

# 面向银行业务的交易监控可视化系统设计与实现

任立男, 段桂华, 谭荻, 王建新

(中南大学 信息科学与工程学院, 湖南 长沙, 410083)

**摘要:** 针对银行日益增长的异构交易系统产生的海量交易日志结构复杂、提取困难、分析效率低以及不能及时发现交易异常等问题, 设计与实现一个面向银行业务的交易实时监控系统, 并提供交易数据的可视化分析。首先, 系统对交易日志中关键共性指标进行采集并结构化存储; 然后, 通过机器学习方法对交易量进行实时预测, 提供交易量异常告警; 最后, 通过可视化方式对异构交易系统实时监控, 并对交易数据提供多角度的可视化分析。研究表明: 该系统界面美观、交互良好、性能优异, 能很好地满足银行工作人员的业务分析需求。

**关键词:** 交易日志; 实时监控; 交易量预测; 可视化分析

中图分类号: TP393

文献标志码: A

文章编号: 1672-7207(2018)10-2490-08

## Design and implementation of transaction monitoring visualization system oriented to banking business

REN Linan, DUAN Guihua, TAN Di, WANG Jianxin

(School of Information Science & Engineering, Central South University, Changsha 410083, China)

**Abstract:** Due to the complex structure, difficulty of extraction, low efficiency of analysis and the failure of timely detection of transaction abnormalities caused by the growing heterogeneous bank transaction system, a real-time transaction monitoring system for banking business that provides the visual analysis of transaction data was designed and realized. Firstly, the system collected the key common indicators from the transaction log and stored them in a structured way, and then predicted the volume of transactions through machine learning and provided alerts of abnormal volume, and finally heterogeneous trading systems were monitored in real time through visualization and multi-perspective visual analysis of transaction data were provided. The system has a beautiful interface, good interaction and excellent performance, which can well meet the analyst's needs of banking analysis.

**Key words:** transaction log; real-time monitoring; trading volume forecasting; visual analysis

近年来, 银行业迅速发展, 银行各项业务已成为国民生活的重要组成部分。用户在银行的交易过程中, 产生了大量的交易日志, 包括用户信息、业务信息等交易的相关信息。如何采集、存储、监控和可视化这类交易数据成为了各大银行分析人员关注的焦点之一。一些研究人员提出了相关的日志数据采集、存储

与可视化方法, 来提高日志信息数据采集和分析的效率。CHAMARTHI 等<sup>[1]</sup>实现了 splunk 在电信部门的应用, splunk 实时采集和存储不同节点的日志数据, 提供交互式检索和可视化分析, 找出移动互联网中的异常电话号码和用户。BIRJALI 等<sup>[2]</sup>提出了基于 Apache Flume 从社交网络中采集社交媒体数据的方法, 并将

收稿日期: 2017-12-12; 修回日期: 2018-01-24

基金项目(Foundation item): 国家自然科学基金资助项目(61572530, 61602171)(Projects(61572530, 61602171) supported by the Natural National Science Foundation of China)

通信作者: 段桂华, 博士, 副教授, 从事计算机网络、网络安全研究; E-mail: duangh@csu.edu.cn

预处理的数据存储在 Hadoop 中,最后使用 InfoSphere BigInsights 对该数据进行分析 and 可视化。BAI<sup>[3]</sup>结合 Flume 提出了一种实时的大数据搜索方法,该方法首先使用 Flume 代理从终端收集日志,然后通过 Elasticsearch 根据需求生成 rowkey 列表,最后 Hbase 根据 rowkey 直接从数据库中获取数据。该方法在大数据日志搜索中具有较好的可行性。SANJAPPA 等<sup>[4]</sup>介绍了一种使用 Logstash 方法收集日志的系统,该系统能够处理多种类型的日志数据,帮助识别网络中的恶意活动。BAJER<sup>[5]</sup>提出了基于 ELK(Elasticsearch, Logstash 和 Kibana)的开源技术在物联网数据中的应用方法,以提取和可视化有关建筑运营的有意义的见解。银行交易日志来源于大量的异构交易系统,为了可实时监控各交易系统的指标,需要构建可交互的系统关联性结构视图,以达到分析人员快速定位到某异常指标所属交易系统的目的,从而进一步追踪和分析异常原因并解决问题。SHAHZAD 等<sup>[6]</sup>介绍和比较了 HTML5, jQuery, D3.js, GOJS, JSmol, Highchart 和 Bootstrap 等基于 Web 的工具和库,用于基于浏览器的交互式可视化。BOONNAVASIN 等<sup>[7]</sup>结合 GOJS 框架提出了构建可交互家系图方案,成为了医生了解病人家庭信息的重要诊断工具,并提供了有效的存储和检索方法。同时,银行交易数据具有很强的时间关联性,属于典型的时序数据,目前对于银行交易量等时序数据的预测方法已比较成熟并广泛应用于各领域<sup>[8]</sup>。综上所述,现有的日志采集、存储和可视化分析方法更多采用主 splunk, Flume 和 ELK 方法。虽然 splunk 整合了大量丰富的数据采集和可视化工具,但是使用成本较高,检索速度较慢且价格昂贵。ELK 栈使用和维护成本较低,被广泛应用于日志数据的采集和存储应用中,但是 Logstash 存在启动慢、不稳定等缺点,需要进一步改进。上述研究的可视化分析多采用统计分析方法,且缺乏交互性。本文围绕上述问题展开研究工作,设计并实现了面向银行业务的交易监控系统,采用 Kafka<sup>[9]</sup>, Logstash<sup>[10]</sup>, Elasticsearch<sup>[11-12]</sup>和 Spark Streaming<sup>[13]</sup>等技术实现交易日志的采集和存储,通过机器学习方法对交易量进行预测并提供告警,完成可交互的监控和可视化分析任务。系统主要贡献如下:1) 基于已有开源日志采集、结构化存储和可视化技术方案,提出了采集过程和数据流处理过程的优化方案;2) 提出了交易量预测方法,能够在监控过程中有效提供异常告警;3) 提供了业务分类、趋势曲线、数据对比分析、具体交易信息定位和交易明细等丰富的可视化方法,提高分析效率;4) 设计了丰富的交互操作,包括交易系统间关联结构图的创建与管理(用于

