**LVS Study Notes**

@author Count.MCristo

@create D20150803

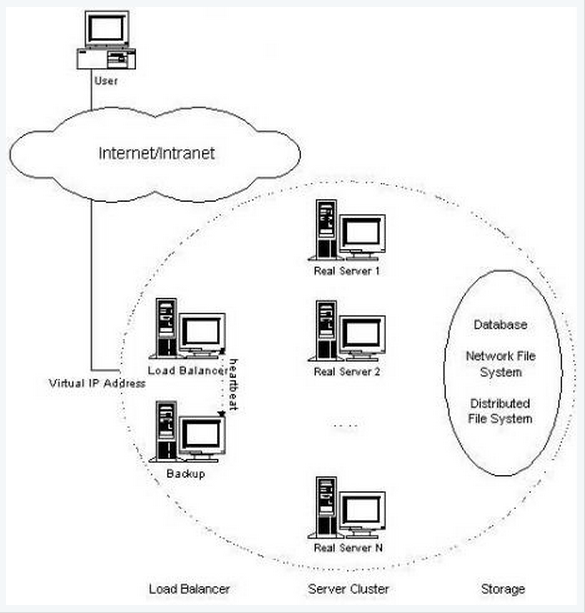
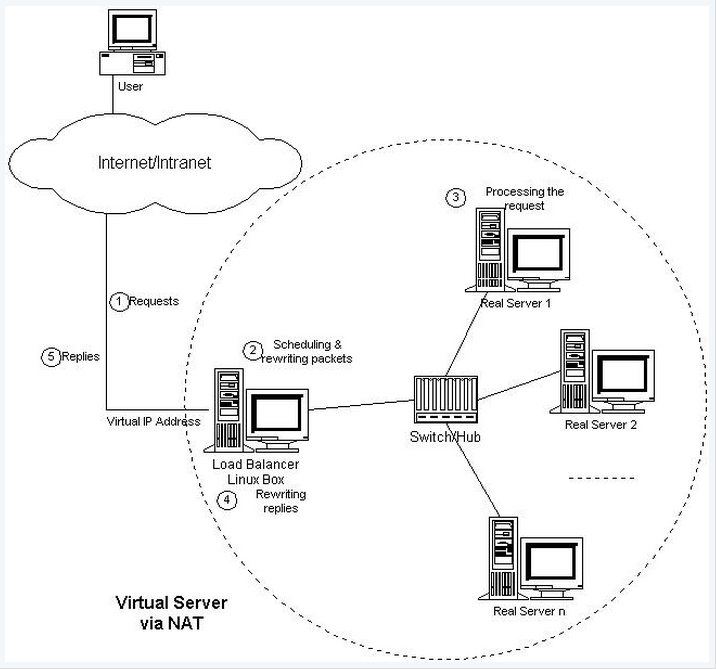
