

服务器负载均衡技术研究*

周莹莲 刘 甫

(湖南涉外经济学院 长沙 410205)

摘 要 负载均衡是建立在网络结构之上的一种调度策略,能有效的扩展服务器带宽和增加吞吐量,增强网络数据处理能力。根据实际层次不同,对负载均衡技术进行了分类,从应用范围、运行原理等角度分析并比较了常用的负载均衡技术和算法,展望了负载均衡的发展。

关键词 负载均衡;负载均衡算法;服务器集群

中图分类号 TP393

Research on Load Balancing of Web-server System

Zhou Yinglian Liu Fu

(Hunan International Economics University, Changsha 410205)

Abstract Load balancing is a kind of dispatch strategy established on network structure, it can extend the bandwidth and increase the throughput of server, enhance the ability of data processing in network. Techniques and algorithms about load balancing are analysed and compared from application field and operation principle, the prospect of load balancing on Web-server system is showed.

Key Words load balancing, load balancing algorithms, service cluster

Class Number TP393

1 引言

随着网络技术的深入应用和网站访问量的飞速增长,如何解决负载急速增长,以确保提供不间断、高质量的服务成为网络运营急需解决的问题。

负载均衡是解决这一问题的可靠技术。采用负载均衡技术的网络服务系统由多台服务器以对称的方式组成一个服务器集合,每台服务器都具有等价的地位,都可以单独对外提供服务而无须其他服务器的辅助。通过某种负载分担技术,将外面发送来的请求均匀分配到对称结构中的某一台服务器上,而接收到请求的服务器独立地回应客户的请求。这种将系统负载分配到不同的服务器上处理,籍此提供解决大量用户并发访问服务,实现并行处理的方法,就是负载均衡(LB, Load Balancing)思想。

负载均衡有两方面的含义:一是将大量的并发访问或数据流量分担到多台节点设备上分别处理,以减少用户等待响应的时间;二是将单个重负载的运算分担到多台节点设备上做并行处理,每个节点设备处理结束后,将结果汇总,返回给用户,系统处理能力得到大幅度提高。负载均衡能够均衡所有的服务器和应用之间的通信负载,根据实时响应时间进行判断,将任务交由负载最轻的服务器来处理,以实现真正的智能通信管理和最佳的服务性能。这种集群技术可以用最少的投资获得接近于大型主机的性能,可以满足不断增长的负载需求。

2 负载状态的评价及获取

2.1 负载状态的评价

目前,负载均衡技术面临三大问题:服务器的

* 收稿日期:2009年12月2日,修回日期:2010年1月5日

基金项目:湖南省教育厅科学研究项目(编号:08C511)资助。

作者简介:周莹莲,女,硕士,研究方向:软件工程,计算机网络,数字视频技术。刘甫,男,博士,副教授,研究方向:多媒体视频技术,数字图像处理。

负载状况的定义、如何获取以及获取后如何处理^[1]。

负载状况的评价由三类组成: 静态的、统计类的和动态的。静态的由资源决定, 可以数字化评价; 统计类的只能作为网站扩充的依据; 动态地使用探测结果来评价。

2.2 负载状况的获取

由于系统中各服务器根据所有服务器的负载情况来决定新的用户访问请求的取舍, 因此, 各服务器必须知道其他服务器的负载状况, 这要求每个服务器必须将自己的当前负载及时通知其他服务器。探知负载状况的技术主要有 3 种:

1) 利用 Agent 技术使用网管协议 Snmp; 在需要的服务器上运行设计的 Agent, 定时通知分配器, 从而得到服务器的确切情况。

2) 第三方的软件: 有一些第三方开发的软件可以完成这种工作, 需要分配器有相应的接口, 从而可以调整分配的结果。

3) 自己开发的软件去主动获取: 最简单的集成方法就是自己去开发, 由分配器去探测服务器的负载状况。

负载状况的收集方式有周期性和非周期性两种。周期性的收集方式会定期询问或广播服务器的负载状态; 非周期性的收集方式只在出现过载或空闲时才询问或广播服务器的负载状态。

选择何种方式要视具体应用而定, 一般来说, 可以根据系统开销和负载状态的实时性这两个因素考虑选择那种收集方式。

负载状况根据收集范围的不同又可以分为全局信息、局部信息和历史纪录。全局信息有可能获得最优解, 但代价可能太大; 局部信息获得局部最优解, 但代价通常较小; 历史纪录通过对服务器过去进行情况的了解来确定服务器的负载信息。

