

一种信令数据高效检索方案^①

陈曦^{1,2}, 王纯^{1,2}, 王晶^{1,2}

¹(北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

²(东信北邮信息技术有限公司, 北京 100191)

摘要: 在网络通信高速发展且得到广泛应用的今天, 如何维护信令网、将信令数据转换为有价值的收益、提高维护效率, 是运行维护部门面临的重要问题。鉴于信令数据是话务控制的重要参考因素, 本文为信令数据的跟踪和分析系统设计了一种高效的检索方案, 更有利于以独立的第三方的视点进行观测。

关键词: 信令; 检索; 索引

Efficient Retrieval Program for Signaling Data

CHEN Xi^{1,2}, WANG Chun^{1,2}, WANG Jing^{1,2}

¹(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

²(EBUPT Information Technology Co. Ltd., Beijing 100191)

Abstract: Nowadays, with the rapid development and wide usage of the network communications, there are important issues that departments are facing, such as how to maintain the signaling network, turn the signaling data into valuable revenue and improve maintenance efficiency. In view of the traffic control signaling data is an important reference factor, this paper designs an efficient retrieval program for signaling data tracking and analysis system, in order to provide independent point of view for observation.

Key words: signaling; search; index

随着软交换网络的部署, 以及 IP (Internet Protocol, 网络互连协议) 业务的蓬勃发展, 令通信网络的全 IP 化成为必然。下一代 IP 网络能够帮助电信运营商构建一个融合的单一网络, 在这个网络上, 电信运营商可以灵活快速的推出新业务, 从而提升服务质量和收益^[1]。但是从目前的运行情况来看, IP 承载系统存在维护难、定位问题难、缺乏相关的监控指标和信令跟踪软件等系列问题, 特别是有针对性的性能和负荷指标, 完善产品的可维护性的要求日益强烈。

在具体应用中我们发现, 针对信令数据的监测需求日趋紧迫, 通信网 IP 化背景下的网络信令跟踪、网络优化、媒体质量分析以及基于分析的业务触发等具有重要的意义和广泛的前景。除了跟踪业务流程外, 信令包含的信息对于后续业务系统的改进甚至于企业的决策都有重要的参考价值^[2], 需要跟踪系统有较强

的定制能力。而信令数据又存在数据量大, 关联性强等特点, 因此我们有必要设计一种针对性的高效的信令数据检索方案。

1 检索面临的问题以及方案的提出

1.1 信令数据的特点

网络通信过程中产生的信令数据, 除了海量数据普遍有的数据量大, 速率快, 实时性高的特点^[2]之外, 还具有其特殊性。首先, 一次完整的信令交互过程包括多个封包^[3], 以 BICC (Bearer Independent Call Control protocol, 与承载无关的呼叫控制协议) 信令为例, 一次完整的信令交互过程包括 IAM (Identity and Access Management, 身份识别与访问管理), ACM (Address Complete Message, 地址全消息), APM (Application Transport Mechanism, 应用传送机制), REL (Release

^① 基金项目: 国家自然科学基金(61072057, 60902051); 国家 973 计划项目(2010CB315802); 中央高校基本科研业务费专项基金(BUPT2009RC0505);

国家科技重大专项(2011ZX03002-001-01, 移动互联网总体架构研究)

收稿时间: 2011-09-21; 收到修改稿时间: 2011-11-10

message, 释放消息)等 10 多种封包, 检索信令流程时需要将这些封包进行关联并呈现^[3]; 其次, 在这一次信令交互过程中, 封包到达的时间顺序与封包发出的时间顺序一致, 是正序的, 第三点, 由于信令各个流程间隔较长以及时延等问题, 一次信令流程中的封包到达的时间点虽为正序但不一定连续。

1.2 检索面临的问题

由于信令数据的特殊性, 如何在信令检索时准确并快速地关联一次信令会话中的所有封包, 是一个非常关键的问题。以 BICC 信令为例, BICC 中的 CIC (Circuit Identification Code, 电路识别码) 用来识别在两个对等 BICC 实体间的一个信令关系, 以及属于这个信令关系的所有封包。即一次 BICC 会话过程中产生的每个封包包含的 CIC 相同, 我们可以根据 CIC 来关联这次 BICC 会话过程的所有封包。

