维普资讯 http://www.cqvip.com

高速网络内容监控系统的设计与实现*)

史永丰[1] 赵燕平[1] 许榕生[2]

(北京理工大学管理与经济学院 北京100081)[1] (中科院高能所计算中心 北京100039)[2]

摘 要 基于CCDGP体系结构,本文提出了一个新颖的基于会话连接的并能预测分析机处理能力的负载均衡算法,该算法综合考虑了分析机的可用性、处理能力、高速网络信息的实时处理及针对应用连接进行内容分析的不可分割性等因素,并结合Wu-Manber快速关键字匹配算法来实现高速网络环境下的网络内容实时监控。实验测试表明该系统能较好地解决高速网络环境下的网络内容监控问题。

关键词 高速网络,CCDGP,内容监控,负载均衡

Network Content Security Monitoring System Based on High-Speed

SHI Yong-Feng ZHAO Yan-Ping XU Rong-Sheng

(School of Management and Economics, Beijing Institute of Technology, Beijing 100081)

(Computing Center, Institute of High Energy Physics, CAS, Beijing 100039)

Abstract A network content security monitoring system suitable to high-speed network environment is discussed in this paper. The system is based on load balance algorithm and Wu-Manber module. The algorithm considers the availability and load of analysis hosts. The system may solve the problem of real-time online monitoring of WWW and E-mail violation information for high-speed networks.

Keywords Content security monitoring, CCDGP, High-speed network, Load balance

1 问题的提出

网络内容监控系统的基本工作原理是根据设置的安全策略,采用关键字匹配或者语义分析等技术手段,对经过被监控点的数据报文进行检查,如果满足安全性要求就进行正常的路由转发,否则根据安全策略的规定进行处理。但这几年来,随着网络的迅速发展,网络带宽增加很快,10M共享的网络迅速被100M、1000M交换网络所取代。当传统的网络内容监控技术被迁移到高速的网络时,随条件不同,会出现大量问题。主要可以归纳为以下两种:

(1)狭义的数据捕获指网络接口从网络上获取通信数据,广义则指从网络上获取数据并且将网络数据传输给CPU进行处理的过程。在这个过程中首先面临的问题是数据获取的位置问题,在基于CSMA/CD技术的100M/10M以太网上,每一个以太网HUB端口能够提供网络上通讯的每一个数据包。在基于交换的以太网或快速以太网上,可以在关键点插入一个HUB或者使用交换设备提供的端口径向来获取必要的信息。但在高速宽带网络上,特别是在主干网上由于没有类似HUB的设备,这些难于实现。一些交换机在千兆以太网上提供端口径向功能,但这常常会降低交换机的性能。在2.5G POS,ATM主干网上多数的路由器上并不提供端口径向。就目前而言,千兆级乃至更高的网络接口技术的发展使网络数据捕获不再是问题,完全可以通过一个网络接口来获取高速网络环境下的所有网络数据[1]。

(2)协议分析策略的优劣将决定数据处理初期的性能。当数据被捕获并传送到监控计算机的缓冲区后,便要对这些数据进行协议分析,在这里需要根据应用功能需求分析制定具体策略。例如,根据应用需求,先对数据包进行过滤,仅将应用功能所需数据包发送给协议分析模块,这将提高处理速度。

本文将提出一种能适应高速网络数据流的网络内容监控系统。第2节分析系统的总体结构,第3节着重探讨它的实现及其关键技术,最后是对系统的性能分析和结论。

2 系统总体结构

高速网络监控技术与传统网络监控技术的不同之处在于前者面对的数据流更快,业务量更大,这就

*)本文得到国家重点基础研究发展规划"973"项目资助,项目编号:G1999035806。史永丰 硕士研究生,研究方向为Web信息挖掘,企业竞争情报分析;赵燕平 教授,研究方向:Web信息检索与数据挖掘,信息管理与信息系统、数理统计;许榕生 研究员,研究方向:网络安全。

维普资讯 http://www.cqvip.com

需要更强的数据捕获和处理能力。虽然千兆级乃至更高的网络接口技术较为成熟，鉴于现有总线，内存的性能限制，快速、大量的网络数据还是难于在一台计算机上进行处理，但可以用分而制之的方法来实现数据处理。我们称之为集中式获取和分布式机群处理机制，用英文描述就是 Centralized Network Traffic Capturing and Distributed Group Processing，简述为 CCDGP 结构。CCDGP 结构的采用不仅简化了数据获取过程，而且分布式处理技术的运用提高了单位时间内所能处理的数据量，提高了总的处理能力。这种方法在应对目前1Gbits/sec 带宽的网络流量方面具有较好的性价比。

