

一种基于查询事件的域名服务器日志模型

乔 婧^{1,2}, 金 键¹, 毛 伟¹

(1 中国科学院 计算机网络信息中心, 北京 100190; 2 中国科学院 研究生院, 北京 100190)

摘 要: 提出了一种基于查询事件的日志模型, 采用查询/应答日志匹配的方法完整的记录了一次查询事件, 利用内存数据结构提高了海量数据写入的 I/O 效率; 在日志分析过程中对日志文件建立二维哈希索引, 利用布隆过滤器减少磁盘 I/O 次数, 提高了分析效率.

关键词: 域名服务器; 日志; 查询/应答配对; 哈希索引; 布隆过滤器

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2010)02-0095-04

A DNS Logging Model Based on Query Event

QIAO Jing^{1,2}, JIN Jian¹, MAO Wei¹

(1 Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China;

2 Graduates School, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In this paper, we investigate a logging model based on a complete DNS query event. The corresponding query/response matching technique is designed, equipped with a facility of memory caching to solve the performance bottleneck of the insertion operations. Furthermore, a mechanism of two-dimensional hash indexing with Bloom Filter is designed to improve searching and retrieving by reducing disk I/O.

Key words: DNS; logging; matching query/response; Hash indexing; Bloom filter

1 引言

域名系统(Domain Name System, DNS)主要用于命名计算机和网络服务等资源, 并提供与其相关的其他信息的解析, 如 IP 地址等. 域名服务器负责存储域名信息并完成对 DNS 查询请求的解析, 不言而喻, 域名服务器的性能, 稳定性和安全性十分重要. 与此同时, 针对 DNS 的攻击层出不穷^[1]. 因此, 对经过域名服务系统的网络事件进行及时捕捉和实时分析, 是监控 DNS 服务质量, 服务器运行状况和及时进行防御响应的基础.

目前普遍使用的 DNS 服务器软件, 例如 ISC 发布的 BIND^[2] 及 Nominum 公司开发的 Foundation^[3] 产品等都有简单的日志功能, 但由于软件实现的限制和服务器解析性能方面的考虑, 其日志功能不能够满足实时监控和分析的要求. 如: 日志信息不全

面, 面对大量的查询请求会降低解析服务的性能; 日志的数据量大, 难以在可接受的时间内完成在线统计和分析. 针对以上问题, 文中设计了一种基于查询事件的域名服务器日志模型.

2 DNS 查询-响应过程

当 DNS 客户机需要查询域名时, 它会查询域名服务器来解析该域名. 客户机发送的每条查询消息都包括 3 条信息, 以指定服务器应答的问题:

(1) 指定的 DNS 域名, 表示为完全合格的域名(FQDN).

(2) 指定的查询类型, 它可根据类型指定资源记录, 或作为查询操作的专门类型.

(3) DNS 域名的指定类别.

域名服务器有两种工作模式: 迭代(iterative)和递归(recursive).

收稿日期: 2009-01-12; 修回日期: 2009-03-15

基金项目: 中国科学院计算机网络信息中心青年基金项目(0814081105)

- 在迭代模式下,域名服务器首先在本地数据库中查找,如果服务器查不到相应的记录,会返回另一个可能知道结果的服务器的地址给查询的发起者,以便它向新的 DNS 服务器发送查询请求。

- 在递归模式下,要求域名服务器一次性的完成全部名字解析,即如果不能从该服务器本地得到解析,就由该 DNS 服务器向其他 DNS 服务器发出请求,直到得到查询结果或出现错误为止,相当于由收到递归请求的 DNS 服务器来完成迭代解析中用户的工作。

【RFC-1035】定义了 DNS 查询和应答消息的标准格式,【RFC-1034】中对域名服务器的应答结果进行了详细描述。

3 基于查询事件的日志模型

文中提出了一种基于查询事件的日志模型,每条日志反应了一次完整的查询-响应,并对其进行分类存储,建立索引,提高实时查找和检索的效率。

该日志模型的工作流程如图 1 所示。日志服务器从网络上获取 DNS 查询和应答消息,在内存中缓存并将每条查询请求和其对应的服务器应答消息进行匹配,一段时间后,将匹配完成的消息对按照一定的映射规则分类写入对应的日志文件,并对这些日志文件建立索引,在索引文件中存储相应的索引信息。该模型主要包括 3 个部分:数据采集、查询/应答消息配对算法和日志文件索引结构设计。

