

# 流式计算模型在废弃话单处理中的应用<sup>①</sup>

郭淑娴, 杨金民

(湖南大学 信息科学与工程学院, 长沙 410082)

**摘要:** 对废弃话单进行有效管理, 是保障通信系统计费可靠性的重要内容之一, 然而对于废弃话单数据所呈现出的实时性、突发性等特征, 现有的大数据处理模式并不能很好地应对. 本文采用动态路由策略实现计算系统运行时的轻量级伸缩, 应对废弃话单实时性和突发性带来的挑战. 在此基础上, 本文给出了一种基于流式计算模型的废弃话单处理系统的实现, 并通过实际数据验证了本文提出的动态路由策略在应对数据突发方面的有效性.

**关键词:** 废弃话单; 流式计算模型; 实时性; 突发性; 动态路由

## Abandoned Bill Handling System Based on Stream Computing Model

GUO Shu-Xian, YANG Jin-Min

(School of Information Science and Engineering, Hunan University, Changsha 410082, China)

**Abstract:** Managing abandoned bills effectively is an important part to guarantee the accounting reliability of communication systems. However, for the distinctive features of the abandoned bills such as real time and burstiness, etc., the existing big data computing models can not deal with them well. In order to meet the challenges of real time and burstiness of abandoned bill system, this paper presents a dynamic routing strategy to realize the run time lightweight scaling of the computing system. On this basis, this paper gives an implementation of the abandoned bill system based on the stream computing model. Finally, the results demonstrate that the proposed dynamic routing strategy is an effective way on handling burstiness of data.

**Key words:** abandoned bill; stream computing model; real time; burstiness; dynamic routing strategy

## 1 引言

通信行业竞争的加剧, 导致竞争领域不断由外部市场领域向内部质量管理领域延伸, 话单计费可靠性目前已成为通信运营商质量管控的核心, 而对 BOSS 计费系统所产生的废弃话单进行有效管理, 是保证计费可靠性的重要内容之一.

废弃话单是指计费系统在预处理环节中废弃的话单, 例如时长为 0、拨出方为空的话单. 湖南移动 BOSS 系统每天处理近 30 亿条话单, 而被废弃的话单每天约有 14 亿. 为何废弃话单的数量如此之多, 什么原因导致了如此大量的废弃话单, 其中是否包含了本该计费, 但因被废弃而未进行计费的话单. 对这些问题, 迫切需要建立废弃话单分析引擎, 对被 BOSS 系

统废弃的话单进行检测和统计分析, 判定 BOSS 对话单的废弃判断是否正确. 同时需要发掘废弃话单产生根源, 识别废弃话单的特征和特性, 定位产生废弃话单的设备, 进而提升计费的可靠性, 减少收入的流失, 实现创收.

对于废弃话单, 其特性可以归结为以下几点:

(1) 实时性. 废弃话单数据实时产生, 处理越及时, 定位问题就越快速. 同时随着时间的流失, 其所包含的价值就越低. 因此, 话单产生之后, 需要及时处理.

(2) 易失性. 每天都会产生数据量巨大的废弃话单数据, 对于所有数据持久化保存会导致很高的存储成本. 注意到大部分废弃话单数据具有共性, 信息重叠性较高, 通常提取共性就可, 原始话单数据无需保存.

<sup>①</sup> 基金项目:国家自然科学基金(61272401)

收稿时间:2016-03-21;收到修改稿时间:2016-04-27 [doi:10.15888/j.cnki.csa.005488]

(3) 突发性. 由于业务特性, 废弃话单的量随着时间的变动并不一致, 即前一时刻数据速率和后一时刻数据速率可能会有巨大的差异. 同时, 由于废弃话单属于异常监控类的应用场景, 当异常发生时, 其数据量本就会突增. 因此, 需要计算系统既要能够在数据量突发时仍然保证计算的低延迟性, 又要能够在数据量降下来时, 快速的释放计算资源.

(4) 无关联性. 通常废弃话单数据之间并不存在关联性, 即不同的废弃话单之间并没有直接的联系. 这个特性可以简化分析的处理逻辑.

