**Hadoop-MapReduce**

**测试规格说明书**

Version 1.0

小组成员：

杨云龙

陈昆度

李嘉艺

马宇晴

**版本变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 |
| 1.0 | 2016/05/09 | 杨云龙 | 陈昆度，李嘉艺，马宇晴 |

目录

[1 前言 5](#_Toc450767262)

[1.1 目的 5](#_Toc450767263)

[1.2 项目背景 5](#_Toc450767264)

[1.3 文档概述 6](#_Toc450767265)

[1.4 测试用例与需求用例参照表 6](#_Toc450767266)

[2基本需求模块 7](#_Toc450767267)

[2.1操作系统测试 7](#_Toc450767268)

[2.1.1测试策略描述 7](#_Toc450767269)

[2.1.2测试用例 7](#_Toc450767270)

[2.2 Hadoop部署 8](#_Toc450767271)

[2.2.1测试策略描述 8](#_Toc450767272)

[2.2.2测试用例 8](#_Toc450767273)

[2.3 Maven部署 9](#_Toc450767274)

[2.3.1测试策略描述 9](#_Toc450767275)

[2.3.2测试用例 9](#_Toc450767276)

[3实现需求模块 10](#_Toc450767277)

[3.1数据爬取 10](#_Toc450767278)

[3.1.1测试策略描述 10](#_Toc450767279)

[2.1.2测试用例 10](#_Toc450767280)

[3.2数据预处理 11](#_Toc450767281)

[3.2.1测试策略描述 11](#_Toc450767282)

[3.2.2测试用例 11](#_Toc450767283)

[3.3 Map-Reduce函数 12](#_Toc450767284)

[3.3.1测试策略描述 12](#_Toc450767285)

[3.3.2测试用例 12](#_Toc450767286)

[3.4输出结果 13](#_Toc450767287)

[3.4.1测试策略描述 13](#_Toc450767288)

[3.4.2测试用例 13](#_Toc450767289)

[3.5源码改进 14](#_Toc450767290)

[3.5.1测试策略描述 14](#_Toc450767291)

[3.5.2测试用例 14](#_Toc450767292)

[4非功能需求模块 15](#_Toc450767293)

[4.1运行速度测试 15](#_Toc450767294)

[4.1.1测试策略描述 15](#_Toc450767295)

[4.1.2测试用例 15](#_Toc450767296)

# 前言

## 目的

本测试文档主要用于测试，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。同时介绍了小组成员之间的工作，设计了测试用例。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，排序分为数据抓取模块，map模块，参数提取模块，reduce模块，价格排序模块和输出模块。由于性能改进是对InputSplit进行了修改，所以只需要测试这一个就可以了。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，我们会将测试结果整合到一个文档当中。

## 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. 人员分工
4. 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 | linux部署 | 操作系统测试 |
| Hadoop部署 | Hdfs和Mapreduce测试 |
| Maven部署 | 编译测试 |
| 实现需求 | 数据爬取 | Python爬虫测试 |
| 数据预处理 | 预处理测试 |
| Map-Reduce函数 | 排序测试 |
| 输出结果 | 结果测试 |
| 源码改进 | Inputsplit测试 |
| 非功能需求 | 运行速度 | 运行速度测试 |

# 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。

### 2.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 操作系统测试 | |
| Brief Description | 检查Linux是否正确安装 | |
| Precondition | VMware安装成功，有Linux的ios光盘 | |
| Primary Actor | 测试员 | |
| Dependency | None | |
| Basic Flow  (Test) | Steps | |
| 1 | 配置虚拟机； |
| 2 | 在虚拟机上安装Linux系统； |
| 3 | 在虚拟机上安装Java环境； |
| 4 | 用Shell命令测试环境 |
| Postcondition  (Test) | Linux安装成功 |
| Specific Alternative Flows  (Test) | RFS basic flow 4 | |
| 1 | 使用shell时发现有功能安装失败（如JVM）； |
| 2 | 重新安装 |
| Postcondition  (Test) | 安装失败 |

## 2.2 Hadoop部署

## 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 | Hadoop部署 | |
| 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安装Hadoop集群 |
| 3 | 修改Hadoop中的配置文件 |
| 4 | 测试Hadoop-MapReduce的example函数 |
| 5 | 测试命令start-all.sh |
| Postcondition  (Test) | Hadoop安装成功 |
| Specific Alternative Flows  (Test) | RFS 4 | |
| 1 | 测试函数无法运行 |
| 2 | ABORT |
| Postcondition  (Test) | Hadoop安装失败 |

**2.3** **Maven部署**

**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。

**2.3.2测试用例**

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Maven部署 | |
| 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源码； |
| 2 | 在Linux环境下编译安装； |
| 3 | 测试编译hadoop源码。 |
| Postcondition  (Test) | Maven搭建成功。 |
| Specific Alternative Flows  (Test) | RFS 3 | |
| 1 | Hadoop编译失败； |
| 2 | 重新搭建Maven |
| Postcondition  (Test) | Maven搭建失败 |

# 3实现需求模块

## 3.1数据爬取

### 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和属性，有序的进行输出。异常情况下会出现数据的缺省。

### 2.1.2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 |
| 2 | 运行python爬虫文件 |
| 3 | 查看数据爬取结果 |
| 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 |

## 3.2数据预处理

### 3.2.1测试策略描述

本测试的目的在于测试对爬取数据执行的预处理过程将数据成功的进行了格式化。正常情况下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。

### 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 | Tester | |
| 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 |
| 3 | 查看输出结果 |
| 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 |

## 3.3 Map-Reduce函数

## 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 |
| 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 |

## 3.4输出结果

### 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。

### 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 | 与真实数据进行比较 |
| Postcondition  (Test) | 数据正确 |
| Specific Alternative Flows  (Test) | RFS 2 | |
| 1 | 数据有明显的错误 |
| 2 | ABORT |
| Postcondition  (Test) | Map-Reduce函数功能性错误 |

## 3.5源码改进

### 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InputSplit函数 |
| 2 | 编译Hadoop-MapReduce |
| Postcondition  (Test) | Hadoop成功编译 |
| Specific Alternative Flows  (Test) | RFS 2 | |
| 1 | 编译错误 |
| 2 | ABORT |
| Postcondition  (Test) | 函数出现语法错误 |

# 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 |
| 2 | 统计计算时间 |
| 3 | 在改进后的Map-Reduce中运行函数 |
| 4 | 比较运行时间 |
| Postcondition  (Test) | 速度明显得到了改善 |