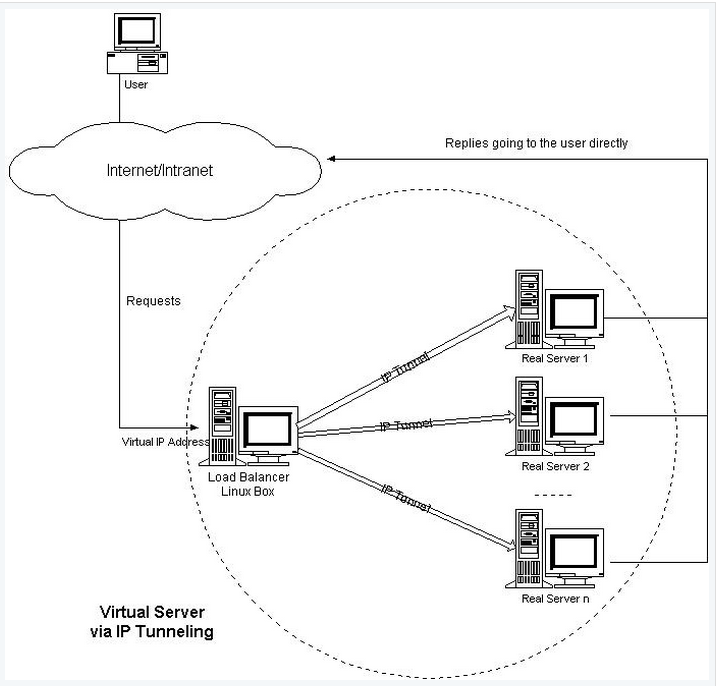
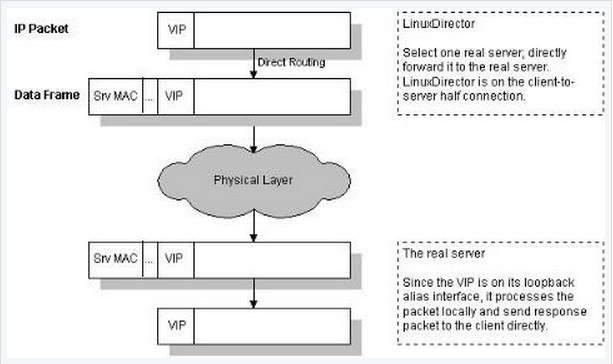
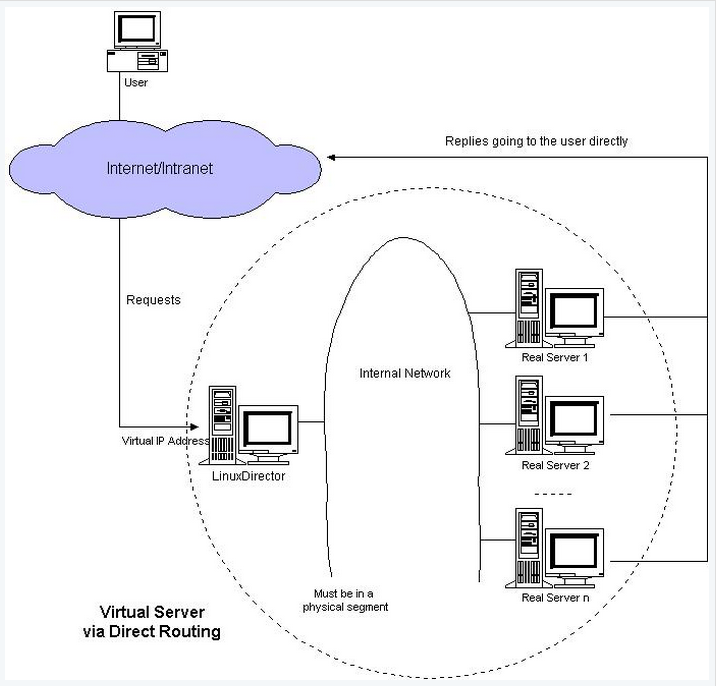
@version D20150824

