



基于 FKS 的信令采集与监控技术

汪保友, 姚健, 张正卿

(中国联合网络通信有限公司上海市分公司, 上海 200050)

摘要: 运营商信令数据具有丰富的应用价值, 但其价值时效性强, 随着时间的流逝而快速衰减。为了更好地服务于实时营销场景, 必须有效提升信令数据的实时处理及分析能力。首先介绍某运营商基于 FKS 的信令采集处理方案, 该方案综合发挥了 Flume、Kafka、Spark Streaming 三组件的各自优势, 对运营商打造实时决策中心, 起着重要支撑作用。此外, 提出针对 Kafka 中消费者组对各主题 (topic) 消费情况的监控方案, 该方案对实时流采集故障及时预警提供了有效的手段, 在某运营商大数据平台运营实践过程中, 有效减少信令消息队列拥堵现象, 规避信令数据丢失风险, 对实际生产运营支撑起着良好的保障作用。

关键词: 实时流; 监控; Kafka; 大数据; 流计算

中图分类号: TP311

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2018096

Signaling collection and monitoring technology based on FKS

WANG Baoyou, YAO Jian, ZHANG Zhengqing

Shanghai Branch of China United Network Communications Co., Ltd., Shanghai 200050, China

Abstract: The signaling data of telecom operators, which have a wide range of applications and great of values, is big data in real sense. In order to serve the real-time marketing scenarios better, the real-time processing and analysis ability of signaling data must be improved effectively. Firstly, the scheme of signaling collection and processing based on FKS was introduced, which combined the advantages of components such as Flume, Kafka and Spark Streaming, and played an important supporting role for operators to build real-time decision center. Furthermore, the Kafka monitoring program for topic consumption of consumer groups was proposed, which provided effective means for warning faults of real-time flow collection. The monitoring solution could effectively reduce the message queue congestion, avoid the risk of data loss, and also could obtain good application effects in practice.

Key words: real-time streaming, monitoring, Kafka, big data, stream computing

1 引言

电信运营商处在大数据产业链的传输与交换的中心地位, 拥有庞大的活跃用户群体和数据资产。其中, 信令数据尤其庞大, 以 700 万活跃用户

为例, 来自运营支撑系统 (operation support system, OSS, 以下简称 O 域) 的信令数据每天就有大约 15 TB, 可以说, 通信信令是真正意义上的大数据。O 域数据具有体量大、实时强、内容精细、贴近网络等特点, 其价值时效性很强, 且其

收稿日期: 2017-07-16; 修回日期: 2018-01-20



价值随着时间的流逝而快速衰减,为了更快地响应市场需求,更好地服务于实时营销场景,必须有效提升信令数据的实时处理及分析能力。Hadoop 分布式并行计算框架,通过 HDFS 核心组件,为用户提供高容错性和高伸缩性的海量数据的分布式存储;通过另一核心组件 MapReduce,为用户提供逻辑简单、底层透明的并行处理框架。它可以解决传统关系型数据库关于海量数据难以处理的问题;但由于其机制是先进行数据积累,再进行数据拆分和聚合计算,难以满足数据实时性处理要求。流式计算框架 Flume、Kafka、Spark Streaming、Storm 等,通过管道流、消息队列等事件机制,接收到消息后马上处理,可以很好地解决数据实时处理问题。但同时实时流信令采集监控的问题显得比较突出,传统运维模式是根据客户投诉或网络显性故障,被动地解决问题,无法做到对故障的主动预判,所以永远都是慢“半拍”的被动式维护。解决客户感知不佳的问题迫在眉睫,无论用户是否投诉,先于客户发现问题、解决问题,是电信运营商亟待解决的难题。

本文提出基于 FKS (Flume、Kafka、Spark Streaming 的简称) 的信令采集处理方案;并针对 Kafka 中消费者组对各 topic (主题) 消费情况,提出有效可行的监控方案,实现了实时流采集监控可视化,方便了运维监控和预警提醒,在实践中取得良好的效果。

2 基于流式计算框架的信令采集方案

2.1 相关定义

本文采集的数据源是网运侧 A 口、S1-MME 口、Gn 口等信令,监控内容主要介绍的是 Kafka 集群中消费者组对信令流的消费情况监控。为说明方便,下面列出信令和 Kafka 组件的基本概念。

信令是通信网络交流的语言,根据其内容和处理过程,有如下几个基本概念。

- CDR (call detail record, 呼叫详细记录) /

TDR (transaction detail record, 事务详细记录) 信令数据: 是指基于全量信令数据进行处理后,生成的供信令检测类应用使用的呼叫及事务详细记录。

- 业务信令数据: 是根据应用的需要对 CDR/TDR 信令数据进行解析、处理,从中提取出对业务应用所需的信令事件数据,例如呼叫事件、位置更新事件、漫游事件、开关机事件等。
- A 口信令: 通俗来说, A 口信令也称 2G/3G 位置信令。A 接口是指 BSC (基站控制器) 和 MSC (移动业务交换中心) 之间的接口,是无线、核心网的交界。A 口信令过程主要有: 开机过程、周期性位置更新过程、主/被叫、短信收发、切换等,内容包括用户位置信息、业务类型信息等。
- S1-MME 信令: 通俗来说, S1-MME 信令也称 4G 位置信令。内容包括: 控制面的网络性能指标 (attach、EPS、TAU 等)、用户信息 (TMSI/IMSI)、终端信息 (IMEI)、承载信息等。
- Gn 信令: Gn 接口是同一 PLMN 中 SGSN 与 SGSN 间以及 SGSN 与 GGSN 间的接口,通俗来说, Gn 信令即上网详单。内容包括: 访问网站、URL 地址、用户信息 (MSISDN/IMSI)、终端类型、用户位置区、协议类型等。