主监控窗口,提高交易实时监控的用户体验)、时间控制、数据指标选择、交易查询等。

## 1 系统设计

### 1.1 数据描述

本文的样本数据来源于某银行网银渠道、EIX(enterprise information exchange server, 企业信息交换服务器)、外联前置和非税前置 4 个交易系统的日志数据。原始数据包含用户信息和交易信息,结合系统的分析目标以及用户隐私,本文只提取了其中的交易信息,并对数据进行了预处理,预处理后的数据共有 1 981.3 万条交易记录,时间跨度为 5 月。

交易信息主要包括交易卡号、交易时间、交易系统、业务编码、业务描述、交易渠道、交易金额、交易响应码(成功码、失败码)、响应码描述、交易延时等,本文针对这些交易信息的属性做了可视化分析。

### 1.2 功能设计

本文采用 B/S 架构设计,主要通过采集多个异构交易系统的日志并提取关键指标结构化存储后,实现了交易系统监控与交易数据可视化分析等功能。系统功能结构包括多系统联合监控模块、单系统监控与可视化分析模块、事件列表模块、统计报表模块、交易追踪模块和系统配置模块,如图 1 所示。

1) 多系统联合监控模块包括多系统关联性结构图创建子模块、多交易系统实时监控与告警子模块以及系统间交易数据对比分析子模块 3 个部分。

多系统关联性结构图创建子模块可根据用户自行构建的系统关联拓扑结构图,实时呈现自定义的监控主界面。

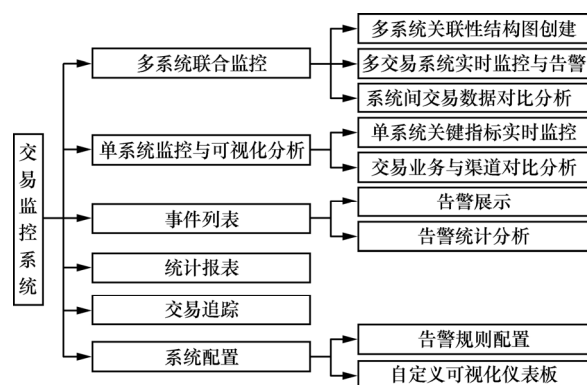


图 1 交易监控系统功能模块结构

Fig. 1 Diagram of transaction monitoring system function module structure

多交易系统实时监控与告警子模块的功能是显示单位时间内的系统名称、交易量、业务成功率、技术成功率和平均交易延时等交易指标信息,根据服务器的推送数据实时完成指标刷新,并对异常数据和服务器推送中断做异常标记,同时向相关人员发送告警信息。

系统间交易数据对比分析子模块可通过图表等可视化方式显示多个交易系统间的交易量、交易延时、交易成功率 3 个指标的对比结果,并提供直观分析环境。

2) 单系统监控与可视化分析模块包含单系统关键指标实时监控、交易业务与渠道对比分析 2 个功能。

单系统关键指标实时监控功能是对单个交易系统的交易量、交易延时、交易成功率这 3 大指标进行可视化展示,并对数据实时刷新以达到实时监控效果。

交易业务与渠道对比分析功能是以可视化的方式展示交易业务和渠道指标在交易中的分布差异,便于分析人员发现问题。

3) 事件列表模块包含告警展示和告警统计分析 2 个功能。

告警展示是在系统指标异常时进行告警,告警信息包含系统名称、异常指标类型、异常值、发生异常时间,该信息以列表的形式按告警时间显示给用户。该模块提供了交易系统和指标类型筛选功能,供分析人员集中了解特定告警信息。