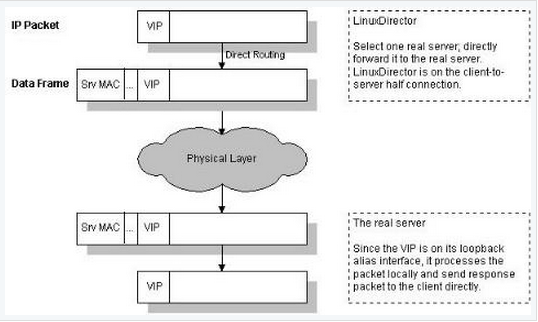
@description 记录LVS重要的知识

@motto VICTORY LOVES PREPARATOIN

@criterion 凯旋基诺

@copyright Insigma HengTian Software Ltd.

1. **LVS介绍** 
   1. @reference
      1. LVS @百度百科
   2. **含义：**
      1. LVS是Linux Virtual Server的简写，意即Linux虚拟服务器，是一个**虚拟服务器**(@001)集群系统。本项目在1998年5月由章文嵩博士成立，是中国国内最早出现的自由软件项目之一。
   3. **宗旨：**
      1. 使用集群技术和Linux操作系统实现一个高性能、高可用的服务器。
      2. 很好的可伸缩性（Scalability）
      3. 很好的可靠性（Reliability）
      4. 很好的可管理性（Manageability）
   4. **特点：**
      1. **可伸缩网络服务**(@003)的几种结构，它们都需要一个前端的负载调度器（或者多个，进行主从备份）。我们先分析实现**虚拟网络**(@004)服务的主要技术，指出IP负载均衡技术是在负载调度器的实现技术中效率最高的。在已有的IP负载均衡技术中，主要有通过网络地址转换（Network Address Translation）将一组服务器构成一个高性能的、高可用的虚拟服务器，我们称之为VS/NAT技术（Virtual Server via Network Address Translation）。在分析VS/NAT的缺点和网络服务的非对称性的基础上，我们提出了通过**IP隧道**(@005)实现虚拟服务器的方法VS/TUN （Virtual Server via IP Tunneling）(@notice 开启IP隧道方式需要重新编译内核)，和通过直接路由实现虚拟服务器的方法VS/DR（Virtual Server via Direct Routing），它们可以极大地提高系统的伸缩性。VS/NAT、VS/TUN和VS/DR技术是LVS集群中实现的三种IP负载均衡技术。
   5. **技术：**
      1. **技术简介**
         1. LVS集群采用IP负载均衡技术和基于内容请求分发技术(@006)。调度器具有很好的吞吐率，将请求均衡地转移到不同的服务器上执行，且调度器自动屏蔽掉服务器的故障，从而将一组服务器构成一个高性能的、高可用的虚拟服务器。整个服务器集群的结构对客户是透明的，而且无需修改客户端和服务器端的程序。为此，在设计时需要考虑系统的透明性、可伸缩性、高可用性和易管理性。
      2. **集群采用的三层结构（一般是三层结构）（有待进一步系统归纳 @version 20150805）**
         1. **负载调度器（Load Balancer）**
            1. 它是整个集群对外面的前端机，负责将客户的请求发送到一组服务器上执行，而客户认为服务是来自一个IP地址（我们可称之为虚拟IP地址）上的。
         2. **服务器池（Server Pool）**
            1. 是一组真正执行客户请求的服务器，执行的服务有WEB、MAIL、FTP和DNS等。
         3. **共享存储（Shared Storage）**
            1. 它为服务器池提供一个共享的存储区，这样很容易使得服务器池拥有相同的内容，提供相同的服务。
      3. **调度器**
         1. 调度器是服务器集群系统的唯一入口点（Single Entry Point），它可以采用IP负载均衡技术、基于内容请求分发技术或者两者相结合。
         2. 在IP负载均衡技术中，需要服务器池拥有相同的内容提供相同的服务。当客户请求到达时，调度器只根据服务器负载情况和设定的调度算法从服务器池中选出一个服务器，将该请求转发到选出的服务器，并记录这个调度；当这个请求的其他报文到达，也会被转发到前面选出的服务器。在基于内容请求分发技术中，服务器可以提供不同的服务，当客户请求到达时，调度器可根据请求的内容选择服务器执行请求。因为所有的操作都是在Linux操作系统核心空间中完成的，它的调度开销很小，所以它具有很高的吞吐率。服务器池的结点数目是可变的。当整个系统收到的负载超过目前所有结点的处理能力时，可以在服务器池中增加服务器来满足不断增长的请求负载。
         3. 对大多数网络服务来说，请求间不存在很强的相关性，请求可以在不同的结点上并行执行（@analysis 相关性弱的话就说明了不需要在原有的性能基础上进一步提高性能处理问题，而是可以从性能零点开始性能的叠加，所以性能与服务器池的节点数目呈正相关），所以整个系统的性能基本上可以随着服务器池的结点数目增加而线性增长。 共享存储通常是数据库、网络文件系统或者分布式文件系统。服务器结点需要动态更新的数据一般存储在数据库系统中，同时数据库会保证并发访问时数据的一致性。静态的数据可以存储在网络文件系统（如NFS/CIFS）中，但网络文件系统的伸缩能力有限，一般来说，NFS/CIFS服务器只能支持3~6个繁忙的服务器结点。对于规模较大的集群系统，可以考虑用分布式文件系统，如AFS、GFS、Coda和Intermezzo等。分布式文件系统可为各服务器提供共享的存储区，它们访问分布式文件系统就像访问本地文件系统一样，同时分布式文件系统可提供良好的伸缩性和可用性。
      4. **分布式锁管理器**
