基于Linux的服务器负载均衡性访问的解决方案

傅 明 程晓恒 王 玮
(长沙交通学院计算机工程系 410076)

摘要: 服务器集群(Cluster)是一种服务器上使用的高端技术,使用一台以上的服务器虚拟成一台服务器对外提供高可用性高性能的服务。本文讨论了在Linux环境下,依据LVS实现的服务器集群技术在Web服务均衡负载方面的应用。LVS采用一定的算法将外部的访问负载均衡分布到多台内部的真实服务器上,并通过心跳侦测和自动接续达到高性能服务器的要求,满足大访问量、长时间不间断服务的需要。

关键词: 服务器集群 LVS 负载均衡 高可用性 GPL许可

1 概述

服务器集群 (cluster) 技术是在服务器上使用的高端技术。它通过采用多台相对低配置的服务器并行工作,来完成高配置服务器的工作,甚至可以突破某些限制,更好地实现服务功能。它能够提供相同服务或实现相同目的,提高服务的稳定性和核心网络服务的性能

Cluster分为High-Availability (HA)、Load Balance和Scientific三个部分,分述如下:

1.1 High-Availability (HA)

HA通过多台计算机使用一定的方式(heartbeat)互相侦听对方的工作情况。平时主服务器对外提供服务,当主服务器发生故障,备份服务器无法侦听到主服务器的heartbeat时,便自动接替主服务器的工作,待主服务器恢复正常后,再把工作交还,这样用户就感觉不到停机的影响。heartbeat能提供HA,但不能提供扩展性,也就是说cluster的性能就是一台服务器的性能。所以heartbeat一般都是和load balance结合起来使用。单纯的load balance而无HA得话,则当负责分配任务的router (switcher) 出故障时,整个cluster都会不工作。

1.2 Load Balance

Load Balance主要的应用是Web服务器,同一个IP地址后面实际采用了多台主机。主服务器响应对于这个IP地址的请求并通过一定的算法把任务分配给多台真实节点的服务器,实现负载的均衡。同时,还具有易扩展性,能够方便地添加新的节点服务器,它还能监测每一台真实节点的工作状态,动态修改自己的真实节点列表。

LVS提供了4种负载平衡方法 (Load-balancing Methods)和3种转发机制 (Traffic Forward Mechanism),参见表1和表2。

表1 负载平衡方法 (Load-balancing Methods)

Name	Description
Round robin	Distribute jobs equally among the real servers.
Least-connec-tions	Distribute more jobs to real servers with fewer active connections. (The IPVS table stores active connections.)
Weighted round robin	Distribute more jobs to servers with greater capacity. Capacity is indicated by the user-assigned weight, which is adjusted upward or downward by dynamic load information.
Weighted least connections	Distribute more jobs to servers with fewer active connections relative to their capacity. Capacity is indicated by the user-assigned weight, which is adjusted upward or downward by dynamic load information.

表2 转发机制 (Traffic Forward Mechanism)

	VS-NAT	VS-TUN	VS-DR
Server	any	Tunneling	non-arp device
Server network	private	LAN/WAN	LAN
Server number	low (10~20)	High	high
Server gateway	load balancer	Own router	own router

1.3 Scientific

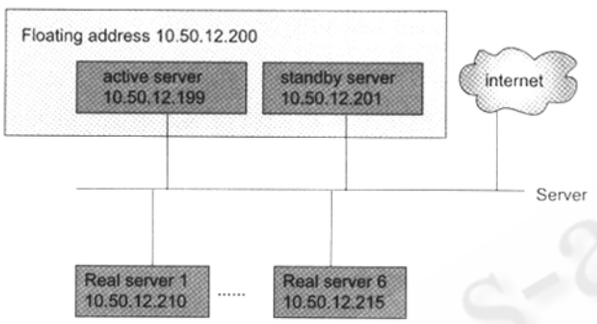
Scientific主要用于计算量大的情形 (如图象处理等)。这种cluster的主要目的是为了提高计算机处理任务的速度。应用程序需要被写成一种分布式处理模式,在运行时被分成分布式处理的进程,同时运行在多个节点

上。在电影《泰坦尼克号》的特效制作过程中，使用了120台Alpha工作站，其中一百余台运行的是Linux,这些Linux工作站正是以Cluster的Scientific方式工作的。