3 负载均衡技术与算法比较

根据负载均衡所采用的设备对象分为软/硬件负载均衡; 根据 OSI 的网络分层体系模型, 可分为基于域名系统的负载均衡, 基于客户端的负载均衡, 基于应用层的负载均衡, 基于 IP 层的负载均衡, 基于 TCP 层的负载均衡; 根据负载均衡算法来划分, 可分为静态负载均衡和动态负载均衡; 从应用的地理结构来分, 可分为本地负载均衡和全局负载均衡。本地负载均衡是指对本地的服务器集群做负载均衡, 全局负载均衡是指对分别放置在不同

的地理位置、在不同的网络及服务器群集之间作负载均衡。

3.1 软/硬件负载均衡

软件负载均衡解决方案是指在一台或多台服务器相应的操作系统上安装一个或多个附加软件来实现负载均衡, 它的优点是配置简单、灵活, 成本低廉。缺点是每台服务器上安装额外的软件运行会消耗系统不定量的资源; 可扩展性不太好, 受到操作系统的限制。

硬件负载均衡设备由真实服务器和负载均衡器两类。真实服务器为用户提供真正的服务, 它的一些性能数据是负载均衡的主要依据之一。负载均衡器由专门的设备完成专门的任务, 独立于操作系统, 其功能是根据用户和真实服务器的各种情况(这些情况要能反映服务器的负载状况或性能状况)通过一定的算法进行调动和分配工作, 从而提高由真实服务器整体构成的网络的稳定性和响应能力。它主要是集中所有的用户请求, 然后分配到多台服务器上处理, 来提高系统的处理效率。有些负载均衡器集成在交换设备中, 置于服务器与因特网链接之间, 有些则以两块网络适配器将这一功能集成到计算机中, 一块连接到因特网上, 一块连接到后端服务器群的内部网络上。

一般而言, 硬件负载均衡在功能、性能上优于软件方式, 不过成本昂贵。

3.2 经典的负载均衡技术

1) 基于域名系统的负载均衡

该方法是最早的负载均衡技术^[2], 在 DNS (Domain Name System, 域名系统) 中为多个地址配置同一个域名, 使得查询这个名字的客户机将得到其中一个地址, 从而不同的客户可以访问不同的服务器, 达到负载均衡的目的, 这就是基于域名系统的负载均衡。很多知名的 Web 站点都使用了这个技术, 包括 Yahoo、Sina 等站点^[3]。DNS 轮循实现起来简单, 无需复杂的配置和管理, 简单, 易行, 并且服务器可以位于互联网的任意位置。但是不能考虑距离远近, 减少记录的有效期 TTL 易造成网络信息冗余。

2) 基于客户端的负载均衡

该方法指的是在网络客户端运行特定的程序, 该程序通过定期或不定期地收集服务器群的运行参数: CPU 占用情况、磁盘 I/O、内存等动态信息, 再根据某种选择策略, 找到可以提供服务的最佳服务器, 将本地的应用请求发向它。如果负载信息采

集程序发现服务器失效,则找到其他可替代的服务器作为服务选择。整个过程对于应用程序来说是完全透明的,所有的工作都在运行时处理,是一种动态的负载均衡技术。服务提供一个 Java Applet 在客户端浏览器中运行,Apple 向各个服务器发送请求收集服务器的负载等信息,再根据这些信息将客户的请求发到相应的服务器。该方法的缺点是透明性不够完善,不具有普遍的适用性。

3) 基于应用层负载均衡调度的负载均衡

该方法将多台服务器通过高速的互连网络连接成一个集群系统,在前端有一个基于应用层的负载调度器^[4]。当用户访问请求到达调度器时,请求会提交给作负载均衡调度的应用程序,分析请求,根据各个服务器的负载情况,选出一台服务器,重写请求并向选出的服务器访问,取得结果后,再返回给用户。该方法存在的问题是:系统处理开销较大,致使系统的伸缩性有限;基于应用层的负载均衡调度器对于不同的应用,需要重写不同的调度器。

4) 基于 IP 层的负载均衡

基于 IP 层的负载均衡方法是现在最常用的方法。用户通过一个单一的 IP 地址(称为虚拟 IP 地址)访问服务器。访问请求的报文先到达负载均衡器,由它进行负载均衡调度,从后台服务器池中选出一个,将报文的目标地址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。真实服务器的响应报文经过负载均衡器时,将报文的源地址和源端口改为虚拟 IP 地址和相应的端口,再把报文发给用户。具体实现的方法有两种。