为了适应高速网络数据流的内容监控，建立了一种基于 CCDGP 结构的网络内容监控系统(如图1)。该系统主要由数据采集器和数据处理机群组成。

(1)数据采集器。数据采集器采用专用的硬件实现，它主要对高速网段上的数据包进行高速采集，通过简单的数据包分析后，根据数据处理机群系统中各处理机的负载情况，按照负载均衡的原则，将高速的网络数据流分发到各处理机。

(2)数据处理机群。数据处理机群一般由多台进行内容监控的处理机组成，多台处理机在功能上构成一个机群系统，共同对海量的网络数据进行关键词快速扫描。各处理机对从采集器分发过来的网络数据包进行扫描，监测是否有不良信息。

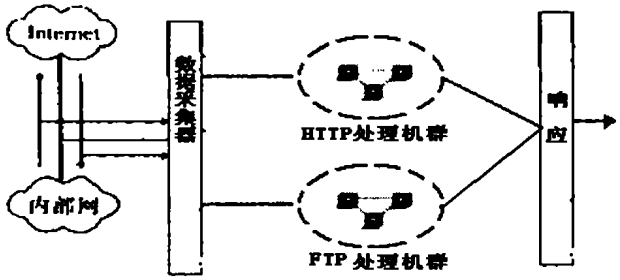

图1 系统结构

3 系统实现及关键技术

3.1 数据采集器

数据采集器主要负责获取高速的网络数据流，并按照负载均衡原则，将高速的网络数据流分发到各处理机。它主要实现3方面的功能：(1)从采集端口获取数据包；(2)对数据包进行预过滤；(3)根据负载均衡的原则将网络数据包分发到数据处理机群系统。

数据包预过滤的主要目的是为了达到缩减数据，去除一些不关心的网络包。为了防止丢包，包过滤只做简单的基于包头内容的过滤，如 IP 地址、TCP/UDP 端口、TCP 标志位等，经过包过滤之后的网络包数据将大大减少。

负载均衡策略是提高处理机集群性能的重要手段。其中最简单的是轮循策略(Round Robin)，但单纯基于包的轮循会破坏连接完整性，导致同一会话中的 IP 包被转发到不同的处理机，从而无法完整地对会话进行监控。为了在 IP 层实现基于会话的负载均衡，本文采用了 HASH 查表的方式，通过建立地址、端口 HASH 表〈源 IP 地址，源端口号，目的 IP 地址，目的端口号〉，通过查表匹配，保证源 IP 地址和端口相同的数据包能够被转发到同一处理机中处理。通过这种方式保证了连接一致性和会话完整性，使得同一会话连接中的 IP 分片能够被转发到同一机器进行 IP 重组和 TCP 流重组并通过查找最小负载的处理机，实现基于流量的动态负载均衡。

为使所有处理机既能共同分担流量，同时又能实现网络内容监控负载的均衡，本系统采用了基于会话连接的动态最小负载优先算法，它能够根据处理机的当前负载、可用性状况及处理机的能力来进行动态调节。具体描述如下：

(1)在每个处理机上配置一个负载指示代理，由这些代理提供处理机的负载情况。我们选择处理机节点等待任务队列中的任务数 t_i 和 CPU 利用率函数为 U_i 来计算第 i 台处理机的负载量 L_i，即

$$L_i=t_i*\alpha+U_i*(1-\alpha) \quad (1)$$

则其当前处理能力 P_i 为：

$$P_i=r_i/(1.25+L_i) \quad (2)$$

其中，i 代表第 i 台处理机，r_i 表示第 i 台处理机的处理能力系数，根据处理机本身的处理性能指标决定；L_i 表示第 i 台处理机的负载量；α 表示影响因子，对处理机负载量的影响系数。

(2)基于会话连接的负载均衡算法动态维护一张数据包分配表，表中包含以下内容：〈源 IP 地址，源端口号，目的 IP 地址，目的端口号，处理机号，会话起始时间〉。为了提高检索的效率，此表采用 HASH 算法实现。当接收到一个 IP 包，利用协议分析取出〈源 IP 地址，源端口号，目的 IP 地址，目的端口号〉，若是新建立连接，找出支持此应用连接的负载量最小又可用的处理机 H，并将地址、端口号、处理机 H 和当前时间存入 HASH 表，同时将该 IP 包发给处理机 H，否则，在 HASH 表中查找此连接所对应的处理机号 H，并将该 IP 包发给处理机 H，同时更新 HASH 表中的时间记录。若连接结束，则删除 HASH 表中的相应记录。

(3)通过负载预测，能满足负载变化的动态性和对资源要求的随机性。负载预测充分考虑到处理机的历史信息和当前负载信息。对处理机的负载预测可用公式表示如下：

$$P_{next}(i)=P_{now}(i)*\alpha+P_{average}(i)*(1-\alpha) \quad (3)$$

其中，$P_{next}(i)$表示处理机 i 在下一个应用连接事务

维普资讯 http://www.cqvip.com

处理的预测处理能力；$P_{now}(i)$表示处理机 i 当前应用连接事务的处理能力；$P_{average}(i)$表示处理机 i 在最近若干应用连接事务处理的平均处理能力；α 表示影响因子，对处理机当前处理能力的影响；$(1-\alpha)$表示对处理机在一段时间内平均处理能力的影响。

