



(12) 发明专利申请

(10) 申请公布号 CN 101719847 A

(43) 申请公布日 2010. 06. 02

(21) 申请号 200910197230. 0

(22) 申请日 2009. 10. 15

(71) 申请人 上海寰雷信息技术有限公司

地址 200241 上海市闵行区东川路 555 号乙  
楼 5052 室

(72) 发明人 龙雷

(74) 专利代理机构 上海科盛知识产权代理有限  
公司 31225

代理人 赵志远

(51) Int. Cl.

H04L 12/26(2006. 01)

H04L 12/24(2006. 01)

H04L 29/12(2006. 01)

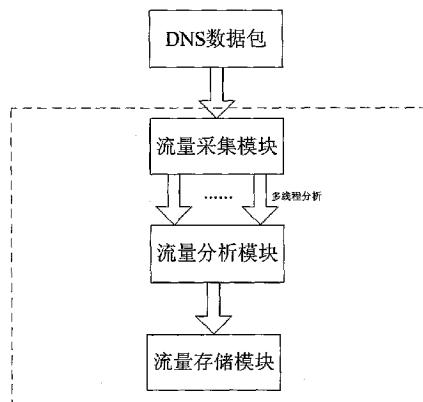
权利要求书 1 页 说明书 5 页 附图 1 页

(54) 发明名称

一种 DNS 流量的高性能监控方法

(57) 摘要

本发明涉及一种 DNS 流量的高性能监控方法，该方法镜像 DNS 流量采集服务器上联通设备的进出流量到 DNS 流量采集服务器，由 DNS 流量采集服务器进行数据包实时抓取、实时分析以及实时存储。与现有技术相比，本发明方法使 DNS 数据分析具有准确性、及时性、实时性以及完整性。



1. 一种 DNS 流量的高性能监控方法，其特征在于，该方法镜像 DNS 流量采集服务器上联通讯设备的进出流量到 DNS 流量采集服务器，由 DNS 流量采集服务器进行数据包实时抓取、实时分析以及实时存储。

2. 根据权利要求 1 所述的一种 DNS 流量的高性能监控方法，其特征在于，所述的上联回线设备包括交换机或者路由器。

3. 根据权利要求 1 或 2 所述的一种 DNS 流量的高性能监控方法，其特征在于，所述的 DNS 流量采集服务器实时抓取的数据包包括：

DNS 服务器收到客户端的查询包 A；

DNS 服务器回复客户端的响应包 B；

DNS 服务器向上层 DNS 服务器发送的递归查询包 C；

DNS 服务器收到上层 DNS 服务器回复的递归响应包 D。

4. 根据权利要求 3 所述的一种 DNS 流量的高性能监控方法，其特征在于，所述的 DNS 流量采集服务器实时分析抓取的数据包，得到关键统计指标值，该关键统计指标值包括：

查询包 A 的每秒查询次数；

递归查询包 C 的每秒递归查询次数；

响应包 B 的每秒响应次数；

查询包 A 以及响应包 B 的解析成功率；

查询包 A 以及递归查询包 C 的递归查询率；

查询包 A 以及递归查询包 C 的缓存命中率；

查询包 A 的查询类型分类；

响应包 B 的响应类型分类；

查询包 A 的 TOP-N 顶级域名；

查询包 A 的 TOP-N 二 / 三级域名；

查询包 A 的 TOP-N 域名；

查询包 A 的 TOP-N 源 IP；

查询包 A 的 TOP-N 源 IP 和域名；

递归响应包 D 的 TOP-N 错误域名；

递归响应包 D 的 TOP-N 错误域名和源 IP。

5. 根据权利要求 4 所述的一种 DNS 流量的高性能监控方法，其特征在于，所述的 DNS 流量采集服务器以 1 分钟为最短时间单位进行实时分析。

6. 根据权利要求 4 或 5 所述的一种 DNS 流量的高性能监控方法，其特征在于，所述的 DNS 流量采集服务器采用多线程技术对 TOP-N 数据进行实时分析。

7. 根据权利要求 6 所述的一种 DNS 流量的高性能监控方法，其特征在于，所述的 DNS 流量采集服务器每整分钟汇总多个线程的统计指标值，得到的统计指标值插入到存储文件或者数据库的数据表中。

## 一种 DNS 流量的高性能监控方法

### 技术领域

[0001] 本发明涉及 DNS 流量监控方法,特别是涉及一种 DNS 流量的高性能监控方法。

### 背景技术

[0002] DNS (Domain Name System) 是域名系统的意思,其作用就是协调 IP 地址和主机名之间的双向切换。DNS 是当今 Internet 的基础架构,众多的网络服务(如 Http、Ftp、Email 等等)都是建立在 DNS 体系基础之上的。各省级运营商(包括固网或者移动运营商)为上网用户提供了 DNS 网络的运营服务,通常情况下,每个省级运营商的 DNS 网络分为若干节点,每个节点是由路由器、交换机和若干台服务器组成,每台服务器运行 DNS 软件,提供 DNS 查询的解析工作。