(1)网络地址转换(NAT)。优点:真实服务器可运行在任何支持 TCP/IP 的 OS 上,能使用私有 IP 地址,仅需要一个合法的 IP 地址分配给前端分配器。缺点:可扩展性不够好,当真实服务器的结点数增加至 25 或更多时,前端服务器会成为整个系统的瓶颈,因为请求包和应答包都须经过前端分配器重写。

(2)IP 隧道(IP tunnel)。前端分配器收到用户请求包后,根据 IP 隧道协议封装该包,然后传给某个选定的真实服务器;真实服务器解包出请求信息直接将应答数据包传给用户。IP 隧道是一种用 IP 数据包封装 IP 数据的技术,它能使发往某一 IP 地址的数据包经过封装后转发到另一 IP 地址。优点:在 IP 隧道实现技术中,前端分配器只将请求送

往不同的真实服务器,真实服务器应答应用户请求。因此前端分配器可以处理大量请求,管理大约 100 个真实服务器且不会成为系统瓶颈,最大流量可达 1Gbps。缺点:要求所有真实服务器支持 IP 隧道协议。但随着 IP 隧道协议成为操作系统的标准,该技术将可以应用于所有的 OS。

5) 基于 TCP 层的负载均衡

该方法又被称为基于分发器或第四层的负载均衡调度^[9],通常对外都有一个公用的虚拟 IP 地址(Virtual IP Address),用户通过这虚拟地址访问服务时,访问请求的报文会到达虚拟服务器主机,由它进行负载均衡调度,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。真实服务器的回应报文经过虚拟服务器主机时,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。该方法用分配器监听 TCP80 端口并转发,但是容易形成分配器瓶颈。

3.3 负载均衡算法

算法是影响负载均衡的重要因素,目前,常用的负载均衡算法可以分为静态、动态两种。静态负载均衡适用于同构并可预知负载量的集群系统,算法简单,易于实现。动态负载均衡适用的范围则广而灵活,需要动态收集服务器的使用情况以及每个任务的执行特征,开销比静态负载均衡大,但可以实时反映集群系统的负载情况。

1) 常用的静态负载均衡算法^[6]主要有 4 种:

(1)轮循调度(Round-Robin Scheduling)算法。多个 IP 地址以轮换的方式返回给客户。优点:简单方便,响应速度快,它无需记录当前所有连接的状态,是一种无状态调度,在一定程度上起到负载均衡的作用。缺点:没有考虑不同服务器提供服务能力的不同,服务器是否可用、服务器的负载情况、客户与服务器之间的距离等因素。(2)加权轮循调度(Weighted Round-Robin Scheduling)算法。用相应的权值表示服务器的处理能力,按权值的高低和轮循方式,分配请求到各服务器,权值高的服务器比权值低的服务器处理更多的连接,一段时间后,各服务器处理的请求数趋向于各自权值的比。优点:算法简单高效,考虑了不同服务器的处理性能差异。缺点:当请求服务时间变化大时,单独的加权轮循调度算法可能会导致服务器间负载不均衡。

(3) 目标地址散列调度 (Destination Hashing Scheduling) 算法。通过一个 Hash 函数将一个目标 IP 地址映射到一台服务器, 它先根据请求的目标 IP 地址作为散列键, 从静态分配的散列表中找出对应的服务器, 如果该服务器是可用的且未超载, 则将请求发到该服务器, 否则返回空。(4) 源地址散列调度 (Source Hashing Scheduling) 算法。与目标地址散列调度算法相反, 它根据请求的源 IP 地址, 作为散列键, 从静态分配的散列表中找出对应的服务器, 若该服务器是可用的且未超载, 则将请求发送到该服务器, 否则返回空。在实际应用中, 目标地址散列调度和源地址散列调度可以结合使用在防火墙集群中, 它们可以保证系统的唯一出入口。

2) 常用的动态负载均衡算法^[7] 主要有 4 种:

