

并行 WEB 服务器机群中实现动态 负载均衡的请求分配策略*

饶志坚

(云南农业大学经济贸易学院, 云南 昆明 650201)

摘要: 随着 Internet 的高速发展, 采用并行 WEB 服务器机群系统是实现高性能 Web 服务器的一个有效途径, 通过请求分配使机群达到负载均衡是提高这种系统运行效率的基本手段。本文在阐述相关技术的基础上提出一种实测与估测相结合的适用于同构和异构机群系统的请求分配策略, 以使整个系统达到动态负载均衡。

关键词: Web 服务器机群系统; 请求分配; 负载均衡

中图分类号: TP 393.03 **文献标识码:** A **文章编号:** 1004 - 390X(2001)02 - 0124 - 04

并行 Web 服务器机群系统 (Parallel Web Servers Cluster) 是由局域网范围内通过高速网络连接的一组通用 Web 服务器构成的, 这些 Web 服务器可以是同构的也可以是异构的, 它运行相应的服务器软件, 将内部所有的 Web 服务器作为整体对用户并行地提供服务, 对外部而言整个系统就相当于一个高性能的 Web 服务器, 客户方以单一地址的方式访问系统。一般由一台特殊的服务器 (请求分配器) 接收客户的请求, 并按某种策略动态地分配到各 Web 服务器上进行处理, 这就是请求分配。

Web 是一种典型的 Client/Server 结构的应用系统。Client 的 HTTP 请求发送到 Web 服务器上, Web 服务器收到请求后对其进行应答服务, 应答过程使 Web 服务器上产生了负载。Web 服务器负载的轻重将直接影响当前及后续请求的应答速度, 为了提高整个系统的效率, 降低平均应答延长时间, 就要使这种负载在各 Web 服务器间进行分布, 实现负载均衡。请求分配是实现 Web 服务器机群系统的一个关键技术, 请求分配的单位是 HTTP 请求, 分配器并不对请求进行实际的应答, 而是将接收到的每一个请求分发到某一台 Web 服务器上, 由收到请求的 Web 服务器完成实际的应

答工作。

机群系统的一个特点就是充分利用每一个结点的计算能力, 为了在并行 WEB 服务器机群系统中实现这一目的, 就需要研究高效的动态负载均衡机制, 使各结点达到动态的负载均衡, 而实现这一目标的基本途径是请求分配, 本文就此问题提出了一种采用实时负载采集与服务器固有处理能力相结合的请求分配策略, 此策略适用于同构或异构的机群系统。

1 请求分配研究现状

1.1 请求分配技术

实现请求分配的技术主要有以下几种:

(1) 动态 DNS: 通过域名解析服务器 (DNS) 对每次的解析请求 (域名到 IP 地址的翻译) 返回不同的 IP 地址从而将请求分发到不同的 Web 服务器。

(2) IP 地址翻译: 类似于路由技术和防火墙技术, 在网络底层对外界来的 IP 包进行分析并修改源地址和目标地址, 将其发送到系统中的某个 Web 服务器, 并将 Web 服务器返回的 IP 包再进行修改后发送回源目的。

(3) TCP Tunnel: 类似于 Proxy Server 技术,

* 收稿日期: 2000 - 08 - 29

基金项目: 云南农业大学青年基金

作者简介: 饶志坚 (1972 -), 男, 云南思茅市人, 讲师, 主要从事高性能 WEB 服务器的研究。

在收到 HTTP 请求时首先与 client 端建立 TCP 连接,再选择系统中的某一个 Web 服务器建立另一个 TCP 连接,然后在生成一个线程或进程负责在两个连接间进行信息交换,直至任何一方中止连接。

(4) HTTP 重定向:HTTP1.0 以后的协议中有一个重定向命令,对客户请求返回一个重定向命令时,客户端(浏览器)就会按返回的新的地址再发出请求,从而实现请求分配。

1.2 主要的请求分配策略

请求分配策略主要有以下几种:

(1) Round Robin 转轮算法:最简单的请求分配方法,不用考虑各 Web 服务器的当前状态,请求分配器只是将系统中所有的服务器地址组成一个循环列表,对到达的请求从列表中循环选择一个进行分配。

(2) Least Connections First 最少连接优先法:选择当前系统中连接数(当前应答数)最少的 Web 服务器作为分配的目标。

(3) Faster Response Precedence 最快响应优先法:选取当前一段时间内响应最快(请求的应答时间短)的 Web 服务器作为分配的目标。

由于 Web 技术的发展,特别是动态网页技术(如 CGI, ASP 等)的成熟,及数据库、多媒体在 Web 上应用,使得不同的请求在 Web 服务上所产生的负载差异很大,不同客户端发出的对同一个脚本的请求在服务器上所产生的负载也会有很大的差异,以上 3 种方法,依据连接数量或请求处理时间来衡量服务器的负载已不精确了,必须采用新的分配策略,提高动态负载平衡的精度,并能实用于异构的机群系统。

