**Spark Streaming的分析与应用**

**测试需求规格说明书**

Version 1.3.0

小组成员：

陈少杰

姜鑫

蒲彦均

邹嘉欣

**版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 | 备注 |
| 1.0.1 | 2017/05/17 | 邹嘉欣、陈少杰、姜鑫、蒲彦均 | 邹嘉欣、陈少杰、姜鑫、蒲彦均 | 初稿 |
| 1.1.0 | 2017/05/23 | 姜鑫 | 邹嘉欣、陈少杰、蒲彦均 | 增加了应用功能与非功能测试 |
| 1.1.1 | 2017/05/23 | 陈少杰、蒲彦均 | 姜鑫、邹嘉欣 | 细化测试内容，调整格式 |
| 1.2.0 | 2017/05/25 | 邹嘉欣 | 陈少杰、姜鑫、蒲彦均 | 根据A组、G组的评审意见修改格式、文字错误等问题,给所有用例加上了测试结果的评价准则 |
| 1.2.1 | 2017/05/26 | 陈少杰 | 邹嘉欣、蒲彦均、姜鑫 | 修改性能调优部分测试 |
| 1.2.2 | 2017/05/31 | 蒲彦均 | 邹嘉欣、陈少杰、姜鑫 | 修改部分测试用例 |
| 1.2.3 | 2017/05/31 | 邹嘉欣 | 陈少杰、姜鑫、蒲彦均 | 修改作业调度测试用例 |
| 1.2.4 | 2017/05/31 | 姜鑫 | 邹嘉欣、陈少杰、蒲彦均 | 修改部分测试用例 |
| 1.3.0 | 2017/05/31 | 陈少杰 | 邹嘉欣、蒲彦均、姜鑫 | 修改部分测试用例 |

目录

[1 前言 5](#_Toc484036715)

[1.1 目的 5](#_Toc484036716)

[1.2 软件测试的主要内容 5](#_Toc484036717)

[1.3 文档概述 5](#_Toc484036718)

[1.4测试用例与需求用例参照表 5](#_Toc484036719)

[1.5术语和缩略语 6](#_Toc484036720)

[2功能需求模块 6](#_Toc484036721)

[2.1 数据的产生与输入测试 6](#_Toc484036722)

[2.1.1测试策略描述 6](#_Toc484036723)

[2.1.2测试用例 6](#_Toc484036724)

[2.2数据流抽象测试 7](#_Toc484036725)

[2.2.1测试策略描述 7](#_Toc484036726)

[2.2.2测试用例 7](#_Toc484036727)

[2.3 Spark Streaming的长时间容错测试 8](#_Toc484036728)

[2.3.1测试策略描述 8](#_Toc484036729)

[2.3.2测试用例 8](#_Toc484036730)

[2.4作业调度测试 9](#_Toc484036731)

[2.4.1测试策略描述 9](#_Toc484036732)

[2.4.2测试用例 9](#_Toc484036733)

[2.5窗口支持测试 10](#_Toc484036734)

[2.5.1测试策略描述 10](#_Toc484036735)

[2.5.2测试用例 10](#_Toc484036736)

[3非功能需求模块 11](#_Toc484036737)

[3.1 实时性测试 11](#_Toc484036738)

[3.1.1测试策略描述 11](#_Toc484036739)

[3.1.2测试用例 11](#_Toc484036740)

[3.2可扩展性测试 12](#_Toc484036741)

[3.2.1测试策略描述 12](#_Toc484036742)

[3.2.2测试用例 12](#_Toc484036743)

[3.3吞吐量测试 12](#_Toc484036744)

[3.3.1测试策略描述 12](#_Toc484036745)

[3.3.2测试用例 13](#_Toc484036746)

[3.4 Spark Streaming的持久化测试 13](#_Toc484036747)

[3.4.1测试策略描述 13](#_Toc484036748)

[3.4.2测试用例 13](#_Toc484036749)

[3.5 性能调优测试 14](#_Toc484036750)

[3.5.1测试策略描述 14](#_Toc484036751)

[3.5.2测试用例 14](#_Toc484036752)

