Web 信息的分布式在线评测技术

毛 昀[1], 汪东升[1], 郑纬民[1], 邓小铁[2]
(1. 清华大学计算机科学与技术系, 北京 100084; 2. 香港城市大学电脑科学系)

摘 要: 通过对 Web 信息的跟踪记录可以实现网上用户行为的分析, 同时也为 Web Server 和 Web Cache 的性能评测提供了原始数据. 本文提出了分布式在线评测解决方案. 该方案通过网络监听产生评测的日志, 并利用分布处理扩大了系统的吞吐率, 提高了系统的可用性. 同时, 通过在线重构、数据内核过滤等关键技术有效地提高了系统的性能.

关键词: Web 信息; 分布处理; 网络监听; 性能评测
中图分类号: TP302 文献标识码: A 文章编号: 0372-2112 (2001) 12A-1836-04

Distributed Online Measurement for Web Traffic

MAO Yun[1], WANG Dong-sheng[1], ZHENG Wei-min[1], DENG Xiao-tie[2]
(1. *Dept. of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*,
2. *Dept. of Computer Science, City University of Hong Kong, Hong Kong*)

Abstract: Detailed measurement results for Web traffic provide the foundations of analysis in both large-scale user's access patterns and performance evaluation of Web servers and Web caches. This paper proposes a distributed online measurement system, which generates rich logs by packet monitoring for Web analysis in the cluster environment. The system gains high throughput and availability by adopting distributed processing technologies. Some new techniques, such as online reconstruction in memory and kernel packet filter, are also adopted to improve the system performance.

Key words: Web information; distributed processing; network monitoring; performance evaluation

1 引言

随着网络技术的发展和 Internet 的普及, Web 信息处理的规模日益增加, 如何构建高性能的 Web 服务器和智能的 Web Cache 成了研究的热点. 为了提供智能信息服务, 网站往往需要进行网络上用户访问行为的分析, 并对每次访问的事件和数据进行详细的记录, 对用户的访问行为建立准确的模型, 以便做进一步的性能分析.

当前得到网络访问行为的方法主要分为四类: (1) 记录 Web server 日志. (2) 修改用户的 Web 浏览器, 对每次访问日志记录. (3) 利用 Web Proxy 的日志. (4) 通过网络监听, 记录用户访问行为. 实际上, 由于网络的复杂性, 只有对网络协议和服务器端、客户端进行综合研究, 才能真正改善网络服务器的性能. 而这样复杂的工作通过网络监听的方式能够完成. 网络监听方式不仅能够记录 HTTP 的请求与回应信息, 而且能够在 TCP 层上记录更加丰富的信息和准确的时间戳. 另外, 由于网络监听方式是被动的, 不会影响当前的网络状况, 更不需要修改客户端或者服务器端的任何软件. 这种方法的缺点是监听系统的开发比较复杂, 须考虑较多因素, 要对监听来的原始数据报文进行 TCP 层和 HTTP 层的重构工作. 为了获得更加丰富的数据, 网络监听系统需要运行在高带宽的网络出入口. 另外如何处理数据, 设计高性能的系统架构, 也是比较复杂的问题.

目前已有一些大学和研究所针对以上问题进行研究. 例如 Virginia Tech 大学的 HTTPDUMP[1] 和 AT&T 研究所的 BLT (Bi-Layer Trace) 项目[2] 分析以上这些研究成果, 可发现此类监听系统仍有以下几点缺陷: (*a*) 不能承受带宽很大的数据 (超过机器网卡限制) 访问. (*b*) 将网络监听下来的数据报文存在磁盘上, 然后进行分析数据的处理方法会导致较多的磁盘 I/O 操作, 影响了性能, 同时涉及泄漏个人隐私的问题. (*c*) 系统的可用性低, 没有容错能力, 一旦遇到电源掉电或者磁盘故障, 系统将全面崩溃. (*d*) 没有考虑用户使用代理服务器访问的情况.

本文设计了一种具有可扩展性的分布式在线评测方案. 该方案提出了分布处理的思想, 从而突破了单机处理数据受到的带宽限制. 该系统同时采用了 heartbeat 和 fake IP 等集群容错技术, 提高了系统的可用性. 在内存中进行在线分析和内核过滤等技术提高了节点的性能. 实验证明, 证明该方法是切实可行的.