Kafka 采用 “producer-broker-consumer (生产者—缓存代理—消费者)” 模型,实现 “消息队列” 的传送,有如下一些基本概念。

- topic (主题): 特指 Kafka 处理的消息源的不同分类标识,用于保证 producer 以及 consumer 能够通过该标识进行对接。
- partition (topic 物理上的分组): 一个 topic 可以分为多个 partition, 但一个 partition 一定属于一个 topic。每个 partition 是一个有序的队列。partition 中的每条消息都会

被分配一个有序的 ID (offset)。

- offset: 记录某个 consumer group 在某个 partition 中当前已经消费到达的位置。
- message (消息): 是通信的基本单位, 每个消息和数据生产者可以向一个 topic (主题) 发布一些消息。
- producer (消息和数据生产者): 向 Kafka 的 topic 发布消息的过程。
- consumer (消息和数据消费者): 订阅 topics 并处理其发布的消息的过程。
- consumer group (消费者组): 为了便于实现 MQ 中的多播、重复消费等引入的概念。如果 consumer A 以及 consumer B 同在一个 consumer group, 那么 consumer A 消费的数据 consumer B 就无法消费了。即同一 consumer group 中的所有 consumer 使用一套 offset。
- broker (缓存代理): Kafka 集群中的服务器节点。Kafka 集群中的一台或多台服务

器统称为 broker。

2.2 基于 FKS 的信令实时采集处理架构

考虑到业务清晰化、功能模块化需求以及实时采集、处理时延、实时场景营销、离线分析等实际生产与应用场景, 信令实时采集处理可以模块化分为: “数据采集—数据接入—流式计算—数据沉淀” 4 个模块。目前, 业内比较流行的实时日志分析 ELK (Elasticsearch、Flume、Kafka 的简称) 三件套, 提供了基于消息的数据同步机制的新思路。借鉴 EFK 三件套的提法, 提出基于 FKS 的信令实时采集处理架构, 如图 1 所示。图 1 中, 数据采集, 利用 Flume 集群, 负责从各个节点上实时收集网运侧传送的 A 口、S1-MME 口、Gn 口等信令流数据, 并用 “push-based (基于推送)” 方式, 向各接收方传送传输数据。数据接入, 使用 Kafka 集群, 通过 “pull-based (基于提取)” 方式订阅取数, 缓存数据; 并通过实时决策引擎, 支撑实时场景营销、事件触发营销等实时应用服务。流式计算, 采用 Spark Streaming 进行数据实

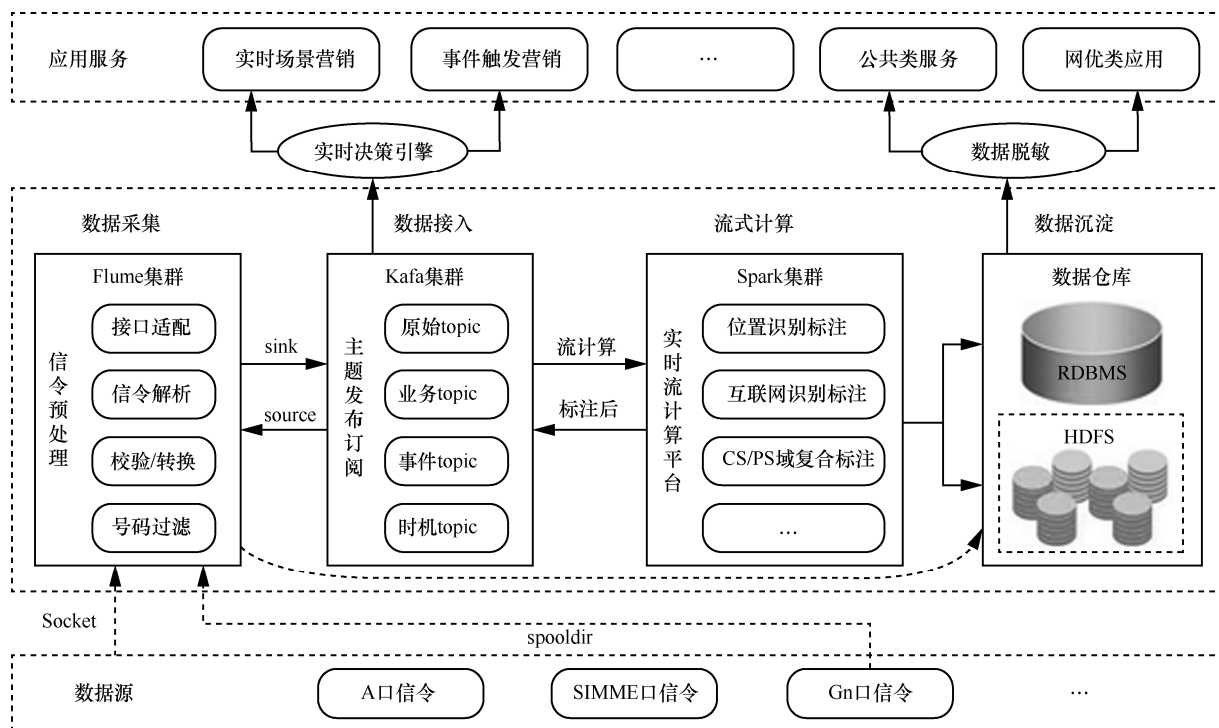


图1 信令采集处理模块化框架



时计算, 对互联网行为识别标注、位置更新识别标注、CS/PS 域复合标注等。数据沉淀, 采用 HDFS + RDBMS, 对信令流加工处理后的数据, 沉淀成数据资产, 为离线统计分析提供服务。