从检索的角度来考虑, 为了能更快关联到一次会话中的所有封包, 我们希望能将一次会话中的所有封包存储在连续的地址。但是, 从实际的角度出发, 正是由于一次会话中存在多个信令封包, 这些封包到达的时间间隔不定, 给信令的连续存储带来了一定难度。例如 BICC 信令, 在拨打电话时会首先发送 IAM 消息进行初始化, 接下来会发送一系列信令, 进行通话的建立, 一直到双方挂机后才拆线, 拆线时会发送 RLC (Release Complete message, 释放完成消息) 信令表示结束。现网上的信令数据是每秒 50 兆的量级, 假设一次会话时长为 5 分钟, 如果我们将当前所有的信令交互过程都存入内存中, 等一次呼叫完成后再进行封包关联和存储, 则需要将这 300 秒的信令数据全部存储于内存中, 也就是 $300 \times 50 \text{ 兆} = 15000 \text{ 兆} = 15\text{G}$, 而且这 300 秒中将会产生大量的其他的信令流程, 如果为了方便信令封包关联, 全部存放在内存, 资源消耗将更大, 因此, 需要及时地将信令数据从内存转移到磁盘上。

1.3 高效检索方案的提出

从上面的分析可以得出, 目前制约检索效率的问题主要有两方面, 一方面是单位时间内信令数据量非常大, 另一方面是信令数据的有效关联。需要从这两方面解决问题。

文献[4]分布式海量文本检索系统研究中提出了一种基于数据库的高效存储检索海量数据的方案, 文中也提到了使用数据库存储的弊端。海量数据的存储和

检索通常采用数据库的方式, 但如果简单使用数据库存储, 需要频繁地向数据库发出插入指令, 这样将消耗大量的时间成本, 加重服务器的处理负担, 造成数据库服务器的 I/O 拥堵^[4]。并且, 在插入同时数据库需要建立索引并对索引进行排序, 会浪费大量效率, 难以满足数据的高即时性^[5]。

目前数据库采用 B 树或 B+树结构来进行索引组织^[6], 当数据量增加到一定数据级时, 索引中叶节点增长, 使得索引的层次增加, 每新插入一个节点会导致更多的树的插入和分裂操作, 随着数据的高速增长, 创建索引的时间将令人难以忍受。再者, 随着索引增加, 执行检索操作, 占用的内存越来越大, 会导致更多的换页和磁盘 I/O 操作。

而且如果不考虑信令数据本身的特点, 仅仅简单地对信令数据进行存储, 会使检索时存在关联困难的问题, 很大程度上制约了检索效率。

所以, 综合两方面问题进行考虑, 并借鉴文献中的方案, 本文提出了一种运用于海量信令数据高效检索的方案。

首先, 因为单位时间内信令数据流量非常大, 需要从数据存储方式入手解决问题。方案对数据封包采取归类存储的方式, 采用多层次的文件夹对其进行分区归类^[2]。首先不同协议的封包放在不同文件夹下, 同一协议的所有封包数据按时间粒度存储在不同的文件中, 文件夹内按时间周期 (如 5 分钟) 分成不同文件, 同一周期点的数据存储在同一文件中。这样的存储划分方式结构清晰, 不仅有利于提高检索维护的效率, 也使得数据清理更便捷。

在写入信令原始数据的同时, 可以结合信令数据本身的特点, 创建索引写入一级索引文件。由于一个信令封包中包含的字段较多, 在生成信令的索引时, 应该筛选出用于辨识封包的关键信息, 以及用于描述该信令与其他信令之间关系的信息^[7], 如信令主键、信令类型、用户号码、传输时间以及在数据文件中的存储位置等关键字, 通过这些关键字, 就能快速的关联出完整的信令流程并迅速在原始文件中定位。并且, 需要分配一个固定大小的缓冲池用于暂存和关联索引数据, 但为了保证内存使用率, 需要及时将缓冲池中的数据进行固化, 写入磁盘, 比如五分钟一次, 充分地利用内存空间, 尽可能将同一流程的信令索引数据进行连续存储。这样就达到了以空间换时间的目的,

有效的为后面的检索步骤提高效率提供了可能。

方案的以上部分都是在存储数据的过程中同步完成的,也就是说原始文件和一级索引文件都是实时生成的,但由于对信令数据的检索的实时性要求不高,在一次分析存储完成后,程序可以对一级索引进行分析处理,以进一步提高检索效率。

所以,在这里我们结合用户的使用习惯以及设置的检索条件,基于一级索引文件的基础上,提出了查询索引也就是二级索引的概念。

二级索引根据检索条件来设计,与一级索引之间的关系是多对一的关系,也就是说,一条一级索引记录,可以根据不同的检索条件进行关联并写入不同的二级索引文件中。根据目前系统设计的检索条件进行分析,二级索引文件主要以用户号码为依据进行划分,包括按主被叫号码,或主被叫号段进行分开存储。