(1) 最小连接调度 (Least-Connection Scheduling) 算法。该算法把新的连接请求分配到当前连接数最小的服务器, 它通过服务器当前所活跃的连接数来估计服务器的负载情况。优点: 能比较准确的反映服务器的负载状况。缺点: 没有考虑客户与服务器之间距离、各服务器提供服务的能力等因素。(2) 加权最小连接调度 (Weighted Least-Connection Scheduling) 算法。克服最小连接算法的不足, 用相应的权值表示服务器的处理能力, 将新的连接请求分配给当前连接数与权值之比最小的服务器。加权最小连接方法也存在不足: 相同的连接数并不能表示相同的负载, 以连接数来表示节点负载情况并不准确, 而且随着负载的增多, 相应服务器节点的处理能力也会发生变化。(3) 基于位置的最小连接 (Locality-Based Least Connections Scheduling, LBLC) 算法。针对请求报文的目标 IP 地址, 找出最近使用的服务器, 若该服务器可用且没有超载, 则将请求发往该服务器, 否则使用最小连接原则进行选择。(4) 带复制的基于位置的最小连接 (Locality-Based Least Connections with Replication Scheduling, LBLCR) 算法。针对请求报文的目标 IP 地址找出最近使用的服务器组, 按最小连接原则进行选择。若服务器没有超载, 则将请求发送到该服务器, 否则按最小连接原则从整个集群中选出一台服务器, 将该服务器加入到服务器组中, 将请求发送到该服务器。

通常情况下, 动态负载均衡较静态负载均衡有 30%~40% 的性能提高^[8]。对一些机器—任务映射策略在执行资源调度期间根据实际情况进行确

定。因此, 动态负载均衡算法要考虑系统在实际运行中的负载变化。但是, 它存在两个值得研究的问题: 一是已有的动态负载均衡算法采用连接数, 任务数等指标来表示负载, 但不能体现当前服务器节点的负载状况。如果集群系统提供了多种服务, 那么每个服务请求带来的负载量就不同, 所以, 相同的连接数或者任务数并不能表示相同的负载; 二是系统在长时间运行下, 计算得到的节点负载量无法得到修正, 必然会背离节点的实际负载状况, 从而导致负载的不均衡。

4 结语

在服务系统中, 负载均衡的目的是根据系统中各个处理机的性能及其负载来分配任务, 服务器节点的处理能力和当前的负载量是影响服务器负载变化的主要因素。负载量是一个动态值, 确定它的参数需要首先确定一个动态调度策略, 负载均衡策略的核心是负载均衡算法。设计算法时要尽量减小算法复杂度。目前, 动态负载均衡技术有取代静态负载均衡技术的趋势, 那么, 设计负载均衡策略应该考虑以下几点: 1) 为了保证系统在长时间运行状态下, 负载不发生较大倾斜, 负载均衡系统每次选择的服务器, 应该是服务器集群中负载较小的; 2) 为了充分利用节点的处理能力, 负载均衡系统在进行决策时, 应该考虑全局的负载状态, 获取的负载信息也要保证是最新的; 由于用户请求的动态变化, 服务器系统各处理节点上的负载也在不断变化, 所以就要求系统能够根据某种动态均衡策略进行服务节点负载的均衡, 为用户提供服务; 3) 在得到每个服务器的负载状态信息之后, 可以在综合考虑系统结构和负载特征等因素的基础上, 对一般的均衡算法进行改进或者把几种方法结合在一起使用, 以更好的适应系统的变化。

参考文献

- [1] 薛军, 李增和, 王云岚. 负载均衡技术的发展[J]. 小型微型计算机系统, 2003, 24(12): 2100~2103
- [2] 常潘, 沈富可. 使用域名负载均衡技术实现校园网对外服务器的高速访问[J]. 计算机应用, 2007, 27(7): 1585~1586
- [3] 肖军弼, 王宇. 应用 DNS 实现 Web 集群服务的负载均衡[J]. 计算机系统应用, 2003(10): 59~62
- [4] 张前进, 齐美彬, 李莉. 基于应用层负载均衡策略的分析与研究[J]. 计算机工程与应用, 2007, 43(32): 138~

区域半径 $r=2$, 无线信号覆盖范围直径为 100m, 初始设定每个节点能量为 1000 个能量单位, Energy 表示共耗费的能量, Survival Nodes 表示剩余节点数目。