在软件部署时, 同一物理设备可以部署多个功能模块, 图 2 列出集群设备拓扑。从图 2 中可看出, Kafka 集群可以复用 HDFS DataNode 设备; Spark 集群可以复用 Flume 集群设备, 也可复用 Kafka 集群设备等。3 台管理节点安装了 Flume MonitorServer、HDFS NameNode、Spark JDBCServer、Zookeeper、YARN ResourceManager 等。

2.3 流式计算框架工作机制

流式计算技术不是批量分析任务的快速版本, 大部分流处理程序是用来实现核心的业务逻辑, 而不是用于对业务进行分析。时间推理工具, 是流式计算的基础, 绝大部分流计算应用场景都对时间有严格的要求。下面对 Flume、Kafka、Spark Streaming 三组件的工作机制做简单介绍。

Flume 是 Cloudera 提供的一个分布式、高可用的、高可靠海量日志聚合系统, 它定制各类数

据发送方, 用于收集数据; 同时提供对数据的简单处理功能, 并写到各种数据接收方。Flume 追求的是数据来源的多样性及数据流向的多样性。Flume 实现机制是“管道流”, 采用“source-channel-sink (数据源-通道-输出地)”模型, 以 agent (代理) 为最小的独立运行单位, 一个 agent (代理) 就是一个 JVM。agent 由 source、sink 和 channel 3 个组件构成, Flume 内置了大量内置的 source、channel 和 sink 类型。不同类型的 source、channel 和 sink 可以自由组合。比如: channel 可以把事件暂存在内存里, 也可以持久化到本地硬盘上。sink 可以把日志写入 HDFS、HBase, 甚至另外一个 source 等。

Kafka 是 LinkedIn 提供的一个高吞吐、分布式的、分区化的、可复制的日志处理框架。适合处理海量日志发布订阅, 提供消息磁盘持久化、支持物理分片存储、多组消费等特性。Kafka 追求的是高吞吐量、高负载, 承担高速数据总线的作用。producer (生产者) 采用 push (推送) 模式将消息发布到 broker (缓存代理), 每条消息都被

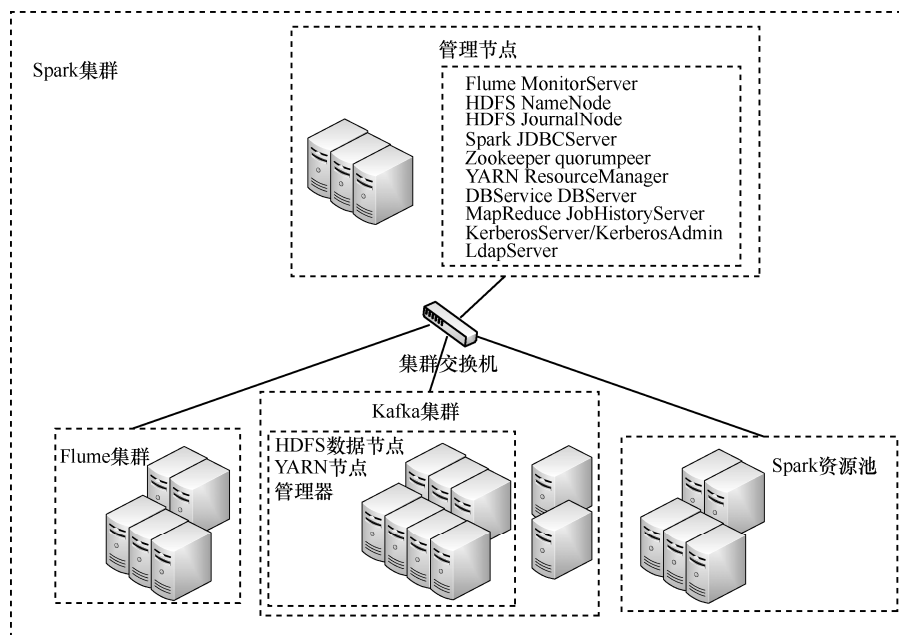


图 2 集群设备拓扑

append (添加) 到 partition (分组) 中, 顺序写磁盘, 保障 Kafka 的吞吐率。broker 按 topic 保存消息队列。每个 topic 下可以有一个或多个 partition, 每个 partition 物理上对应一个文件夹 (该文件夹存储该 partition 的所有消息和索引文件), 同一个 partition 可能会有多个 replica (副本), 在这些 replica 之间选出一个 leader (领导者), producer 和 consumer (消费者) 只与这个 leader 交互, 其他 replica 作为 follower (跟踪者) 从 leader 中复制数据。consumer 采用 pull (提取) 模式从 broker 中读取数据, consumer 可自主控制消费消息的速率; 也可以自己控制消费方式, 即可批量消费也可逐条消费。一个 partition 只能被同一个 group (组) 内的一个 consumer 消费, 但是多个 group 可以同时消费这个 partition。图 3 列出了 Kafka 的拓扑结构。