2 解决方案及测试

由于NAT方式有瓶颈的限制，本文采用了LVS/Tunneling方式进行测试，这种方案需要每个真实节点都有一个公共IP。

下面是配置LVS/TUN的结构图。



在这样的工作状态下，Active server的网卡绑定IP10.50.12.199，standby server的网卡IP是10.50.12.201。在正常工作的时候，active server网卡alias虚拟出第二个IP10.50.12.200，所有发往10.50.12.200的请求都被转向10.50.12.210和其他真实的server.而真实server的返回结果不需要经过10.50.12.200那条返回路径，就可以直接返回客户端，这样，就没有瓶颈的问题。standby server随时通过心跳检测在监视active server的工作状况，一旦active server出现故障，在一个心跳周期内，无法检查到active server心跳存在，standby server就会自动接替active server的工作，alias出10.50.12.200这个IP，从而保证了服务的高可用性。一旦active server恢复正常，standby server会重新交出控制权，继续监听。整个服务器集群对外表现正如一个虚拟出来的服务器一样，这就是LVS这个名称的来历。服务器内部的切换过程，对于用户来说是不可见的，是完全透明的。

在实现过程中，选用了新版的Redhat Distribution 6.2(Zoot) with :

Kernel 2.2.14-5.0 ;Piranha 0.4.12 ;Apache 1.3.12; PHP 3.0.15; X window 11R6.

在active server、standby server和real nodes都正常工作的情况下，打开piranha,运行三个dos窗口来ping -t，对10.12.50.199-201三个IP进行监视，此时可以发现，server上运行nanny进程分别监视每一台real node的工作情况,三个IP的ping值正常。拔掉standby server的网线则可见10.50.12.201的ping窗口不通，其余正常。表示此时standby server处于后备状态，没有工作。接上standby server,拔掉active server网线，此时IP=10.50.12.200和10.50.12.199的ping不通，standby server通过heartbeat监测到active server没有响应，立即alias10.50.12.200(经过了一个heartbeat周期，这里设置为10秒)10.50.12.200可以ping到，浏览一切正常，实现了HA的容错性。再把active server插上，用ipconfig查看，10.50.12.199重新alias出10.50.12.200，standby server交出控制权。拔掉任意real node的网线，从piranha的浏览器monitor状态中可以看到nanny进程检测到real node没有响应，已经把该node从node list中删除。再把real node接回，nanny进程检测到real node的响应，将node重新添加入node list.

通过一些感性直观的测试，证明服务器已经可以正常工作，但是为了在系统性能方面得到一个更加清楚的认识，本文采用量化的方式来进行测试。借助了一个第三方软件,微软公司的Web Application Stress Tool(WAS，Web应用负载测试工具）进行服务器的测试评估。

2.1 测试一

内容：测试LVS在两个server,六个real node组成的LVS via Tunneling系统性能。

测试方法：利用WAS，模拟200个并发的http访问，持续10分钟，前后各有10秒的热身和冷机的时间，模拟生成64K的ISDN（单口）速率进行访问，在其过程中，为了不影响测试用的windows 2000机器性能，进而对测试结果产生影响，我们关闭多余的后台程序，关闭节能方式，关闭屏幕保护，通过另外一台windows 98 SE机器telnet到随机的real node上，top观察系统负载，CPU占用率，内存和交换分区的使用情况；测试完成后，WAS生成报告，存档备用。测试结果：

第一次抽样：

4:58pm up 2 days, 22 min, 1 user, load average: 0.00, 0.49, 0.79

56 processes: 55 sleeping, 1 running, 0 zombie, 0 stopped

CPU states: 0.0% user, 1.5% system, 0.0% nice, 98.4% idle

第二次抽样:

4:59pm up 2 days, 23 min, 1 user, load average: 0.00, 0.40, 0.74

55 processes: 54 sleeping, 1 running, 0 zombie, 0 stopped

CPU states: 0.7% user, 1.5% system, 0.0% nice, 97.6% idle

随机取值两次，单机real node CPU空闲率平均为98%

WAS reports(部分)