[4设计实现测试 16](#_Toc484036753)

[4.1数据的产生模块功能测试 16](#_Toc484036754)

[4.1.1测试策略描述 16](#_Toc484036755)

[4.1.2测试用例 16](#_Toc484036756)

[4.2 文本预处理与分类模块功能测试 17](#_Toc484036757)

[4.2.1测试策略描述 17](#_Toc484036758)

[4.2.2测试用例 17](#_Toc484036759)

[4.3分类速度测试 18](#_Toc484036760)

[4.3.1测试策略描述 18](#_Toc484036761)

[4.3.2测试用例 18](#_Toc484036762)

# 前言

## 目的

本文档主要用于分析本软件工程综合实验小组（研究项目为：对Spark Streaming的分析与应用）测试方面的主要内容，在此基础上对测试用例进行了初步的设计。

## 软件测试的主要内容

软件测试是一种实际输出与预期输出间的审核或者比较过程。软件测试的经典定义是：在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。本次软件测试阶段的主要工作如下：

* 总结项目实现的内容，结合需求分析阶段的内容，分析测试需求，编写测试计划及测试规格说明书；
* 编写有效的、覆盖面广的测试用例；
* 研究相关测试技术；
* 按计划实施测试工作，提交测试报告；

## 文档概述

本次测试需求规格说明书主要参照《Spark Streaming的分析与应用-需求规格说明书》以及已经实现的项目内容，给出了需求用例与测试用例的对照表，对于每个测试用例，先给出测试策略的描述，然后按照RUCM4test标准说明测试用例。主要按照功能需求、非功能需求以及应用功能三个方面进行测试。

## 测试用例与需求用例参照表

本次测试严格遵照《Spark Streaming的分析与应用-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表1所示：

表 1测试用例与需求用例参照表

|  |  |  |
| --- | --- | --- |
| 模块 | 需求用例 | 测试用例 |
| 功能需求 | 数据的产生与输入 | 数据的产生与输入测试 |
| 数据流抽象DStream | 数据流抽象测试 |
| 长时间容错 | 长时间容错测试 |
| 作业调度 | 作业调度测试 |
| 窗口支持 | 窗口支持测试 |
| 非功能需求 | 实时性 | 实时性测试 |
| 扩展性与吞吐量 | 扩展性测试；  吞吐量测试 |
| 持久化 | 持久化测试 |
| 性能调优 | 性能调优测试 |
| 设计实现 | 软件基本功能 | 数据的产生模块功能测试  文本预处理与分类模块功能测试 |
| 性能指标 | 分类速度测试 |

## 1.5术语和缩略语

# 2功能需求模块

## 2.1 数据的产生与输入测试

### 2.1.1测试策略描述

测试员对Spark Streaming 的一种常见输入源Kafka进行简单的发送与接收测试，使用Kafka自带的工具脚本将一段数据发送至Kafka的指定topic，并读取此topic，查看输入数据与输出数据的一致性。

### 2.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 数据的产生与输入测试 | |
| Brief Description | 发送数据后启动数据接收，对照输入输出，检查数据一致性 | |
| Precondition | Kafka程序正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送微博文本文件 |
| Description | 准备待发送微博文本文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 检查Kafka是否处于运行状态 |
| Postcondition | Kafka功能正常，数据准备完毕 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 将微博文本上传服务器 |
| 2 | 启动Kafka数据发送脚本 |
| 3 | 启动接收数据脚本 |
| 4 | 查看接收到的数据数量、顺序，与预期结果进行比较 |
| Postcondition | Kafka发送与接收基本功能正常 |
| Evaluation Criterion | 接收到的数据与发送的数据条数相同，顺序一致。 | |

## 2.2数据流抽象测试

### 2.2.1测试策略描述

本测试目的为检验Spark Streaming接收Kafka数据，构建数据流抽象DStream的过程是否正常，如果正常，程序将直接将读取到的数据写入HDFS，经过对HDFS文件中的数据与输入数据进行比对，若一致则可说明数据流抽象正常。