(5) 无限性. 废弃话单每天都在产生, 是实时产生、动态增加的. 潜在的数据量是无限的, 无法用一个具体确定的数据实现对其进行量化.

综上所述, 对于废弃话单, 其价值是有时间效应的, 因此需在数据的有效时间内完成数据的计算, 而不应等待后续全部数据到来再进行计算. 同时, 为了应对废弃话单突发的特性, 计算架构必须具有较好的伸缩特性: 一方面, 当数据量突增时, 系统能够扩展计算节点; 另一方面, 当数据量恢复时, 系统能够回收计算节点, 防止计算资源的浪费. 最后, 由于废弃话单具有无限的特性, 因此需确保计算系统的长期可靠性, 即需自动应对对节点出错的情况. 因此, 理想的废弃话单处理系统应具备如下功能特性:

数据处理低延迟: 应对数据实时和易失的特性;

数据负载均衡和节点扩展性: 应对数据突发的特性;

节点容错性: 应对数据无限的特性;

实际上, 废弃话单的分析可以归结为海量数据的处理. 对于海量数据的处理, 通常有两种计算模式——批量计算模式<sup>[1]</sup>和流式计算模式<sup>[1-3]</sup>. 批量计算采用的是先存储再集中计算的模式, 无法满足实时性的要求. 现有的流式计算模式大多采用拓扑结构进行计算节点的组织, 动态扩展性较差, 无法达到应该数据突发的目的.

本文以废弃话单处理系统的设计为核心, 对现有大数据处理系统中的一些不足之处进行优化, 为大数据流式计算系统的设计以及应用提供一些指导性原则, 以弥补当前关于大数据流式计算在实践应用层面欠缺的局面.

## 2 研究现状

批量计算采用的是先存储再集中计算的模式. 当

数据记录来到时, 并不立即进行计算, 而是等到数据量达到一定规模之后集中进行计算. 因此除了计算之外, 还涉及到海量数据的存储. 目前, 对于海量数据批量分析技术的研究已相当完善, 并形成了以 Google 的 MapReduce 编程模型<sup>[4]</sup>为理论基础、开源的 Hadoop 计算架构为其代表的稳定高效的批量计算系统.

然而数据记录的价值通常随着时间的流逝而降低, 如果不能立即对其进行处理, 就可能丢失其价值. 例如对于许多组织机构, 尤其是金融服务、国防、情报和税收部门、电信、电子商务等, 需要的是不间断并且实时获取情报的能力, 进而能够分析大流量的实时事件, 及时洞察事件的始末, 并及时且通常是自动的响应变化. 例如对于股票的交易数据记录等, 数据实时性强, 数据量巨大, 并且无止境, 传统的批量计算模式已很难适用.

为了应对海量数据实时计算的要求, 流式计算模式被提出. 所谓流式计算<sup>[5-7]</sup>, 是指将数据记录当作是数据流的形式来处理, 而非是静态的一批数据. 数据流是指在时间分布和数量上无止境的一系列数据记录的集合, 其最小的组成单元是每条数据记录. 流式计算的本质, 就是在数据记录不断流动的过程中对其进行及时的分析计算, 发现其蕴含的内在价值, 并将其内在价值通过一定的途径展示给用户. 例如对于大型电商网站, 其对于流式数据的应用则非常常见, 这些流式数据包含用户搜索了什么关键字、查看了什么商品、购买了什么商品等, 对于这些数据的计算分析, 可以实时的对用户作出商品的推荐, 在提供更好的用户体验的同时, 也可以实现创收.

对于流式计算的早期研究, 通常集中在数据库环境中开展数据计算的流式化, 其数据记录的规模往往较小, 数据记录的类型也比较单一. 然而新时期的流式大数据呈现出实时性、易失性、突发性、无关联性、无限性等特征, 对系统提出了更高的要求. S4 流式计算系统<sup>[8,9]</sup>, Storm 流式计算系统<sup>[10]</sup>的相继推出, 在一定程度上推动了大数据流式计算技术的发展和应用. 这些系统的本质是以拓扑的结构来完成节点的组织, 每个节点在自己所处的链路上完成数据的处理并传输给下一个节点. 然而这些系统在自由伸缩性、计算容错和可靠性、负载均衡等诸多方面仍然存在着明显不足. 例如对于废弃话单, 其突发的特性需要计算系统具备运行时伸缩节点的特性, 而这是现有流式系统无法完

成的. 如何构建低延迟、高伸缩性且持续可靠运行的大数据流式计算系统, 是当前亟待解决的问题.