2 系统结构与与设计

根据以上的方案分析,本文对数据检索进行了设计。

2.1 总体结构图

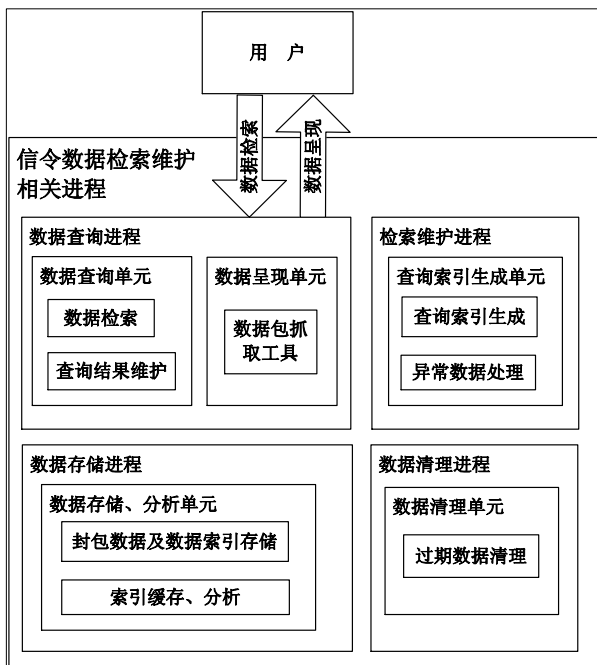


图 1 信令检索方案总体结构图

图 1 是信令检索以及相关功能的总体结构,从图可以看到,与检索相关的主要有以下几个模块:数据

检索、数据存储、检索维护、数据清理。

下面对各个模块的主要设计进行简单的介绍。

2.2 数据检索模块

数据检索模块负责处理用户的查询请求以及维护查询结果,主要包括两个功能单元:一、数据检索单元,二、数据呈现单元。

数据查询单元为用户提供信令数据查询接口,用户可以使用号码、协议类型以及时间等关键词或多个关键词的组合对信令数据进行检索,模块可同时接收多个用户的查询请求,通过启动不同的线程为用户提供统一的检索服务^[6]。为了尽可能提高检索效率,检索结果除了提供给用户之外,系统还额外分配一定大小的内存空间对近期检索结果进行缓存,并对其进行维护管理,如果下次的检索与内存中保存的结果相关,则不需要进行一次完整的检索,可有效节省时间。

数据呈现单元将封包重新组织成 wireshark 工具可识别的文件形式,以便于用户直观的了解封包信息,确定信令通讯过程是否存在问题。

2.3 数据存储模块

数据存储模块不仅包括信令数据的存储,还包括一级索引数据的生成和存储,以及对异常信令数据的处理。

在对信息进行检索前,首先需要对数据建立索引,建立索引的过程是一个非常耗时的操作,这里主要讨论如何能够高效地建立索引。在信令数据存储的同时,模块生成并存储一级索引,系统从数据封包中提取某些关键字将其作为索引存储,如 BICC 中,根据每次呼叫的时间、号码、CIC 等关键字为其建立索引。一级索引与原始信令数据是一一对应的关系,每一条索引记录了相应的信令封包数据的主要信息,主要是信令封包数据存储的位置,便于快速的定位并提取相应的信令封包。

以 BICC 信令为例,如图 2 所示,当收到现网的信令封包并由其他功能模块解析后传入存储模块,存储模块将封包存入数据存储文件,然后根据关键字以及位置信息建立一级索引。首先扫描内存 cache,查看封包的 CIC 是否已存在。如果已存在,则将该封包数据放入相应节点下,并更新该节点状态为脏数据,更新位置信息以及最后加入时间等;如果不存在,需要检查封包是否是起始封包 IAM,如果是 IAM 则新建一个以该 CIC 为主键的节点,如果不是,需要进行异常

