

Linux 下基于 B/S 的高并发 Web 服务优化研究

朱明伦, 胡金初

(上海师范大学 数理信息学院, 上海 200234)

摘要: B/S 模型是软件开发的主流技术, 在实际应用中也日益成为主导技术。部署在 Linux 系统下的 B/S 系统在高并发的应用环境中往往会产生 2 种类型的服务瓶颈问题。通过分析这些问题, 提出了一套比较完整的解决方案, 使在硬件条件不变的情况下, 增加 Web 服务器提供服务的能力。

关键词: Linux; 反向代理; 缓存; Web 优化

中图分类号: TP311.52

文献标识码: A

文章编号: 1673-629X(2006)05-0140-03

Research of Using Software Method to Optimize High Load B/S Model Web Service in Linux Environment

ZHU Ming-lun, HU Jin-chu

(Mathematics & Science College of Shanghai Normal University, Shanghai 200234, China)

Abstract: B/S model is the leading technology of the software development. It is used widely in many Web applications. A high load B/S application running in Linux environment will bring two kinds of performance issues. The article analyses these two kinds of problems and gives an intergraded solution. The solution promotes the performance a lot without any hardware upgrade.

Key words: Linux; reverse proxy; cache; Web optimization

0 引言

Web 是目前计算机网络中最炙手可热的技术, 随着 Browser/Server 技术的日益成熟, 基于 Web 的 B/S 软件开发模型已经成为了软件开发中的主流技术。一台或多台运行着 Web 服务的服务器, 在各种技术 (如 CGI, ASP, PHP, Java) 的支撑下成为了计算中心, 而客户端 (Browser, 即浏览器) 仅仅承担了用户界面显示、数据交互等简单的功能。

在大中型应用中, B/S 模型的瓶颈问题在于主要集中在服务器所能承受的用户数量上。每个客户端用户的访问, 都会占用服务器一定的资源。因此, 在硬件条件和软件算法不变的情况下, 每台服务器一般都存在一个最高用户并发数, 一旦访问的用户数量超过这个最高用户并发数, 就会引起系统性能急剧下降、程序执行超时、用户访问无响应等服务器拒绝服务的情况的出现。即使采用了高性能的主机, 也不能解决上万甚至上十万次的用户并发访问, 在这种情况下即使硬件条件再高, 也满足不了用户增长的需求^[1]。

因此, 如何优化 Web 服务器的性能, 使之能满足高强度并发, 是急需解决的问题。

造成 Web 服务器响应缓慢甚至停止响应的主要原因有 2 大类:

- (1) Web 服务器软件进程达到上限;
- (2) I/O 设备或 CPU 超负荷运行。

解决高并发 Web 服务器的性能, 主要有 2 种解决途径: 硬件和软件。虽然硬件方案在性能和可操作性上优于软件方案, 但是硬件价格昂贵, 性价比远低于软件的优化方案。当软件进行了优化, 仍然不能满足业务发展需要时, 再考虑增加硬件, 才是比较科学和经济的做法, 因此, 必须首先研究如何利用软件方式, 通过分析造成 Web 服务器瓶颈的原因, 来对服务进行优化。

文中着重以 Linux 操作系统为环境, 研究问题产生的根源和如何使用各类软件的优化手段使问题得到顺利的解决。

1 使用反向代理技术解决 Web 服务器软件进程达到上限的问题

在 Linux 系统中一般使用 Apache 作为 Web 服务器软件, 通过 Apache 服务器, 用户可以方便地构建各种 Web 应用环境。目前广泛使用的 Apache 1.3.x 有最大进程数限制, 在默认情况最大可设置为 255。用户可以通过修改 Apache 的源代码来增加这个最大进程数, 使之能处理更多的并发用户, 但是这样的做法会造成操作系统进程调度繁忙, 系统负载增大, 并不能真正解决问题。

收稿日期: 2005-08-30

作者简介: 朱明伦 (1981-), 男, 上海人, 硕士研究生, 研究方向为网络和多媒体技术; 胡金初, 教授, 研究方向为网络和多媒体技术。

这个问题造成的主要原因是,Web服务器的并发用户数达到了软件设置的上限,而且由于服务器至客户端的网络传输存在延时,造成Web服务器的进程集中在数据传输的处理上,而不是程序或数据的处理。假设Web服务器需要花5秒钟把数据传输给客户端,如果在5秒内陆续有若干用户进行了连接,使Web服务器的进程数达到所设置的极限,而且每个连接在数据处理上只花费了0.01秒,但需要花5秒的时间把生成的数据发送给客户端。在这种情况下,并发用户数已经达到了最大进程数限制,服务器将不能接受新的服务请求,但CPU却十分空闲,这是一种很大的资源浪费。

反向代理是代理中的一类,它能提供透明的代理服务。当客户端访问反向代理时,反向代理将按事先设置好的策略到真正的Web服务器上取回客户端所需要的内容,并负责传送回客户端。

