ISSN1000-9825 Journal of Software 软件学报 2000,11(1):122~125

一个虚拟 Internet 服务器的设计与实现*

章文嵩 吴婷婷 金士尧 吴泉源

(国防科学技术大学计算机学院 长沙 410073)

E-mail: wensong@iinchina.net

摘要 针对已有的解决 Internet 服务器性能瓶颈和可靠性问题的方法所存在的不足,提出基于 IP 层负载平衡调度的解决方法,将一组服务器构成一个可伸缩的、高可用的虚拟 Internet 服务器.通过在服务机群中透明地加入和删除结点以实现系统的伸缩性;通过检测结点或服务进程故障和正确地重置系统达到高可用性.详细讨论了虚拟 Internet 服务器的体系结构、设计方法和实现技术,并给出了相应的性能测试结果.

关键词 Internet 服务器,负载平衡,高可用性,网络地址转换.

中图法分类号 TP393

当今计算机技术已进入以网络为中心的计算时期,大量的应用都围绕着网络进行,对服务器的性能和可靠性提出了越来越高的要求.例如,随着 Internet 的飞速发展和用户的剧烈增长,比较热门的 Web 站点会因为被访问次数急剧增长而不能及时处理用户的请求,导致用户长时间地等待,大大降低了服务质量.这时,Web 服务器的性能往往成为整个系统中的瓶颈.当在 Web 服务器上大量使用 CGI 和数据库等 CPU 密集型应用的情况下,服务器性能瓶颈问题更加突出.另外,随着电子商务等关键性应用在网上的运行,任何例外的服务中断都将造成不可估量的损失,因此服务器的可靠性也越来越重要.

为了有效地解决 Internet 服务器的性能瓶颈和可靠性问题,我们给出了基于 IP 层负载平衡调度的解决方法,将一组服务器构成一个可伸缩的、高可用的虚拟 Internet 服务器.通过使用网络地址转换,使得服务器组的复杂结构对用户是透明的.系统的伸缩性通过在服务机群中透明地加入和删除一个节点来达到;通过检测节点或服务进程故障和正确地重置系统达到高可用性.

1 相关的解决方法

解决服务器性能瓶颈问题的现有方法主要分为以下 3 类.

(1) 单服务器解决方法.当服务器性能会成为瓶颈时,最简单的方法是将其升级为更高档的服务器.但该方法有以下不足:① 升级过程繁琐,机器切换会使服务暂时中断,并造成原有计算资源的浪费;② 越往高端的服务器发展,所花费的代价越大;③ 一旦该服务器或应用软件失效,会导致整个服务的中断.

(2) 基于 RR-DNS 的多服务器解决方法.NCSA(National Center of Supercomputing Applications)的可伸缩的 WEB 服务器系统就是最早基于轮转域名系统 RR-DNS(round-robin domain name system)的原型系统[1~3].其基本思想是:通过 RR-DNS 服务器把域名轮流解析到这组 Web 服务器的不同 IP 地址,将负载分到各台服务器上,从而提高整个系统的性能.然而,该方法存在以下一些问题.① 域名服务系统是按层次结构组织的,各级域名服务器都会缓冲已解析的名字到 IP 地址的映射,它会妨碍 Round-Robin 方法在客户端生效,会导致不同 WEB 服务器间严重的负载不平衡.另外,域名到 IP 地址映射的 TTL(time to live)值较难设定,太大,则负载不

---

* 作者章文嵩,1973 年生,博士生,主要研究领域为并行与分布计算,Internet 技术和对象数据库系统.吴婷婷,女,1975 年生,博士生,主要研究领域为移动数据库技术,Internet 技术.金士尧,1937 年生,教授,博士生导师,主要研究领域为计算机体系结构,仿真技术和性能评价.吴泉源,1942 年生,教授,博士生导师,主要研究领域为人工智能,分布式计算及软件技术.