收稿日期: 2001-04-02; 修回日期: 2001-08-30
基金项目: 973 国家重点基础研究发展规划项目 (No. G1999032702); 国家自然科学基金 (No. 60003003)

2　分布式 HTTP 协议重构

2.1　整体设计

分布式HTTP 协议重构的系统结构如图1 所示. 数据首先通过数据分配器将监听到的数据分配给各个分析节点. 分析节点通过高速局域网相连, 他们将原始的数据报文重构为TCP 层和HTTP 层的信息, 并将恢复完成的数据保存到后台的存储服务器中. 备份服务器用来处理主存储服务器发生意外事件的情况.

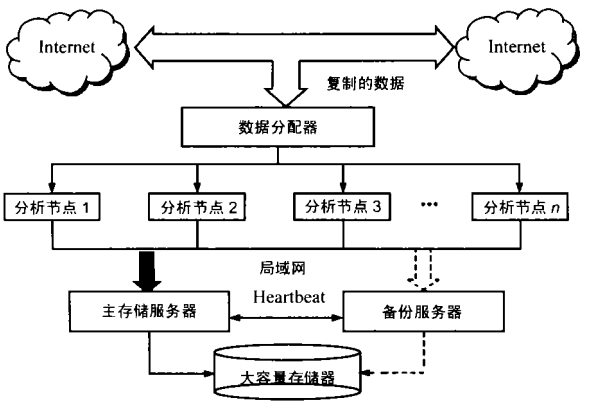

图1　分布式HTTP 协议重构结构图

在分析节点上, 当数据由网卡获得后, 先通过核心过滤器, 过滤掉无用的数据帧, 然后转交给程序的重组进程. 主线程获得数据链路帧后, 去掉帧头, 形成IP 报文, 交给IP 层重组. IP 层需要判断是否为IP 分片, 如果不是, 去掉IP 头, 直接交给TCP 层重组. 否则把IP 分片拼成一个完整的IP 报文, 然后交给TCP 层重组. TCP 层则根据TCP 协议, 将无连接、无顺序的IP 报文恢复成为有连接的TCP 报文. 以上整个的过程就是在模拟一个TCP/IP 堆栈. HTTP 层将HTTP 协议的请求和回应信息抽取出来, 存入本地的硬盘. 转移进程专门负责周期性的将硬盘上的数据传送到存储服务器中. 如果需要对HTTP 的请求和响应进行匹配, 则可以选择离线进行或者在数据分析的时候进行, 因为日志中已经记录了足够的信息可以保证离线匹配的正确性.

2.2　分配算法

由于要处理的对象是高带宽的数据, 因此对数据分配器的性能要求非常高. 通常普通的PC 机不能胜任, 必须采用高性能的路由器.

数据分配器的分配算法必须满足HTTP 重构的要求, 同时保证数据流量大致的负载平衡. HTTP 协议重构需要将同一单向TCP 连接的数据分配到同一分析节点上. 如果算法不能保证这一点, 那么集群节点之间的通讯量将会很大, 使得系统失去了好的可扩展性, 同时降低了整体性能.

路由器的分配策略分为静态路由和动态路由两种. 静态路由可按照数据报文中的目的IP 进行报文转发, 保证同一TCP 连接中的数据路由策略相同; 动态路由则根据当前网络负载状况进行动态分配, 其优点是保证各个分析节点的数据流量相等, 实现负载平衡. 因此, 虽然动态路由的算法有利于数据分配的负载平衡, 但是只有静态路由策略满足HTTP 重构的要求.

本方案设计了一种半静态的分配算法来弥补静态路由策略在负载平衡方面的不足. 路由表每24 小时根据网络流量的变化改变一次. 首先, 假设一共有 n 个分析节点, 将整个网络分成 m 个网段, $\{S_1, S_2, \ldots, S_m\}$, $m \geq n$ 例如 S_1 代表目的IP 地址为 166.111.0.0 到 166.111.255.255 的网段, S_2 代表162.105.0.0 到162.105.255.255 的网段. 假设网段 S_j 第 i 天的流量记做 $T_{j,i}$, 则在第 k 天的开始, 对每个网段 S_j 的流量 $E_{j,k}$ 估计为 $E_{j,k} = (T_{j,k-1} + T_{j,k-2} + T_{j,k-7})/3$. 即当日某网段的流量估计由昨天、前天和七天前的流量进行估计. 这样也将星期几的影响因素考虑在内, 例如周末的访问和平时就有较大不同.

新的路由表根据估计值产生. 显然, 得到最佳分配是一个NP 难题. 这里给出一个求近似解的算法, 可在多项式时间内完成. 算法如图2 所示.