告警统计分析可分别按交易系统和告警类型以图表等可视化的方式展示告警量的统计,供分析人员直观分析告警规律和趋势。

4) 统计报表模块提供了将监控数据快速生成报表的功能,分析人员可根据时间范围选择任意时间段的报表结果,如日报、月报等。

5) 交易追踪模块提供了关键字查询功能,分析人员可根据当前需要对业务、渠道、系统、金额、时间等已知交易属性进行查询,并根据查询结果快速定位到不明确交易的详细记录。

6) 系统配置模块包括告警规则配置和自定义可视化仪表板 2 个功能。

告警规则配置为分析人员提供了配置告警指标类型、频率、最大告警次数、阈值、告警时间范围等规则的功能,系统将根据给定规则提供告警。

自定义可视化仪表板为分析人员提供了自定义可视化分析扩展接口,系统根据交易数据分析需求提供大量组合的可视化仪表板接口,分析人员可按照添加规则配置接口,从而获取满足需求的可视化结果;

若没有接口可满足,则向系统管理员发送申请,管理员审核通过后将为其开放可视化接口。

### 1.3 架构设计

根据以上功能需求,本文提出了如图 2 所示的系统架构,系统由 Kafka 系统<sup>[10]</sup>、ELK 协议栈<sup>[12]</sup>、Spark Streaming 数据流处理器<sup>[14]</sup>、数据库存储服务器和 Web 应用服务器组成。

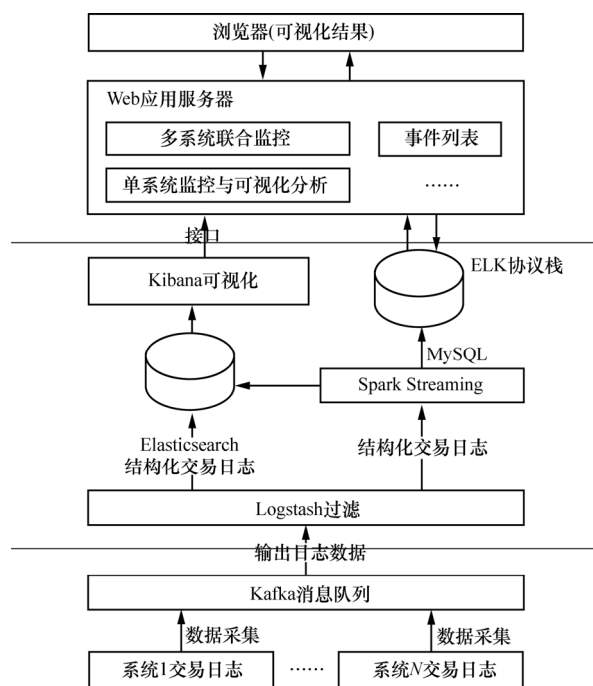


图 2 交易监控系统架构图

Fig. 2 Architecture transaction monitoring system

1) Kafka 系统。采用分布式订阅消息队列方式监听和收集各交易应用系统的交易日志数据,完成数据聚合和传输过程,保证采集过程中的高效性、易维护性和稳定性。

2) ELK 协议栈。由 Kafka 采集的数据传输至日志采集解析工具 Logstash, Logstash 通过正则规则对系统名称、交易时间、交易类型、交易机构、外联单位、交易时长、响应码、响应码描述、交易金额共 9 个指标进行过滤完成数据结构化后,分别存储至 Elasticsearch 和输出至 Spark Streaming; Elasticsearch 为 Kibana 提供数据监听和访问接口, Kibana 通过内部封装的多种图表工具,如表格、柱状图、饼状图、折线图等,对存储在 Elasticsearch 中的数据进行深入挖掘分析与可视化。

3) Spark Streaming 数据流处理器。Spark Streaming 负责对上述 Logstash 结构化的数据进行统计和预测等实时计算处理,最终将处理结果汇总后存

入 Elasticsearch 和 MySQL 数据库。

4) 数据库存储服务器。数据库存储服务器负责系统数据的持久化存储,主要存储用户信息、交易量预测分析结果以及为用户交互频繁的数据信息等。系统采用了较主流的 MySQL 数据库,MySQL 具有质量小、高效、易扩展和易维护的优点。

