

面向实时业务监控的流式数据分析方法研究

林云婷¹ 唐乐红^{*1} 黄长泉² 陈秋彤¹ 吴启晗¹

1. 阳光学院, 福建 福州 350015

2. 中科数创(厦门)智能科技研究院, 福建 厦门 361015

[摘要]针对传统批处理监控存在数据更新滞后、指标口径固化、乱序事件导致统计偏差以及突发流量下告警延迟升高等问题, 提出一种面向实时业务监控的流式数据分析方法。该方法以消息队列和分布式流处理引擎为基础, 构建数据接入、事件时间治理、增量指标计算、动态异常识别、结果服务和运行保障一体化链路; 通过水位线与迟到数据旁路机制提高窗口统计完整性, 利用可组合状态实现多粒度指标增量更新, 并结合动态规则、鲁棒统计阈值和告警抑制策略降低误报与告警风暴。仿真原型结果表明, 与定时批处理和基础流式方案相比, 本文方法在峰值负载下能够保持较低的端到端延迟, 并改善乱序窗口完整率和故障恢复效率。

[关键词]实时业务监控; 流式数据; Apache Flink; 事件时间; 增量计算; 异常告警

DOI: 10.64635/ja.2026.1415

中图分类号: TP274

文献标识码: A

Research on Streaming Data Analysis Methods for Real-Time Business Monitoring

Lin Yunting¹, Tang Lehong^{*1}, Huang Changquan², Chen Qiuotong¹, Wu Qihan¹

1. Yangguang University, Fuzhou, Fujian 350015, China

2. Zhongke Shuchuang (Xiamen) Intelligent Technology Research Institute, Xiamen, Fujian 361015, China

Abstract: To address problems in traditional batch-processing monitoring, such as delayed data updates, fixed indicator definitions, statistical deviations caused by out-of-order events, and increased alarm latency under burst traffic, this paper proposes a streaming data analysis method for real-time business monitoring. Based on message queues and distributed stream processing engines, the method constructs an integrated chain covering data access, event-time governance, incremental indicator calculation, dynamic anomaly identification, result services, and operational support. Window statistics completeness is improved through watermarking and a bypass mechanism for late-arriving data. Multi-granularity incremental updating of indicators is realized by using composable states, while false alarms and alarm storms are reduced through dynamic rules, robust statistical thresholds, and alarm suppression strategies. The simulation prototype results show that, compared with scheduled batch processing and basic streaming schemes, the proposed method can maintain lower end-to-end latency under peak loads and improve out-of-order window completeness and fault recovery efficiency.

Keywords: real-time business monitoring; streaming data; Apache Flink; event time; incremental computing; anomaly alarm

引言

数字化业务系统持续产生订单、交易、库存、设备状态和应用日志等事件数据。对于运营调度、风险识别和故障处置而言, 数据价值通常随时间衰减, 若仍采用分钟级或小时级离线汇总, 异常发现与决策执行之间将形成明显时差。已有工业大数据平台研究表明, 将 Kafka 与 Flink 等组件用于数据接入、计算和查询预警, 能够缩短数据处理链路并增强工业现场的实时分析能力^[1]。然而, 实时业务监控并非简单地将离线 SQL 迁移到流处理引擎, 还需

同时解决多源事件语义不一致、事件乱序、指标状态持续增长、规则频繁变化以及高峰负载下系统稳定性等问题。

现有实时监控系統常見兩類不足: 一是偏重技術組件拼接, 缺少從業務指標定義到異常處置反饋的閉環; 二是將固定閾值作為主要判斷依據, 難以適應周期波動、業務增長和分群差異。為此, 本文提出一種面向實時業務監控的流式數據分析方法, 主要工作包括: 建立統一事件模型與事件時間治理機制; 設計可組合的增量指標計算方法; 構建動態規則與統計異常融合的識別策略; 利用檢查點、

幂等输出和弹性调度保证处理连续性。

1 实时业务监控需求与总体架构

实时业务监控的核心目标可概括为“看得见、算得准、报得快、可追溯”。“看得见”要求系统接入数据库变更、消息事件、设备采样和日志等异构数据；“算得准”要求统一指标口径，并正确处理重复、缺失与乱序数据；“报得快”要求从事件产生到告警输出保持稳定的低延迟；“可追溯”则要求保存原始事件、计算版本和告警处置记录。云原生实时数据平台的相关研究指出，作业全生命周期管理、状态优化和弹性调度是降低实时任务运维负担的重要手段^[2]。因此，本文将监控方法划分为数据源层、接入缓冲层、流式分析层、结果存储层和业务应用层，如图 1 所示。