         1. 不同服务器上的应用程序同时读写访问分布式文件系统上同一资源时，应用程序的访问冲突需要消解才能使得资源处于一致状态。这需要一个分布式锁管理器（Distributed Lock Manager），它可能是分布式文件系统内部提供的，也可能是外部的。开发者在写应用程序时，可以使用分布式锁管理器来保证应用程序在不同结点上并发访问的一致性。
         2. 负载调度器、服务器池和共享存储系统通过高速网络相连接，如100Mbps交换网络、Myrinet和Gigabit网络等。使用高速的网络，主要为避免当系统规模扩大时互联网络成为整个系统的瓶颈。
      5. **监视器**
         1. Graphic Monitor是为系统管理员提供整个集群系统的监视器，它可以监视系统的状态。Graphic Monitor是基于浏览器的，所以无论管理员在本地还是异地都可以监测系统的状况。为了安全的原因，浏览器要通过HTTPS（Secure HTTP）协议和身份认证后，才能进行系统监测，并进行系统的配置和管理。
   6. **优缺点：**
      1. **优点**
         1. 开源，免费
         2. 在网上能找到一些相关技术资源
         3. 具有软件负载均衡的一些优点
      2. **缺点**
         1. 最核心的就是没有可靠的支持服务，没有人对其结果负责；
         2. 功能比较简单，支持复杂应用的负载均衡能力较差，如算法较少等；
         3. 开启隧道方式需重编译内核；
         4. 配置复杂；
         5. 主要应用于LINUX，目前没有专门用于WINDOWS的版本，不过可以通过配置，使Windows成为LVS集群中的Real Server（Win2003、Win2008中）。
   7. LVS负载均衡集群基本原理和功能拓扑
2. **LVS配置知识**
   1. @reference <http://os.51cto.com/art/201202/319979.htm>
   2. **术语**
      1. Director：前端运行LVS的负载均衡服务器，目前只能是Linux，可以针对Web、FTP、Cache、MMS、MySQL做负载均衡。
      2. RealServer：后端需要负载均衡的服务器，可以是各类操作系统，Linux､Solaris､Aix､BSD､Windows都可，甚至Director本身也可以作为RealServer使用｡
      3. LVS：Linux Virtual Server，Linux下的负载均衡器，不多作介绍了，跟DNS轮询以及一些商业产品的比较等不做论述，具体看看LVS网站，支持LVS-NAT､ LVS-DR（@description此方式适合所有的RealServer同一网段下，即接在同一个交换机上）､LVS-TUNL（@description 此方式RealServer可以任意了，完全可以跨地域､空间，只要系统支持Tunnel就OK，所以此种方式更利于扩展，当然此种扩展是指物理上的扩展）三种不同的方式，NAT不常使用主要使用DR､TUNL方式（其实TUNL方式也不常使用，此种方式需要编译Linux内核）｡
   3. **三种转发机制优缺点**
      1. @reference <http://my.oschina.net/leeypp1/blog/294807>
      2. **LVS-NAT(Virtual Server via NAT)**
         1. VS/NAT 的优点是服务器可以运行任何支持TCP/IP的操作系统，它只需要一个IP地址配置在LVS主机上，服务器组可以用私有的IP地址。缺点是它的扩充能力有限，当服务器结点数目升到20时，LVS主机本身有可能成为系统的新瓶颈，因为在VS/NAT中请求和响应封包都需要通过负载平衡LVS主机。在 Pentium 166主机上测得重写封包的平均延时为60us，假设TCP封包的平均长度为536 Bytes，则LVS主机的最大吞吐量为8.93 MB/s。再假设每台服务器的吞吐量为600KBytes/s，这样一个LVS主机可以带动16台服务器。
         2. 拓扑结构
      3. **LVS-TUNL(Virtual Server via IP Tunneling)**
         1. 在VS/TUN 的集群系统中，负载平衡LVS主机只将请求分配到不同的实际服务器，实际服务器将应答的资料直接返回给用户。这样，负载平衡LVS主机就可以处理巨量的请求，而不会成为系统的瓶颈。即使负载平衡LVS主机只有100Mbps的全双工网卡，虚拟服务器的最大吞吐量可以达到几Gbps。所以，VS/TUN可以极大地增加负载平衡LVS主机分配的服务器数量，它可以用来构建高性能超级服务器。VS/TUN技术对服务器的要求是所有的服务器必须支持"IP Tunneling"或者"IP Encapsulation"协议。目前，VS/TUN 的后端服务器主要运行Linux操作系统。因为"IP Tunneling"正成为各个操作系统的标准协议，所以VS/TUN也会适用运行其它操作系统的后端服务器。
         2. 拓扑结构
         3. 工作流（有问题 @version D20150805）
      4. **(LVS-DR) Virtual Server via Direct Routing**
         1. 同VS/TUN 一样，VS/DRLVS主机只处理客户到服务器端的连接，响应资料可以直接从独立的网络路由返回给客户。这可以极大地提高LVS集群系统的伸缩性。同 VS/TUN相比，这种方法没有IP隧道的开销，但是要求负载平衡LVS主机与实际服务器都有一块网卡连在同一物理网段上，服务器网络设备或者设备别名不作ARP响应。
         2. 拓扑结构
         3. 工作流