[0003] 鉴于 DNS 网络的重要性,通过监控 DNS 流量得出统计指标对于掌握 DNS 网络的日常运行状况,特别是及时发现异常 DNS 流量是非常必要。现有的监控方式主要是在每台 DNS 服务器通过抓包或者记录日志的方式抽样分析,得出相应统计指标。这种方式存在一个显著的缺点 - 抽样分析。因为 DNS 流量非常大,每秒有几十万个数据包,抽样分析虽然可以减少数据分析量,但是抽样分析可能造成的结果是刚刚抽样完一个时间点,异常的 DNS 流量进入,却没有分析到。另外该方式还存在另外一个缺点是在 DNS 服务器本身进行分析,额外的分析工作会对 DNS 服务器正常的 DNS 查询解析工作造成影响。

### 发明内容

[0004] 本发明所要解决的技术问题就是为了克服上述现有技术存在的缺陷而提供一种 DNS 流量的高性能监控方法。

[0005] 本发明的目的可以通过以下技术方案来实现:一种 DNS 流量的高性能监控方法,其特征在于,该方法镜像 DNS 流量采集服务器上联回线设备的进出流量到 DNS 流量采集服务器,由 DNS 流量采集服务器进行数据包实时抓取、实时分析以及实时存储。

[0006] 所述的上联回线设备包括交换机或者路由器。

[0007] 所述的 DNS 流量采集服务器实时抓取的数据包包括:

[0008] DNS 服务器收到客户端的查询包 A;

[0009] DNS 服务器回复客户端的响应包 B;

[0010] DNS 服务器向上层 DNS 服务器发送的递归查询包 C;

[0011] DNS 服务器收到上层 DNS 服务器回复的递归响应包 D。

[0012] 所述的 DNS 流量采集服务器实时分析抓取的数据包,得到关键统计指标值,该关键统计指标值包括:

[0013] 查询包 A 的每秒查询次数;

[0014] 递归查询包 C 的每秒递归查询次数;

[0015] 响应包 B 的每秒响应次数;

[0016] 查询包 A 以及响应包 B 的解析成功率;

- [0017] 查询包 A 以及递归查询包 C 的递归查询率；
- [0018] 查询包 A 以及递归查询包 C 的缓存命中率；
- [0019] 查询包 A 的查询类型分类；
- [0020] 响应包 B 的响应类型分类；
- [0021] 查询包 A 的 TOP-N 顶级域名；
- [0022] 查询包 A 的 TOP-N 二 / 三级域名；
- [0023] 查询包 A 的 TOP-N 域名；
- [0024] 查询包 A 的 TOP-N 源 IP；
- [0025] 查询包 A 的 TOP-N 源 IP 和域名；
- [0026] 递归响应包 D 的 TOP-N 错误域名；
- [0027] 递归响应包 D 的 TOP-N 错误域名和源 IP。
- [0028] 所述的 DNS 流量采集服务器以 1 分钟为最长时间单位进行实时分析。
- [0029] 所述的 DNS 流量采集服务器采用多线程技术对 TOP-N 数据进行实时分析。
- [0030] 所述的 DNS 流量采集服务器每整分钟汇总多个线程的统计指标值，得到的统计指标值插入到存储文件或者数据库的数据表中。
- [0031] 与现有技术相比，本发明具有以下优点：
- [0032] 1、本发明是对于所有 DNS 数据包的分析，没有采用抽样分析，保证了 DNS 数据分析的准确性；
- [0033] 2、本发明采用多线程分析 DNS 数据包，保证了 DNS 数据分析的及时性；
- [0034] 3、本发明以 1 分钟为最长时间单位进行分析，保证了 DNS 数据分析的实时性；
- [0035] 4、本发明每分钟存储分析后的数据到文件或者数据库，保证了 DNS 数据分析的完整性。

## 附图说明

- [0036] 图 1 为本发明的原理图；
- [0037] 图 2 为本发明的硬件结构简单示意图。

## 具体实施方式

- [0038] 下面结合附图对本发明作进一步说明。
- [0039] 一种 DNS 流量的高性能监控方法，该方法镜像 DNS 流量采集服务器上联回传设备的进出流量到 DNS 流量采集服务器，由 DNS 流量采集服务器进行数据包实时抓取、实时分析以及实时存储。
- [0040] 所述的上联回传设备包括交换机或者路由器；所述的 DNS 流量采集服务器实时抓取的数据包包括：
  - [0041] DNS 服务器收到客户端的查询包 A；
  - [0042] DNS 服务器回复客户端的响应包 B；
  - [0043] DNS 服务器向上层 DNS 服务器发送的递归查询包 C；
  - [0044] DNS 服务器收到上层 DNS 服务器回复的递归响应包 D。
- [0045] 所述的 DNS 流量采集服务器实时分析抓取的数据包，得到关键统计指标值，该关