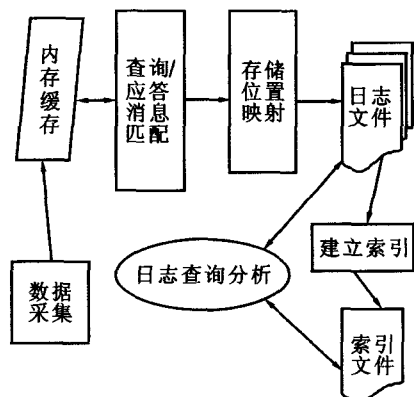


图 1 基于查询事件的日志模型图

3.1 数据采集

将 DNS 解析服务器与日志服务器分离。通过在交换机上配置端口镜像,将日志服务器与镜像端口连接,这样日志服务器就可以捕获解析服务器的所有通讯数据。对 DNS 查询包和应答包的监听过滤可以使用现有的抓包工具,例如 libpcap 函数库,然后

通过对报文进行协议解析获得所需的数据。

3.2 查询/应答消息匹配

3.2.1 匹配原理

DNS 消息头中含有一个 16 位的 ID,它是由发起查询的客户端生成的,服务器将其原封不动的复制到应答消息中,被客户端用来对应所发起的查询和收到的应答。在一定时间段内,可以假设同一个客户端生成的 ID 不发生重复,因此可以通过发起查询的客户端 IP 地址结合 DNS 消息头的 ID 号来唯一确定一次查询事件,也就可以通过比较查询消息和应答消息中的这两个属性来进行匹配。

3.2.2 匹配算法描述

由于查询和应答消息的接受和处理有一定的时间延迟,并且通过网络和服务器的处理后不一定严格按照时间先后顺序排列,所以需要先将消息缓存在内存中,等待进行匹配。由于单位时间内的消息量比较大,对这些消息进行实时匹配的关键是要快速地查找“ip # port + id”相同的消息,这里需要使用一种支持快速查找的数据结构。

Hash 表在 Hash 结果分布较为均匀的时候,能够提供时间复杂度为 $O(1)$ 的查找性能。但是,当源数据经过 Hash 函数之后的结果不均匀的时候,Hash 表就有着较差的性能。此时,设节点总数为 n ,则查找节点的最坏情况下的时间复杂度为 $O(n)$ 。

红黑树是一种自平衡二叉查找树。它能够动态调整各个子树的相对位置和结构,从而保证在任意源数据分布的情况下,都能提供时间复杂度为 $O(\log n)$ 的稳定查找性能。

文中采用 Hash 表和红黑树两者相结合的方式,将“ip # port + id”哈希值相同的消息映射到 Hash 表的同一个桶,每个桶内都用一颗红黑树来组织里面的数据,如图 2 所示。

查找关键字 key 由“IP 地址 # 端口号”和消息 ID 号组成;变量 record-count 表示当前节点中所存在的日志记录个数;变量 log_data 表示日志记录队列,按照时间顺序排序,队列长度即为 record_count 的值。

如图 3, DNS 查询/应答消息匹配算法的执行步骤如下:

- (1) 对每一条消息提取 ip # port 和消息头 id 作为关键字,通过哈希计算映射到哈希桶内的红黑树;

- (2) 以关键字 key 查找红黑树。若查找失败,则向红黑树插入新节点,将新日志记录插入 log_data 中,record_count 为 1。若查找成功,对于应答消息,在 log_data 队列里查找 QNAME/QTYPE/

