

PaperPass专业版检测报告

简明打印版

比对结果（相似度）：

总体：6 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：6 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

期刊库：2 %（期刊库相似度是指论文与学术期刊库的比对结果）

学位库：5 %（学位库相似度是指论文与学位论文库的比对结果）

会议库：2 %（会议库相似度是指论文与会议论文库的比对结果）

互联网：0 %（互联网相似度是指论文与互联网资源的比对结果）

编号：58B90A7116417FUAT

版本：专业版

标题：基于Node-red与Redis的实时流数据处理模型的设计与

作者：王江波

长度：3306 字符(不计空格)

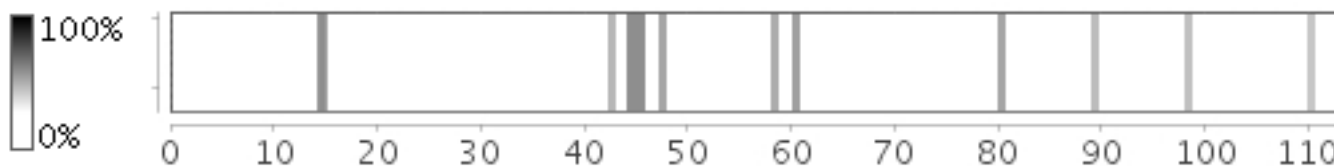
句子数：113句

时间：2017-3-3 14:17:21

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

1. 相似度：1 % 篇名：《基于流数据挖掘的入侵检测研究》

来源：会议论文 2009年西南地区网络与信息系统学术年会 2009-08-12 作者：姜毅 王翰虎 陈梅

2. 相似度：1 % 篇名：《基于InfoSphere Streams的流数据分析及应用》

来源：学位论文 武汉理工大学 2014 作者：辛波

3. 相似度：1 % 篇名：《宜宾移动农村市场网格化管理平台设计与实现》

来源：学位论文 电子科技大学 2014 作者：杜耀

4. 相似度：1 % 篇名：《面向Web的企业竞争情报获取研究》

来源：学位论文 中国科学技术大学 2013 作者：赵洁

5. 相似度：1 % 篇名：《基于Web用户日志的电子商务领域竞争对手分析--以11家电子商务网站...》

来源：会议论文 第八届中国管理学年会——中国管理的国际化与本土化 2013-11-02 作者：赵洁 温润

6. 相似度：1 % 篇名：《测试工程的研究及在电子图书商务系统测试中的应用》
来源：学位论文 北京工业大学 2006 作者：王莹
7. 相似度：1 % 篇名：《Node.js:一种新的Web应用构建技术》
来源：学术期刊 《现代电子技术》 2015年6期 作者：王金龙 宋斌 丁锐
8. 相似度：1 % 篇名：《面向软件可靠性的测试用例生成技术的研究与实现》
来源：学位论文 电子科技大学 2007 作者：唐乐
9. 相似度：1 % 篇名：《基于软件能力测试模型的大型复杂装备控制系统软件质量管理研究》
来源：学位论文 重庆大学 2009 作者：孟宪葵
10. 相似度：1 % 篇名：《行政事业单位人事工资管理信息系统的设计与实现》
来源：学位论文 电子科技大学 2010 作者：冯德福
11. 相似度：1 % 篇名：《基于SDF中间件的数字电视交互应用集成开发环境的设计与实现》
来源：学位论文 中山大学 2009 作者：叶炳宏
12. 相似度：1 % 篇名：《软件本地化测试的研究》
来源：学术期刊 《信息通信》 2012年3期 作者：徐雄飞
13. 相似度：1 % 篇名：《软件测试计划与测试用例在嵌入系统中的应用》
来源：学术期刊 《舰船电子工程》 2005年1期 作者：黄清清 许宏斌

互联网相似资源列表：

没有找到与互联网相似度高的资源！

全文简明报告：

第六章 系统测试与性能分析

评定一个模型与系统的好与坏，有两个基本标准，一个就是该系统的功能是否满足需求，另一个就是该系统的性能是否满足需求， 本文将基于这两个标准，对所设计的流式处理模型和应用系统展开测试。

