第 33 卷 第 3 期 计 算 机 工 程 2007 年 2 月
Vol.33 No.3 Computer Engineering February 2007
·网络与通信· 文章编号：1000—3428(2007)03—0136—02 文献标识码：A 中图分类号：TP393

无负载均衡器的 Linux 高可用负载均衡集群系统

谢作贵，戚骁亚
(广东省 Linux 公共服务技术支持中心，广州 510057)

摘 要：针对由交换机连接的双机(或多机)负载均衡集群系统，将负载均衡功能转移到真实服务器节点上，从而不需要单独的负载均衡器节点，而且服务器节点之间互为备份，在一个节点失效后，另外一个备份节点将接管其工作，从而构建一个不需要负载均衡器的 Linux 高可用负载均衡集群系统。

关键词：虚拟 MAC；负载均衡；高可用；Linux

High-availability and Load-balance Cluster System without Load-balancer

XIE Zuogui, QI Xiaoya
(Guangdong Linux Technology Service Center, Guangzhou 510057)

【Abstract】 For the load-balance cluster system connected by a switch, this paper proposes moving the load-balance function to the real-server nodes, so that there is no need for a load-balancer. And real-server nodes will backup each other. When one node fails, another backup node will take over its work. As a result, it realizes a high-availability and load-balance Linux cluster system without a load-balancer.

【Key words】 Virtual MAC; Load-balance; High-availability; Linux

目前市场上非常成功的一些负载均衡集群软件，如开源的 Linux Virtual Server(LVS)、商用的 F5 公司的 Big-IP 产品，它们的架构都是前端存在一个负载均衡器来接收用户请求，然后根据调度算法，选择后端的一台真实服务器进行用户请求的处理。这也就意味着用户如果需要负载均衡集群，那么他们就需要额外购买一个单独的负载均衡器来完成该功能，但是这对于一般的中小企业来说，无疑是增加了成本。此外，在政府和企业信息化过程中还需要同时具备负载均衡和高可用功能的集群系统，而传统的高可用集群系统仅实现 failover 功能，没有负载均衡功能，从而并不能完全满足当前需求。所以市场迫切需要一个低成本的融合高可用与负载均衡功能的集群系统。

针对于这种实际的市场需求，本文提出了无负载均衡器的 Linux 高可用负载均衡集群系统：系统中两个服务器节点同时工作，运行同一服务，并各自承担负载的一部分，同时两节点之间相互备份，当检测到其中一个节点失效后，另外一个节点将接管它的工作。本系统可以无缝地扩展到支持多个服务器节点。

1 无负载均衡器的高可用负载均衡系统架构

图 1 中两个对外提供服务的服务器(A 和 B)通过交换机相连接，每个服务器至少拥有 3 块网卡，其中一块网卡用于对外提供服务，两台服务器在该网络接口上拥有相同的 IP 地址和 mac 地址，一块网卡用于心跳连接(可以为心跳连接配置 2 块网卡，或者一块网卡，一个串口线，从而构建冗余的心跳路径)，另外一块网卡用于连接数据库(也可以是网络文件系统，或者是分布式文件系统)。

当用户请求从外部网络到达时，交换机将数据包同时转发给服务器 A 和 B(需要给 Linux 内核打补丁)。A 和 B 收到了同样的数据包后，由于对于任何用户请求，仅需一台服务器进行响应，因此需要一个过滤模块进行处理，不属于某个节点处理的数据包，该节点应该将该数据包扔掉，当然，属于自己应该处理的数据包节点应该允许通过，这样，就达到了负载均衡。为了达到高可用，还需要在两个节点上运行高可用软件，它们之间相互监测对方的状态，一旦发现对方失效，那么就应该接管对方目前正在进行的工作。