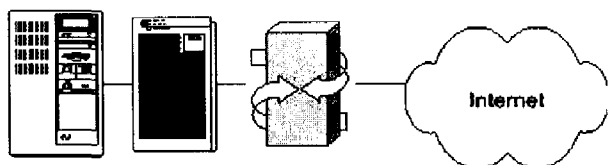
在Linux系统中,可选择Squid作为反向代理软件。Squid使用线程进行工作,不存在Apache中的进程限制问题,由其代替Apache担任传输数据的角色。使用反向代理的好处主要有:

(1)使Apache专心于数据处理,充分利用CPU空闲进行工作。

(2)可担当Web防火墙的功能,可对一般的攻击进行防御。

(3)可作为Web服务器的高速缓存,加快系统的响应。

反向代理典型的网络结构如图1所示。



Web服务器 反向代理 Internet网

图1 反向代理的典型网络结构图

Squid和Apache可共享在一台物理服务器上,也可以安装在二台独立的服务器上。反向代理的主要原理是,当反向代理收到用户请求后,对请求进行一定的判断,如果是正常的访问,则把访问请求向Web服务器提交。Web服务器收到反向代理提交的请求后,根据请求的内容执行相应的操作,生成数据结果,通过本机地址(localhost)或通过以太网把生成的数据反馈给反向代理。至此,Web服务器的工作完成,Apache可接受下一个新的请求,负责把Web服务器生成的数据传输给客户端的工作则由反向代理去完成。

Squid在收到用户的请求后,会对用户提交的HTTP头进行一定的判断和处理,如果用户递交的HTTP头不符合规范,或存在问题,则Squid将扮演Web服务器的角色,直接把出错信息返回给客户端。此外,Squid还能根据管理员事先设置好的策略,把一部分静态内容保存在本地内存或硬盘中。如果客户端访问这些静态内容(如网页上

的JPG图片),则直接从内存或它所在的服务器的硬盘中取出数据直接传输给客户端,而不再向Web服务器索要,进一步节省了Web服务器的开支。

在默认情况下,Squid作为代理服务器运行,要开启Squid的反向代理功能,需要在squid.conf文件中添加下述语句^[2]:

```
http_port 80 //使 Squid 运行在 Web 标准的 80 端口
acl Safe_ports port 81//Apache 运行在本机 81 端口,则定义 81 端口为安全端口,允许 Squid 访问此端口
acl rule1 dst 127.0.0.1//允许 Squid 访问本机地址
acl url1 urlpath_regex php//如果访问的资源扩展名为 .php, 则不作缓存处理,否则不同客户对同一程序的不同请求会得到一致的结果
no_cache deny url1
http_access allow rule1
httpd_accel_host 127.0.0.1//Apache 运行在本机,定义反向代理的 Web 请求转发到本机
httpd_accel_port 81//Apache 运行在 81 端口,所以定义反向代理的 Web 请求转发到 81 端口
httpd_accel_with_proxy off//在开启反向代理的同时,不允许 Squid 作为正向代理运行
```

2 使用负载均衡技术解决 I/O 设备或 CPU 超负荷运行的问题

这问题产生的根源是 I/O 设备或 CPU 不堪重负,导致服务器响应变慢或没有响应。负载均衡技术的主要思想是把高度并发的请求利用算法或策略平均地分配到多台服务器上,这些服务器组成服务器群,就能为高并发要求的服务提供可靠性和高性能性。

以下是2种使用Linux解决负载均衡的方案。

2.1 使用DNS轮询方法解决负载均衡

使用DNS方法解决负载均衡的基本思想是通过把不同的IP地址赋值给同一个域名,使客户端在对域名解析的过程中随机得到一个IP地址,从而访问集群中的某一个服务器。不同的客户端会得到不同的IP地址,从而达到把高并发的请求分散到各个服务器的目的^[3]。

在Linux中使用BIND作为DNS服务器软件,假设现在某网站拥有3台内容一样的Web服务器,其域名为www.foo.com,3台服务器的IP地址分别为202.129.1.1,202.129.1.2和202.129.1.3,则可在BIND的配置文件中设置为:

```
www IN A 202.129.1.1
www1 IN A 202.129.1.2
www2 IN A 202.129.1.3
www IN CNAME www1
www IN CNAME www2
```

利用DNS解决负载均衡的优点是简单、易配置,而且服务器可以分布在不同的物理位置上,不必要求处在同一网段内或处于同一个交换器下。其缺点也十分明显,如果

要增加或减少一台服务器, DNS 更新的速度相对较慢, 有时会花几小时甚至超过 24 小时来同步全球的 DNS 服务器。此外, 利用 DNS 解决负载均衡无法知晓各个服务器的工作情况, 不能知道各个服务器之间的差异, 可能会出现服务请求在某一台服务器上相对集中的情况^[4]。