* + 1. **归纳**

1. **LVS + Keepalived（DR模式）高可用负载均衡集群配置实例**
   1. @reference
      1. @鸡啄米 <http://www.jizhuomi.com/software/351.html>
      2. @CSDN <http://blog.csdn.net/m582445672/article/details/7670015>
   2. **配置详情**
      1. **环境简介**
         * 1. 虚拟机软件

@resource VMware Workstation 11.1.2

* + - * 1. 操作系统

@resource CentOS-6.4-x86\_64

* + - * 1. Keepalived

@resource Keepalived-1.2.9

* + - * 1. IPVS管理工具

@resource IPVSadm-1.26

* + - * 1. Webbench

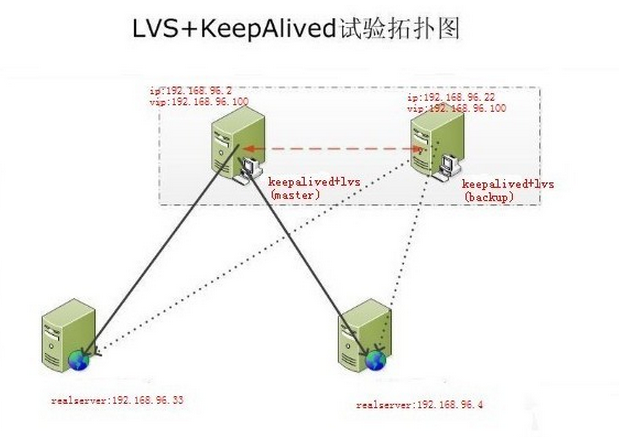
@resource webbench-1.5

@reference <http://www.cnblogs.com/kgdxpr/archive/2013/05/07/3064134.html>

* + - * 1. iKende的TCP、UDP压力测试工具

TcpUdpTest

@reference <http://my.oschina.net/ikende/blog/204874>

* + 1. **拓扑结构**
       - 1. **拓扑图**
         2. **说明**

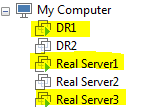
Master是主DR（负载均衡服务器），Backup是辅DR（目前该版本没有配置辅DR）

各个服务器目前采用的是动态获取IP的方式

VIP代表DR对其他服务器（主要指RealServer）提供的IP

IP是用来进行DR们之间的通信所用

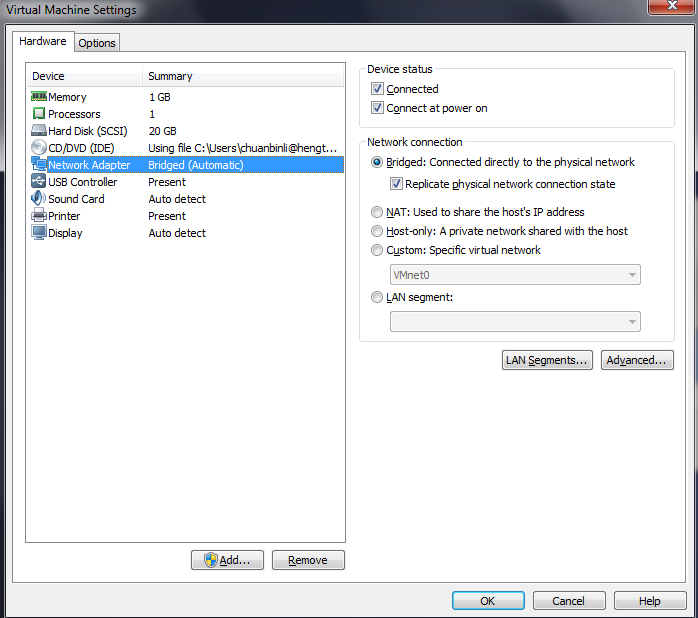
* + 1. **系统基本的安装和配置**
       - 1. **安装好系统**

图示

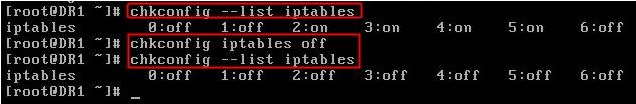
只需要安装DR1（负载均衡服务器）、Real Server1（真实服务器，以下简称RS1）、Real Server3（真实服务器，以下简称RS3）