### 3 基于流式计算的废弃话单处理模型

对于废弃话单的处理, 其本质是发现废弃话单的废弃共性, 对于异常的废弃共性予以的及时发现和快速处理, 进而减少损失. 根据废弃话单数据所呈现的特性, 本文设计了一种具备高伸缩性和计算可靠性的流式计算模型来完成废弃话单的处理. 其实现过程有两个方面需要考虑: 一是如何应对废弃话单突发性的需求; 二是如何设计废弃话单处理的流式处理模型. 本节分别从路由策略、系统架构以及实现两个方面进行详细的描述.

#### 3.1 高伸缩性的动态路由策略

为了满足废弃话单处理系统对于突发性的要求, 需要计算系统具备计算节点的高伸缩性. 为了实现高度伸缩的特性, 本文放弃了之前的静态路由方式, 转而采用动态路由的方式来管理计算节点之间的数据通信. 对于静态路由, 在任务开始时已固定计算节点个数以及数据和计算节点之间的路由关系. 而动态路由则是使用专门的管理节点来对计算节点的路由关系进行管理, 并实现路由关系的实时下发. 动态路由的运行机制如图 1 所示.

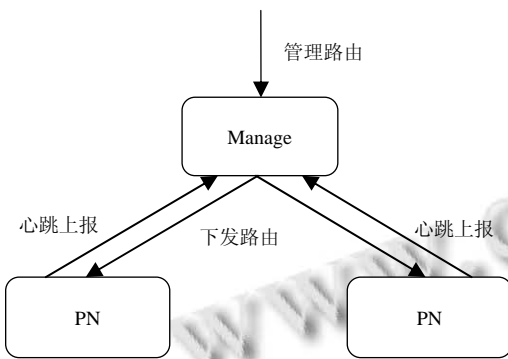


图 1 动态路由机制

其中 ProcessNode 表示计算节点, Manage 表示路由管理节点. 当某个 ProcessNode 节点发生故障时, Manage 会动态的剔除该节点的相关路由关系以实现节点容错. 同时, 为了防止 Manage 引起的单点故障, 每个 ProcessNode 节点会缓存下发的路由信息, 这样即使 Manage 发生故障, 也不会影响拓扑关系的正确执行, 从而可以实现 Manage 节点的容错.

对于动态路由策略, 除了运行机制之外, 还需设计对应的路由信息获取机制. 为了实现这种动态路由机制, 本文将路由信息采用组的方式来组织, 每个路由组具有一个编号, 当节点需要路由信息时需根据路由组编号, 然后通过一定的策略(随机性策略, 一致性 hash, 权重策略等)从组内获取一条路由信息. 其中获取路由的参数包括: 路由组编号, 路由策略以及参数. 由于是按组编号的方式来获取路由信息, 而组内的路由信息又可以自由的进行组织, 因此可以实现路由信息的动态伸缩. 动态路由的组织方式如图 2 所示.

路由组 ID	路由信息
ID1	Ip1
	Ip2
ID2	Ip3
	Ip4
...	...

图 2 路由表的组织方式

当需要节点伸缩时, 可以简单的修改 Manage 的路由信息, 然后由 Manage 完成路由信息的下发. 通过这种机制, 可以自由的对节点进行伸缩. 结合心跳机制和动态路由策略, 同样可以有效的实现节点级容错. 动态路由机制的运行实例如图 3 所示.

#### 3.2 系统架构和实现

本文结合提出的动态路由机制来实现废弃话单的处理, 并从系统架构、系统实现三个方面进行描述.

##### 3.2.1 系统架构

系统架构如图 4 所示.