6.1 测试条件准备

在进入系统测试与性能分析之前，需要做一些测试前的准备工作，最基本的准备工作就是测试数据与测试平台的准备。

1.测试数据的准备：

系统所用的测试数据，是实际生产线上截取的某政府的政务网站一天的访问流量，已经通过日志的方式保存下来，通过一个readline.js的程序来模拟http_tracer的功能。 目的，就是将这个日志发布到redis的通道中，供Node-red通过redisSub节点去订阅这些数据。 为了对系统进行压力测试，以此来检验系统的性能与数据的吞吐量，本文将采集到的实时数据分为5个独立文件， 同时利用 readline.js并发进行，发布到 http_trace通道，在 Node-red中去订阅这些数据。 这样通过提高数据的输入速度来检验系统的数据吞吐量以及数据的处理和计算能力。 图6-1展示了实际的部分样本数据。

图6-1 实际系统中的样本数据

2.测试环境的准备：

{ 57 %：测试环境检验一个系统最基本的需求，包括硬件环境和软件环境。 }

硬件环境： 一台装有两台独立虚拟机的PC机，一台虚拟机上部署本文所设计的实时流数据处理模型，另一台部署数据可视化系统也就是本文所提到的myboard。 表6-1所展示的是虚拟机的硬件配置信息。

表6-1测试的硬件环境

名称VMware 虚拟机

CPUIntel Core i7-4720HQ 四核CPU @ 2.60GHz(2601 MHz)

内存8.00 GB (1600 MHz)

硬盘32M硬盘缓存1 TB硬盘存储

网卡Killer e2200 Gigabit Ethernet Controller (NDIS 6.30)

软件环境： 部署实时流数据处理模型最基本的软件需求，基本信息显示如表6-2所示：

表6-2测试的软件环境

机器名软件名称软件版本

Core 01

(192.168.1.113)操作系统Ubuntu LTS 14.04

数据库 (active) Redis 2.8.19

Node-redV0.10.6

node.jsV5.0.4

数据库mongodb2.4.6

Core 02

(192.168.1.114)操作系统Ubuntu LTS 14.04

数据库 (active) Redis 2.8.19

myboard (可视化系统) V1.0

数据库mongodb2.4.6

node.jsV5.0.4

6.2 系统功能测试

本文所设计的网站监控系统主要提供了用户行为分析和网站群页面监控这两大功能，针对这两大功能展开测试工作。

{ 45 % : 首先是用户行为分析，包括独立访问的IP统计、热点页面统计、用户浏览器类型统计以及热门关键词统计。} 图6-2，图6-3所展示的就是对该功能的测试结果。

{ 57 % : 图6-2 用户行为分析模块测试图 (A) }

{ 57 % : 图6-3 用户行为分析模块测试图 (B) }

网站群页面监控包括网站访问量统计，错误类型统计，错误网站统计等，测试结果如图6-4和图6-5所示。

{ 50 % : 图6-4 网站群页面监控模块测试图 (A) }

图6-5 网站群页面监控模块测试图 (B)

从上面的测试结果可以看出本文所设计的系统，在功能上基本满足用户的需求，下一小节我们将对系统进行一次性能分析，检测模型在低时延、高并发的条件下的数据吞吐量和数据计算能力。

6.3 系统性能分析

在系统性能分析过程中，我们首先将原始数据报以日志的方式存储到本地硬盘里，通过编写日志读取程序来读取日志文件来模拟大量的数据流，样本数据如图6-1所示。读取的方式为按条读取，通过控制信息的条数来观察系统的性能。性能分析主要分为两个方面：一方面是数据从接收到处理再到存入数据库进行统计计算所消耗的处理时间，另一个方面是可视化界面中数据的更新频率以及响应时间。

表6-3显示了实时流数据处理模型处理数据时所需要的处理时间，其中 messages表示模拟读入日志数据到实时流数据模型的记录条数，{ 48 % : time表示实时流数据处理模型处理这些数据所需要的时间。} 通过测试发现，随着消息条数的指数增加，实时流数据处理模型的数据处理时间的增长速度低于线性增长速度。