Spark 是一个类似于 MapReduce 的分布式计算框架, 其核心是弹性分布式数据集 (resilient distributed dataset, RDD), 提供了比 MapReduce 更丰富的模型, 可以在内存中对数据集快速进行多次迭代, 以支持复杂的数据挖掘算法和图形计算算法。Spark Streaming 是一种构建在 Spark 上的实时计算框架, 它扩展了 Spark 处理大规模流

式数据的能力。Spark Streaming 以时间片 Δt (比如 1 s) 为单位, 将输入数据分成一段段的离散数据流 (Discretized Stream, DStream), 这些流具有与 RDD 一致的核心数据抽象, Spark Streaming 会把每块数据作为一个 RDD, 并使用 RDD 操作处理每一小块数据。每个块都会生成一个 Spark job 处理, 最终结果也返回多块。Spark Streaming 提供了丰富的 API 和基于内存的高速计算引擎, 它的 output (输出) API 操作可将 DStream 数据输出到外部数据库或文件系统中, 因此, 它适合一些需要历史数据和实时数据结合分析的应用场合, 为用户结合流式处理、批处理和交互查询等应用提供方便。当然, 对于实时性要求不是特别高的应用也能完全胜任。因为需要在不同粒度上对信令数据进行加工分析, 涉及聚合、去重、连接等较为复杂的统计需求, 若是采用 Storm 这样的流式框架, 实时性虽可以得到保证, 但需求的实现复杂度也大大提高了。Spark Streaming 在实时性与复杂统计需求之间的权衡中找到了一个平衡点, 能够满足大多数用户的流计算需求。Spark Streaming 支持 3 种应用场景, 包括无状态操作、状态操作、window (窗口) 操作。“无状态操作”, 只关注当前新生成的小批次数据, 所有计算都只

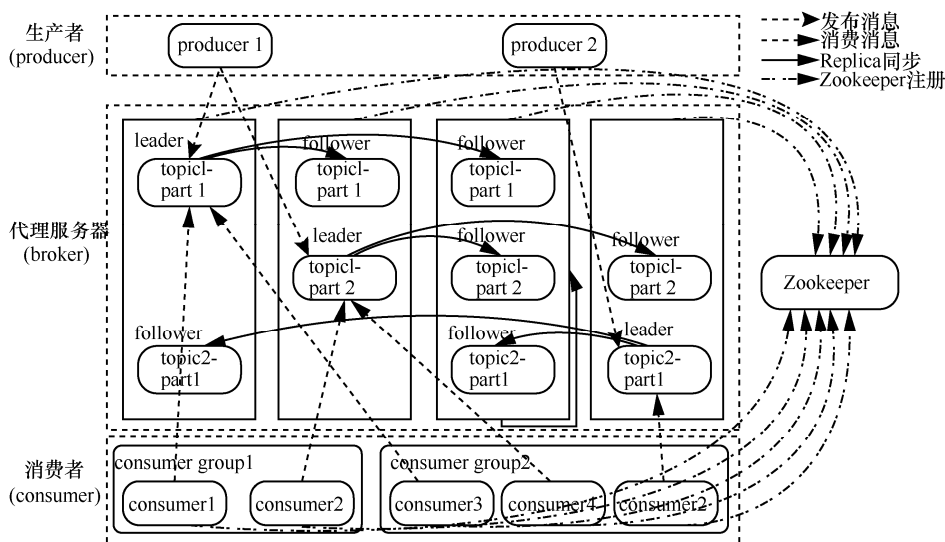


图 3 Kafka 拓扑结构



是基于这个批次的数据进行处理；“状态操作”，是基于全量数据进行操作，除了当前新生成的小批次数据，还需要用到以前生成的所有历史数据，新生成的数据与历史数据合并成一份流水表的全量数据；“window 操作”，允许在一个滑动窗口数据上应用 transformation（转换）算子，如图 4 所示。