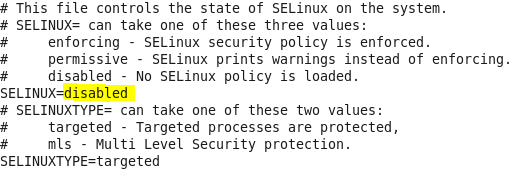
@description 因为虚拟机的安装不是重点且较为简单，在此就不做赘述

@tips 可以安装好一台CentOS，然后进行虚拟机的clone操作，生成其他的服务器

@notice 虚拟机的网络连接方式请选择桥接方式

* + - * 1. **禁用服务器（包括DR1、RS1、RS3）的防火墙和SELinux**

禁用防火墙

禁用SELinux

重启服务器

shutdown –r now

* + - * 1. **测试服务器之间连通性**

使用ping命令测试连通性

测试对象

DR1 <-> RS1

DR1 <–> RS3

* + 1. **安装并配置LVS+Keepalived环境**
       - 1. 安装IPVS模块

介绍

LVS已经是Linux标准内核的一部分,直接被编译在内核中

安装位置

DR1、RS1、RS3

具体查看方法

方法一

方法二

可以看到Linux内核已经集成IPVS，所以此时不需要单独安装该模块

* + - * 1. **安装IPVSadm**

介绍

IPVS管理工具

安装位置

DR1

方法（yum安装）

yum –y install ipvsadm

检查是否安装成功

ipvsadm – help

如果显示了ipvsadm的各种用法，则说明该软件安装成功，否则失败

* + - * 1. **安装Keepalived**

介绍

用于IPVS的健康检查，防止单点故障等

安装位置

DR1

方法

下载Keepalived到/usr/local/src目录下

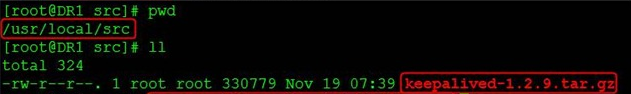
切换到root用户

su

输入密码：123456

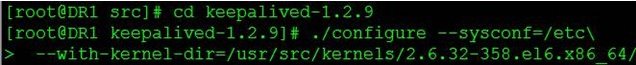
cd /usr/local/src

wget <http://www.keepalived.org/software/keepalived-1.2.9.tar.gz>

图示

安装Keepalived

解压（tar -zxvf）

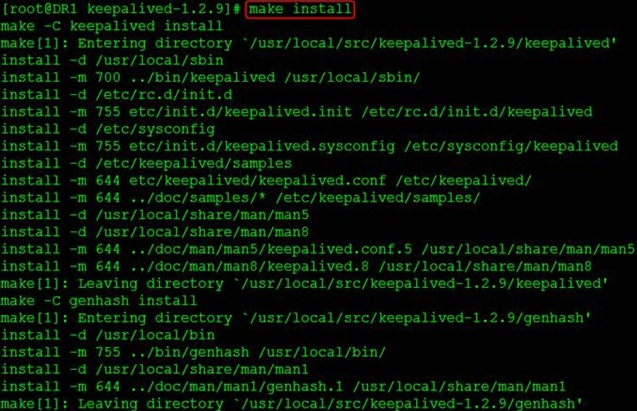
编译(./configure)

注意：确保下面3行是YES



安装(make&&make install)

make

make install

* + - * 1. **安装apache服务和eclipse**

**安装位置**

RS1、RS2

安装过程比较通用且不是重点，在此就不做说明

安装好之后，请开放80端口作为HTTP默认端口

eclipse导入TCP\_UDP\_Communication\_Integration\_MessageQueue和HelloWrold项目

* + - * 1. **配置文件路径（软链接）**

cp /home/keepalived/etc/rc.d/init.d/keepalived /etc/init.d/

cp /home/keepalived/etc/sysconfig/keepalived /etc/sysconfig/

cp /home/keepalived/etc/keepalived/keepalived.conf /etc/keepalived/

cp /home/keepalived/sbin/keepalived /sbin/

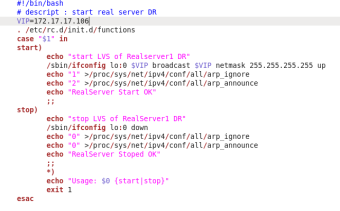
* + - * 1. **配置Keepalived（含HTTP、TCP和UDP通信服务）**
        2. **配置RS1和RS2**