键统计指标值包括：

- [0046] 查询包 A 的每秒查询次数；
- [0047] 递归查询包 C 的每秒递归查询次数；
- [0048] 响应包 B 的每秒响应次数；
- [0049] 查询包 A 以及响应包 B 的解析成功率；
- [0050] 查询包 A 以及递归查询包 C 的递归查询率；
- [0051] 查询包 A 以及递归查询包 C 的缓存命中率；
- [0052] 查询包 A 的查询类型分类；
- [0053] 响应包 B 的响应类型分类；
- [0054] 查询包 A 的 TOP-N 顶级域名；
- [0055] 查询包 A 的 TOP-N 二 / 三级域名；
- [0056] 查询包 A 的 TOP-N 域名；
- [0057] 查询包 A 的 TOP-N 源 IP；
- [0058] 查询包 A 的 TOP-N 源 IP 和域名；
- [0059] 递归响应包 D 的 TOP-N 错误域名；
- [0060] 递归响应包 D 的 TOP-N 错误域名和源 IP。
- [0061] 所述的 DNS 流量采集服务器以 1 分钟为最短时间单位进行实时分析。
- [0062] 所述的 DNS 流量采集服务器采用多线程技术对 TOP-N 数据进行实时分析。
- [0063] 所述的 DNS 流量采集服务器每整分钟汇总多个线程的统计指标值，得到的统计指标值插入到存储文件或者数据库的数据表中。
- [0064] 如图 2，本发明部署一台“DNS 流量采集服务器 2”与上联回线设备 1（交换机或者路由器）相联，通过镜像交换机或者路由器上联端口的所有进出流量到“DNS 流量采集服务器”，由“DNS 流量采集服务器”进行实时抓包，实时分析，实时存储。
- [0065] 如图 1，本发明包括三大模块：
- [0066] 流量采集模块，进行实时抓包；
- [0067] 流量分析模块，进行实时分析；
- [0068] 流量存储模块，进行分析后的数据存储。
- [0069] 1、流量采集模块
- [0070] 按照源地址和目的地址以及源端口和目的端口的不同，DNS 数据包分为四类：
- [0071] DNS 服务器收到客户端的查询包
- [0072] DNS 服务器回复客户端的响应包
- [0073] DNS 服务器向上层 DNS 服务器发送的递归查询包
- [0074] DNS 服务器收到上层 DNS 服务器回复的递归响应包
- [0075] DNS 数据包进入“DNS 流量采集服务器”的流量采集模块，流量采集模块根据过滤规则，分别取出符合规则的四种 DNS 数据包，送入流量分析模块。
- [0076] 2、流量分析模块
- [0077] 流量分析模块主要作用是分析四种 DNS 数据包，得出关键的统计指标值。具体的统计指标值以及所分析的 DNS 数据包类（A、B、C、D）如下：
- [0078] 每秒查询次数（A）

- [0079] 每秒递归查询次数 (C)
- [0080] 每秒响应次数 (B)
- [0081] 解析成功率 (A、B)
- [0082] 递归查询率 (A、C)
- [0083] 缓存命中率 (A、C)
- [0084] 查询类型分类 (A)
- [0085] 响应类型分类 (B)
- [0086] TOP-N 顶级域名 (A)
- [0087] TOP-N 二 / 三级域名 (A)
- [0088] TOP-N 域名 (A)
- [0089] TOP-N 源 IP (A)
- [0090] TOP-N 源 IP+ 域名 (A)
- [0091] TOP-N 错误域名 (D)
- [0092] TOP-N 错误域名 + 源 IP (D)
- [0093] 统计上述指标最耗时的部分是如下几种 :
- [0094] TOP-N 顶级域名
- [0095] TOP-N 二 / 三级域名
- [0096] TOP-N 域名
- [0097] TOP-N 源 IP
- [0098] TOP-N 源 IP+ 域名
- [0099] TOP-N 错误域名
- [0100] TOP-N 错误域名 + 源 IP
- [0101] 因为在统计 TOP-N 的数据时需要进行 hash 表的插入操作,这个操作在数据量大的时候非常费时,所以这个部分使用多线程技术。
- [0102] TPool 类初始化的时候会产生一定数量的线程,并维护它们各自的一个 TaskThread 类用以传递数据。每个线程初始化之后会被插入线程池的空闲线程队列 idle\_queue 上,同时自己把自己锁在一个 self\_lock 的锁上,进入休眠状态。一旦系统有工作要做了,比如抓到了一个 A 类包,那么就会调用 TPool 的 assignWork 方法,这个函数会从空闲队列中取出一个线程,为它准备好相应的工作,然后解开它的 self\_lock 让它开始工作。线程工作结束之后会自己把自己插入空闲队列,然后再一次通过加锁 self\_lock 进入睡眠等待下一个任务。这样就能够保证负载在各个线程之间是均衡的,不会出现一个线程比较繁忙一个线程比较空闲的情况。
- [0103] 为了提高效率避免过多的加锁操作带来的等待,多个线程工作时并不是往同一个 MonitorInterface 表里插入,而是每个线程有自己的一个 MonitorInterface 表,工作线程每次只在自己的 MonitorInterface 表插入数据。由于插入数据多了之后 hash 表效率会下降,并且要保证每一分钟要把这一分钟收集到的数据写入到文件或数据库中,因此在 hash 表中增加了一个功能,即统计平均每次插入时要寻找的链表长度,以此来衡量 hash 表的效率。当线程自己的 MonitorInterface 的 hash 表的插入长度超过一个指定的值(一般设为 1)或者是设置的时间到了(一般设为 5 ~ 10 秒的一个间隔),线程就会把自己的