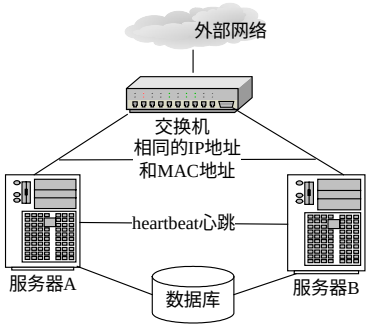

图 1 系统拓扑结构

2 系统设计

从上述系统架构可以看出，该系统需要 3 部分功能：使交换机按照广播方式发送数据包的模块，数据包过滤模块(负载均衡模块)，高可用模块。

2.1 改造交换机工作方式的模块

通常情况下，交换机会按照 mac-port 的映射关系进行数据包的转发，而集线器会以广播的方式进行转发(如果用集线器来连接服务器 A 和 B 就不需要该模块，但是目前集线器的使用已经越来越少了)。同时交换机在目的 mac 地址没有和

作者简介：谢作贵(1980－)，男，硕士生，主研方向：计算机网络，集群和存储；戚骁亚，博士生

收稿日期：2006-03-24 **E-mail：**xie.zuogui@gd-linux.com

port 建立起映射关系之前以广播的方式发送。

现在我们希望服务器 A 和 B 能够同时收到用户请求。首先 A 和 B 两台服务器需要有相同的 IP 地址和 mac 地址，即使这样，如果交换机已经建立起该 mac 地址和 port 的映射关系，也不会把数据包同时转发到这两台服务器。

交换机通过自动学习机制来建立 mac-port 的映射关系，也就是它将端口号与从该端口发出的数据包的源 mac 地址对应起来。为了不让交换机建立起 vmac(即两台服务器对外提供服务的网卡上设置的相同 mac 地址)和 port 的映射关系，在发送数据包时采用另外一个 mac 地址，譬如说 outgoing_mac(不等于 vmac)来封包，这样，交换机在自动学习时，将建立的是 outgoing_mac 和 port 的映射关系，而不是目前实际的 vmac 和 port 的映射关系。

当数据包从外网到达时，无论是到达 A 或 B，其数据包的目的 mac 地址将为 vmac(因为在进行 arp 应答时，还是采用 vmac 应答，只是在数据链路层进行封包时采用 outgoing_mac)，那么由于交换机没有建立 vmac 的映射关系，因此它将该数据包以广播的方式同时发送给服务器 A 和 B。

综上所述，通过改变操作系统中 TCP/IP 协议栈的数据链路层，在不需要特殊的交换机的前提条件下，就可以实现服务器 A 和 B 同时接收到相同的数据包。

2.2 过滤策略

服务器上通过运行一个包过滤策略来实现负载均衡。目前服务器都收到了数据包，它们需要采取一定的包过滤策略来决定数据包是继续处理还是丢弃。过滤策略的设计原则应该是简单和高效(基于 Linux 内核中提供的 netfilter 框架以内核模块的形式实现从而保持其高效性)。目前采用数据包的源 IP 地址的低 3 位来进行过滤。注意为了确保服务的连续性，同一连接(或会话)上的数据包必须发送到同一台服务器上进行处理。具体的过滤策略定义如下：

$$f(ip)=MOD(ip_{low},n)$$

其中，IP代表数据包的源IP地址，而IP_{low}代表IP地址的低 3 位，这也就意味着该集群系统可扩展到 8 台服务器(理论上可以扩展任意数目的服务器来提供任意规模的用户服务，但是当用户服务流量非常大时，数据包的广播对网络带宽的消耗太大，该架构的实际性能会随着规模的扩大而降低)，n代表当前集群系统中服务器的个数。通过对源IP地址的低 3 位用n求模，每一个IP地址将映射成为 0~n-1 之间的某个值。各服务器只接收特定的f(IP)值而丢弃其它，这样，来自外网的数据包将根据它们的源IP地址被分配到各服务器上进行处理，从而，在每个服务器上处理的负载平均为总负载的 1/n。