5) Web 应用服务器。用户通过 Internet 与 Web 应用服务器进行交互。Web 应用服务器采用 MVC(Model View Controller)模式将表示层、数据模型和控制器解耦合,表示层向用户提供友好的可视化界面,控制器用来接收用户提交的请求,将请求中传来的 json 数据转换成模型中对应的数据模型,然后调用相应的业务方法并根据调用结果选择返回的视图。本文表示层基于 MVC 的设计思想,采用 MVVM(Model-View-ViewModel)模式<sup>[15-16]</sup>将 HTML DOM 视图层(View)与数据模型(Model)分离,ViewModel 负责处理业务逻辑,并分别于 View 和 Model 双向通信,该模式使视图层功能单一化,用于大量交易日志的可视化中,极大提高了页面的加载和渲染效率。

## 1.4 核心模块设计

### 1.4.1 多系统联合监控

银行各异构应用系统之间的交易具有一定的关联性,如网银渠道内的交易信息会包含一部分外联前置和非税前置的交易流水。因此,基于多个关联系统的联合监控可以让用户更直观地了解全局监控信息,提高用户体验。本文通过对各类交易系统日志的分析,提取交易笔数、交易延时、技术成功率和业务成功率 4 个共性指标,为分析人员提供创建多系统关联性结构图的交互性过程,最后通过数学统计与机器学习等方法实现多系统的联合监控。

1) 创建多系统关联性结构图。本文采用基于 HTML5 的交互图形库 GOJS<sup>[17]</sup>来实现多系统关联性结构图的创建,GOJS 是一种模型-视图结构,视图是模型的可视化,模型是视图的数据体现,模型通过 2 个数组结构存储,分别为节点数组和连接线数组,节点数组存储应用系统的基本信息和关键指标数据,连线数组存储应用系统之间的关联关系。图 3 所示为多系统关联性结构图创建过程。

用户通过拖拽组件菜单将父容器组件拖拽到画布指定位置,添加应用系统名称后,客户端会向服务器发送请求,获取该系统的交易量、交易延时、技术成功率和业务成功率 4 个关键指标,然后系统将指标信息按系统规范放入子容器组件中,应用系统节点(NodeDataArray)创建完成。用户可通过连接线(LinkDataArray)绘制应用系统之间的拓扑关系,并

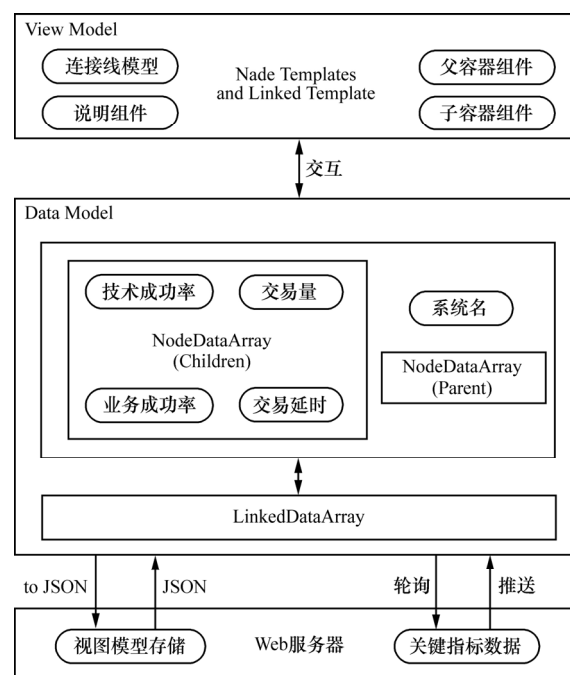


图3 多系统关联性结构图创建过程

Fig. 3 Creation process of multi-system correlation chart

为应用系统添加备注说明,完成拓扑图(view model)的创建,系统将创建的拓扑图结构(Data Model)存储在数据库中。用户完成拓扑图创建后,即可实时监控所有应用系统的交易状态。本文采用轮询方式监听服务器中指标数据变化,实时将返回数据通过刷新 Data Model 的方法更新节点模板(NodeTemplates)和连接线模板(LinkTemplate),以达到实时监控的效果。

2) 多系统联合监控与告警。本文采集的交易数据源是离散的,需经过 Spark Streaming 实时计算将各交易系统的交易数据统计和汇总,并有序地存储到 MySQL 数据库服务器中,Web 应用服务器以轮询的方式向 MySQL 请求实时数据并显示在多系统关联结构图中,以达到监控的目的。本文结合对时序数据交易量的实时预测和人为配置告警规则的干预提供告警机制,告警类型包括 4 个关键指标,其中交易延时、技术成功率和业务成功率的基线通过人为配置来定义,交易量的基线值通过机器学习方法的预测结果得到。

本文通过统计分析外联前置系统某一周内的交易流水数据,发现该系统的交易量和时间存在明显的相关性,并具有一定周期性,最后选定 Spark 决策树回归模型来进行交易量的预测。图 4 所示为外联前置系统 2017-06-04—12(包含上一个周日和下一个周一)的交易量趋势图。