其中 ProcessNode 负责消息处理逻辑, 同时接收 Manage 下发的路由信息. MessageCollector 是特殊的 ProcessNode 节点, 负责消息收集. Manage 负责节点管理和路由信息管理;

其中, ProcessNode 节点又被细分为输入模块, 心跳检测和路由模块, 逻辑处理模块, 输出模块四个部分: 输入模块负责接收请求数据; 逻辑处理模块负责数据处理逻辑; 输出模块负责按照路由信息将消息传送到其他节点; 心跳检测和路由模块负责向 Manage 发送心跳包, 并接收 Manage 下发的路由. ProcessNode 节点的组成如图 5 所示.

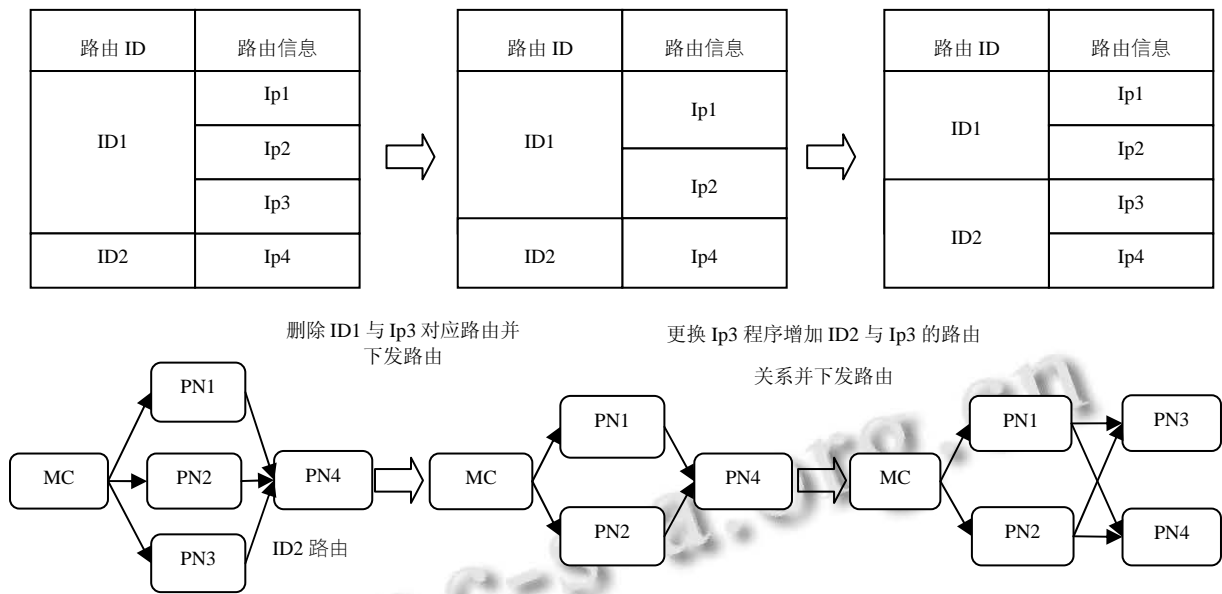


图 3 动态路由机制运行实例

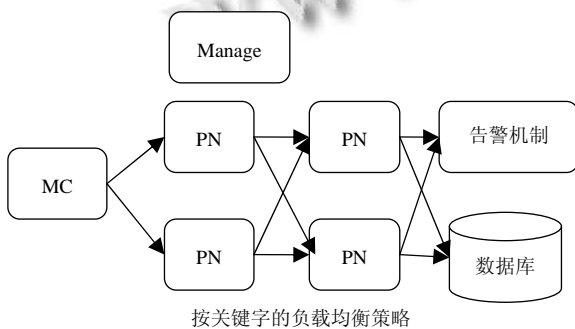


图 4 系统架构图

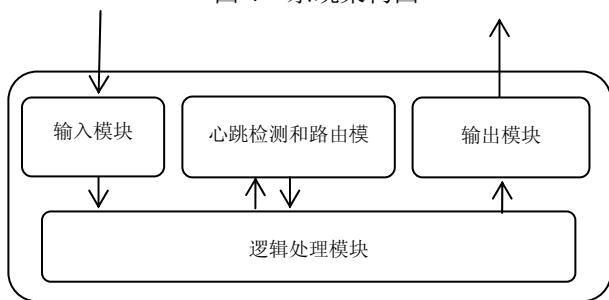


图 5 ProcessNode 构成

节点完成处理和重要信息入库，并将重要信息实时推送给用户。

### 3.2.2 系统实现

为了方便实现各个模块，本文提取了各个模块的共性，提供了一种通用的服务架构。通过这种架构，可以很方便的构建各个模块。通用架构的主要功能是构建服务器架构，完成消息的收发，并在不同的阶段挂接不同的函数完成逻辑的处理。通用架构如图 6 所示。

