDATE THAT测试Hadoop-MapReduce的example函数 |
| 5 | VAILDATE THAT测试命令start-all.sh |
| Postcondition  (Test) | Hadoop环境配置成功而且完整 |
| Specific Alternative Flows  (Test) | RFS 3,4,5 | |
| 1 | 测试的程序无法正常运行 |
| 2 | ABORT |
| Postcondition  (Test) | Hadoop环境安装和配置失败 |

### 2.2.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | Hadoop部署 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 获取Hadoop的安装包 | 0.2 |
| 2 | 在指定的数台虚拟机中安装Hadoop集群 | 1 |
| 3 | 修改Hadoop中的配置文件以符合测试需求（如host） | 1 |
| 4 | 测试Hadoop-MapReduce的example函数 | 0.3 |
| 5 | 测试命令start-all.sh | 0.1 |
| 总工时 | 2.6 | | |

**2.3 编译测试**

**2.3.1测试策略描述**

用例的目的是测试Maven开发工具是否能够正常工作。正常情况下，Maven能够完成Hadoop源码的编译；失败情况下，编译过程会抛出异常。

**3.3.2测试用例**

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 编译测试 | |
| Brief Description | 测试Maven是否能够编译； | |
| Precondition | 完成Hadoop集群的搭建 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 下载Maven3.3.9； |
| 2 | 在Linux环境下打开Hadoop源码 |
| 3 | VAILDATE THAT测试编译hadoop源码。 |
| Postcondition  (Test) | Maven环境搭建成功，能够成功编译。 |
| Specific Alternative Flows  (Test) | RFS 3 | |
| 1 | Hadoop编译不通过 |
| 2 | ABORT |
| Postcondition  (Test) | Maven环境搭建不完全或者失败，或缺少Hadoop源码安装依赖 |

**2.3.3测试用例所需工时**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | Maven部署 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 下载Maven3.3.9 | 0.2 |
| 2 | 在Linux环境下打开Hadoop源码 | 0.1 |
| 3 | 测试编译hadoop源码。 | 1 |
| 总工时 | 1.3 | | |

# 3实现需求模块

## 3.1 Python爬虫测试

### 3.1.1测试策略描述

本次测试的目的是验证python爬取的数据是否正确而且完整。正常情况下抓取的数据应该按照关键词-属性的格式，如[pic] = ”4333”，进行完整的输出。异常情况下会出现数据的缺省，如[CPU型号] = ””。

### 3.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 数据爬取测试 | |
| Brief Description | 测试爬取数据的正确性 | |
| Precondition | Python3.5安装成功 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 确定需要抓取的网页地址，观察网页是否能正常登陆 |
| 2 | 运行python爬虫文件VAILDATE THAT观察运行情况 |
| 3 | 人工查看数据爬取结果 VAILDATE THAT抽样检查结果。 |
| Postcondition  (Test) | 数据爬取结果正确 |
| Specific Alternative Flows  (Test) | RFS 3 | |
| 1 | 数据发生了缺省或者出现异常标签 |
| 2 | ABORT |
| Postcondition  (Test) | 爬取的数据难以处理，爬取失败 |
| Specific Alternative Flows  (Test) | RFS 2 | |
| 1 | Python运行中出现trackback错误 |
| 2 | ABORT |
| Postcondition  (Test) | 爬虫程序存在没有考虑到的编写问题 |

### 3.1.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | 数据爬取测试 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 确定需要抓取的网页地址 | 0.01 |
| 2 | 运行python爬虫文件 | 0.01 |
| 3 | 人工查看数据爬取结果，并抽样检查 | 1 |
| 总工时 | 1.02 | | |

## 3.2数据预处理测试

### 3.2.1测试策略描述

本测试的目的在于验证预处理模块成功的将原始数据进行了格式化。正常情况下预处理程序应该提取出原始数据中的关键数据（如CPU，硬盘等），异常情况下出现数据缺省或者出现了意外的标签。

### 3.2.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 数据预处理 | |
| Brief Description | 将原始数据中关键的部分提取出来 | |
| Precondition | 原始数据 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 获取原始数据的输出文件 |
| 2 | 运行Java环境下的预处理程序，观察程序运行状况 |
| 3 | VAILDATE THAT 检查输出结果 |
| Postcondition  (Test) | 数据提取成功 |
| Specific Alternative Flows  (Test) | RFS 3 | |
| 1 | 数据发生了缺省或者出现异常标签 |
| 2 | ABORT |
| Postcondition  (Test) | 提取函数出现问题，提取失败 |

### 3.2.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | 数据预处理 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 获取原始数据的输出文件 | 0.1 |
| 2 | 运行Java环境下的预处理程序 | 0.2 |
| 3 | 查看输出结果 | 0.2 |
| 总工时 | 0.5 | | |

## 3.3函数执行测试

## 3.3.1测试策略描述

本测试的目的在于将预处理程序分布到集群中进行分散，并且将数据保存到key-value的结构中，同时在Reduce过程中进行性价比计算并且排序。正常情况下可以成功传递给reduce函数，并且得到排序结果。异常情况下将会编译不通过。