本文通讯联系人:章文嵩,长沙 410073,国防科学技术大学计算机学院研究生队

本文 1998-10-26 收到原稿,1999-03-01 收到修改稿

平衡更严重;太小,则频繁的域名解析使 RR-DNS 成为系统中一个新的瓶颈.② 由于用户访问请求的突发性和访问方式不同,即使 TTL 值为 0,各服务器间的负载仍存在较严重的负载不平衡问题.③ 系统的可靠性和可维护性差,一台服务器失效或管理员对其进行维护,均会导致域名已被解析到该服务器上的用户出现服务中止.

(3) 基于应用层负载平衡调度的多服务器解决方法.EDDIE[4],Reverse-Proxy[5]和 pWEB[6]都使用基于应用层调度的方法来建立一个可伸缩的 WEB 服务器.它们都将到达的 HTTP 请求转发到不同的 Web 服务器,取得结果后,再返回给用户.该方法也存在一些问题.首先,系统处理开销特别大,致使系统的伸缩性有限.从请求到达至处理结束,调度器需要进行 4 次从核心与用户空间的切换以及从用户到调度器和调度器到真实服务器的两次 TCP 连接,还需要对请求进行分析和重写.一般地当服务器组数目增加时,调度器会很快成为新的瓶颈.文献[7]在 Linux 1.3 版本上应用快速报文插入技术,使得进行负载平衡调度的用户进程访问网络设备接近核心空间的速度,降低了上下文切换的处理开销,但并不彻底.其次,基于应用层的负载平衡调度器与应用协议密切相关,对于 HTTP,Proxy 和 SMTP 等应用协议,需要写不同的调度器.

2 虚拟 Internet 服务器的体系结构

虚拟 Internet 服务器采用基于 IP 层负载平衡调度技术,在操作系统核心空间中将 IP 层上的 TCP 连接负载均衡地转移到不同的服务器上,且调度器自动屏蔽掉服务器的故障,从而将一组服务器构成一个高性能的、高可用的虚拟 Internet 服务器.虚拟 Internet 服务器的体系结构如图 1 所示,整个服务器组的复杂结构对用户是透明的,用户看到的是单个服务器,且无需修改客户端和服务器端的程序.为此,在设计时需要考虑系统的透明性、负载均衡性、容错性和易管理性.

2.1 透明性和网络地址转换

透明性是通过网络地址转换实现的.当用户通过 Virtual IP Address(即调度器的外部地址)访问服务器时,请求报文到达调度器,调度器以负载均衡方法从一组真实服务器选出一个,将报文的目标地址 Virtual IP Address 改写成选定服务器的地址,报文的目标端口改写成选定服务器的相应端口,最后将报文发送给选定的服务器.同时,调度器在 Hash 表中记录这个连接,当这个连接的下一个报文到达时,从 Hash 表中可以得到原选定服务器的地址和端口,进行同样的改写操作,并将报文传给原选定的服务器.真实服务器的回应报文经过调度器时,将报文的源地址和源端口改为 Virtual IP Address 和相应的端口,再把报文发给用户.当连接终止或超时时,调度器将这个连接从 Hash 表中删除.这样,用户所看到的只是在 Virtual IP Address 上提供的服务,而虚拟服务器的结构对用户是透明的.对改写后的报文,应用增量调整 Checksum 的算法调整 TCP Checksum 的值[8],避免了扫描整个报文来计算 Checksum 的开销.