数据处理流程。这样就完成了对一个封包索引的建立 以及关联。

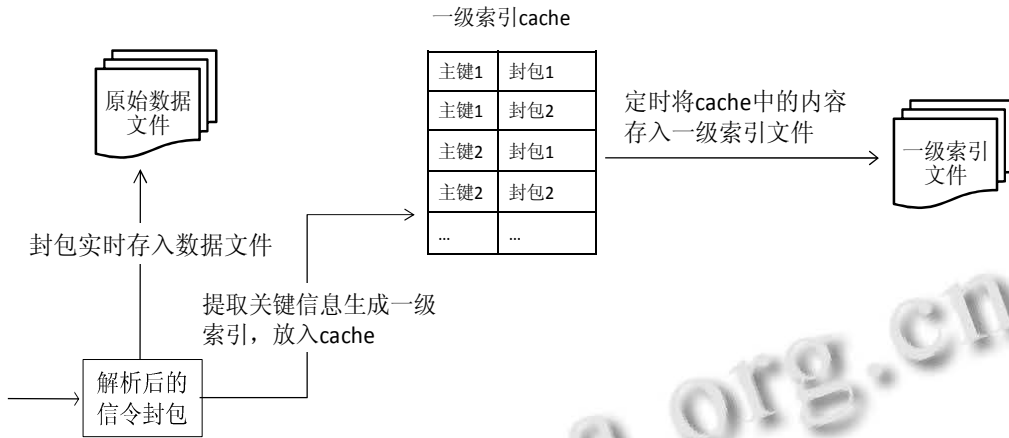


图 2 一级索引文件生成流程

当到达系统设定的检查周期时，模块检查当前内存中的所有节点，对于所有的脏数据，如果其信令流程已完成，即收到结束封包，则将节点信息存入一级索引文件，删除其在内存中的节点；如果未完成信令流程，则将节点现有信息存入索引文件，删除位置信息等，并将节点状态更新为干净数据。

2.4 检索维护模块

检索维护主要是对二级索引的生成和存储，以及对异常数据的处理。

二级索引是指在新生成的数据索引的基础上，系统定期生成的多份包含不同查询条件的查询索引，比如按照号码、时间段来查询用户在一个时间段的所有信令封包，或者根据 CIC、时间段来查询某个信令流程的完整性等等。二级索引的建立是为了进一步提高

数据查询的效率，有针对性的用户各种查询请求，当用户有查询请求的时候，通过查找其对应类型的查询索引，能够进一步减少查询定位时间。

考虑到对内存空间的限制以及存储效率的要求，一级索引在数据存储的同时进行建立，二级索引的建立不需要考虑实时性，是通过其他进程，定期扫描新生成的一级索引文件来建立^[8]。

如图 3 所示，进程每隔一段时间对一级索引的目录进行扫描，将新文件路径加入到作业列表的末尾。进程扫描索引文件，定位到第一条数据，读取其关键字，找到本次索引关心的内容，加入到内存节点并整理相关的信息，读取完信息完整后将内存中的节点写入相应的二级检索文件。

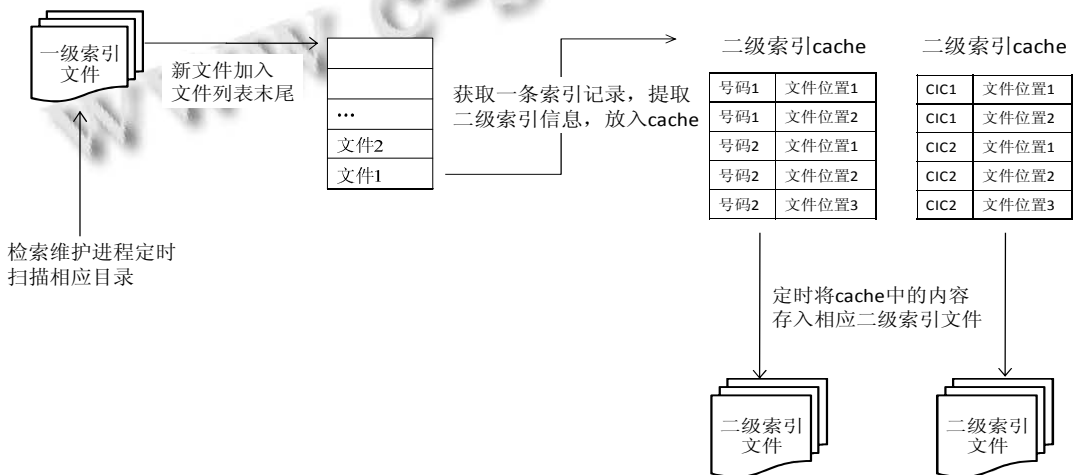


图 3 二级索引生成流程

2.5 数据清理模块

由于硬盘空间的问题，系统只存储一定时间内的文件，可以根据不同的用户需求以及容量等综合条件进行配置，对系统中过期的文件进行定时清理，以释放存储空间。信令数据是按照日期不同来分文件夹存储的，因此数据清理模块只需要每天定时扫描文件夹日期，清理超过有效期的文件夹下的相应文件即可。