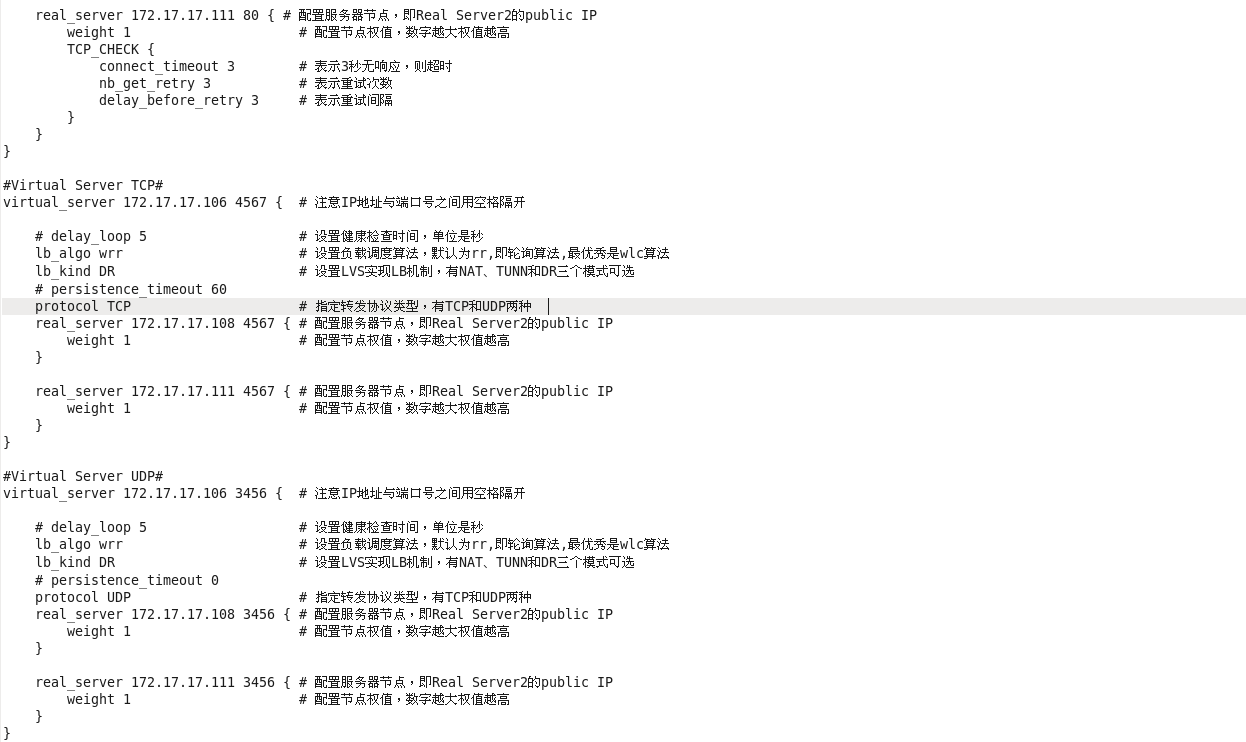
cd /etc/init.d/

配置脚本文件lvsMode.sh

为脚本文件添加权限

chmod 777 lvsMode.sh

****

* + 1. **测试**
       - 1. **负载均衡测试**

**HTTP**

RS1和RS2上启动HelloWrold

启动lvsMode.sh

cd etc/init.d/

./lvsMode.sh

启动DR1的Keepalived

/etc/init.d/keepalived start

使用正在使用的Window系统进行

访问<http://RS1> 的IP/HelloWrold，出现访问成功字样

访问<http://RS2> 的IP/HelloWrold，出现访问成功字样

多次访问<http://DR1> 的IP/HelloWrold，出现访问成功字样

在DR1上通过IPVSadm查看访问情况（ipvsadm –Ln）(工具具体使用请参考IPVSadm使用简介)

注意：目前服务器的地址还是动态获取，所以可能造成两次开机IP不同，可以使用ifconfig查看服务器IP，进行动态使用

TCP&&UDP

使用上述的TCP、UDP测试工具

输入DR1的IP和分别输入TCP、UDP端口（见上面的配置文件）

在DR1上通过IPVSadm查看访问详情（ipvsadm –Ln && ipvs –list stats）

* + - * 1. **压力测试**

@description 测试之前请先参考Linux环境下基于Socket的HTTP、TCP、UDP的通信设计与开发

HTTP

安装Webbench

@reference <http://www.linuxidc.com/Linux/2013-11/92386.htm>

测试实例

webbench –c 1000 –t 10 DR1的IP

参数解释

–c：每秒并发的数量

–t：并发的时间（单位：秒）

IP：需要进行压力测试的主机IP

TCP&&UDP

使用上述的TCP、UDP测试工具

输入DR1的IP和分别输入TCP、UDP端口（见上面的配置文件）

在DR1上通过IPVSadm查看访问详情（ipvsadm –Ln && ipvs –list stats）

1. **IPVSadm使用简介**
   1. ipvsadm –Ln
      1. 作用
         * 1. 查看负载关系
      2. 参数
         * 1. forward

转发方式

* + - * 1. weight

权值

* + - * 1. ActiveConn

当前活动的连接数目

* + - * 1. InActConn

当前非活动连接数目

* 1. ipvsadm –list –stats
     1. 作用
        + 1. 查看连接详情
     2. 参数
        + 1. Conns