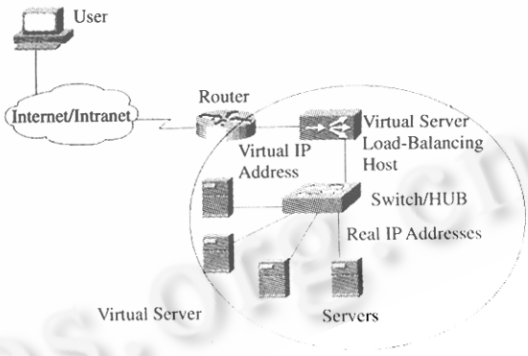

Fig. 1 Architecture of a virtual Internet server

图 1 虚拟 Internet 服务器的体系结构

2.2 负载平衡

负载平衡调度是以 TCP 连接为粒度的.根据 HTTP 协议[9],从 WEB 服务器上获取每个对象都需要建立一个 TCP 连接,同一用户的不同请求会被调度到不同的服务器上,所以这种细粒度的调度完全避免了因用户访问的突发性引起的负载不平衡问题.在调度算法上,我们采用轮转调度(round-robin scheduling)算法和基于权值的轮转调度(weighted round-robinscheduling)算法.轮转调度算法是假设所有服务器处理性能均相同,依次将请求调度到不同的服务器,算法简单,但不适用于服务器组中处理性能不一的情况.为此,使用基于权值的轮转调度算法,用相应的权值表示服务器的处理性能,将请求数目按权值的比例分配到各服务器.调度器需要记录各个

服务器已建立 TCP 连接的数目,当某台服务器被调度时,其连接数加 1;当连接中止或超时的时候,其连接数减 1.假设每台服务器的权值为 $W_i(i=1,\ldots,n)$,TCP 连接数目为 $T_i(i=1,\ldots,n)$,依次选 T_i/W_i 最小的服务器为调度对象.算法实现也比较容易,并且与服务器无关.

2.3 故障检测

我们采用两种方法来检测故障.一种方法是,资源监测器每隔 t 毫秒对每个服务器发 ARP(address resolve protocol)请求,若有服务器过超 r 毫秒仍没有响应,则说明该服务器已发生故障,资源监测器通知调度器将该服务器的所有服务进程调度从调度列表中删除.另一种方法是,资源监测器定时地向每个服务进程发请求,若不能返回结果则说明该服务进程发生故障,资源监测器通知调度器将该服务进程调度从调度列表中删除.资源监测器能通过电子邮件或传呼机向管理员报告故障,一旦监测到服务进程恢复工作,通知调度器将其加入调度列表进行调度.

2.4 服务器管理

服务器可运行任何支持 TCP/IP 的操作系统,可采用任何 Internet 服务器软件,也无需对客户程序作任何修改,可适用于所有 Internet 站点.通过系统提供的管理程序,管理员可发命令,随时将一台机器加入服务或切出服务.

3 系统实现

我们在 Linux 2.0 操作系统源代码上加入和改写了 2 000 多行 C 语言代码,在 IP 层截取和改写 IP 报文,实现了可伸缩的虚拟 Internet 服务器,并提供了一个 ippfvsadm 程序进行虚拟服务器的管理.系统的源程序和使用说明已在网上发布*,至今已经被访问了 6 000 多次.就我们所知,该系统已在美、英、德、澳等国的十几个站点上正式使用.

系统的主要功能模块如图 2 所示,其中 PFVS_Module 是虚拟服务器的主控模块,用于截取和改写 IP 报文;PFVS_table 表存放虚拟服务器的规则;Hash of Connections 表是用于记录当前连接的 Hash 表.ippfvsadm 管理程序通过 setsockopt()函数将虚拟服务器的规则写入 PFVS_table 表中,通过/proc 文件系统把 PFVS_table 表中的规则读出.

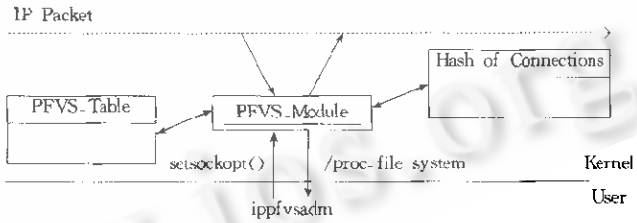

Fig. 2　Mail modules of the system
图2　系统的主要模块

4 性能测试

我们对单台 Web 服务器和多台 Web 服务器构成的虚拟 Web 服务器的最大吞吐率和延时进行了测试对比.测试软件是美国 ZD LABS 的 WebBench 1.1,服务器都是 Pentium 166、32M 内存和 100M Intel EtherExpress 网卡的 PC,运行 Linux 操作系统,Web 服务器是 Apache 1.3.1.客户机是一般的 PC,共有 6 台,运行 Windows 95/NT 操作系统.它们通过 10/100M 自适应的 3Com Switch 3000 交换机相连.在 WebBench 测试时,有一个主控进程和多个模拟访问的客户进程.主控进程生成工作负载和参数,发给客户进程,各个客户进程连续地访问服务器并记录测试数据,等测试完毕后向主控进程汇报,最后由主控进程进行统计.测试结果见表 1.