### 2.2.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 数据流抽象测试 | |
| Brief Description | Spark Streaming从Kafka读取数据，构建DStream输入流，直接写入HDFS | |
| Precondition | Kafka、Spark均正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送微博文本文件 |
| Description | 准备待发送微博文本文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 检查Kafka是否处于运行状态 |
| 2 | 检查Spark是否处于运行状态 |
| Postcondition | Kafka和Spark功能正常，数据准备完毕 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动Kafka数据发送程序，设置发送数据批数 |
| 2 | 启动SparkStreaming测试程序 |
| 3 | 等待数据处理完毕，下载查看HDFS文件 |
| Postcondition | Spark Streaming数据流构建正常 |
| Evaluation Criterion | HDFS文件中的数据条数与发送的数据条数相同，说明Spark Streaming完成了数据流抽象，并且准确地将输入数据逐条写入HDFS，没有遗漏。 | |

## 2.3 Spark Streaming的长时间容错测试

### 2.3.1测试策略描述

用例的目的是测试系统的各个模块在长时间运行状态下对发生的错误情况的应对能力，分别对数据导入导出模块、数据抽象和作业调度模块测试容错能力。正常情况下，当这三个模块出现错误时，系统可以处理错误并继续正常的运行；异常情况下，模块会出现问题无法继续正常运行。

### 2.3.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 长时间容错测试 | |
| Brief Description | 在运行正常的系统中将某个作业置错，检测系统能否对错误进行纠正 | |
| Precondition | 系统运行正常 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 运行一次正常的处理任务 |
| Description | 运行一次正常的处理任务 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备待发送微博文本文件 |
| 2 | 检查Kafka和Spark是否处于运行状态 |
| 3 | 运行执行指令 |
| Postcondition | 任务正常执行 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 关闭分布式节点中一台机器的网络 |
| 2 | 检测系统是否发现错误的作业 |
| 3 | 检测系统是否对错误的作业进行纠正 |
| Postcondition | 系统完成了本次处理任务并没有报错 |
| Evaluation Criterion | 实际结果与预期结果一致 | |

## 2.4作业调度测试

### 2.4.1测试策略描述

用例的目的是测试系统是否正确地进行作业调度，脚本分别测试作业调度正常和作业调度异常两种情况。正常情况下，系统可以运行相关资源并进行调度；异常情况下，将出现系统异常。

### 2.4.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 作业调度测试 | |
| Brief Description | 测试系统是否正确地进行作业调度 | |
| Precondition | 完成了数据流的抽象 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 运行spark streaming系统 |
| Description | 运行spark streaming系统，准备启动任务 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 输入测试数据； |
| 2 | 系统验证输入数据的完整性； |
| 3 | 将输入的数据流进行离散化； |
| 4 | 将离散化的数据按时间序列划分为RDD； |
| Postcondition  (Test Oracle) | 完成数据流的抽象； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 将每个作业都分隔成多个阶段； |
| 2 | 将作业放入调度队列； |
| 3 | 系统检测系统是否有空闲的资源； |
| 4 | 开始后续作业的第一个阶段； |
| Postcondition  (Test Oracle) | 完成阶段的执行； |
| Specific Alternative Flows  (Test Sequence) | RFS 3 | |
| 1 | 正在执行的作业占用了所有的资源； |
| 2 | 继续在队列中等待； |
| Postcondition  (Test Sequence) | 系统的队列继续监测正在执行的任务的资源占用情况； |
| Evaluation Criterion | 实际结果与预期结果一致 | |

