**Spark Streaming分析与应用**

**软件问题报告**

Version 1.2

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 | 20170531 | 蒲彦均 | 邹嘉欣、陈少杰、姜鑫 | 初稿 |
| 1.1 | 20170531 | 姜鑫 | 邹嘉欣、陈少杰、蒲彦均 | 增加姜鑫负责的测试部分报告 |
| 1.2 | 20170601 | 邹嘉欣、陈少杰 | 姜鑫、蒲彦均 | 增加作业调度、窗口支持和性能调优部分的测试报告 |

目录

[1 Spark Streaming功能需求测试 4](#_Toc484106227)

[1.1数据的产生与输入测试结果 4](#_Toc484106228)

[1.2数据流抽象测试结果 4](#_Toc484106229)

[1.3 Spark Streaming的长时间容错测试结果 5](#_Toc484106230)

[1.4作业调度测试结果 5](#_Toc484106231)

[1.5窗口支持测试结果 5](#_Toc484106232)

[1.6结论 6](#_Toc484106233)

[2 Spark Streaming非功能性需求测试 6](#_Toc484106234)

[2.1实时性测试结果 6](#_Toc484106235)

[2.2可扩展性测试结果 7](#_Toc484106236)

[2.3吞吐量测试结果 7](#_Toc484106237)

[2.4 Spark Streaming的持久化测试结果 7](#_Toc484106238)

[2.5性能调优测试结果 8](#_Toc484106239)

[2.6结论 10](#_Toc484106240)

[3设计实现测试 10](#_Toc484106241)

[3.1数据的产生模块功能测试 10](#_Toc484106242)

[3.2文本预处理与分类模块功能测试 11](#_Toc484106243)

[3.3分类速度测试 11](#_Toc484106244)

[3.4结论 12](#_Toc484106245)

# 1 Spark Streaming功能需求测试

## 1.1数据的产生与输入测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 数据的产生与输入测试 | | | |
| **简要描述** | 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。 | | | |
| **前提和约束** | Kafka系统状态正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动Kafka数据发送脚本 | 脚本启动成功 | 数据成功发送到topic中 |  |
| **2** | 启动接收数据脚本 | 接收数据脚本接收到的数据与发送的数据一致 | 数据的数量与顺序一致 | 通过 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170523 | | | |

## 1.2数据流抽象测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 数据流抽象测试 | | | |
| **简要描述** | 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。 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动本小组编写的Kafka数据发送程序，设置发送数据批数为100 | 数据成功发送到指定topic——weibo | topic中发现数据 |  |
| **2** | 启动SparkStreaming测试程序 | 程序成功启动 | 在Spark应用管理界面中发现测试程序的信息 |  |
| **3** | 等待数据处理完毕，下载查看HDFS文件 | 接收数据脚本接收到的数据与发送的数据一致 | 数据的数量与顺序一致 | 通过 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170523 | | | |

## 1.3 Spark Streaming的长时间容错测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | Spark Streaming的长时间容错测试 | | | |
| **简要描述** | 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 | | | |
| **前提和约束** | 系统状态正常，并完成了一次正常的处理任务 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 关闭分布式节点中一台机器的网络 | 任务的某个作业发生故障 |  |  |
| **2** | 检测系统是否发现错误的作业并对错误的任务纠正 |  |  |  |
| **3** | 查看任务结束时的任务完成状态 | 系统完成了本次处理任务并没有报错 | 实际结果与预期结果一致 | 通过 |
| **测试人员** | 蒲彦均 | | | |
| **测试时间** | 20170530 | | | |

## 1.4作业调度测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 作业调度测试 | | | |
| **简要描述** | 给系统输入多个作业，检测系统能否正常进行作业调度 | | | |
| **前提和约束** | 系统状态正常，并完成了数据流的抽象 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 给系统输入多个作业 | 成功输入作业 |  |  |
| **2** | 查看系统是否正常地进行了作业调度 | 各个作业均正常执行 | 实际结果与预期结果一致 | 通过 |
| **测试人员** | 邹嘉欣 | | | |
| **测试时间** | 20170524 | | | |