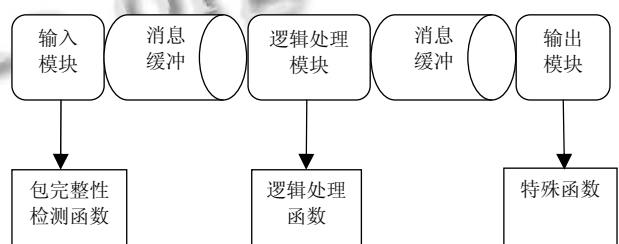


图 6 通用服务架构

注意到对于废弃话单的处理，其处理信息的传播策略应该是主动的，而非用户被动拉取，因此本文在设计中加入了告警机制，实现重要消息的主动推送。废弃话单的处理过程为：MessageCollector 从其他消息源接收信息，然后采用流式的方式和随机路由策略分散到一级处理节点，一级处理节点完成关键字段的抽取并按照关键字段执行负载均衡策略，并分发到二级

每个模块在实现时，可以简单的通过定制挂接的函数来实现各自的功能。例如 ProcessNode 的实现如图 7 所示。

## 4 系统实验与分析

为了验证本文提出的动态路由策略的有效性，本文使用实际的废弃话单数据，分别测试动态路由策略

和静态路由策略在数据突发的场景下,话单数据的处理延迟情况,来验证本文提出的动态路由策略的有效性.所谓处理延时,是指从消息发出到消息最终完成处理并提交所花费的时间.

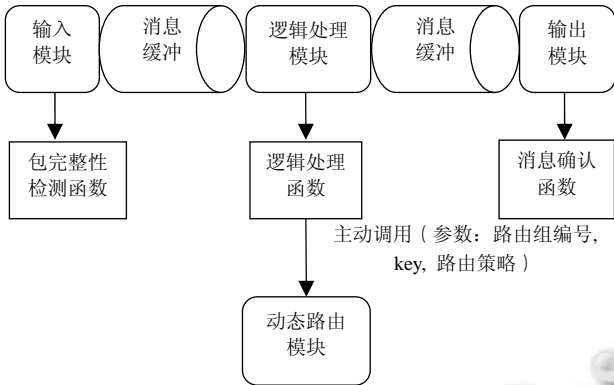


图7 ProcessNode 实现

在实验中,为了充分发挥动态路由机制的性能,本文开发了一个辅助系统,该辅助系统可以根据系统负载情况,按照预先设计的策略自动启用资源池中的备用计算节点并下发路由,以实现系统的自动扩展.

实验环境为 Xeon E5-2680 V2, linux 操作系统, 16GB 内存,使用进程模拟计算节点.为了模拟突发性,实验中使用 1GB 总量的废弃话单数据先以较小的速率发送,然后以较高的发送速率发送到流式计算系统,在处理过程中统计话单消息的平均处理延时.两种路由策略在不同突发速率下的平均延时如图8所示.

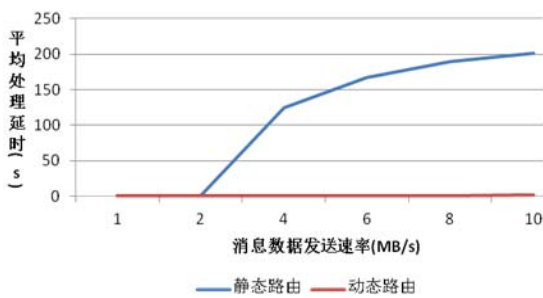


图8 静态路由策略和动态路由策略在不同突发速率下的平均延时

从实验结果可以看出,对于静态路由策略,由于缺乏扩展性,随着消息数据产生速率的增长,其消息的平均处理延迟也越来越高.对于动态路由策略,在数据突发的情况下,其仍能维持较低的处理延迟.