利用 DNS 解决负载均衡除了可以利用在各种 B/S 系统上, 还可以应用在文件服务中, 用途比较广泛。

2.2 基于 NAT 的方法解决负载均衡

NAT 负载均衡是一种利用将 IP 地址转化为其它 IP 地址(一般是内网 IP 地址)的负载均衡技术, 它的性能较好。NAT 负载均衡可以使用软件或硬件完成。客户端访问的服务器是一台做 NAT 转换的服务器, 该服务器收到用户请求后, 会根据事先设置好的算法自动把请求数据包中的目的地址更改为服务器集群中的某个承担服务的 IP 地址, 从而分散请求^[5]。

在基于 Linux 的 B/S 系统下, 使用该方法可能会遇到问题: 在同一个会话(Session)过程中, 由于会话信息存放在服务器端的内存或硬盘中, 而客户端的每次连接可能将分布到不同的服务器中, 服务器之间无法确定用户的会话是否有效, 以及上次会话是由哪一台服务处理的, 所以这时用户的会话状态会丢失, 也就无法完整地为用户提供服务。

在应用中, 主要有 2 种方案可以解决以上问题:

(1) 使用数据库记录会话状态, 代替原有的使用内存或文件系统存储的方式。使用这种方式需要对源程序作改动。如果源程序抽象性非常强, 则只需更改少量代码实现此功能, 否则对程序将作大规模改动。在数据库负载高的应用中, 如果使用数据库记录会话状态, 会使数据库系统成为系统新的瓶颈。

(2) 使用共享文件系统, 如 NFS。首先把各类应用系统的会话存储设置为使用文件系统的形式, 然后把集群中的一台服务器作为会话共享服务器, 开启 NFS 共享, 其它服务器则把该共享目录使用 NFS 等手段远程挂上本地目录, 实现会话信息的全局共享。

基于 NAT 的负载均衡是一种比较完善的负载均衡方法, 其硬件实现方案也十分成熟。该类负载均衡方法所

有的数据交换必须通过 NAT 负载均衡服务器, 所以它适用于计算密度较高、对 CPU 或磁盘要求较高的应用场合, 不适合类似文件服务等对带宽要求很高的应用。

衡量一个负载均衡方法好坏的主要依据是轮询算法和消息反馈机制。以上几种方案都有相对成熟的轮询算法, 但没有消息反馈机制。一旦服务器集群中的某台服务器出现故障, 也无法知晓, 用户端的请求依然会被分配到出现故障的服务器上。

3 基于 Linux 的软件解决方案总结

综合以上 2 个问题及其解决方案, 可以给出基于 B/S 模型的 Web 服务器的优化解决方案, 其结构是:

(1) 使用多台服务器负载均衡, 选择 DNS 轮询作为负载均衡的手段。

(2) 每台服务器的 Web 服务器运行在非标准端口(如 81), 在 HTTP80 端口运行反向代理服务器。

(3) 在有条件的情况下分离 Web 服务器和反向代理, 以获得最佳性能。

以上结构特别适用于高并发、服务器群安放在异地的大型门户网站。通过以上结构, 可使系统的整体性能有质的飞跃。

参考文献:

- [1] Azar Y, Feder M, Lubetzky E, et al. The Multicast Bandwidth Advantage in Serving a Web Site[J]. Computer Science, 2003, 2233:180-188.
- [2] Oskar Pearson c/o Qualica Technologies (Pty) Ltd. Squid Documentation[EB/OL]. <http://www.squid-cache.org>, 2003.
- [3] Cardellini V, Colajanni M, Yu P S. DNS dispatching algorithms with state estimators for scalable Web-server clusters[J]. World Wide Web, 1999, 2:67-77.
- [4] 王木年, 曹先彬. 一种域分布合作 Web 缓存系统[J]. 计算机研究与发展, 2002, 39(3):275-279.
- [5] 周振. WWW 缓存技术的研究与实现[D]. 大连: 大连海事大学, 2004.

(上接第 133 页)

4 结束语

提出融合漏洞扫描功能的 IDS 系统模型, 通过漏洞扫描与 IDS 联动, 一方面增强了被监控主机的抗入侵能力; 另一方面提高了入侵检测系统的检测效率, 在网络入侵检测中是一种新的尝试。目前笔者根据该模型设计的网络入侵检测原型系统正在开发中, 系统开发平台采用 Windows 2000, 开发工具采用 Visual C++, 前台采用 JSP 技术实现基于 Web 的用户管理界面, 提供查询信息、控制和报警等功能。

参考文献:

- [1] 姚立红, 谢立. IPSEC 与防火墙协同工作设计与实现[J]. 小型微型计算机系统, 2004, 25(2):138-186.
- [2] Kumar G. Classification and detection of computer intrusion [D]. Indiana: Purdue University, 1995.
- [3] 徐漫江. 一种主机与网络相结合的漏洞扫描工具的设计与实现[J]. 雷达与对抗, 2002(2):65-68.
- [4] 杨英鹏, 马建峰. 一种基于代理的分布式抗攻击的入侵检测体系结构[J]. 计算机工程, 2003, 29(13):71-72.
- [5] 陈铁明, 蔡家楣, 蒋融融, 等. 基于插件的安全漏洞扫描系统设计[J]. 计算机工程与设计, 2004, 25(2):194-196.