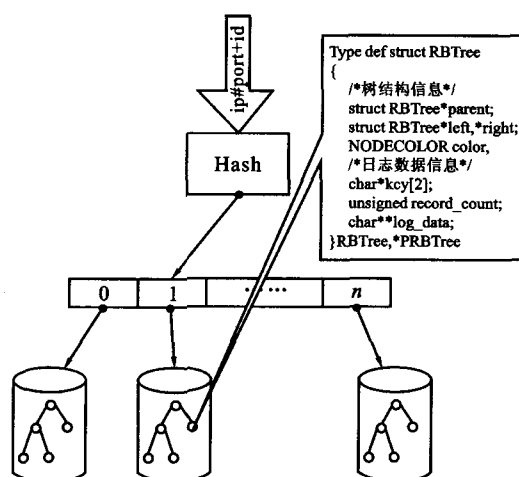


图2 内存数据结构示意图

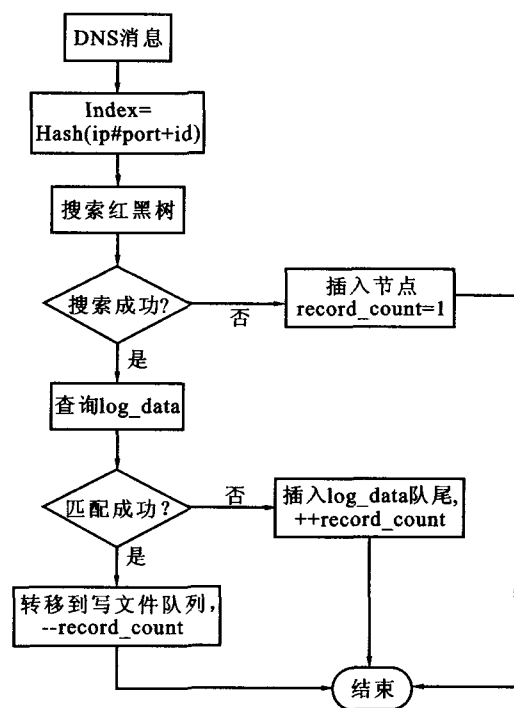


图3 DNS查询/应答消息匹配算法流程图

QCLASS 完全相同的第一条查询记录进行匹配,若匹配成功,则将匹配的日志记录插入到写文件队列中,并从 `log_data` 中删除该记录, `record_count` 减 1;若匹配失败则在队列尾部插入该条记录, `record_count` 加 1。对于查询消息的处理与之相同。

(3) 对于超时未匹配的日志,定时进行清理并插入到写文件队列。

DNS 日志包含了大部分的 DNS 服务信息,为了服务审计和安全需要,必须要对原始的 DNS 服务日志信息在磁盘上进行存储备份。当定时器超时或写文件队列达到一定长度时,将队列中的日志记录全部写入日志文件,并清空队列。日志文件中每条记

录的格式为:

timestamp ip # port query: name class type response: rcode responsetime

每个字段的含义分别为:

timestamp: 日志记录的时间;

ip # port: 请求端 ip 地址和端口号;

query: name: 查询的域名;

class: 表示资源记录的类别,如 IN 表示互联网。

type: 表示资源记录的类型,如 A, MX, NS 等。

response: rcode: DNS 应答包中的回应码,用来标识服务器的应答状况^[4]。

responsetime: 表示解析服务器响应时间,可通过应答时间戳与请求时间戳相减得到。

3.3 基于二维哈希索引的日志存储结构

对于 DNS 系统的实时监控功能,通常需要以下实时统计数据:(1)指定时间段内访问指定域名的所有查询事件;(2)指定时间段内指定 IP 来源的所有查询事件;(3)指定时间段内指定 IP 访问指定域名的所有查询事件。

获取以上信息均需要根据给定条件对日志文件实现查询事件的快速定位。当数据量很大的时候,显然采用文本方式按照时间顺序存储,以逐行扫描方式进行查找的日志模式无法满足要求。现有的数据库系统提供了多种索引结构和查询优化方法,然而对于格式单一的海量 DNS 原始日志,使用数据库的通用接口进行存储和分析的开销太大。文中根据 DNS 日志数据的特点和基于查询事件的日志分析的特定需求,在文件系统之上设计了一种索引结构,易于实现且可灵活地适用于不同的文件系统。

文件在磁盘上是分块存储的,效率主要决定于 I/O 访问次数,由于内存有限,无法将所有数据一次加载到内存。论文提出一种存储结构,按照查询事件分析的关键字段将日志进行分类存储在多个小文件中,并在这些文件上建立索引,查找时能够快速定位目标日志所在文件,避免对其他无关日志文件的访问,可大大缩小需要加载到内存的数据,从而减少 I/O 次数。

首先根据时间段建立不同的文件夹,比如将 1 个小时产生的日志数据存放在同一文件夹下,将该文件夹下的日志记录按照“ip # port”和“name”两个属性通过哈希计算映射到不同的文件。日志文件的索引由 ip 和域名两个属性的哈希值组成,因此称为

二维哈希索引^[5]. 此索引结构能够很好地支持基于查询事件的部分匹配查询和精确匹配查询. 部分匹配查询是指给定 ip 和域名两个属性之一, 查找含有给定值的日志记录, 例如, “查找源 ip 地址为 202.106.196.229 # 43331 的所有日志记录”, 或 “查找查询域名 www.cnnic.cn 的所有日志记录”. 精确匹配查询是指同时指定 ip 和域名查询满足条件的所有记录, 例如 “查找源 ip 地址为 10.10.1.31 # 35878, 查询域名为 www.cnnic.cn 的所有日志记录”.