## 1.5窗口支持测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 窗口支持测试 | | | |
| **简要描述** | 测试能否以不同大小的窗口对数据进行操作 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常，测试数据完全入库，设置多个窗口大小 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 选择一个预先设置的窗口 | 窗口设置完成 | 是否完成窗口大小设置 |  |
| **2** | 窗口在存储处理后数据的数据块上滑动 | 每次滑动，窗口选中一定量的数据 | 是否每次滑动都能选中正确的数据部分 |  |
| **3** | 合并窗口内的数据块 | 得到合并后的窗口内的数据块 | 是否得到合并后的数据块 |  |
| **4** | 判断合并后的数据块是否正确 | 合并后的数据块内容正确 | 每次滑动，合并后的数据块是否均正确 |  |
| **5** | 循环执行1-4，直至每个选定大小的窗口测试完毕 | 对于不同大小的窗口都能在入库的数据上滑动，并正确地合并窗口内的数据块内容 | 是否每个大小的窗口都能支持窗口操作，并合并得到正确的窗口中的数据 | 对于设置的每个不同大小的窗口，应用都能正确的进行窗口操作，并合并得到每个窗口中的数据块 |
| **测试人员** | 陈少杰 | | | |
| **测试时间** | 20170531 | | | |

## 1.6结论

通过本章测试，可以得到结论如下：Kafka的数据发送与接收功能运行正常；Spark Streaming的数据流抽象功能正常；并且系统可以满足长时间容错、作业调度及窗口支持等功能性需求。

# 2 Spark Streaming非功能性需求测试

## 2.1实时性测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 实时性测试 | | | |
| **简要描述** | 设置多个批次的微博数据，记录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。 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动本小组编写的Kafka数据发送程序，设置发送数据批数10 | 数据成功发送到指定topic——weibo | topic中发现数据 |  |
| **2** | 启动SparkStreaming直接读写程序，每批次数据处理完毕，记录该批次处理时间 | 程序成功启动 | 在Spark应用管理界面中发现测试程序的信息 |  |
| **3** | 查看发送程序与直接读写程序的日志，计算每条数据走完流程的总用时 | 得到每条数据走完流程的总用时 |  | Spark Streaming在特定场景下数据从输入到输出延时平均在2.3s左右 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170527 | | | |

## 2.2可扩展性测试结果

此部分具有对Spark集群稳定的直接威胁，风险较大，故放弃测试。

## 2.3吞吐量测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 吞吐量测试 | | | |
| **简要描述** | 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 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动本小组编写的Kafka数据发送程序，设置发送数据批数100 | 数据成功发送到指定topic——weibo | topic中发现数据 |  |
| **2** | 等待一段时间后，启动Spark Streaming直接读写程序 | 程序成功启动 | 在Spark应用管理界面中发现测试程序的信息 |  |
| **3** | 查看运行日志，记录运行时间与处理数据量的关系 | 得到Spark在当前场景下的吞吐量 |  | SparkStreaming在特定场景下吞吐量达到了40余万每分钟 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170530 | | | |

## 2.4 Spark Streaming的持久化测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | Spark Streaming的持久化测试 | | | |
| **简要描述** | 测试Spark Streaming对数据的持久化情况 | | | |
| **前提和约束** | 无 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 执行一次正常的任务 | 任务开始执行 |  |  |
| **2** | 监测内存中Spark模块相关的数据 | 成功读取到内存中Spark相关的数据 |  |  |
| **3** | 在完成抽象DStream之后查看内存中是否增加了Dsteam数据 | 内存中成功增加了Dsteam数据 |  |  |
| **4** | 在完成任务之后查看HDFS | HDFS中有结果数据 |  | Spark Streaming可以正常完成对数据的持久化 |
| **测试人员** | 蒲彦均 | | | |
| **测试时间** | 20170531 | | | |