从图 4 所示的趋势图可以发现:交易量在 1 d 内 8 点至 10 点持续增长,之后开始有下降趋势,下午 1 点到 3 点又一次持续增长后逐渐下降,凌晨左右交易量处于 1 d 中较低状态并持续到 8 点左右。1 周内工作日交易量总体较高,周末交易量相对较低。在节假日,交易量相对于非节假日较低,节假日结束后的第 1 d 的交易量也会相对较低。因此,本文提出了一种告警机制中的交易量预测方法,其过程如图 5 所示。

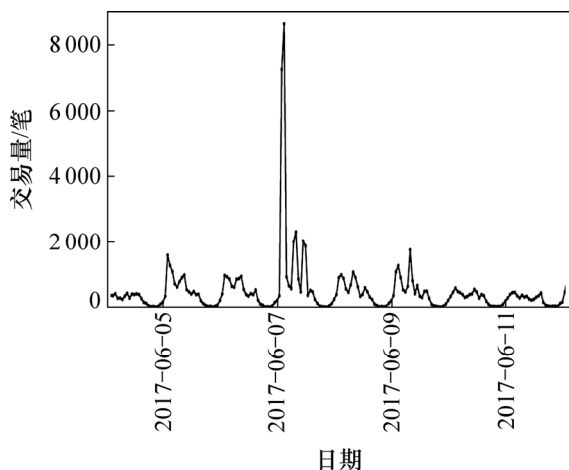


图 4 外联前置系统一周内交易量趋势图

Fig. 4 Trending chart of trading volume of outbound-front system within a week

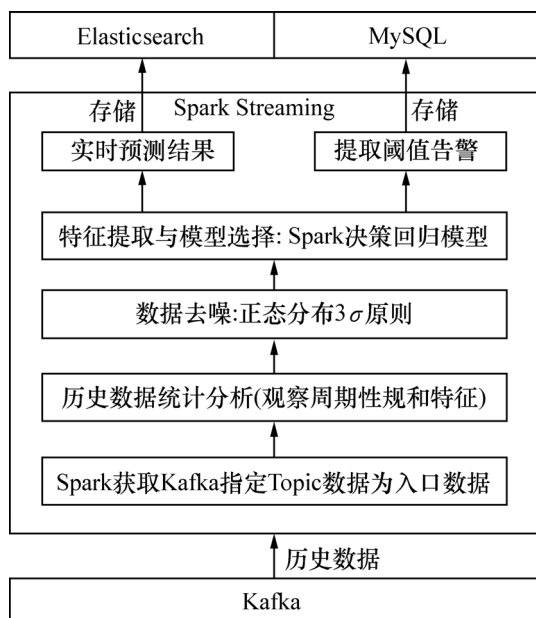


图 5 交易量预测过程

Fig. 5 Process of trading volume forecast

首先,将交易系统、1 周中的第几天、1 d 中的小时值、小时中的第几个十分钟、是否节假日、是否临

近节假日这个 6 个特征作为历史数据的特征标签。然后,通过进一步分析外联前置系统的交易量发现,在系统正常的情况下交易量在某个时段有激增情况,该激增点会对预测产生一定影响,使得对应时刻的预测值也会激增,而实际交易量远低于预测值产生误报警。所以本文根据正态分布的  $3\sigma$  原则对历史数据进行去噪。最后,对历史数据打上特征标签并去噪后,采用 Spark 决策树回归模型<sup>[18-20]</sup>对历史数据进行训练并将结果实时存储至 Elasticsearch,将需要告警的数据信息存储至 MySQL。

#### 1.4.2 单系统监控与可视化分析

本文采用 Kibana 可视化工具监听 Elasticsearch 中的交易数据,使用其内部丰富的图表库与 Web 应用服务器相结合实现数据的实时监控与可视化分析。图 6 所示为基于 Kibana 的交易数据可视化过程。

Kibana 可视化依赖 Elasticsearch 数据聚合的两种类型:量聚合和刻度聚合。量聚合用于面积图、饼图的分析结果,刻度聚合用于折线图、柱形图等以坐标轴为刻度的分析结果。Kibana 通过配置索引匹配模式实时监听 Elasticsearch 相关数据集,运用面积图,以线性时间为横轴、交易量为纵坐标、按响应码分类对交易量可视化;运用折线图,以线性时间为横轴、交易量为纵坐标、按交易分类对交易延时可视化;运用饼图对交易业务占比可视化;运用表格对交易详情可视化。以上指标的可视化图表是独立的 Visualize,最后将多个 Visualize 整合到 Dashboard 仪表板,生成 URI 接口,供 Web 应用服务器调用,分析人员可设定筛选指标和时间范围得到可视化的查询结果。