3.3.1 分类存储均衡问题

使用简单的哈希算法将日志进行分类, 数据在各文件中的分布均衡程度受所使用的哈希函数和 DNS 查询分布的影响. 由于不同时间段的查询分布情况(各域名和 IP 来源的流量)不同, 无法找到一个哈希函数可以保证将不同的源数据进行均匀的分散, 因此, 分类不均匀的情况无法避免, 当大部分日志都映射到同一个文件时, 对该文件中的日志的查找性能将大大降低. 为了解决这一问题, 可按照如下步骤控制日志文件大小的均匀程度: (1) 限制文件的最大尺寸, 设为 M , 具体值根据服务器硬件配置和 DNS 查询分布情况进行设置; (2) 将大于 M 的文件进行分裂, 可采用类似线性哈希的方法, 参见参考文献[5], 这种方法要不断调整原有的索引结构, 增加了复杂度, 带来了一定的开销. 为了满足高性能的要求, 论文简单地将大文件分割为多个不大于 M 的文件, 即采用溢出桶的方法, 将溢出数据存入新的文件; (3) 将小文件合并成一个文件, 进行合并的文件尺寸上限可设为 $M/2$, 以减少文件数目.

3.3.2 布隆过滤器(Bloom Filter)的使用

由哈希冲突的原理可知不同的查询条件(IP 或域名)可能对应同一个哈希索引号, 通过哈希索引只能缩小查找范围, 并不能确定文件中一定存在所查日志, 因此, 对于实际上并不存在的日志仍要扫描文件进行查找, 这样会增加不必要的磁盘 I/O 操作, 从而降低系统的性能. 为了解决这个问题, 给每个文件定义一个布隆过滤器, 相当于该文件的摘要, 在查找该文件之前, 先通过布隆过滤器判断该文件是否含有所查询的记录, 从而可以省去一些不必要的文件操作(布隆过滤器存放在内存当中, 因此对它的计算开销比磁盘查找小). 尤其是当需要查找多个溢出文件时, 可以通过布隆过滤器过滤掉一些文件, 进一步缩小查找范围^[1].

综上所述, 文中提出一种优化的二维哈希索引用于日志的磁盘存储, 如图 4. 索引号用 4 位表示,

前两位对应 IP 地址的哈希值, 后两位对应域名的哈希值, 由下面的式子求得:

$$ip_index = Hash(ip + port) \% iip_num$$

$$name_index = Hash(name) \% iname_num$$

这里设 ip 和 name 的哈希桶数目 iip_num 和 $iname_num$ 均为 4, 所有的组合数目为 16. 映射为 0000 和 0001 的日志数据被合并到同一个文件中, 共享一个布隆过滤器; 0010 索引没有对应的数据; 映射到 0011 索引的数据被分散到了两个文件中.

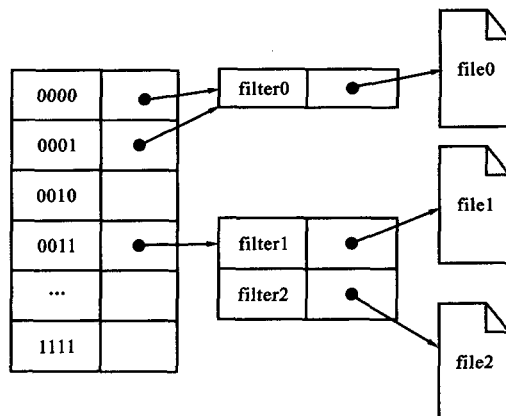


图4 日志文件索引结构图

当查询某个时间段的日志时, 首先找到指定时间段的日志所在的文件夹, 将该文件夹下文件的索引信息和布隆过滤器加载到内存中, 然后根据给定条件(ip 或 name)计算哈希值, 根据索引号找到对应的文件名和布隆过滤器列表, 依次根据布隆过滤器判断是否含有所查 ip 和域名, 并在对应的文件中查找符合查询条件的日志.

4 结束语

文中在提高日志信息的关联性和可读性, 节省存储空间, 建立索引以优化日志的查询分析等方面做了相应的研究, 提出了一种基于查询事件的域名服务器日志模型, 并对模型的设计框架和关键技术的实现方法进行了详细的阐述.