* Linux Virtual Server Project 项目的网址:http://proxy.iinchina.net/~wensong/ippfvs/

Table 1 Brief testing results of Webserves

表 1 Web 服务器性能测试结果

	Requests per second (GETs/s)①	Throughput (Bytes/s)②	Processing delay③ (ms)
Single Web Server④	97	648.958	35.516
Virtual Web Server⑤(2 set)	190	1269.245	38.474
Virtual Web Server(3 set)	282	1930.934	38.928

①每秒处理请求数,②吞吐率,③处理延时,④单台 Web 服务器,⑤虚拟 Web 服务器.

另外,若用 tcpdump 程序测试得在现有配置下调度器重写一个报文的平均延时为 60μs.设一个报文的平均长度为 536 个字节(非本地 IP 连接的最大段长度),则重写报文的最大吞吐率为 8.93MBytes/s.假设 Web 服务器的平均吞吐率为 600KBytes/s,当服务器的数目升到 15 台时,才达到虚拟服务器的最大吞吐率.可见,它具有良好的伸缩性.

5 结束语

本文对解决 Internet 服务器瓶颈问题的已有方法进行分析比较,指出了它们存在的不足;并给出了基于 IP 层调度的多服务器解决方法,通过网络地址转换、负载平衡调度、故障检测技术,将一组提供并行服务的服务器构成一个高性能、高可用的虚拟 Internet 服务器.该方法具有良好的伸缩性,也无需对客户机和服务器作任何修改,可适用于任何 Internet 站点.

参考文献

1 Katz E D, Butler M, McGrath R. A scalable HTTP server: the NCSA prototype. Computer Networks and ISDN Systems, 1994,27(687):155~163

2 Kwan T T, McGrath R E, Reed D A. NCSA's world wide web server: design and performance. IEEE Computer, 1995,28(11):68~74

3 Brisco T. DNS support for load balancing. RFC 1794, 1995. http://andrew2.andrew.cmu.edu/rfc/rfc1974.html

4 Dahlin A, Froberg M, Walerud J *et al*. EDDIE: a robust and scalable Internet server. 1998, http://www.eddieware.org/

5 Engelschall R S. Load balancing your Web site: practical approaches for distributing HTTP traffic. Web Techniques Magazine, 1998,3(5). http://www.webtechniques.com

6 Walker E. pWEB——a parallel Web server harness. 1997, http://www.ihpc.nus.edu.sg/STAFF/edward/pweb.html

7 Anderson E, Patterson D, Brewer E. The magicrouter: an application of fast packet interposing. 1996, http://www.cs.berkeley.edu/~eanders/magicrouter/

8 Rijsinghani A *et al*. Computation of the Internet checksum via incremental update. RFC 1624, 1994. http://www.internic.net/ds/

9 Fielding R, Gettys J, Mogul J *et al*. Hypertext Transfer protocol——HTTP/1.1. 1997, Http://www.w3.org/Protocols/

Design and Implementation of a Virtual Internet Server

ZHANG Wen-song WU Ting-ting JIN Shi-yao WU Quan-yuan

(*School of Computer National University of Defense Technology Changsha* 410073)

Abstract The shortcomings of the existing solutions to the performance bottleneck and reliability problem of Internet servers are analyzed, and a solution based on IP-level load balancing is given in this paper, in which a scalable and highly available virtual Internet server can be built on a cluster of servers. Scalability is achieved by transparently adding or removing a node in the cluster. High availability can be provided by detecting the node or daemon failures and reconfiguring the system appropriately. The architecture, design and implementation of the virtual Internet server are discussed in detail. Brief performance testing results are also provided.

Key words Internet server, load balance, high availability, network address translation.