数据源层产生业务事实与运行状态；接入缓冲层利用 CDC 或日志采集组件完成数据捕获，并通过 Kafka 按业务键分区，隔离上游波动；流式分析层负责清洗、时间治理、维度关联、窗口聚合、异常识别和状态管理；结果存储层分别保存明细、聚合结果与热点指标；业务应用层提供实时大屏、告警推送和查询接口。处置结果反向写入规则与标签体系，使阈值、抑制周期和责任路由能够持续修正。



图 1 面向实时业务监控的流式数据分析方法架构

2 流式数据分析方法

2.1 统一事件模型与数据预处理

为避免不同系统字段含义不一致，定义统一事件 $E = \langle id, type, key, eventTime, processTime, dimensions, measures, version \rangle$ 。其中 id 用于幂等去重， $type$ 表示事件类别， key 用于分区和状态定位， $eventTime$ 为业务实际发生时间， $processTime$ 为进入计算引擎的时间， $dimensions$ 和 $measures$ 分别保存维度与度量， $version$ 记录模式或规则版本。数据进入流处理层后，依次执行格式校验、字段标准化、重复事件过滤、敏感字段脱敏和维度补全。对于低频变化的维度数据，采用广播状态；对于高频变化的账户、商品或设备状态，采用异步 I/O 或变更日志流关联，防止同步查询阻塞主链路。

2.2 事件时间与乱序治理

业务事件经过网络、缓存和重试后可能晚于其后产生的事件到达。若直接按处理时间统计，会造成跨窗口错位。相关研究表明，延迟容忍度应依据数据流的乱序特征设置，过小会丢失迟到事件，过大则会增加计算等待时间^[3]。本文采用事件时间窗口，并按数据源维护自适应水位线：在滑动观测区间内计算到达延迟分位数，以延迟的高分位值加安全裕量作为允许乱序时间。正常迟到事件在水位线推进进入原窗口；超过阈值的事件进入旁路流，由补偿任务修正结果并记录修订版本。该机制在时效性与窗口完整性之间形成可配置平衡。

2.3 多粒度增量指标计算

实时监控指标通常同时包含秒级瞬时值、分钟级趋势值和小时级累计值。若每次窗口触发均扫描全部明细，计算量将随窗口长度快速增长。本文将指标拆分为可组合状态，包括计数、求和、最值、均值的二元状态以及近似去重状态。对可加指标，窗口状态按“旧状态+新增事件-过期事件”更新；对均值类指标同时维护样本数与总和；对去重指标采用分桶位图或概率集合，并根据业务允许误差确定精度。设第 t 时刻新增事件集合为 A_t 、移出窗口的事件集合为 R_t ，则可加指标状态更新为： $S_t = S_{(t-1)} + \sum f(A_t) - \sum f(R_t)$ 。通过状态复用，1min、5min 和 1h 指标可由基础粒度状态逐级合并，减少重复计算。

2.4 动态规则与异常融合识别

固定阈值适合边界明确的业务规则，但面对明显周期性或分群差异时容易误报。已有基于 Flink 与规则引擎的监控告警研究证明，动态更新告警口径、阈值、格式和推送方式能够提高系统灵活性^[4]。本文将异常识别划分为规则判定和统计判定两条支路。规则支路处理余额小于零、库存超卖、接口错误率超过上限等确定性条件；统计支路对订单量、转化率、响应时间等波动指标计算中位数与中位绝对偏差，得到鲁棒标准分数，并结合环比偏离度判定异常。两类结果经融合评分后进入分级告警模块。为控制告警风暴，系统设置持续时间、恢复阈值、冷却时间和同类事件聚合键，仅在异常持续或等级升级时再次推送。

2.5 状态可靠性与负载自适应

流式任务需要长期运行，状态规模、热点键和突发流量会影响稳定性。针对上下游负载不均问题，相关研究采用负载预测和容器伸缩方式调整 Flink 资源，能够降低等待时间并提高计算效率^[5]。本文从三个方面实施运行保障：第一，采用周期检查点和外部化保存，故障后从一致性状

态恢复；第二，输出端使用事务提交或业务主键幂等写入，避免重放导致重复结果；第三，监测反压比例、消费滞后、CPU 利用率和状态增长率，当连续多个观测周期超过阈值时提高并行度，负载回落后延迟缩容。对于热点键，先进行局部随机拆分，再在下游合并，防止单个分区成为瓶颈。