3.2 处理机的设计与实现

处理机中关键词的匹配速度可以说是决定系统性能的关键，解决不好，势必成为系统瓶颈，严重影响系统的性能，甚至导致失败。字符串的匹配算法直接影响系统的检测效率。当对网络数据包匹配特定字节串的特征时，需要一个有效的字符串匹配算法。为了提高处理机扫描关键词的效率，本文采用了一种多关键词匹配算法——Wu-Manber 来对信息流进行监测和过滤。Wu-Manber[2]算法是 Boyer-Moore[3]算法处理多关键词问题的派生形式，是一种快速实用的多关键词匹配算法。它采用了 Boyer-Moore 算法的框架，使用块字符(block character)来计算不良字符移动距离(bad-character shift)表(Shift[])。此外，在进行匹配的时候，它使用 HASH 表选择关键词集合中的一个子集与当前文本进行匹配，减少无谓的匹配运算。Wu-Manber 方法的执行时间不会随着关键词集的增加而成比例增长，而且要远少于使用每一个关键词和 Boyer-Moore 算法对文本进行匹配的时间总和。Wu-Manber 算法的时间复杂度在最好的情况能达到 O(B * n/m)(B 是块字符的长度，是算法在每一个入口点计算块字符的时间)。

Wu-Manber 算法框架：

```
计算 Shift[],Hash[],Prefix[]表 //预处理
while(text < textend)//开始匹配
{ hashval = hashBlock(text); //计算当前的块字符
//查找块字符的不良字符移动表得到下一个匹配开始位置
  shift-distance = Shift[hashval];
  if(shift-distance == 0)
//当前块字符出现在某个关键词的末尾(可能的匹配开始位置)
    { shift-distance = 1;
      P = Hash[hashval];
//得到可能与当前文本匹配的所有关键词的集合的开始位置
      while(P)} //检验子集中的关键词是否匹配;
  text += shift-distance; }//选择下一个可能的匹配入口点
```

测试结果表明，Wu-Manber 算法在10000个关键字的情况下，匹配速度达到175.4Mbps。要处理1Gbps 的数据流量，如果不考虑除关键字匹配以外的其它操作，采用合理的数据分流和负载平衡策略，6台同等性能的计算机并行计算就能够满足要求。

4 实验测试

在实验环境里，我们采用一台运行 Linux 的服务器(Intel P III 900，1GB 内存)，作为安全交换机，在其上插了一块千兆网卡和多块百兆网卡，同时通过改造网络上免费的 Web 服务器的源代码实现了对 HTTP 应用协议的检测分析，实际测试表明，当采用 6 台分析主机做负载均衡时，其检测分析能力已接近 400Mbit/s，见图 2，如果增加分析主机的数量，系统的分析能力仍可进一步提高，基本满足了高速网络实际应用需求。

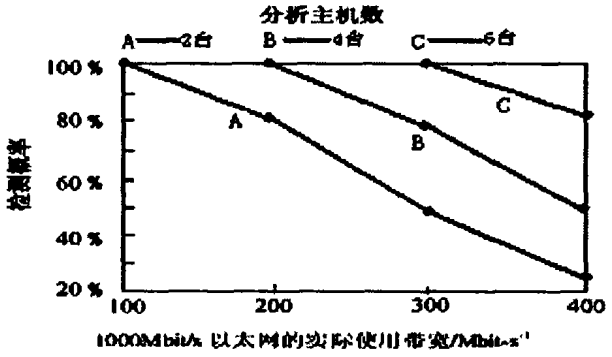

图2 测试结果

结束语 与传统的网络内容监控系统相比，本文实现的高速网络内容监控系统，采用了一种基于 CCDGP 的体系结构，它主要包括高性能的宽带数据采集器技术和基于应用的最小负载均衡技术。该系统可实时完整地获取高速网络数据流，并可将网络数据流及时地分发到后续的处理机；通过采用可扩展的处理机群，该系统可以实时全面地监控和分析高速的网络数据流。但是本系统是基于文本内容的，当敏感信息以图片等其他形式出现时则无法识别。

参考文献

1 MOORE A W, Neugebauer R, Hall J. Network Monitoring with Nprobe [EB/OL]. Http://www.cl.cam.ac.uk/users/iap10/nprobe2002.pdf.

2 Sun W, Manber U. A Fast Algorithm For Multi-pattern Searching[D]. The Computer Science Department of The University of Arizona, 1994

3 Boyer R S, More J S. A fast string-searching algorithm[M]. Communications of the ACM20, 1977.762～772