## 2.5性能调优测试结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 批处理时间测试 | | | |
| **简要描述** | 测试批处理时间是否合理 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常，并准备待发送的微博文本文件，设置不同的批处理时间； | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动kafka数据发送程序 | Kafka数据发送程序正常启动 | Kafka数据发送程序是否正常启动 |  |
| **2** | 启动spark测试程序 | Spark测试程序启动 | Spark测试程序是否正常启动 |  |
| **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； | 得到每批数据处理的处理时间和等待处理时间 | 得到每批数据处理的处理时间和等待处理时间 |  |
| **4** | 计算批次平均处理时间、平均等待处理时间 | 得到当前批处理时间设置下的平均处理时间和平均等待时间 | 得到当前批处理时间设置下的平均处理时间和平均等待时间 |  |
| **5** | 更改批处理时间，重复1-4； | 得到每个批处理时间下的平均处理时间和平均等待时间 | 得到每个批处理时间下的平均处理时间和平均等待时间 |  |
| **6** | 判断各数据处理时间和等待处理时间的比值是否合适 | 得到每个批处理时间下，处理时间和等待处理时间的比值 | 判断哪个批处理时间下，处理时间和等待处理时间比值更合适 | 当前环境下，设置批处理时间为1s左右较为合理 |
| **测试人员** | 陈少杰 | | | |
| **测试时间** | 20170531 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 并行化程度测试 | | | |
| **简要描述** | 测试kafka获取数据的并行能力； | | | |
| **前提和约束** | Kafka与Spark系统状态均正常，准备待发送的文本，设置多个kafka发送端实例 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动kafka数据发送程序； | Kafka数据发送程序正常启动 | Kafka数据发送程序是否正常启动 |  |
| **2** | 启动spark测试程序； | Spark测试程序启动 | Spark测试程序是否正常启动 |  |
| **3** | 记录运行是否正常以及当前处理时间； | 得到当前并行程度下的处理时间 | 得到当前并行程度下的处理时间 |  |
| **4** | 增加并行kafka发送端实例的数量，重复1-3； | 得到增加并行化程度后的处理时间 | 得到增加并行化程度后的处理时间 |  |
| **5** | 判断kafka数据获取部分能否很好实现并行化 | 得到各个并行化程度下的处理时间 | 比较增加并行化前后的批处理时间，确定当前并行化程度是否合理 | 根据数据量，当前的3路并行发送较为合理，增加并行化处理时间并没有很大提高 |
| **测试人员** | 陈少杰 | | | |
| **测试时间** | 20170531 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 过期数据清理测试 | | | |
| **简要描述** | 测试过期数据能否及时清除 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常，测试数据完全入库，设置多个窗口大小 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动kafka，不停获取数据 | Kafka数据发送程序正常启动，并不停获取数据 | Kafka数据发送程序正常启动，并不停获取数据 |  |
| **2** | 设置定期删除时间（为了测试方便，可设置为1h） | 设置数据删除时间 | 设置数据删除时间 |  |
| **3** | 每隔一段时间记录过期数据的删除情况 | 过期数据已被删除 | 是否每次查看，过期数据都被删除 |  |
| **4** | 循环1-3，直至多个删除时间全部测试 | 不同的定期删除时间下，过期数据都能被删除 | 是否不同的定期删除时间下，过期数据都能被删除 | 不同的定期删除时间下，过期数据都能被删除 |
| **测试人员** | 陈少杰 | | | |
| **测试时间** | 20170601 | | | |

## 2.6结论

通过本章测试，可以得到结论如下：Spark Streaming的实时性、持久化等非功能性需求得到了满足，并且在特定场景下吞吐量达到了40余万每分钟。

# 3设计实现测试

## 3.1数据的产生模块功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 数据的产生模块功能测试 | | | |
| **简要描述** | 读取本地微博文本文件，将每一行数据发送至Kafka | | | |
| **前提和约束** | Kafka系统状态均正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动本小组编写的Kafka数据发送程序，设置发送数据批数10 | 数据成功发送到指定topic——weibo | topic中发现数据 |  |
| **2** | 启动本小组编写的Kafka数据消费程序。 | 程序成功启动 | 控制台有读取出的信息 |  |
| **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 | 发送数据量与接收到的数据量一致 | 发送数据量与接收到的数据量一致 | 通过 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170524 | | | |

## 3.2文本预处理与分类模块功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 文本预处理与分类模块功能测试 | | | |
| **简要描述** | 启动Spark分类程序，检查Kafka中分类结果的数量、质量 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动本小组编写的Kafka数据发送程序，设置发送数据批数10 | 数据成功发送到指定topic——weibo | topic中发现数据 |  |
| **2** | 启动Spark文本分类模块 | 程序成功启动 | 在Spark应用管理界面中发现测试程序的信息 |  |
| **3** | 启动Kafka数据接收模块 | 发送数据量与接收到的数据量一致并且均带有标签 | 发送数据量与接收到的数据量一致并且均带有标签 | 通过 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170524 | | | |

## 3.3分类速度测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例名称** | 分类速度测试 | | | |
| **简要描述** | 启动Spark分类程序，检查Kafka中一段时间后分类结果的数量 | | | |
| **前提和约束** | Kafka与Spark系统状态均正常 | | | |
| **序号** | 测试步骤 | 预期结果 | 评价准则 | 测试结论 |
| **1** | 启动本小组编写的Kafka数据发送程序，设置发送数据批数100 | 数据成功发送到指定topic——weibo | topic中发现数据 |  |
| **2** | 启动Spark文本分类模块 | 程序成功启动 | 在Spark应用管理界面中发现测试程序的信息 |  |
| **3** | 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 | 根据Spark程序日志中分类数量与日志打印时间，计算分类程序的分类速度 | 根据Spark程序日志中分类数量与日志打印时间，计算分类程序的分类速度 | 分布式分类程序平均分类速度达到每分钟7万余条 |
| **测试人员** | 姜鑫 | | | |
| **测试时间** | 20170524 | | | |

## 3.4结论

通过本章测试，可以得到结论如下：数据的产生模块功能正常，并且能够正常地对文本进行预处理和分类。此外，分布式分类程序平均分类速度达到每分钟7万余条。