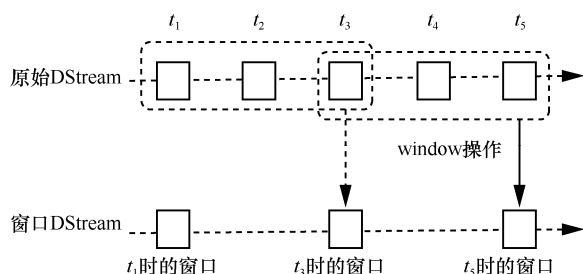


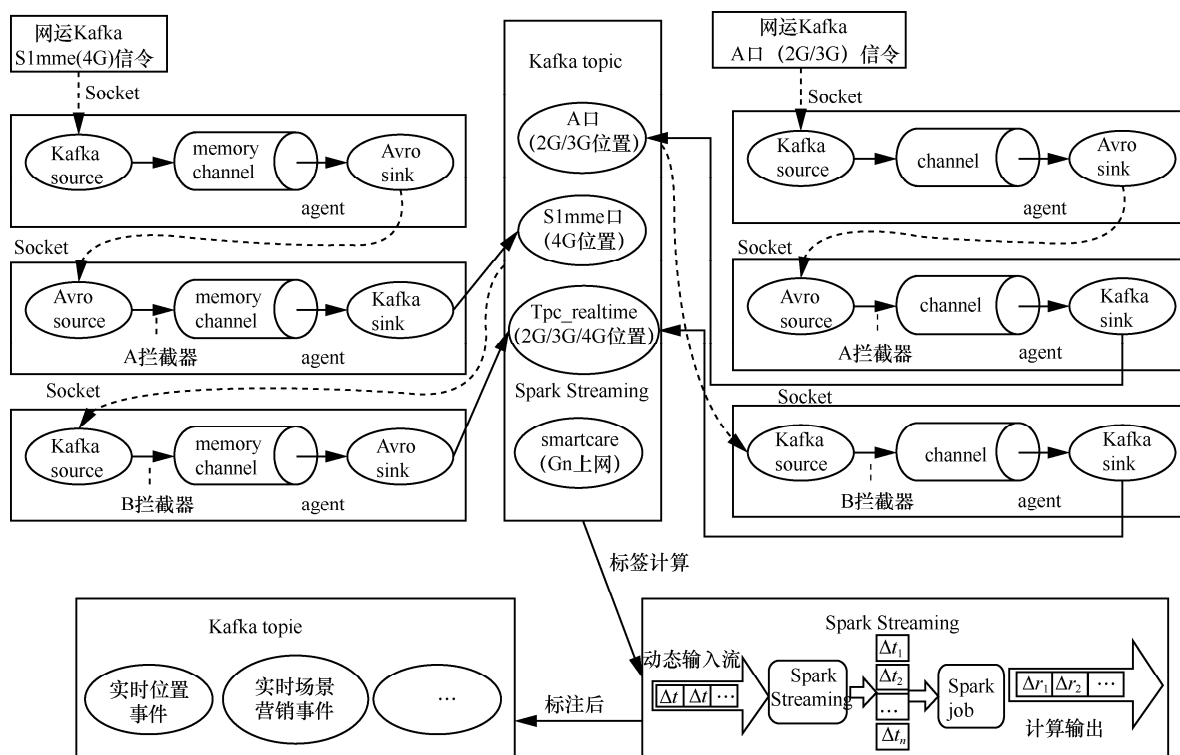
图 4 Spark Streaming window 操作过程

图 4 中，右边箭头向下粗实线表示窗口当前的滑动位置，中间箭头向下细虚线表示前一次窗口位

置，窗口每滑动一次，落在该窗口中的 RDD 被一起同时处理，生成一个窗口 DStream（windowed DStream），窗口操作需要设置两个参数：窗口长度（window length），即窗口的持续时间；滑动间隔（sliding interval），窗口操作执行的时间间隔。这两个参数必须是原始 DStream（original DStream）批处理间隔（batch interval）的整数倍。

2.4 实时决策中心业务流程

图 5 列出了某运营商实时流信令采集处理流程。考虑到信令解析、时间格式转换、区分主被叫、号码标准化、号码过滤以及接口适配等因素，Flume 采用级联处理模式。经过 Flume 预处理后的实时信令，形成 Kafka 的原始 topic。Spark Streaming 对这些原始 topic 进行实时标签计算，再回写给 Kafka，形成业务 topic、事件 topic、时机 topic，比如实时位置事件、实时营销场景事件等。这些 topic 经过实时决策引擎，支撑着实时场



注 1：A 拦截器作用：解密等

注 2：B 拦截器作用：时间格式转换、区分主被叫、号码标准化、号码过滤等

图 5 实时信令流采集处理流程

景营销、事件触发营销等实时应用服务。

图6列出某运营商实时决策中心对异步信令事件营销场景的支撑流程。适用于原始事件量较大、需要进行清洗/转换/过滤或复杂计算来形成触发事件、基于用户被动接触渠道进行营销推荐的场景,使用HTTP、SDTP、JMS、文件等流数据接口。