## 2.5窗口支持测试

### 2.5.1测试策略描述

用例的目的是测试spark streaming应用能否支持不同的窗口大小，在数据入库后，设置多种不同大小的窗口对应用程序的数据进行操作，查看其正确性。

### 2.5.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 窗口支持测试 | |
| Brief Description | 测试能否以不同大小的窗口对数据进行操作 | |
| Precondition | Kafka、spark可以正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 测试数据完全入库，设置窗口大小 |
| Description | 将要进行窗口化操作的数据经过spark streaming应用处理，并存入数据库，并设置1s、6s、10s、20s等多个大小的测试窗口 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备测试数据用到的微博文本文件，包含微博数据数量在5000条左右 |
| 2 | 启动kafka和spark streaming，进行数据处理并入库 |
| 3 | 设置要测试的窗口大小（1s、6s、10s、20s） |
| Postcondition | 测试数据准备完成，各个窗口大小已选定； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 选择一个预先设置的窗口； |
| 2 | 窗口在存储处理后数据的数据块上滑动； |
| 3 | 合并窗口内的数据块； |
| 4 | 判断合并后的数据块是否正确 |
| 5 | 循环执行1-4，直至每个选定大小的窗口测试完毕 |
| Postcondition | 窗口支持功能是否正常； |
| Evaluation Criterion | 实际结果与预期结果一致 | |

# 3非功能需求模块

## 3.1 实时性测试

### 3.1.1测试策略描述

用例的目的是spark streaming应用能否满足实时性的要求，即测试系统输入数据后，到处理完成所需的时间。设置多个批次的微博数据，对每批次数据发送到处理完成的用时，进而判断其实时性是否满足要求。

### 3.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 实时性测试 | |
| Brief Description | 测试从数据输入开始到数据处理结束的时间 | |
| Precondition | Kafka和spark运行正常 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送的文本 |
| Description | 准备待发送的微博文本文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备待发送的微博文本文件，包含微博数据数量在5000条左右 |
| 2 | 确保kafka和spark处于运行状态 |
| Postcondition | Kafka和Spark功能正常，数据准备完毕 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动本小组编写的kafka数据发送程序，设置发送数据批数 |
| 2 | 启动Spark Streaming直接读写测试程序 |
| 3 | 每批次数据处理完毕，记录该批次处理时间 |
| 4 | 计算批次平均处理时间 |
| Postcondition | 得到实时性指标 |
| Evaluation Criterion | 本测试没有通过与不通过之分，仅仅为了测试Spark Streaming的实时处理速度的大小。 | |

## 3.2可扩展性测试

### 3.2.1测试策略描述

在集群现有规模下再添加一台机器，测试是否可以通过简单地配置，快速将新的节点加入到集群中，实现集群的扩展。

### 3.2.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Spark Streaming的可扩展性测试 | |
| Brief Description | 向Spark集群中新增一台机器，经少量配置，完成集群扩展，达到可用 | |
| Precondition | Spark集群正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 确认Spark集群运行情况 |
| Description | 确认Spark集群运行情况 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 对新加入的节点进行安装配置 |
| 2 | 修改现有集群的配置 |
| 3 | 重新启动新集群 |
| Postcondition | 新集群运行正常 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动Kafka数据发送程序，设置发送数据批数 |
| 2 | 启动Spark数据接收测试程序 |
| 3 | 查看Spark Application管理页面，检查新节点是否成功运行 |
| Postcondition | 整个集群，包括新节点，运行正常 |
| Evaluation Criterion | 通过简单地配置，快速将新的节点加入到集群中，集群正常运行，实现集群的扩展。 | |

## 3.3吞吐量测试

### 3.3.1测试策略描述

在集群正常运行状态下编写Spark Streaming程序，对输入数据直接进行输出，并打印日志，从而统计系统吞吐量。

### 3.3.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Spark Streaming的吞吐量测试 | |
| Brief Description | 在集群正常运行状态下对输入数据直接进行输出，从而统计系统吞吐量 | |
| Precondition | Spark集群正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 确认Spark集群、Kafka运行情况 |
| Description | 确认Spark、Kafka集群运行情况 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备测试用数据 |
| 2 | Kafka发送程序部署到多台机器 |
| Postcondition | 保证发送速度不低于Spark Streaming处理速度 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动Kafka发送数据程序，以最快速度发送数据 |
| 2 | 等待一段时间后，启动Spark Streaming直接读写程序 |
| 3 | 查看运行日志，记录运行时间与处理数据量的关系 |
| Postcondition | 根据实际测试情况得出集群吞吐量 |
| Evaluation Criterion | 本测试没有通过与不通过之分，只希望获得在特定集群上Spark Streaming的吞吐量。 | |