当前连接数

* + - * 1. InPkts

输入的数据包

* + - * 1. OutPkts

输出的数据包

* + - * 1. InBytes

输入的字节流量

* + - * 1. OutBytes

输出的自己流量

1. **虚拟服务器**
   1. 虚拟服务器也称为VPS（Virtual Private Servers）主机，相对于真实主机而言，采用特殊的软硬件技术把一台完整的服务器主机分成若干个主机。实际上是将真实的硬盘空间分成若干份，然后租给不同用户，每一台被分割的主机都具有独立的域名和IP地址，但共享真实主机的CPU、RAM、操作系统、应用软件等。运行时由用户远程操作属于自己的那一块，而这一块对任何用户而言，就是一台“完整”的服务器，与真实独立的主机功能完全一样，网民一般在空间域名网络申请虚拟服务器。用户只需对自己的信息进行远程维护，而无需对硬件、操作系统及通信线路进行维护。 @resource 百度百科
2. [**负载均衡**](http://baike.baidu.com/view/51184.htm)**集群**
   1. 负载均衡集群为企业需求提供了更实用的系统。负载均衡集群使负载可以在计算机集群中尽可能平均地分摊处理。负载通常包括应用程序处理负载和网络流量负载。这样的系统非常适合向使用同一组应用程序的大量用户提供服务。每个节点都可以承担一定的处理负载，并且可以实现处理负载在节点之间的动态分配，以实现负载均衡。对于网络流量负载,当网络服务程序接收了高入网流量，以致无法迅速处理的时候，网络流量就会发送给在其它节点上运行的网络服务程序。同时，还可以根据每个节点上不同的可用资源或网络的特殊环境来进行优化。与科学计算集群一样，负载均衡集群也在多节点之间分发计算处理负载。它们之间的最大区别在于缺少跨节点运行的单并行程序。大多数情况下，负载均衡集群中的每个节点都是运行单独软件的独立系统。但是，不管是在节点之间进行直接通信，还是通过中央负载均衡服务器来控制每个节点的负载,在节点之间都有一种公共关系。通常，使用特定的算法来分发该负载。@resource 百度百科
3. **可伸缩性网络服务**
   1. @reference LVS中文站点 <http://zh.linuxvirtualserver.org/node/9>
   2. 可伸缩性（Scalability）是在当今计算机技术中经常用到的词汇。对于不同的人，可伸缩性有不同的含义。现在，我们来定义可伸缩网络服务的含义。
   3. 可伸缩网络服务是指网络服务能随着用户数目的增长而扩展其性能，如在系统中增加服务器、内存或硬盘等；整个系统很容易被扩展，无需重新设置整个系统，无需中断服务。换句话说，系统管理员扩展系统的操作对最终用户是透明的，他们不会知道系统的改变。
   4. 可伸缩系统通常是高可用的系统。在部分硬件（如硬盘、服务器、子网络）和部分软件（如操作系统、服务进程）的失效情况下，系统可以继续提供服务，最终用户不会感知到整个服务的中断，除了正在失效点上处理请求的部分用户可能会收到服务处理失败，需要重新提交请求。Caching和复制是建立高可用系统的常用技术，建立多个副本会导致如何将原件的修改传播到多个副本上的问题。
   5. 实现可伸缩网络服务的方法一般是通过一对多的映射机制，将服务请求流分而治之（Divide and Conquer）到多个结点上处理。一对多的映射可以在很多层次上存在，如主机名上的DNS系统、网络层的TCP/IP、文件系统等。虚拟（Virtual）是描述一对多映射机制的词汇，将多个实体组成一个逻辑上的、虚拟的整体。例如，虚存（Virtual Memory）是现代操作系统中最典型的一对多映射机制，虚存建立一个虚拟内存空间，将它映射到多个物理内存上。
4. **虚拟网络**
   1. 虚拟网络是一种包含部分是虚拟网络链接的计算机网络。虚拟网络链接是在两个计算设备间不包含物理连接，而是通过**网络虚拟化**来实现。
   2. 两种最常见的虚拟网络形式为基于协议的虚拟网络（如VLAN、VPN和VPLS等）和基于虚拟设备（如在hypervisor内部的网络连接虚拟机）的虚拟网络。
   3. **网络虚拟化**的内容一般指虚拟专用网络 (VPN)。VPN 对网络连接的概念进行了抽象，允许远程用户访问组织的内部网络，就像物理上连接到该网络一样。网络虚拟化可以帮助保护 IT 环境，防止来自 Internet 的威胁，同时使用户能够快速安全的访问应用程序和数据。
   4. “Virtual Private Network”。VPN被定义为通过一个公用网络(通常是因特网)建立一个临时的、安全的连接，是一条穿过混乱的公用网络的安全、稳定隧道。使用这条隧道可以对数据进行几倍加密达到安全使用互联网的目的。
5. **IP隧道技术**
   1. @reference IP隧道技术@百度百科
   2. 是路由器把一种网络层协议封装到另一个协议中以跨过网络传送到另一个路由器的处理过程。
   3. 隧道技术是一种数据包封装技术，它是将原始IP包（其报头包含原始发送者和最终目的地）封装在另一个数据包（称为封装的IP包）的数据净荷中进行传输。
   4. 所谓隧道，实际上是路由器把一种网络层协议封装到另一个协议中以跨过网络传送到另一个路由器的处理过程。发送路由器将被传送的协议包进行封装，经过网络传送，接受路由器解开收到的包，取出原始协议；而在传输过程中的中间路由器并不在意封装的协议是什么。这里的封装协议，称之为传输协议，是跨过网络传输被封装协议的一种协议，IP协议是IOS唯一选择的传输协议。而被封装的协议在此为IPX协议或者AppleTAlk协议，通常可以称之为乘客协议。需要特别注意的是：隧道技术是一种点对点的链接，因而必须在链接的两端配置隧道协议。
6. **内容分发**
   1. @reference 内容分发@百度百科
   2. 内容分发 (Content Delivery)，它通过实现用户对网站的就近访问及网络流量的智能分析，将本节点流媒体资源库中的指定内容，根据业务运营商定义的内容分发策略向下层节点推送（PUSH）。下层节点控制系统通知下层内容管理系统登记接收，该节点以内容注入的方式接收分发的内容。从技术上解决网络带宽小、用户访问量大、网点分布不均等对用户访问效果的影响，大大提高了网络的响应速度。