图6的业务流程处理过程描述如下。

(1) 信令系统产生信令事件。

(2) Flume 实时/准实时采集信令事件。

(3) Flume 将原始信令事件写到 Kafka。

(4) Flume 从 Kafka 读取原始信令事件,对原始事件进行清洗/转换/过滤,并将处理后事件写回 Kafka。

(5) PME 位置事件识别: 从 Kafka 读取位置信令事件,结合预配置位置区域信息,识别驻留/进入/离开等位置事件。

(6) PME 将识别后的位置事件写回 Kafka。

(7) Spark 处理过程如下。

- 实时互联网行为标注: 从 Kafka 读取上网数据,结合预配置互联网知识库,进行互

联网行为标注(URL标注为App或网站)。

- 实时互联网事件识别: 结合预定义互联网访问规则识别互联网事件(访问某App或网站)。

- 复杂事件处理(位置和互联网事件结合、特殊用户信息判断等)。

(8) Spark 将识别后互联网事件写回 Kafka。

(9) 实时决策引擎从 Kafka 读取识别后事件,基于营销策略生成营销推荐信息。

(10) 实时决策引擎发布营销推荐信息。

(11) 营销推荐信息推送给渠道。

(12) 渠道将营销推荐信息发送给用户。

(13) 渠道通过离线方式基于营销反馈接口反馈营销推荐结果。

(14) 实时决策引擎处理营销反馈结果。

3 Kafka 中消费者组对信令流的消费监控实施方案

3.1 问题提出

由于 Flume 组件类似于“管道流”,信令传

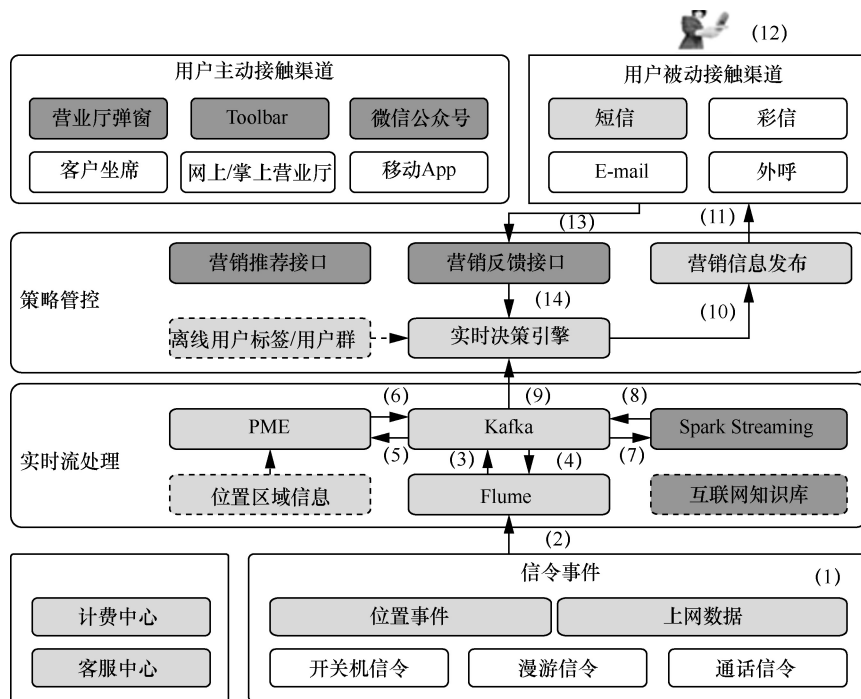


图6 实时决策中心业务流程



输后不保存，如果下游环节未及时拉取数据或消费数据，就容易造成数据丢失。数据一旦流出，再想追溯弥补这些丢失的数据就很困难。同样，如果网运侧信令流长时间中断而未及时发现，就会造成信令流积压拥堵，影响后续处理的时效，不利于实时营销场景的数据支撑。实时流信令采集监控的问题比较突出，传统运维模式是根据客户投诉或网络显性故障，被动地解决问题，无法做到对故障的主动预判，所以永远都是慢“半拍”的被动式维护。解决客户感知不佳的问题迫在眉睫，无论用户是否投诉，先于客户发现问题、解决问题，是电信运营商亟待解决的难题。

3.2 监控方案

实时流信令采集监控措施分为如下3个层次。

第一层是平台性能监控，包括：平台组件（Flume/Kafka/Zookeeper/HDFS/Spark 等）进程状态监控；平台主机性能（磁盘利用率、内存利用率、CPU 利用率等）监控；平台访问审计监控。

第二层是实时流消费监控。

由于基于 FKS 的信令采集处理系统，Flume 要往 Kafka 传送数据，Spark Streaming 要往 Kafka 回写数据，Kafka 处在中枢环节，Flume 或 Spark Streaming 的采集处理故障，也会体现在 Kafka 处理环节中。为了避免消息队列拥堵，规避数据丢失风险，一种比较便捷高效的方式是在 Kafka 中加入探针，定时监控 Kafka 中 consumer group（消费者组）对 topic 的消费状况。目前主流的 3 种 Kafka 监控程序分别为：Kafka Web Console、Kafka Manager、KafkaOffsetMonitor。综合借鉴上述 3 种监控程序的思路，结合运营商自身业务模式及现有服务架构，设计实施了基于 Kafka 的实时流信令采集监控方案。

第三层是消费端与数据来源端定期一致性对比校验。

从对某运营商实时流大数据平台运行维护持续跟踪结果看，影响对外正常业务支撑的原因（排

除外部数据源故障外），主要是磁盘容量不足、CPU 使用率过高等导致采集进程（线程）故障。这些都会反映在 Kafka topic 的消费进度异常或增长趋势上。第二层实时流消费监控设计理念，可类比为海鸥。海鸥是海上航行安全的“预报员”，如果海鸥贴近海面飞行，那么未来的天气将是晴好的；如果它们沿着海边徘徊，那么天气将会逐渐变坏；如果海鸥离开水面，高高飞翔，成群结队地从大海远处飞向海边，或者成群的海鸥聚集在沙滩上或岩石缝里，则预示着暴风雨即将来临。

限于篇幅，本文重点介绍第二层次监控的实施方案和实施效果。

基于 Kafka 的实时流信令采集监控，设计思路是重点关注 topic，尤其是对数据量特别庞大的原始 topic 加强监控，定期采集展示 consumer group 对这些 topic 当前消费情况、消费时序图以及 offset 在时序上的变化情况。对于某个 topic 有多个消费者组同时消费时，可以视需要，对这些消费者组同时监控。一般情况下，不同消费者组对同一 topic 的消费情况和变化趋势是基本一致的，如果差异太大，就需要对系统参数做相应修改调整。

实施办法是按固定时间间隔（比如间隔 10 min），对 Kafka 消费情况进行打点采样，获取某 consumer group 对某 topic 当前消费情况（offset），生成时序文件。正常情况下，此时序文件可以获取，文件行数固定，且其中 offset 是逐步累积增长的。图 7 列出实时流消费情况监控的流程。根据时序文件生成过程中出现的异常情况，可及时告警提醒，为核查解决问题赢得时间。主要异常情况分如下 4 种。

- 监控文件不能生成，初步判定 Kafka 平台故障，发出告警提醒。
- 监控文件行数不符合预期，将不合规文件复制至问题目录待查。
- offset 在 t 个周期连续中断，初步判定 Kafka