## 3.4 Spark Streaming的持久化测试

### 3.4.1测试策略描述

用例的目的是测试系统把临时数据保存到可多次读取的设备上的能力。主要测试少量的临时数据持久化到内存的能力和大量数控持久化到磁盘的能力。

### 3.4.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Spark Streaming的持久化测试 | |
| Brief Description | 测试Spark Streaming对数据的持久化情况 | |
| Precondition | 数据持久化成功 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 测试数据的持久化 |
| Description | 测试数据的持久化 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备待发送微博文本文件 |
| 2 | 检查Kafka和Spark是否处于运行状态 |
| 3 | 运行执行指令 |
| Postcondition | 任务正常执行 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 监测内存中Spark模块的数据 |
| 2 | 在完成抽象DStream之后查看内存中是否增加了Dsteam数据 |
| 3 | 在完成任务之后查看HDFS中是否有结果数据 |
| Postcondition | 临时数据成功持久化 |
| Evaluation Criterion | 实际结果与预期结果一致 | |

## 3.5 性能调优测试

### 3.5.1测试策略描述

用例的目的是测试系统的性能调优方面的能力，主要测试批处理时间、并行化程度和过期数据清除三部分测试。批处理时间主要测试每批次数据处理时间与批处理时间间隔设置是否合理，并行化程度主要测试kafka发送端的并行发送能力，清除过期数据主要测试过期数据是否能按时清除。

### 3.5.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 批处理时间测试 | |
| Brief Description | 测试批处理时间是否合理； | |
| Precondition | Kafka和spark运行正常 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送的文本 |
| Description | 准备待发送的微博文本文件； |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备待发送的微博文本文件； |
| 2 | 确保kafka和spark处于运行状态； |
| 3 | 设置多个不同批处理时间（200ms，500ms，1s，5s，10s） |
| Postcondition | Kafka和Spark功能正常，数据准备完毕； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动kafka数据发送程序； |
| 2 | 启动spark测试程序； |
| 3 | 每批次数据处理完毕，记录该批次数据处理时间、等待处理时间； |
| 4 | 计算批次平均处理时间、平均等待处理时间 |
| 6 | 更改批处理时间，重复1-4； |
| 5 | 判断各数据处理时间和等待处理时间的比值是否合适 |
| Postcondition | 判断当前批处理时间是否合适； |
| Evaluation Criterion | 得到最佳批处理时间 | |

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 并行化程度测试 | |
| Brief Description | 测试kafka获取数据的并行能力； | |
| Precondition | Kafka和spark运行正常 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送的文本，设置多个kafka发送端实例 |
| Description | 准备待发送的微博文本文件，设置多个kafka发送端实例； |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 准备待发送微博文本文件； |
| 2 | 确保kafka和spark处于运行状态； |
| Postcondition | Kafka和Spark功能正常，数据准备完毕； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动kafka数据发送程序； |
| 2 | 启动spark测试程序； |
| 3 | 记录运行是否正常以及当前处理时间； |
| 3 | 增加并行kafka发送端实例的数量，重复1-3； |
| 4 | 判断kafka数据获取部分能否很好实现并行化 |
| Postcondition | 根据kafka发送端增加并行化后正确性、运行速度与之前的差异，判断kafka发送端是否有很好的并行化处理能力； |
| Evaluation Criterion | 并行化前后，是否均运行正确，以及增加并行前后的运行速度差异。 | |

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 过期数据清理测试 | |
| Brief Description | 测试过期数据能否及时清除 | |
| Precondition | Kafka和spark运行正常 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 数据不停入库，设置删除时间 |
| Description | Kafka不停地获取微博数据，并设置要删除数据的时间； |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 启动kafka，不停获取数据 |
| 2 | 设置定期删除时间（为了测试方便，可设置为1h） |
| Postcondition | Kafka正常运行，设置了不同的数据清除时间； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动kafka，不停获取数据 |
| 2 | 设置定期删除时间（为了测试方便，可设置为1h） |
| 3 | 每隔一段时间记录过期数据的删除情况 |
| 4 | 循环1-3，直至多个删除时间全部测试 |
| Postcondition | 确保过期数据能及时清理； |
| Evaluation Criterion | 每个测试时间，是否都能保证过期数据及时清理 | |