2 实现高效动态负载平衡的请求分配策略

2.1 基本术语

平均应答延迟 MRD(Mean Response Delay):定义为大量请求从建立网络连接到应答结束并断开连接的平均时间。主要用于评价 Web 服务器的平均性能,即广大用户对这个站点的评价的期望值。

最大稳定连接数 MSC(Maximum Stable Connections):定义为系统在效率下降不明显的前提下能够支持的最大并发连接数。此值与 Web 服务器的硬件、软件及系统配置有关,可通过测试得到,但由于每个请求的内容差异及服务器脚本(CGI, ASP 等)的执行在服务器上产生的负载是不同的,

因此 MSC 只能是作为一个估测值。

负载(Load)计算机的忙闲程度,即指当前任务占用 CPU、内存、I/O 等系统资源的比例,这是一个即时量,通常以 0~1 之间的数表示,如 0.8 表示在一定时间有 80%的系统资源正在被占用。

剩余最大稳定连接数 RSC(Remain Stable Connections):对于具有负载为 L 的服务器来说, $RSC = (1 - L) \times MSC$,即系统在效率下降不明显的前提下能够再支持的最大并发连接数。

2.2 HTTP 请求的行为特征

可以将 HTTP 请求分为两类,一类是对服务器固有资源的访问请求,如访问 Html 网页、Gif 图片等,另一类是执行 Scripts 服务器脚本,如 CGI, PERL, ASP, ISAPI 等。不考虑服务器方的 Cache 时,两种请求相同的基本动作有:建立 TCP 连接、分析 HTTP 请求信头、打开访问文件、传送请求内容、关闭文件和断开 TCP 连接。而对于后一种 Scripts 程序在应答时还要执行程序,并将程序运行结果返回给客户,其过程就更为复杂,通常是先运行程序,将程序运行结果放于一临时文件中,最后进行数据传输,如一般的 CGI 方式,也有一边执行程序一边回传运行结果的,如 ISAPI。

由于不同 Client 的网络接入方式的不同(ISDN 或拨号连接),每一个连接就可能具有不同的网络传输速度,在服务器方进行请求应答时将请求内容通过网络传送给客户方的速度也不同。对于一个请求,服务器在应答的过程中是间断地占有系统资源的,在服务器与客户方进行数据传送时,服务器方必需得到客户方对收到数据的回应后才进行下一数据块的传送,这与 TCP/IP 的网络传输协议有关^[1]。这种间隔时间的长短还与网络路由有关,经由不同的路由也会有不同的网络延时时间,占用系统资源的间隔也不同。

在服务器上每一个请求都对对应着一个处理该请求的线程或进程,操作系统在这些线(进)程间进行调度,每一个线(进)程将应答信息写入相关 TCP 的缓冲区中,当缓冲区满时,线(进)程就处于挂起状态,从而不参与 CPU 的调度,当 TCP 的缓冲区数据发送出去后,挂起线(进)程又继续参与调度。总的来说,HTTP 请求有以下 3 个的特点:

(1) 数据的传输在客户方与服务器方是间断传送的,需要两方的确认后才能进行下一数据块的发送。

(2) 不同的客户端发出的请求与服务器所建立的 TCP/IP 连接的网络传输速度有很大差别的。

(3) 一个请求与服务器建立连接以后,其网络传输速度在一定时间段内是相对稳定的。

扩展到多个并发的 HTTP 请求,当服务器同时有多个 HTTP 请求处于应答过程中时,每一个请求分时享用服务器的资源,在单位时间内,网络连接速度慢的连接线(进)程获得服务器的资源相对较少,而总的应答时间也相对较长。正因为如此,服务器的当前应答数与服务器的负载不是一致的。

对于异构的 WEB 服务器机群系统来说,要达到较高的系统效率就应通过请求分配使各服务器所承担的负载与其具有的处理能力成比例。下面的算法就是按照每一台服务器的剩余处理能力的比例进行请求分配。

2.3 分布式负载采集与系统可用性

服务器的负载采集可通过在每台 Web 服务器上运行一个监控进程 Monitor,在将一台服务器加入到系统中时,首先由 Monitor 与请求分配器建立永久的 TCP/IP 连接,并从请求分配器获得负载采集频率,接着就以此频率采集 Web 服务器的负载数据,计算出 RSC(剩余稳定连接数),通过建立的连接发送到请求分配器。

对整个系统来说,如果一定时间内某台 WEB 服务器上的监控程序没有将 RSC 信息送到请求分配器上,则可假定此 Web 服务器出现故障,从而不再将新的客户请求分配到这台服务器上,以提高系统的可用性。

2.4 算法描述

首先需要获得每一台 Web 服务器的固有处理能力,在此我们用 MSC(剩余最大并发连接数)来表示系统的固有处理能力,这可通过测试获得,现有一些测试软件就具有此功能^[2]。