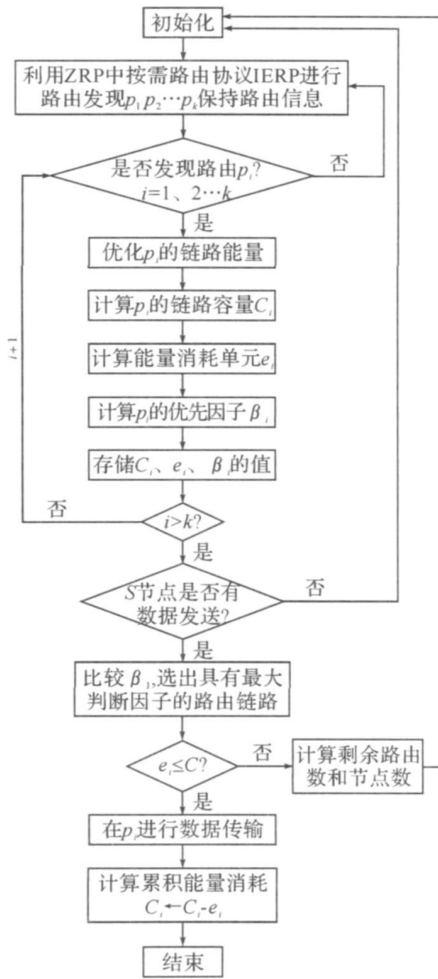


图3 一次数据传输过程中能量优化的主要步骤

用 gawk 命令对仿真输出的能量变化 trace 跟踪文件中的进行分析, 得到一组相关数据, 用 gnuplot 把数据用图形表示出来^[8], 图 4 表示 ZRP 协议和 T-ZRP 协议耗费的能量比较, 图 5 表示 ZRP 协议和 T-ZRP 协议的网络剩余节点数比较, 通过分析发现采用 T-ZRP 路由协议, 不仅可以节省网

络能量, 而且可以推迟路由更新, 减少失效的节点数, 从而延缓网络分裂。

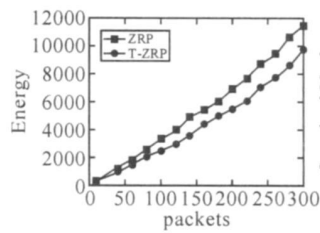


图4 两种路由协议的能量消耗的比较

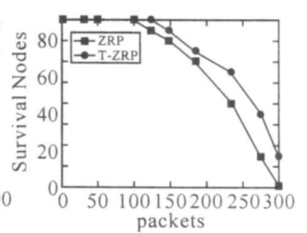


图5 两种路由协议的剩余节点数目比较

4 结语

本文在原有的 ZRP 路由协议基础上作了一些改进, 在其区域间按需驱动路由算法内加入一种基于判断因子的节能算法, 对路由的链路能量进行优化, 并设置判断因子对路由进行选择, 从而达到节能的目的, 通过仿真表明 T-ZRP 协议将会较好的延长网络的存活时间。

参考文献

- [1] 陈林星, 曾曦, 曹毅. 移动 Ad Hoc 网络: 自组网分组无线网络术[M]. 北京: 电子工业出版社, 2006, 4
- [2] IETF Internet Draft. The Zone Routing Protocol (ZRP) for Ad Hoc Networks [J/OL]. 2002, 1
- [3] 段斌, 冯军. 半径自适应算法在 ZRP 协议中的应用[J]. 电力系统通信, 2008, 29(7)
- [4] 马明辉, 武晓庆, 武穆清. ZRP 路由协议的 NDP 优化与仿真分析[J]. 无线电工程, 2007, 37(7)
- [5] 李志远. 嵌入服务通告与发现的 ZRP 协议[J]. 通信技术, 2008, 41(5)
- [6] I. Stojmenovic, X. Lin. Power Aware Localized Routing in Wireless Networks [J]. IEEE Trans. Parallel and Distrib. Sys, 2001, 12(11): 1122 ~ 1133
- [7] 于斌, 孙斌, 温暖, 等. NS2 与网络模拟[M]. 北京: 人民邮电出版社, 2007, 4
- [8] 柯志亨, 程荣祥, 邓德隽. NS2 仿真实验[M]. 北京: 电子工业出版社, 2009, 3

(上接第 14 页)

- [5] 黎哲, 郭成城, 陈亮. 一个基于 TCP 迁移机制的第七层负载均衡系统[J]. 计算机应用研究, 2005(5): 116 ~ 118
- [6] REYNOLDS D A, QUATIERI T F, DUNN R B. Speaker verification using adapted Gaussian mixture models[J]. Digital Signal Processing, 2000, 10(1): 19 ~ 41

- [7] 王晓川, 叶超群, 金士尧. 一种基于分布式调度机制的集群体系结构[J]. 计算机工程, 2002, 28(3): 131 ~ 133
- [8] Keren A, Barak A. Adaptive Placement of Parallel Java Agents in a Scalable Computing Cluster [C] // Proc. of the Workshop on Java for High Performance Network Computing. Palo Alto, CA, USA: ACM Press, 1998