# 4设计实现测试

## 4.1数据的产生模块功能测试

### 4.1.1测试策略描述

测试人员对系统的数据产生模块进行基本功能测试，测试目标是确认数据产生模块可以

1. 顺利读取文本文件
2. 成功发送至Kafka的weibo topic。

### 4.1.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 数据的产生模块功能测试 | |
| Brief Description | 读取本地微博文本文件，将每一行数据发送至Kafka | |
| Precondition | Kafka正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送微博文本文件 |
| Description | 准备待发送微博文本文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 检查Kafka是否处于运行状态 |
| Postcondition | Kafka运行正常，数据准备完毕 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动本小组编写的Kafka数据发送程序，设置发送数据批数 |
| 2 | 启动本小组编写的Kafka数据消费程序。 |
| 3 | 等待数据处理完毕，下载查看数据接收程序接收到的数据条数 |
| Postcondition | Kafka数据产生模块功能运行正常 |
| Evaluation Criterion | 比对发送数据与接收到的数据的总数，若相等则说明我们的数据发送模块功能正常。 | |

## 4.2 文本预处理与分类模块功能测试

### 4.2.1测试策略描述

测试人员对系统的文本预处理与分类模块进行基本功能测试，测试目标是确认文本分类模块可以

1. 顺利读取Kafka数据流。
2. 成功构建DStream数据流。
3. 完成预处理与分类（打标签）流程。
4. 完成写入Kafka weiboLabel topic。

### 4.2.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 文本预处理与分类模块功能测试 | |
| Brief Description | 启动Spark分类程序，检查Kafka中分类结果的数量、质量 | |
| Precondition | Kafka正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送微博文本文件 |
| Description | 准备待发送微博文本文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 检查Kafka和Spark是否处于运行状态 |
| 2 | 启动数据产生模块 |
| 3 | 启动Spark文本分类模块 |
| Postcondition | 数据流经过Kafka、Spark，完成分类打标签工作，写入Kafka |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动Kafka数据接收模块 |
| 2 | 等待数据发送模块发送数据完毕，并再等待一段时间，直到没有新的数据被接收到为止。 |
| 3 | 下载查看数据接收程序接收到的数据，与发送数据进行比对，检查数据数量、质量。 |
| Postcondition | 分类模块功能正常 |
| Evaluation Criterion | 每条格式正确的微博数据能够被成功打上标签，并且数据没有异常丢失则说明文本预处理与分类模块功能正常。 | |

## 4.3分类速度测试

### 4.3.1测试策略描述

测试人员对系统的文本分类模块进行分类性能（速度）测试，测试目标是确认文本分类模块的分类速度。

### 4.3.2测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | Spark分类模块性能测试 | |
| Brief Description | 启动Spark分类程序，检查Kafka中一段时间后分类结果的数量 | |
| Precondition | Kafka正常运行 | |
| Tester | 测试员 | |
| Dependency | None | |
| Test Setup | Name | 准备待发送微博文本文件 |
| Description | 准备待发送微博文本文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 检查Kafka和Spark是否处于运行状态 |
| 2 | 启动数据产生模块，以最大速率发送数据（根据实际情况可以在多态机器上启动多个数据产生模块，以确保数据产生速度高于分类速度） |
| 3 | 启动Spark文本分类模块 |
| Postcondition | 数据流经过Kafka、Spark，完成分类打标签工作，写入Kafka |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 启动Kafka数据接收模块 |
| 2 | 查看Spark文本分类程序产生的日志，根据两次写日志时间间隔和两次已分类总数进行数据分类速度计算 |
| Postcondition | 成功测试出分类模块速度 |
| Evaluation Criterion | 本测试没有通过与不通过之分，只为测试Spark在特定场景下的文本处理速度，此速度既可能受限于分类器的分类速度，也可能受限于Spark的I/O速度。 | |