### 3.3.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Map-Reduce函数 | |
| Brief Description | 测试Map-Reduce的传递，性价比计算和排序 | |
| Precondition | 预处理数据 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 读取与预处理后的数据 |
| 2 | 将数据导入到Map函数中进行变换 VAILDATE THAT 运行测试 |
| 3 | 将key-value值传入Reducer |
| 4 | 将数据按照规则进行性价比计算 |
| 5 | 利用性价比计算结果进行排序 |
| 6 | 输出排序结果 |
| Postcondition  (Test) | 函数成功执行，得到输出文件 |
| Specific Alternative Flows  (Test) | RFS 2 | |
| 1 | 编译不通过 |
| 2 | ABORT |
| Postcondition | MapReduce函数语法错误 |

### 3.3.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | Map-Reduce函数 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 读取预处理后的数据 | 0.1 |
| 2 | 将数据导入到Map函数中进行变换 |
| 3 | 将key-value值传入Reducer |
| 4 | 将数据按照规则进行性价比计算 |
| 5 | 利用性价比计算结果进行排序 |
| 6 | 输出排序结果 |
| 总工时 | 0.1 | | |

## 3.4输出结果测试

### 3.4.1测试策略描述

本测试的目的在于检查输出结果的正确性，正常情况下数据应该按照真实的笔记本性价比进行排序，异常情况下将出现排名错误，比如说性价比更低的笔记本电脑反而排名靠前。

### 3.4.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 输出结果 | |
| Brief Description | 测试输出结果的正确性 | |
| Precondition | MapReduce函数成功执行 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 准备测试结果 |
| 2 | VAILDATE THAT 与真实数据进行比较 |
| 3 | VAILDATE THAT 更换价格区间，继续进行比较 |
| Postcondition  (Test) | 数据正确 |
| Specific Alternative Flows  (Test) | RFS 2,3 | |
| 1 | 数据有明显的错误 |
| 2 | ABORT |
| Postcondition  (Test) | Map-Reduce函数功能性错误 |

### 3.4.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | 输出结果 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 准备测试结果 | 0.01 |
| 2 | 与真实数据进行比较 | 0.2 |
| 总工时 | 0.21 | | |

## 3.5 Inputsplit测试

### 3.5.1测试策略描述

本测试的目的在于对改进后的MapReduce函数进行测试，正常情况下能够编译通过。

### 3.5.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 源码改进 | |
| Brief Description | 对Hadoop的源代码改变进行测试 | |
| Precondition | Maven成功配置 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 重载文件分割函数InputSplit- |
| 2 | VAILDATE THAT 编译Hadoop-MapReduce |
| 3 | 运行一个Hadoop的示例（比如本程序） VAILDATE THAT 观察是否正常运行。 |
| Postcondition  (Test) | Hadoop成功编译 |
| Specific Alternative Flows  (Test) | RFS 2 | |
| 1 | 编译Hadoop的过程中发现错误 |
| 2 | ABORT |
| Postcondition  (Test) | 函数出现错误或者冲突 |
| Specific Alternative Flows  (Test) | RFS 3 | |
| 1 | 运行Hadoop的时候出现了运算错误或者语法错误 |
| 2 | ABORT |
| Postcondition  (Test) | 需要修改Inputsplit函数 |

### 3.5.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | 源码改进 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 重载InputSplit函数 | 0 |
| 2 | 编译Hadoop-MapReduce | 1 |
| 3 | 运行一个Hadoop的示例 | 0.1 |
| 总工时 | 1.1 | | |

# 4非功能需求模块

## 4.1运行速度测试

### 4.1.1测试策略描述

为了测试我们的Map-Reduce函数在修改了代码之后能够以更快的速度运行，所以进行该测试，正常情况下，Map-Reduce的运行速度会比标准的Hadoop-MapReduce执行的更快；失败情况下，系统的功能测试存在未能通过的项目。

### 4.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 运行速度测试 | |
| Brief Description | 测试改进后的Map-Reduce函数的运行速度 | |
| Precondition | Map-Reduce函数正常工作 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 在改进前的Map-Reduce中运行函数处理数据 |
| 2 | 统计计算时间 |
| 3 | 在改进后的Map-Reduce中运行函数处理数据 |
| 4 | VAILDATE THAT 比较运行时间 |
| Postcondition  (Test) | 程序的运行速度有了提高，说明性能得到了保证 |
| Specific Alternative Flows  (Test) | RFS 4 | |
| 1 | 运行速度相较于修改之前反而更慢 |
| 2 | ABORT |
| Postcondition  (Test) | 程序的高性能无法得到保证，需要修改。 |

### 4.1.3测试用例所需工时

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case Specification** | | | |
| Name | 数据爬取测试 | | |
| Basic Flow  (Test) | Steps | | 所需工时（小时） |
| 1 | 在改进前的Map-Reduce中运行函数 | 0.1 |
| 2 | 统计计算时间 | 0 |
| 3 | 在改进后的Map-Reduce中运行函数 | 0.1 |
|  | 4 | 比较运行时间 | 0.1 |
| 总工时 | 0.2 | | |

# 5.测试方案自我评价

本文的测试方案主要关注的重点在于对我们实际制作的功能进行重点测试，环境配置等基本功能其实在制作阶段已经基本完成了，所以测试的必要性并不突出，所以我们针对基本功能只设置了3个用例，对功能实现设置了6个用例。

对于非功能需求的模块，我们只设置了一个最重要的测试，就是Hadoop在改进了之后是否得到了速度上的提升同时数据结果依然准确。

总而言之，我们的测试方案与需求规格说明书中做到了一一对应，能够全面的测试软件的需求。