(1) $d_i = 0, I = 1, 2, \cdots, n$. $D = \{d_i\}$
(2) $G = \{E_{j,k}\}, j = 1, 2, \cdots, m$
(3) Repeat
　(a) 令 $E_{c,k} = \max_{\arg j} G$.
　(b) $G = G - \{E_{c,k}\}$
　(c) 选择 i 使得 $d_i = \min D$
　(d) 在路由表中加入规则: 网段 $S_c \rightarrow$ 第 i 个节点
　(e) $d_i = d_i + E_{c,k}$
(4) Until $G = \Phi$

图2　路由表生成算法

3　关键技术

3.1　内存在线重构

BLT 和HTTPDump 是基于TCPDump 的结构, 因此在重构前先将网络监听的数据存入磁盘中, 所有的用户浏览内容, 包括私人信件、信用卡号甚至密码都记入了磁盘中, 这样的策略引入了一些安全问题和不必要的I/O 操作. 为了解决这个问题, 本方案使用内存在线重构技术. 首先, 并不是所有的网络数据都是Web 信息. 校园网内收集到的网络流量数据表明, 在约1 周的时间内, HTTP 协议流量为557.69GB, 约占总流量1266.39GB 的44.0%. 能在分析前提前过滤掉的数据有: (1) 非TCP 的数据报文. 比如UDP, ICMP 或其他类型. 随着视频点播和IP 电话业务量的增长, UDP 协议的数据量增长很快. (2) 某些已知端口的数据. 比如20、21 端口的FTP 协议数据, 23 端口的TELNET 协议数据, 25 端口的SMTP 协议数据等. 这些数据可以通过核心态的过滤器直接过滤. 其次, 协议头的开销, 例如TCP 协议和IP 协议头, 以及链路帧的帧头, 也不应该记录在内存中. 最后, 分布式的系统结构保证了整个网络带宽被分配到了各个分析节点, 每个节点处理的数据量可以控制. 注意, 这里并不能简单的只留下80 端口的数据而抛弃其他所有端口的数据, 因为并不是所有的HTTP 协议都采用80 端口, 而且还要考虑使用代理服务器的情况. 然而, 一个单向TCP 连接

在前 200 字节即可根据它是否遵守 RFC2616[3] 而判定其是否为 HTTP 协议. 这样, 这样内存中实际需要保留的数据绝大多数都是 Web 信息.

因此, 随着普通工作站内存容量的增长, 现有条件下内存在线重构方案是完全可行的. 我们的试验和模拟测试结果证明了这一点.

3.2 容错处理

分布式处理的优势之一就是可以利用其硬件设备的冗余来获得高可用性. 一旦发现节点或守护进程发生故障, 系统能够自动将出错节点删除, 利用剩余节点继续工作.

分析节点的错误处理如下: 存储服务器上运行一个监视进程, 它来检测每个分析节点是否工作正常. 分析节点上的服务进程会每隔 t 秒钟报告工作状态, 如果监视进程收到某节点异常工作状态信息或者工作状态汇报超时, 则该节点被认为发生故障, 路由表中关于该节点的规则都被删除, 并将其数据分配到剩余的 $n-1$ 个节点机中.

排除掉分析节点故障带来的问题后, 存储服务器的可用性决定了整个系统的可用性. 为了减小主存储服务器崩溃带来的问题, 我们为其设立了备份服务器, 并采用 Fake IP[4] 和 heartbeat[5] 技术, 将整个系统的故障率降到最低. 在主服务器和备份服务器上同时运行着 heartbeat 守护进程, 它们通过网络互相发送工作正常的信号. 当备份服务器在预定的时间内检测不到主服务器的正常信号时, 它将采用 fake IP 技术, 设定虚拟的 IP 地址来接替主服务器的存储以及监视的功能; 当主服务器恢复之后, 备份服务器重新收到工作正常的信号, 立刻释放虚拟 IP, 将工作交回给主服务器.

4 性能评价

本系统的测试环境选用 8 个节点的"清华同方探索 108"并行集群计算机. 其中每个节点有 2 个 Pentium III 700E CPU, 512M 内存, 40G SCSI 硬盘, 2 块 Intel EEPRO 100Mbps 以太网卡. 8 个节点均安装 Linux 操作系统, 内核版本为 2.2.16. 8 个节点中, 分别用一个 100Mbps 的 Hub 和 100Mbps 的交换机组成了内网和外网.

由于所用实验硬件设备的限制, 我们选择 Hub 作为网络数据复制器. 7 个节点分别用作模拟 Web 服务器和客户端. 客户端利用 apache bench tool 模拟并发的 Web 访问数据. 余下的节点通过网路监听记录 Web 访问行为. 表 1 为测试结果.