2.3 高可用模块

因为为了防止一个服务器节点宕机之后，属于该服务器的用户请求得不到处理，所以在每个服务器节点上运行高可用软件。每个节点都监控其它节点的状态，一旦发现某个节点出现故障(如服务出现故障、对外服务链路出现故障)，那么将选择集群中处于活动状态的另外一个节点接管失效服务器的工作。譬如说，在接管之前，该节点只接收 f(IP)=0 的数据包，而失效服务器接收 f(IP)=1 的数据包，那么在接管之后，该节点将接收 f(IP)=0 和 1 的数据包，从而开始处理失效服务器以前处理的工作。

通过这种方式，即使集群中有节点出现故障，一旦高可用软件检测到该情况，它将迅速地(几秒之内)把该工作迁移到其它节点上，从而保证整个集群对外提供服务的高可用。

3 结论

综上所述，本文针对于中小企业中希望使用带负载均衡功能的高可用集群系统，但是不想增加负载均衡器所带来成本的这种情况，设计并实现了无负载均衡器的 Linux 高可用负载均衡集群系统。对于用交换机连接的服务器集群，改造了服务器的协议栈，使得交换机能够以广播的方式发送数据包；通过一个包过滤策略，每个节点处理总负载的 1/n(n 为服务器的总数)；并且节点之间相互备份。

我们利用 5 台机器对系统进行测试，2 台作为测试机(源 IP 地址设为一奇一偶)，2 台做为服务器对外提供服务，1 台做为数据库(为服务器提供一致的数据)。试验表明，正常状态下所有服务器都能收到测试机发过来的用户请求，但是只处理属于自己的工作；在一台服务器宕机之后，另外一台服务器迅速接管其工作，开始处理所有的工作，用户现在的请求都能够在处于活动状态的节点上得到应答；当宕掉的机器恢复正常之后，两台服务器又恢复到原先的状态：即各自处理属于自己的一部分工作。

本文方案设计及其实现对于不需要负载均衡器的高可用负载均衡集群系统的设计具有较好的参考和实用价值。

参考文献

1 Robertson A.The Evolution of the Linux-HA Project[Z]. 2004-02. http://www.linux-ha.org/TechnicalPapers.

2 Russell R, Welte H. Linux Netfilter Hacking HOWTO[Z]. 2002-07. http://www.netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO.html.

3 杨晓宁, 伍卫国, 刘爱华, 等. 多负载均衡器集群系统中负载均衡器故障恢复机制[J]. 计算机工程, 2004, 30(23).

（上接第 132 页）

空闲的 DEV，然后产生一个 UDP 套接字，将套接字与 DEV 建立映射关系。完成了模块与 DEV 的绑定后，启动一个监视任务(Moniter_Task)监视 UDP 端口。到此，通信模型建立完毕。最后启动模块的其他工作任务开始正常的业务流程。

5 结束语

本文提出了一种不同于传统通信机制的模块间通信模型。将用于网络通信的 UDP 方式引入到程序内的模块间通信中，避免了传统方式可能引起的降低效率与资源耗费过多问题。通过送耦合的连接方式增强了程序的灵活性。在实验中，这种基于 UDP 方式的虚拟设备绑定的通信模型取得了较好的效果，在对实时性要求较高的嵌入式系统开发中有较高的价值。

参考文献

1 陈智育, 温严军, 陈 琪, 等. VxWorks 程序开发实践[M]. 北京: 人民邮电出版社, 2004-05: 107-109.

2 孔祥营. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado[M]. 北京: 中国电力出版社, 2001-11: 23-30.

3 任 行, 任方洁, 王婉南. 基于 VxWorks 嵌入式操作系统的 C/S 模式网络编程[J]. 现代电子技术, 2003, 26(19).

4 郑泽胜. 嵌入式系统以及实时软件开发[Z]. 2003-05-15. http://www.pocketix.com.cn.