Page	Hits	TTFB Avg	TTLB Avg	Auth	Query
GET /	13400428.9		8809.7	No	No

2.2 测试二

内容：测试LVS在直接访问一个原LVS的real node的系统性能。

测试结果：随机取值两次，单机real node CPU空闲率平均为:78.45%

WAS reports

Page	Hits	TTFB Avg	TTLB Avg	Auth	Query
GET /	13400786.27		8807.33	No	No

从200个并发http访问LVS和单机的情况来看，单机的CPU占用率在200个httpd（http daemon）进程运行以后，CPU空闲资源迅速降低到78.45%，而LVS(六个real node)由于采用load balance技术，平均每个node分配到三十余个httpd进程，系统资源状况没有明显的下降，空闲资源仍有98%。当然，我们不能由此认为linux系统单机作为Web Server服务不可取，由于http进程的面向无连接的特性，一个http进程在传输结束以后也就立即结束，而不是一直保持连接状态，因此，200个并发的http访问，在经过了多达每秒22.34个并发请求，经历10分钟时间，在实际web serve的过程中，与中等以上的网络浏览量。在PC相同硬件配置下，服务器性能表现较好的操作系统是FreeBSD和Linux。

下面具体的分析WAS reports的关键数据（黑体部分）。页面摘要部分提供了页面的名字，接收到第一个字节的平均时间（TTFB），接收到最后一个字节的平均时间(TTLB)，以及测试脚本中各个页面的命中次数。TTFB和TTLB这两个值对于计算客户端所看到的服务器性能具有重要意义。TTFB反映了从发出页面请求到接收到应答数据第一个字节的时间总和（以毫秒计），TTLB包含了TTFB，它是客户机接收到页面最后一个字节所需要的累计时间。

通过比较TTFB和TTLB的值，可以很明显的看出，TTFB的值在LVS和Single两次实验中的值有着明显的差别:428.9/786.27 (ms)，在单节点的情况下，TTFB的值是LVS的1.83倍。但是略有意外的是TTLB分别是8809.7和8807.33(ms)。也即TTLB的值没有差别。

经过分析，原因如下：一个http的浏览请求是从客户端(browser)发出一个请求,在得到服务器的响应之后，确定服务器可以通过通过路由器到达后，再发出一个内容为“GET / HTTP/1.0/r/n/r/n”的http request,服务器在port 80侦听到http request后，产生一个httpd的守护进程，发送请求的页面(/)给客户端，结束进程，同时http浏览过程完成。

在这里，我们认为单机服务器在负载大量并发的http request的情况下，对于新的http request，产生新的httpd以响应其请求的响应时间相对于LVS的时间要长，这是因为LVS每一台real node的系统负载比单机的要小得多。这是为什么TTFB有明显的差别的原因。

但是对于设置的/index.html页面，由于内容非常简单，信息量很小，可以从WAS给出的详细统计报表中看出在1 0分钟内,客户端一共接受到941707.12KB(Total Bytes Recv),平均速率是1569．80KB／S（Bytes Recv rate）。对于测试采用的内部百兆局域网并使用交换机的情况下，传输速率不是瓶颈，传输质量在LAN内部也很好，不会出现packet lost的情况。因此，一旦httpd生成，开始响应客户端的http request，传输问题对于上面内部百兆局域网的理想状况不是制约速率的瓶颈。据推测，在更接近于真实网络状况的测试条件下，应该会有更加理想的测试结果。

3 结论

LVS能够虚拟出一台能够满足大流量多进程情况下的具有高可用性的Web Server。LVS的优势在目前的测试条件下，暂时没有表现出速度上面有比较令人惊奇的结果，原因是多方面的，最大的可能性是网络条件过于理想，并且流量和进程数仍然不多。但是可以看到，对于Web Server来说，最重要的要求，LVS已经出色的完成了。在两百个并发访问的情况下，系统性能几乎没有变化，CPU空闲保持在98%以上，这是令人满意的。

目前LVS/Piranha计划的缺憾，在于暂时只能支持http和ftp请求。我们也希望能找到在LVS下支持多种协议的LVS实现。另外，LVS/Piranha不支持real node之间的数据自动同步。这就要求用其他方法来完成。目前考虑了NFS、rsync等方法，经过衡量，考虑到安全性等方面，我计划采用rsync+cron来实现。此外，Piranha没有提供数据容错方面的支持。■

参考文献

1 Wensong Zhang, "Creating Linux Virtual Servers", LinuxExpo 1999 Conference。

2 http://www.linuxvirtualserver.org。

3 news://news.freesoft.cei.gov.cn。

4 http://elinux.uhome.net。

5 陆涛涛等, Red Hat Linux 6.x 入门与提高，北京清华大学出版社，2000。6 中科红旗软件技术有限公司编著，红旗Linux网络管理教程，北京电子工业出版社，2001年。