各台 Web 服务器上运行一个监控进程 Monitor,用于定时获取服务器的实时负载信息,获取频率可由请求分配器统一管理。监控进程获取服务器负载后,通过事先测定的 MSC 计算出 Web 服务器的剩余处理能力 RSC,然后将计算结果送至请求分配器,对于在两个 RSC 数据到达的时间段接收的 HTTP,请求分配器将按当前 RSC 的比例进行请求分配,这样不断的通过监控程序进行修正,最终使系统维持动态的负载平衡。

2.5 请求分配算法

在请求分配器上需要维护一个服务器状态表 WST,该表的结构如下:

IP	Port	RSC	RATE
----	------	-----	------

IP: Web 服务器的 IP 地址,对客户请求通过查表可获得相应 Web 服务器的 IP 地址,从而进行请求分配;

RSC:存放监控进程送来的相应 Web 服务器的剩余最大稳定连接数;

RATE:当前状态下各服务器 RSC 的比例;

系统运行过程中,请求分配器实时接收各临控进程送来的 RSC,并将其存放于服务器状态表 WST 中,对 WST 的状态转变时间间隔内到达的请求将按 RSC 的比例进行分配。为高效实现按比例分配请求,请求分配器中需要两个链表,表中结点的数据结构的结构如下:

Index	Count	Next
-------	-------	------

其中 Index 是相应的 WEB 服务器在 WST 表中的索引,Count 用于对每一轮分配的计数器,即本轮分配中对应的 WEB 服务器分配数额,Next 是指向下一结点的指针。

通过负载采集,从 WEB 服务状态表中可以获得当各 WEB 服务器可再接收的请求数量之比,若两个 RSC 数据到达的时间为 T,在 T 这段时间内对到达的请求按比例进行分配,具体分配时,是根据 RSC 的值计算出 Rate 的比例进行分配的,Rate 的计算方法如下:

$RSC_1, RSC_2, \dots, RSC_n$ (RSC_i :第 i 台服务器的 RSC, n 为系统的当前规模)

$$\text{Max Rsc} = \max(RSC_1, RSC_2, \dots, RSC_n)$$

$$\text{Rate}_i = \left[\frac{RSC_i \times R}{\text{Max Rsc}} \right]$$

通过处理后每一个 Rate_i 小于等于 R , R 通常可取值为 10。

系统中结点的数量与系统中可用的 WEB 服务器数一致,每增加一台 WEB 服务器时,就创建一个结点。

这些结点由 2 个链表来管理,1 个是循环链表

LD,由 Count 不为零的结点组成,另 1 个是链表 LE,回收本轮分配中 count 为零的结点。

LD 链表的整理:依次扫描 WST 中的每 1 行,对 Rate 不为零的项从 LE 表中取出 1 个结点,将 count 值设为 Rate,将 Index 值设为 WST 中对应的索引,通过 Next 域形成 1 个循环链表,同时设置 1 个指针 Cur 指向链表中的第 1 项。

当有 1 个新的请求到达时,分配算法如下:

(1) 从 Cur 指向的结点中取出 Index 域,在 WST 中找出对应的 WEB 服务器的 IP 和端口,进行请求分配。

(2) 将结点的 Count 值减 1。

(3) 如果减一之后 Count 为零,则表明此服务器在本轮内的分配数量已用完,把结点从 LD 表中移出到 LE 表中待用。

(4) 如果 LD 表为空,则表明本轮分配结束,重新整理链表 LD,进入下一轮分配。否则将 Cur 指向 LD 表的下一个结点。

当有新的 RSC 数据到达时,就需要重新计算

出 Rate,并根据 Rate 的值整理 LD 链表,进行新的请求分配。

3 结论

分配策略通过将实测与估测的方式结合起来,将负载与 MSC 作为请求分配的依据,避免了采用服务器的当前应答数或应答速度进行负载估计而出现的误差。将服务器的剩余负载转变为 RSC,实现了在同构或异构的并行 WEB 服务器机群系统中通过请求分配达到动态负载平衡的目的。

[参 考 文 献]

- [1] KRIS JAMSA, BEN COPE. Internet Programming [M]. 北京:电子工业出版社,1998.
- [2] 邸烁.异构 Web Server 集群请求分配和负载平衡研究[D][博士论文].北京:清华大学计算机系,1999.
- [3] YEAGER N J, MCGRATH R E. The Advanced Guide for World Wide Web Information Providers[M]. San Francisco, CA: Morgan Kaufmann, 1996.

A Strategy of Request Dispatching for Load Balancing in Parallel Web Server Clusters

RAO Zhi-jian

(Faculty of Trade and Economy, Y A U, Kunming 650201, China)

Abstract: With the thriving of the Internet, Applying the Parallel Web Server Clusters system is an efficient way to achieving the high performance of the Web Server. And a basic method of improving the performance of the system is to balancing the Server Load by dispatching the requests. Based on the relevant technologies, this paper proposes an efficient algorithm of request dispatching to balance the whole system load dynamically. It dispatches the requests by analyzing the real-time load and the Maximum Stable Connections (MSC). This algorithm can be applied to both the heterogeneous system and the homogeneous system.

Key words: Web Server Clusters; Request dispatch; Load-balancing