{ 52 % : 表6-3实时流数据处理模型的数据处理时间测试结果 }

编号messages (条) time (ms)

110222880

210323891

310426980

410534655

510639891

表6-4 显示了数据的读入速度与页面数据的更新频率以及页面响应速度的测试结果。

表6-4监控页面响应时间测试数据

编号报文数（个）处理时间

（ms）数据推送

频率（s）最长响应

时间（ms）页面数据更新

时间（s）

12525151901

23189162972

327221421202

430121821182

537171821092

6.4 本章小结

{ 50 % : 本章主要介绍了是对系统主要功能测试和性能方面的分析。 } 首先介绍测试条件以及原始数据的获取，然后分别从功能和性能两个方面对系统进行了测试并得出结论。

第七章 总结与展望

7.1 本文总结

实时流数据处理，已经成为大数据时代下一种重要的计算模式，不论是在学术界还是在商业界，对实时流数据处理的研究和应用都十分广泛，需求也越来越明显。 本文是在科研团队与四川省欧润特软件科技有限公司的合作项目中展开叙述的，结合实时流数据处理的相关理论与实际项目的具体需求， 设计出一套新的基于 Node-red 与 Redis 的实时流数据处理模型，同时将该模型应用到实际的生产环境中加以验证。 对该模型的设计与应用

, 本文主要完成了一下几方面的工作。

1.总结分析了当前流行的实时流数据处理框架, 分析了各自的编程模型以及优缺点, { 43 % : 同时结合 node.js的事件驱动和非阻塞机制对 Node- red的编程模型做了重点阐述。 }

2.从总体架构上给出了基于 Node- red与 Redis的实时流数据处理模型的设计方案, 对 Node- red的节点进行扩展设计, 增加了原始没有的 redisSub、redisPub数据输入输出节点, 以及 redis数据库访问节点, 同时对这些节点加以实现, 重新部署安装到 Node- red中。

3.对基于 redis有序集合的去重统计方法进行研究, 通过分析 redis有序集合的源码并结合 Skip List的基本原理, 提出了在实时流数据计算中基于 redis有序集合 zset的去重统计方法。

4.利用所设计的流式数据处理模型, 实现网站访问实时监控系統。对整个系統进行各个模块的设计与实现, 包括数据实时采集模块、用户行为分析模块、网站群页面监控模块以及数据可视化模块, 流式数据处理过程集中在用户行为分析模块和网站群页面监控模块。

{ 43 % : 5.本文最后还对该模型和所设计的应用系統进行功能和性能上的测试, 以此来验证模型的可行性和有效性。 }

7.2 对未来工作的展望

本文对Node-red与redis做了许多研究和实践工作, 设计出了一套新的流式计算模型, 并将该模型应用到网站访问实时监控系統中。相比于其他流式计算框架而言, 该模型能够快速, 便捷地进行数据流程的管理, 可以实现业务代码的重用。但是对于大数据背景下的实时流数据计算, 该模型仍然有需要改进的地方, 所设计的系統也有待完善的地方, 但这是一个长期的需要不断坚持的应用研究领域。基于本文的研究, 对未来有如下展望:

1.本文对 Node- red的节点开发还比较单一, 虽然设计了新的数据输入、输出和数据计算节点, 但是这些节点进行数据交换的中间桥梁是 redis数据库, 所以后期的一个重要工作就是对节点的通用化设计和扩展。

2.本文所设计的模型只是采用了单一节点的redis server, 没有利用redis 集群来解决复杂的流式计算。所以, redis集群的引入也是今后工作的一个重点。

3.在本文所设计的应用系統中, 也还存在一些不足。今后的另一项重要工作就是不断地完善该网站访问监控系统, 引入更加复杂的流式计算, { 40 % : 降低模块之间的耦合程度, 完善可视化模块的展示效果, 不断提升系统的运行效率。 }

总之, 系统的功能有待丰富, 模型的性能也有待更深的优化, 今后的工作会研究更有效、更适合的实时流数据处理模型, 设计出功能更丰富、性能更优的实时流数据监控系统。