3 方案分析

对于特定的软硬件环境，本方案有几个需要进行实验和分析来确定的变量，主要包括，一级索引文件的存储周期，二级索引文件的存储周期，索引存储粒度。假设现网的信令数据速率为 X 字节/秒，原始封包存储周期为 M 秒，二级索引的存储周期为 N 秒，二级索引按照用户号码的末位数对用户进行分文件存储，共分为 P 份。

现假设用户查询某个用户号码在一个时长为 T 的时间段内的 BICC 信令封包，如果按照原始封包的存储方式不做任何处理，进行顺序检索，则平均检索时间复杂度约为 $X \cdot T \cdot M$ ，使用二级索引后，平均的检索时间复杂度约为 $X \cdot N / P$ ，由此可以看出，信令流程起始封包到结束封包的时间间隔 T 越大，二级索引存储周期 N 越小，分文件存储的粒度 P 越大，效率的提升空间越大。如果用户查询的数据在检索结果缓存中，则对性能的优化效果更为明显。

为了测试本方案的性能，在实验室模拟环境下设计实现了一个检索系统，并模拟出信令流程，对检索的性能进行测试比较。实验的物理环境如下：64 位 Linux 服务器，CPU 主频 3.0HZ，内存 8G，swap 空间 4G。

图 4 是在实验室模拟的每秒 50 兆流量环境下的平均检索时间，横轴表示检索时输入的查询时间跨度，纵轴表示一次检索耗费的总时间。系统对二级索引进行按主叫号码分文件存储的方式，测试了两种粒度下的性能，虚线粒度为 100，实线粒度为 1000。由图 4 可以看出，分文件的粒度越大，检索效率越高，但是由于查询前需要进行线程启动以及其他预处理的步骤，检索时间并不完全与分文件的粒度成反比。在实际应用中，可以根据效率需要，对粒度进行调整。

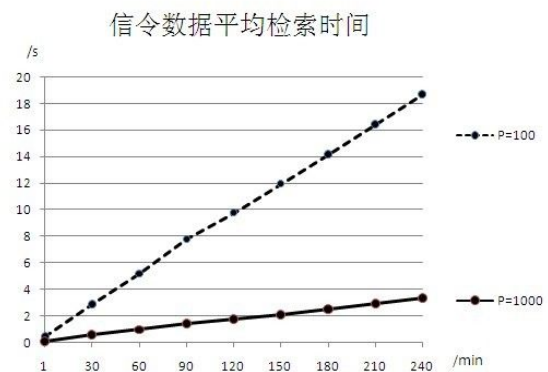


图 4 信令数据平均检索时间图

4 结语

本文针对信令数据的特点，为信令监控分析系统提供了一种高效的检索方案，使得对信令数据的分析更加方便快捷，提高了维护人员进行错误排查、问题定位的效率，且在一定程度上有助于后续业务系统的改进以及企业进行商业决策。

参考文献

- 1 廖建新.移动智能网技术的研发现状及未来发展.电子学报,2003,31(11):1725-1731.
- 2 姚书怀,刘兴伟.大型数据中心海量数据存储解决方案的设计.四川工业学院学报,2004,23(12):27-30.
- 3 孙浩.以软交换为核心的 NGN 网络信令监控系统.现代电信科技,2008,3(3):6-9.
- 4 孙英武,杜凯,杨树强,韩伟红.分布式海量文本检索系统研究.微电子学与计算机,2006,23(12):32-34.
- 5 孙璐.关于数据库索引的初步探讨.经营管理者,2009,12(23):344.
- 6 张刚,孙建,丁国栋,米嘉,王斌.海量数据的索引与检索系统.全国网络与信息安全技术研讨会 2004 论文集.2004:366-370.
- 7 Xia Hong-xia,Zhang Ming-yuan,Wu Jun.Granular Computing Applied to Heuristic Search Algorithm. The 1st International Conference on Information Science and Engineering (ICISE 2009).2009:3804-3807.
- 8 Mei Li, Wang-Chien Lee, Anand Sivasubramaniam, Jing Zhao. SSW: A Small-World Based Overlay for Peer-to-Peer Search. IEEE Trans. on Parallel and Distributed Systems, 2008,19(6):735-749.