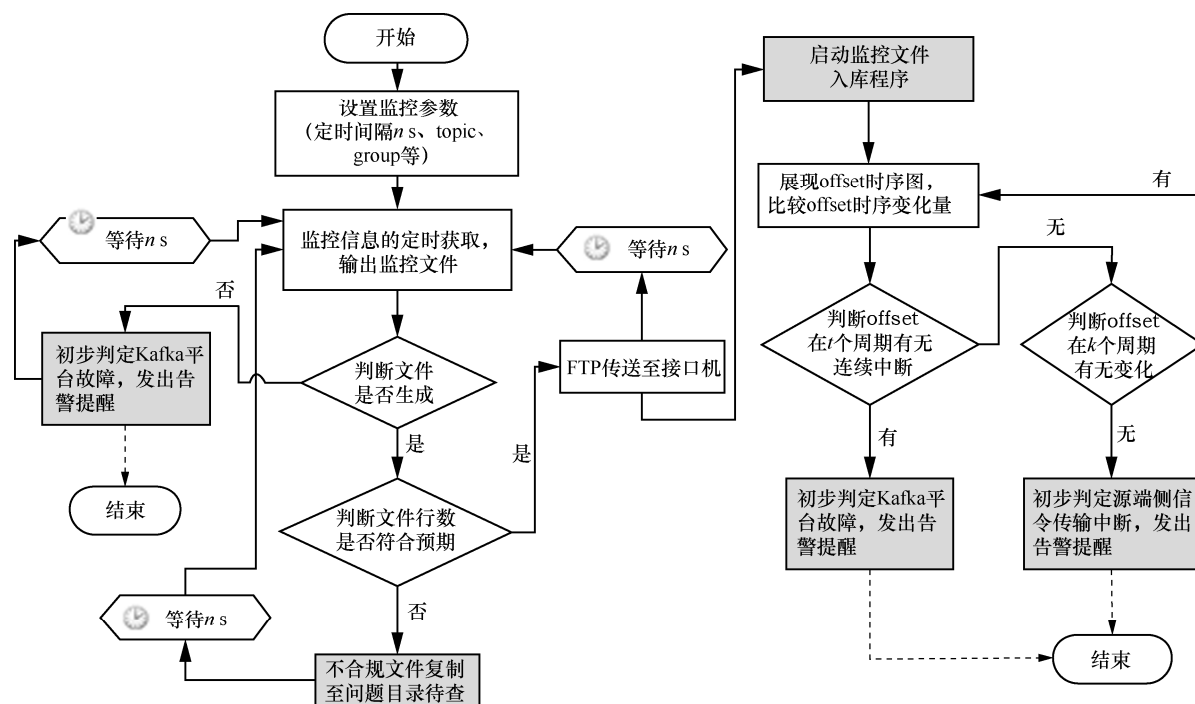


图7 实时流消费监控的流程

平台故障，发出告警提醒。

- offset 在 k 个周期连续无变化，初步判定源端侧信令传输中断，发出告警提醒。

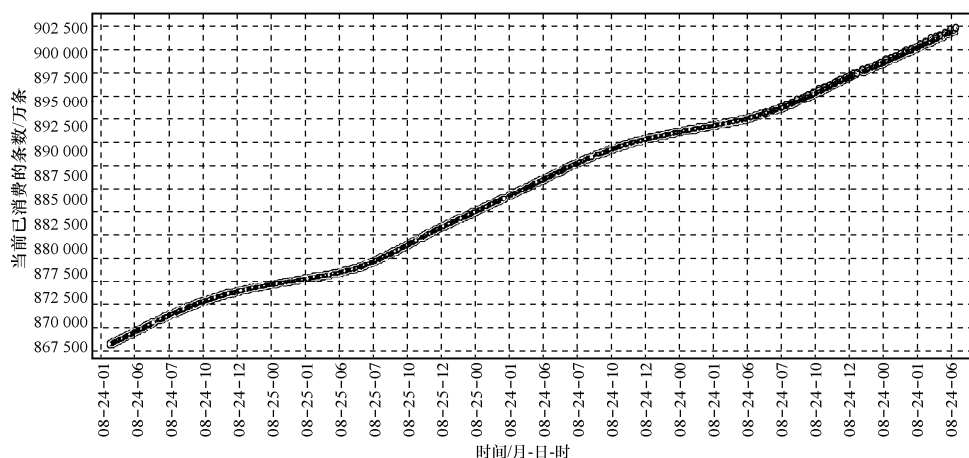
3.3 实施效果

实时流采集监控系统的实施分为如下环节：监控文件的定时获取、监控文件的规整化与自动入库、监控信息可视化展现、预警与告警提醒等。

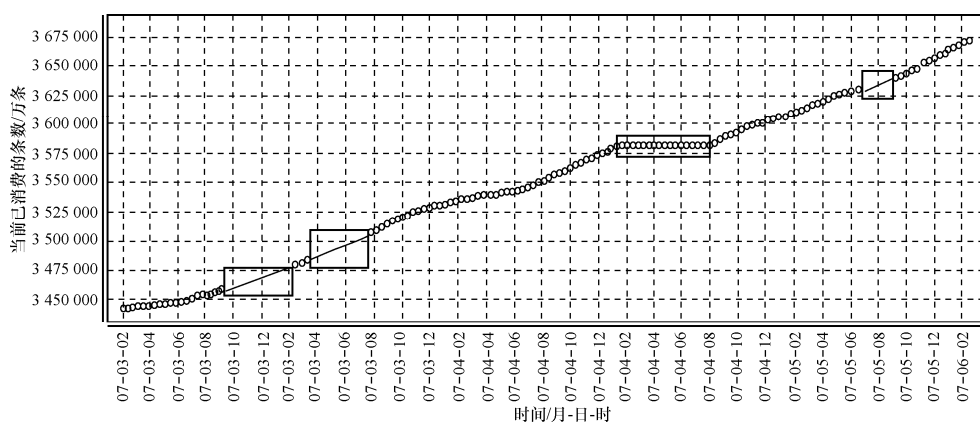
每个环节的处理时间都在秒级，整个流程处理时间在分钟内。其中，监控文件采用每隔 10 min 打点采样方式定时获取，包含 group（消费者组）、topic、PID（分区 ID）、offset、LogSize（总条数）、Lag（未消费的条数）、owner 等信息。该文件的行数与 topic 的分区数直接相关。图 8、图 9 展现监控信息的可视化效果。



图8 实时流消费监控—最近消费查看



(a) A口位置信令消费趋势



(b) 4G 信令消费趋势 (有故障时)

图9 监控系统的图形可视化功能

图8展现监控系统的表格查询功能,查看实时流最近消费情况。正常情况下,表中日期时间与当前时间差距不大(假定监控文件采集间隔为10 min,则此差距不超过15 min)。如果时间差异过大,则意味着是异常现象,需要检查修复。

图9(a)、图9(b)展现监控系统的图形可视化功能,展现最近一段时间A口信令、4G位置信令消费趋势图,正常情况下,“当前已消费的条数”趋势是逐步累积增长的。图9(b)中,左下角矩形方块中断部分表示监控文件打点采样时未生成(或生成文件不合规);中间矩形方块表示offset连续无变化。这些都是异常情况,需要及时告警提醒、分析处理。

实践运行表明:提出的基于Kafka的实时流信令采集监控方案,通过对offset的数值监控,实现了实时流采集监控可视化,起到了行之有效的监控作用,方便了运维监控,极大改善了运维的被动性,绝大多数故障都能做到及时预警提醒,为故障分析处理赢得了有利时间窗口,在实践中取得良好效果。

4 结束语

目前,大数据实时流技术发展迅速、应用较广,但对实时流的采集监控手段相对缺乏。基于FKS的信令采集处理方案,综合发挥了Flume、Kafka、Spark Streaming三组件的各自优势,可快速实现信

令数据的实时采集处理,对运营商打造实时决策中心,开展基于位置的实时营销场景促销活动起着重要支撑作用。本文提出的针对 Kafka 中消费者组对各 topic 消费情况监控方案,是一种简单易行的监控措施,对实时流采集故障及时预警提供了有效的手段,在某运营商大数据平台建设的实践过程中,有效减少消息队列拥堵现象,规避数据丢失风险,对实际生产运营支撑起着良好保障作用。

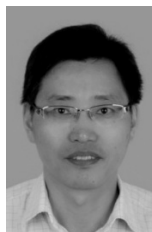
参考文献:

- [1] 许佳捷,郑凯,池明旻,等. 轨迹大数据:数据、应用与技术现状[J]. 通信学报, 2015, 36(12): 97-105.