文中为网络服务系统的实时监控提供了一种解决思路, 今后的研究方向主要集中于如何将日志系统集成到域名系统的在线日志分析和安全事件的实时检测与防御响应中.

参考文献:

- [1] Mitzenmacher M. Compressed bloom filters[J]. Networking, IEEE/ACM Transactions, 2002, 10(5): 604-612.

(下转第 102 页)

检测算法能够得到的最小增益. 当存在 10 次相位跳变时, 可以得到的最小增益也有 19.78dB, 会比采用非相干积分得到的增益多 3.52dB. 但是根据捕获卫星数据的经验表明, 如果连续使用 200ms 数据, 其中出现 10 次位跳变的可能性很小, 因此不考虑这种情况. 当存在 1 次相位跳变时, 可以使用 100ms 的数据, 因此可以得到 20dB 的增益. 带四方块的折线是由不同跳变次数下能够得到的最大增益连接而成的折线. 实际上使用 200ms 数据所得到的增益一般都是处于带圈折线和带下三角折线之间的, 因此可以得到的增益是可观的. 同样, 当都使用 100ms 数据时得到的增益可以比使用非相干积分提高 1.45~5.23dB.

6 结束语

考虑到 GPS 信号中存在导航数据位跳变引起的伪码相位改变情况发生, 伪码相位的变化又会导致在 GPS 接收终端处在环境恶劣情况下不能进行长时间的相干积分运算. 非相干积分运算又会导致积分增益的损失, 这都是由于导航数据位的变化引起的. 因此, 文中研究了如何检测导航数据位的跳变及其位置并利用该跳变来实现长时间的相干积分运算, 以提高信号的处理增益. 研究结果表明, 在使用同样长度数据的情况下, 使用这种方法得到的处理增益要比纯粹使用非相干积分高很多, 可以大大提高接收机的接收灵敏度. 该方法特别适合在软件接

收机中使用.

参考文献:

- [1] 王惠南. GPS 导航原理与应用[M]. 北京: 科学出版社, 2003.
- [2] Psiaki M L. Block acquisition of weak GPS signals in a software receiver[C]// Proceedings of ION GPS 2001. Salt Lake City, Cornell University, 2001: 2838-2850.
- [3] 李玉红, 寇艳红, 张其善. 微弱 GPS 信号捕获算法研究[J]. 遥测遥控, 2005, 26(4): 61-65.
- [4] Tsui J B Y. Fundamentals of global positioning system receivers a software approach[M]. 2nd ed. New York: John Wiley & Sons Inc, 2005.
- [5] 李菊, 陈禾. 直接序列扩频信号快速捕获[J]. 北京理工大学学报, 2005(10): 905-908.
- [6] Gérard Lachapelle. Pedestrian navigation with high sensitivity GPS receivers and MEMS[J]. Personal and Ubiquitous Computing, 2007, 11(6): 78-83.
- [7] Liang Y Lie, Hanzo L. Efferential acquisition of msequences using recursive soft sequential estimation[J]. IEEE Transactions on wireless communications, 2005, 4(1): 128-136.
- [8] Iyappan Ramachandran. On acquisition of wideband direct-sequence spread sepectrum signals[J]. IEEE Transactions on communications, 2006, 5(6): 1537-1546.

作者简介:

何秋生 男, (1974-), 博士, 讲师. 研究方向为兼容 GPS/Galileo 数字接收机基带信号处理.

(上接第 98 页)

- [2] Paul Albitz, Cricket Liu. DNS and BIND[M]. 5th ed. Beijing: O'Reilly, 2006.
- [3] 方蕾, 钱华林. DNS 安全漏洞以及防范策略研究[J]. 微电子学与计算机, 2003, 20(10): 53-57.
- [4] Nominum foundation products[EB/OL]. [2009-03-10]. <http://www.nominum.com/>
- [5] 胡勇其, 侯紫峰. 面向语义的日志系统构架设计[J]. 微电子学与计算机, 2006, 23(8): 17-20.

作者简介:

乔婧女, (1984-), 硕士研究生. 研究方向为计算机网络应用技术、互联网寻址技术.

金键男, (1976-), 硕士. 研究方向为计算机网络系统、互联网寻址技术.

毛伟男, (1968-), 研究员, 博士生导师. 研究方向为下一代互联网技术、资源定位与寻址技术、网络安全技术.