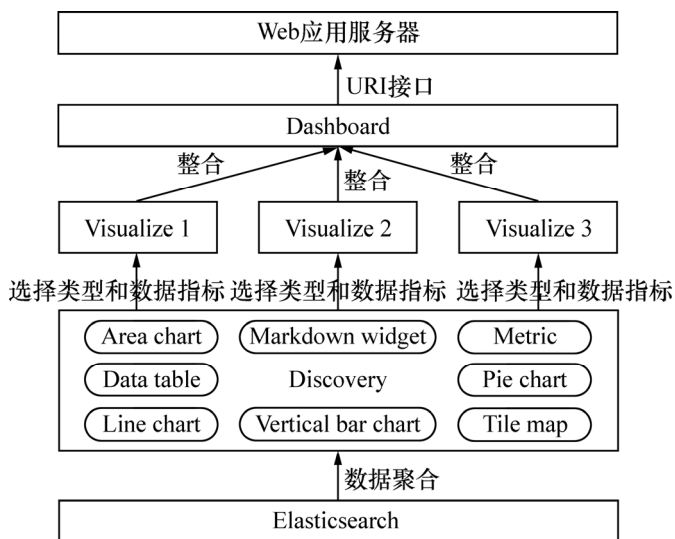


图 6 基于 Kibana 的交易数据可视化过程

Fig. 6 Transaction data visualization process based on Kibana

### 1.4.3 交易追踪

本文采用关键指标检索机制,为分析人员制定交易可视化结果的灵活查询。系统提供交易系统名称、交易金额、交易时间、交易描述(成功、失败等)、渠道(机构)、交易延时6个指标的查询接口,根据输入的查询条件进行过滤筛选,返回可视化结果,如图7所示。首先,Kibana根据查询指标制定3个Visualize(可视化图表),一是对交易量制定统计标签,二是对交易量趋势制定柱形图,三是对满足查询条件的数据制定详细列表。这3个Visualize组合生成Dashboard(可视化仪表盘)并创建格式规范的接口。然后,当用户请求查询时,Web服务器接收查询条件,根据条件拼接查询语句,调用Kibana接口,最后返回给用户可视化结果。

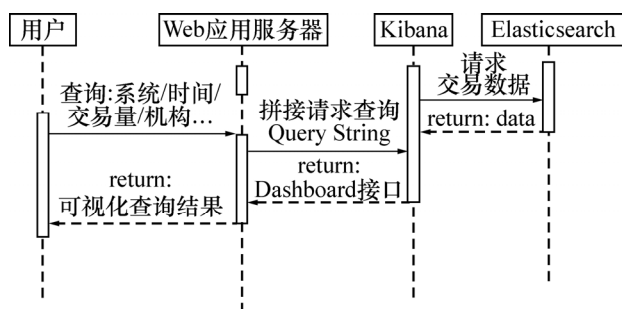


图7 交易追踪时序

Fig. 7 Timing diagram of transaction tracking

## 2 系统实现

本节根据系统设计,实现了面向银行业务的交易监控系统。分析人员可通过Web浏览器的主监控界面观察数据指标变化,当交易出现异常,监控主界面会给出告警标记,分析人员可点击进入出现异常的系统,对当前交易数据进行分析并定位异常交易区域,随后可在交易追踪中迅速定位交易明细,分析异常原因。

### 2.1 多系统联合监控

多系统联合监控模块为分析人员提供了根据异构系统间的关联性构建系统拓扑结构图的交互过程,以明确交易数据流向,系统则根据分析人员构建的关联关系显示实时监控内容。图8所示为针对网银渠道、EIX、外联前置和非税前置4个交易系统的实时监控仪表盘,仪表盘会在交易量异常和系统故障时标出告警标记。

### 2.2 交易量预测分析

图9所示为外联前置系统12h内的交易量预测结

果与实际交易量值的对比分析曲线。结果表明该段时间内交易量预测结果和实际值变化趋势基本拟合,预测值也在阈值浮动范围内,效果良好。

### 2.3 单系统监控与可视化分析

下面以外联前置系统4h内的交易监控数据可视化为例,对单系统监控与可视化模块的实现进行描述。如图10所示,可视化数据按照用户自定义的频率进行更新,默认每10s更新。交易量和交易延时以折线图按时间为横轴显示交易趋势,如图10(a)~(b)所示,交易量曲线中不同曲线代表不同交易响应码(成功或失败等描述);交易延时中不同曲线代表不同业务类型,表现不同业务在交易操作时所需时间的趋势;交易量在业务中的分布情况用饼图展示,如图10(c)所示。分析人员可按照需求选中感兴趣的区域,选中后系统会只提供所选区域的可视化结果,供用户做进一步分析,同时可查看监控视图所提供的交易明细表(见图10(d)),用户根据选择刷新频率和时间范围可切换实时监控模式和报表模式。