XU J J, ZHENG K, CHI M M, et al. Trajectory big data: data, applications and techniques[J]. Journal on Communications, 2015, 36(12): 97-105.
- [2] 陆钢,王哲,区洪辉,等. 基于开源软件的电信业务平台重构设计和分析[J]. 电信科学, 2017, 33(5): 99-105.
LU G, WANG Z, OU H H, et al. Design and analysis of telecom service platform reconstruction based on open source software[J]. Telecommunications Science, 2017, 33(5): 99-105.
- [3] 刘炳涛,王达,叶笑春,等. 基于数据流块的空间指令调度方法[J]. 计算机研究与发展, 2017, 54(4): 750-763.
LIU B T, WANG D, YE X C, et al. The data-flow block based spatial instruction scheduling method[J]. Journal of Computer Research and Development, 2017, 54(4): 750-763.
- [4] 蒋仲安,王明,陈雅. 基于地理坐标和轨迹数据的路径推荐方法[J]. 通信学报, 2017, 38(5): 165-171.
JIANG Z A, WANG M, CHEN Y. Path recommendation based

on geographic coordinates and trajectory data[J]. Journal on Communications, 2017, 38(5):165-171.

- [5] 汤雅妃,魏进武,张云勇. 基于大数据的信令监测系统研究[J]. 邮电设计技术, 2014(7): 47-52.
TANG Y F, WEI J W, ZHANG Y Y. Research on big data based signaling monitoring platform[J]. Designing Techniques of Posts and Telecommunications, 2014(7): 47-52.
- [6] 孟红涛,余松平,刘芳,等. Spark 内存管理及缓存策略研究[J]. 计算机科学, 2017, 44(6): 31-35.
MENG H T, YU S P, LIU F, et al. Research on memory management and cache replacement policies in spark[J]. Computer Science, 2017, 44(6): 31-35.

[作者简介]



汪保友(1968-),男,博士,中国联合网络通信有限公司上海市分公司高级工程师,主要研究方向为数据科学、数据挖掘、数据签名。

姚健(1981-),男,中国联合网络通信有限公司上海市分公司工程师、大数据中心总经理,主要研究方向为数据科学、移动互联网、通信网络规划。

张正卿(1978-),男,中国联合网络通信有限公司上海市分公司工程师,主要研究方向为数据科学、移动互联网、通信网络规划。