和批量计算模式相比,采用流式计算模式的废弃话单系统具有以下特征:(1)可以对废弃话单消息进行实时处理,因此可以更快的发现问题;(2)可以连续进行计算,而非集中计算,因此可以有效的利用计算资源;(3)流式计算模型对于废弃话单进行实时处理,无需存储大量信息,而批量模式需要存储到一定数据才会计算,需要大量的存储资源;(4)流式计算模型可以在运行时对节点进行伸缩,而批量计算通常只能在任务创建时完成节点的负载均衡.两者的对比如表1所示.

表1 基于流式计算模式的废弃话单系统和批量计算模式的对比

比较方面	流式计算模式	批量计算模式
发现问题时间	实时	比较慢
计算资源利用	连续计算,利用 率较高	集中计算,计算资源 的空闲率较高
存储资源利用	实时计算,存储 资源占用很少	缓存后批量计算,存 储资源占用较大
伸展性	能够运行时动态 伸展	在任务创建时静态 分配

### 5 总结

对 BOSS 计费系统所产生的废弃话单进行有效管理,是保证计费可靠性的重要内容之一.本文总结了废弃话单数据所呈现出的实时性、易失性、突发性、无关联性、无限性等特征,给出了理想的废弃话单处理系统应该具有的关键功能特征.阐述了已有的大数据计算模式的研究现状,并就其在节点伸缩性方面的缺点进行了分析.在此基础上,本文设计了一种动态路由策略来实现计算系统的运行时伸缩,以应对废弃话单突发性的需求.本文提出的动态路由策略,通过路由信息分组和按组编号获取路由信息的方式对路由进行动态管理.通过提供的接口可以自由的对组内的路由信息进行增加和修改,极大的提高了系统的热伸缩性.最后,本文给出了一种基于流式计算模型的废弃话单处理系统的实现,并通过实验验证了本文所提出的动态路由策略在应对突发性方面的有效性,进而有效的满足了废弃话单处理的需求.

### 参考文献

1 孙大为,张广艳,郑纬民.大数据流式计算:关键技术及系统实例.软件学报,2014,25(4):839-862.

- 2 Sun DW, Zhang GY, Yang SL, Zheng WM, Khan SU, Li KQ. Re-Stream: Real-time and energy-efficient resource scheduling in big data stream computing environments. *Information Sciences*, 2015, 319: 92–112.
- 3 孟小峰, 慈祥. 大数据管理: 概念、技术与挑战. *计算机研究与发展*, 2013, 50(1): 146–169.
- 4 Dittrich J, Quiané-Ruiz JA. Efficient big data processing in Hadoop MapReduce. *Proc. of the VLDB Endowment*, 2012, 5(12): 2014–2015.
- 5 亓开元, 赵卓峰, 房俊, 马强. 针对高速数据流的大规模数据实时处理方法. *计算机学报*, 2012, 35(3): 477–490.
- 6 Wang LZ, Tao J, Ranjan R, Marten H, Streit A, Chen JY, Chen D. G-Hadoop: MapReduce across distributed data centers for data-intensive computing. *Future Generation Computer Systems*, 2013, 29(3): 739–750.
- 7 顾昕. 分布式流式计算框架关键技术的研究与实现[硕士学位论文]. 北京: 北京邮电大学, 2012.
- 8 Ajwani D, Ali S, Katrinis K, Li CH, Park AJ, Morrison JP, Schenfeld E. Generating synthetic task graphs for simulating stream computing systems. *Journal of Parallel and Distributed Computing*, 2013, 73(10): 1362–1374.
- 9 刘子英, 唐宏建, 肖嘉耀, 张赛. 基于流式计算的 Web 实时故障诊断分析与设计. *华东交通大学学报*, 2014, 31(1): 119–123.
- 10 张振华, 吴开超. 一种分布式 Twitter 数据处理方案及应用. *计算机应用研究*, 2015, 32(7): 2073–2077.