3 仿真原型与结果分析

3.1 实验设计

为验证方法可行性，构建三节点仿真原型，每个节点配置 8vCPU、16GB 内存和千兆网络，软件链路由 Kafka、Flink 和 ClickHouse 组成。测试数据模拟订单、支付、库存和应用日志，共生成约 1000 万条事件，平均事件大小约 1KB；常态输入速率为 1.2 万条/s，并设置两段突发流量，峰值接近 4 万条/s；5%的事件附加 0~8s 随机到达延迟。对比方案包括 5s 定时批处理、仅采用固定窗口与固定并行度的基础流式方案，以及本文方法。评价指标为平均端到端延迟、P95 延迟、峰值处理能力、乱序窗口完整率和故障恢复时间。

3.2 结果与分析

表 1 给出了仿真原型结果。定时批处理的窗口统计较完整，但其延迟主要受轮询周期限制，无法满足秒内告警需求。基础流式方案在常态负载下具有较低延迟，但突发阶段因固定并行度产生反压，P95 延迟明显上升，且部分迟到事件被分配到错误窗口。本文方法通过自适应水位线、增量状态和弹性并行控制，将平均延迟降至 103ms，P95 延迟控制在 329ms；乱序窗口完整率达到 99.5%，故障恢复时间缩短至 10.8s。实时预警系统研究同样表明，采用流处理框架组织业务判断与告警输出有利于增强预警时效性^[6]。

表 1 不同数据处理方案的仿真结果对比

方案	平均延迟/ms	P95 延迟/ms	峰值处理能力/(万条·s ⁻¹)	乱序窗口完整率/%	恢复时间/s
5s 定时批处理	2586	4978	2.41	99.8	33.6
基础流式方案	193	1435	3.60	92.6	24.9
本文方法	103	329	4.60	99.5	10.8

注：数据来自受控仿真原型，用于比较方法趋势。

从结果看，本文方法的优势并非来自单一算子，而是数据缓冲、时间语义、状态计算、规则识别和运行控制的协同。增量计算降低了长窗口重复扫描，水位线减少了乱序误差，弹性调度则避免队列持续堆积。需要说明的是，

表 1 为受控仿真环境结果，其数值用于验证方法趋势，不应直接替代具体企业的生产压测结论。

4 讨论

在工程应用中，方法参数需结合业务代价调整。金融风控更关注漏报成本，可适当延长水位线并提高异常召回；运维监控更关注告警时效，可采用较短窗口和更严格的冷却规则；经营分析则需优先保证口径一致和历史可追溯。面向 Flink 流式分析的高吞吐研究指出，乱序耦合、负载不均和跨节点数据交换均可能限制系统扩展能力^[7]。因此，大规模部署时还应依据键分布、状态大小和网络交换量进行算子链拆分与分区优化。本文方法目前主要采用统计阈值和规则融合，尚未充分利用多变量因果关系；后续可引入在线学习模型，并通过概念漂移检测自动更新基线，同时建立指标血缘与规则版本审计机制。

5 结论

本文针对实时业务监控中更新滞后、乱序偏差、重复计算、规则固化和高峰反压等问题，提出一套流式数据分析方法。该方法以统一事件模型为基础，通过事件时间和自适应水位线治理乱序数据，利用可组合状态实现多粒度增量指标，融合动态规则与鲁棒统计识别异常，并以检查点、幂等输出和弹性调度保障持续运行。仿真原型表明，该方法能够在突发负载下兼顾低延迟、窗口完整性和故障恢复效率。后续研究将面向真实行业数据开展长期测试，并进一步研究在线模型更新、跨区域容灾和资源成本约束下的自适应优化。

[参考文献]

- [1]赵润发,姜渊胜,叶枫,等.基于 Flink 的工业大数据平台研究与应用[J].计算机工程与设计,2022,43(3):886-894.
- [2]郭炜,傅正斌,王新东,等.云原生实时数据湖平台设计与实现[J].计算机技术与发展,2026,36(2):188-194.
- [3]施国欢,宋吉,李江华.不同特征的流数据对 Flink 性能影响研究[J].计算机科学与应用,2022,12(11):2599-2607.
- [4]陈丽,杨大龙,高中纤,等.基于 Flink+QLEXPRESS 处理海量数据的实时监控告警系统[J].信息通信技术,2022(1):64-69.
- [5]艾力卡木·再比布拉,甄妞,黄山,等.基于深度学习的容器化 Flink 上下游负载均衡策略研究[J].大连民族大学学报,2023,25(1):47-52.
- [6]韦统边,吴江波,苏德,等.基于 Flink 实时预警功能的实现[J].汽车电器,2023(8):49-50,53.
- [7]秦政,许利杰,陈伟,等.面向 Apache Flink 流式分析应用的高吞吐优化技术[J].软件学报,2025,36(7):3184-3208.