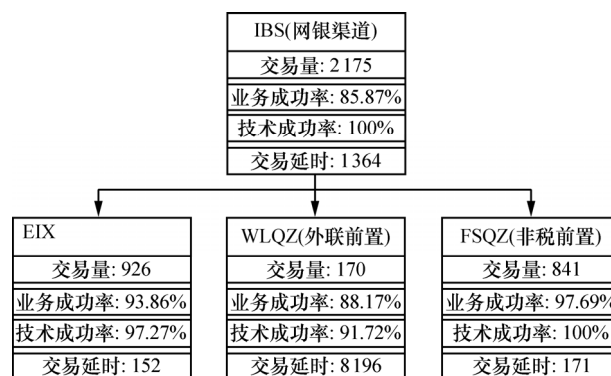
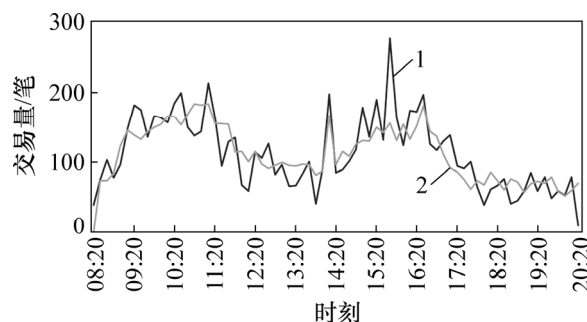


图8 系统实时监控图

Fig. 8 System real-time monitoring chart



1—交易量预测值; 2—交易量实际值。

图9 关键指标预测与实际值对比分析曲线

Fig. 9 Curve comparison of key indicators forecast and actual value

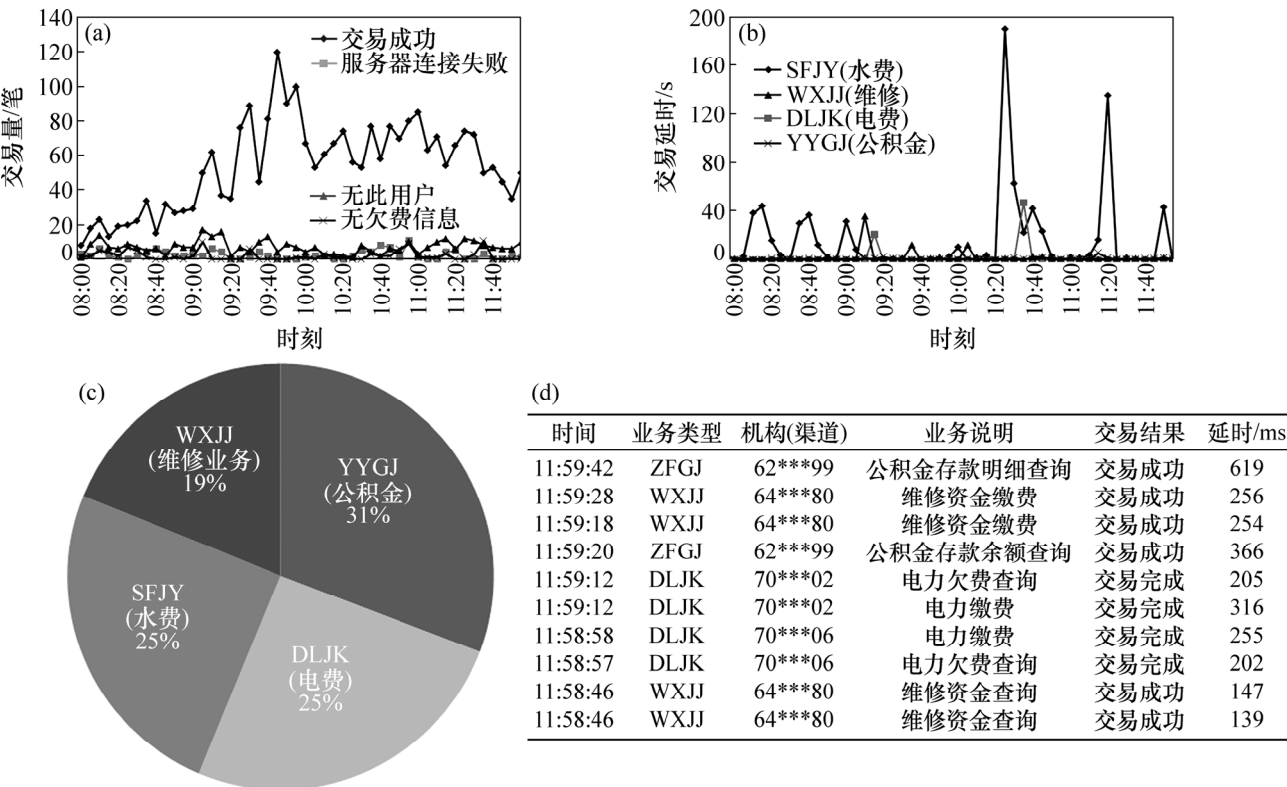


图 10 单系统实时监控与可视化分析

Fig. 10 Single-system real-time monitoring and visual analysis

3 结论

- 1) 本文设计的面向银行业务的交易监控可视化系统针对银行交易类型日志进行采集、结构化存储、多维度可视化分析与交易量预测和告警，为业务分析人员提供了方便直观的监控环境，解决了以前对交易问题定位慢、反馈慢以及分析和解决不及时等问题，极大地提高了工作效率。
- 2) 本文提出的系统架构和可视化方法可用于多种类型日志的采集与可视化分析中，具有较高的移植性、扩展性与复用性。