表 1 单节点测试结果

页面大小 (Bytes)	并发连接数	占用带宽 (Mbps)	实际请求数	捕获的请求数
4301	20	8.56	1,000,000	1,000,000
4301	100	33.6	2,000,000	2,000,000
4301	280	43.6	2,800,000	2,800,000
10456	280	86.4	2,800,000	2,800,000

测试结果表明单节点可以很好的适应这种测试环境. 但是, 由于 Internet 上的平均传输速率较低导致在相同带宽下并发连接数目比局域网多, 所以这样的测试环境并不能代表真正的 Internet 上的情况. 而且由于硬件环境的限制, 实验中也没产生超过 100 兆带宽的数据.

为了去除试验环境带来的限制, 我们设计了一套模拟环境来模拟实际 Internet 出入口上的数据流. 模拟程序以在每台节点机上运行, 模拟数据分配器发送到该节点的数据链路帧, 并通过底层接口将这些数据直接传送给核心态过滤器. 这样的模拟环境不需要在真实的网络中传输数据, 因此也避免了硬件环境带来的限制. 通过这种模拟的方法, 可以对系统中的网络流量、平均页面大小和并发连接数进行控制, 实现在大并发量、高带宽的环境中的测试. 通过模拟测试可以计算出系统最大吞吐率, 当网络带宽超过系统吞吐率之后, 将会有一部分信息因处理瓶颈而丢失.

表 2 的数据是我们模拟高带宽的 Internet 网络出入口真实环境的测试结果. 模拟中重要的参数经过了精心选择: 根据文[6]中的调查结果, 我们选择网页的平均大小为 18.7k 字节. 另外, 根据校园网中的流量分析, 共在 412070 秒中内监听到 557.69GB 的 Web 数据. 因此平均的带宽为 557.69 * 1024 * 8/412070 = 11.08Mbps. 同时, 我们发现网络中的并发连接数大约为 8000, 由于希望每个节点能够处理 100Mbps 带宽的数据, 因此平均并发连接数约为 100 * 44.0%/11.08 * 8000 = 32000.

表 2 真实环境模拟结果

节点数	系统吞吐率(Mbps)	每节点平均吞吐率(Mbps)
1	8.498	8.498
2	34.42	17.21
3	96.31	32.10
4	185.9	46.47
5	289.9	57.98
6	418.3	69.71
7	544.4	77.78
8	673.8	84.23

测试结果表明单节点的吞吐率和节点数目有关. 这是因为节点少的时平均每节点分配到数据量大, 并发连接数也很高, 导致吞吐率下降. 当 8 个节点全部投入使用时, 系统的性能完全可以满足真实环境的需要.

5 结语

利用网络监听进行 Web Server 和 Web Proxy 的性能研究是一个很有前途的方法. 这种方法提供了 HTTP 协议和 TCP 协议的信息, 并且能够准确的提供时间戳, 为分析协议和服务器的性能打下了良好的基础. 同时, 这种方法还可以根据记录的 URL 和 Cookie 信息, 分析用户的访问行为, 设计更为准确的用户访问模型. 本文在详细分析了已有的同类网络监听方案的特点和不足之后, 给出了在内存中进行在线的实时过滤的基于集群系统的分布式解决方案. 实验结果表明, 该方案还原 HTTP 协议的内容准确(包括非 80 端口和使用代理等特殊情况), 具有容错机制, 吞吐能力强和不需要附加硬件设备的特点. 将此技术用于网络出入口进行 Web 信息评测具有良好的应用前景.

参考文献:

[1] R Wooster, S Williams, P Brooks. HTTPDUMP: a network HTTP packet snooper [A]. 1996.

[2] Anja Feldmann. BLT: Bi-Layer Tracing of HTTP and TCP/IP [A]. 9th International World Wide Web Conference 2000 [C].

[3] R Fielding, J Gettys, J Mogul. RFC2616: Hypertext Transfer Protocol-HTTP/1.1 [S]. Network Working Group, 1999.

[4] Simon Horman. Creating redundant linux servers [A]. 4th Annual Linux Expo [C], May, 1998.

[5] Alan Robertson. Linux-HA heartbeat system design [A]. 4th Annual Linux Showcase & Conference [C], Atlanta, Oct. 2000.

[6] Steve Lawrence, Lee Giles. Accessibility and distribution of information on the Web. [J] Nature, 400, 1999: 107-109.

作者简介:

毛　昀　男. 1977 年出生于北京市. 清华大学计算机科学与技术系硕士研究生, 研究方向为分布式系统, 参加了国家重点基础研究发展规划项目的研究.

汪东升　男. 1966 年出生于黑龙江省肇东市. 博士, 副教授. 主要从事分布/并行处理、容错计算等方面的教学和科研工作. 近年来在国内外刊物上发表论文 30 余篇.