MonitorInterface 表的内容插入总的 MonitorInterface 表中。

[0104] 另外,处于性能考虑,这里用 set\_cpu\_affinity 函数手动指定抓包线程工作在 CPU 核心 1 上,工作线程分布在 CPU 核心 2 ~ 4 上,用 set\_cpu\_mask\_affinity 函数指定每分钟的汇集线程不要工作在 CPU 核心 1 上,这样能够保证抓包线程不会被其它工作线程争夺 cpu 时间。同时还用 block\_sig 函数设定抓包和工作线程不要响应 SIG\_ALARM,防止它们去做汇集的工作。

[0105] 3、流量存储模块

[0106] 流量存储模块每个整分钟汇总多个线程的统计指标值,得到汇总后的统计指标值,直接插入统计指标值到文件或者数据库的数据表中。

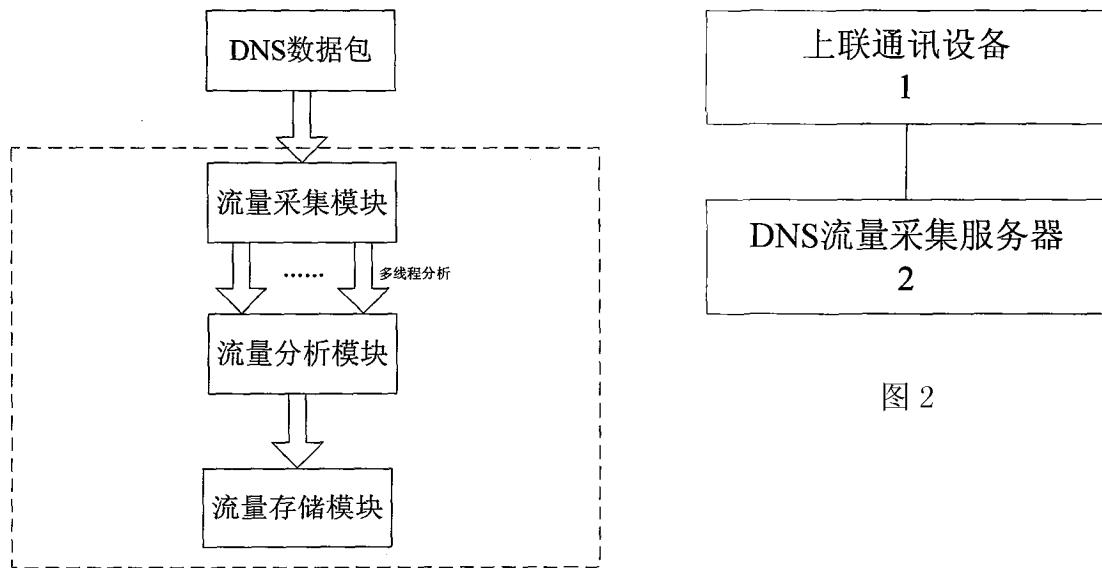


图 1

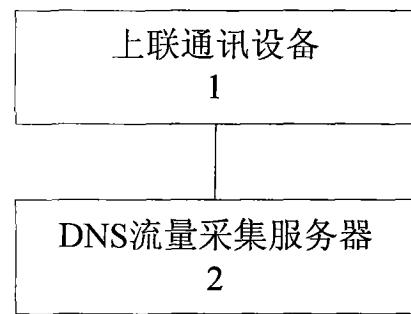


图 2