参考文献：

[1] CHAMARTHI, PRASAD S, MAGESH S. Application of splunk towards log files analysis and monitoring of mobile communication nodes[J]. International Journal of Applied Science and Engineering Research, 2014, 3(2): 478–483.

[2] BIRJALI M, BENI-HSSANE A, ERRITALI M. Analyzing social media through big data using infosphere biginsights and apache

flume[J]. Procedia Computer Science, 2017, 113: 280–285.

[3] BAI J. Feasibility analysis of big log data real time search based on Hbase and ElasticSearch[C]//Ninth International Conference on Natural Computation (ICNC). Shenyang, China: IEEE, 2013: 1166–1170.

[4] SANJAPPA S, AHMED M. Analysis of logs by using logstash[C]//Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications. Singapore: Springer, 2017: 579–585.

[5] BAJER M. Building an IoT Data Hub with Elasticsearch, Logstash and Kibana[C]//IEEE 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW). Prague, Czech Republic: IEEE, 2017: 63–68.

[6] SHAHZAD F, SHELAMI T R, SHAKSHUKI E M, et al. A review of latest web tools and libraries for state-of-the-art visualization[J]. Procedia Computer Science, 2016, 98: 100–106.

[7] BOONNAVASIN M, RATTANATAMRONG P. EnGeno: Towards enabling a medical genogram library for supporting home-visit patient diagnosis[C]//IEEE 5th Global Conference on Consumer Electronics. Kyoto, Japan: IEEE, 2016: 1–4.

[8] AIGNER W, MIKSCH S, SCHUMANN H, et al. Visualization of time-oriented data[M]. London: Springer, 2011: 45–68.

[9] KREPS J, NARKHEDE N, RAO J. Kafka: A distributed messagi

- ng system for log processing[C]//Proceedings of the NetDB. Athens, Greece: ACM, 2011: 1–7.
- [10] WANG G, KOSHY J, SUBRAMANIAN S, et al. Building a replicated logging system with Apache Kafka[J]. Proceedings of the Vldb Endowment, 2015, 8(12): 1654–1655.
- [11] GORMLEY C, TONG Z. Elasticsearch the definitive guide: a distributed real-time search and analytics engine[M]. Sebastopol California, USA: O'Reilly Media Inc, 2015: 61–127.
- [12] ANDREASSEN O Φ, CHARRONDIÈRE C, DE DIOS FUENTE A. Monitoring mixed-language applications with elasticsearch, logstash and kibana (ELK)[C]//Proceedings of the 15th International Conference on Accelerator and Large Experimental Physics Control Systems. Melbourne, Australia, 2015: 786–789.
- [13] LIAO Xinyi, GAO Zhiwei, JI Weixing, et al. An enforcement of real time scheduling in spark streaming[C]//Sixth International Green and Sustainable Computing Conference. Las Vegas, NV, USA: IEEE, 2016: 1–6.
- [14] BIFET A, MANIU S, QIAN J, et al. StreamDM: advanced data mining in spark streaming[C]//IEEE International Conference on Data Mining Workshop. Atlantic City, NJ, USA: IEEE, 2015: 1608–1611.
- [15] LI Xiaolong, CHANG Daliang, PENG Hui, et al. Application of MVVM design pattern in MES[C]//Cyber Technology in Automation, Control, and Intelligent Systems (CYBER). Shenyang, China: IEEE, 2015: 1374–1378.
- [16] SORENSEN E, MIKAILESC M. Model-view-viewmodel (MVVM) design pattern using windows presentation foundation (WPF) technology[J]. MegaByte Journal, 2010, 9(4): 1–19.
- [17] GoJS-Interactive Diagrams for JavaScript and HTML[EB/OL]. [2017–08–30]. <https://gojs.net/latest/index.html>
- [18] GULLER M. Machine Learning with Spark[M]. Berkeley, CA, USA: Apress, 2015: 153–205.
- [19] MENG X, BRADLEY J, YAVUZ B, et al. Mlib: machine learning in apache spark[J]. The Journal of Machine Learning Research, 2016, 17(1): 1235–1241.
- [20] GUAN Lei, HU Guangjun, WANG Zhuan. Research on network security situational awareness technology based on big data[J]. Netinfo Security, 2016, 16(9): 16–22.

(编辑 杨幼平)