

# 使用 LVS 和 Squid 构建代理服务器集群系统<sup>①</sup>

薛建, 王信堂, 曲守宁

(济南大学 信息网络中心, 济南 250014)

**摘要:** LVS 是优秀的集群及 IP 负载均衡开源软件, 同样 Squid 是目前最通用的开源代理服务系统软件. 本文讲述如何利用 LVS 和 Squid 构建高效可靠的代理服务器集群系统.

**关键词:** 负载均衡; Proxy; LVS 软件; Squid 软件

## Building Cluster of Proxy Based on LVS and Squid

XUE Jian, WANG Xin-Tang, QU Shou-Ning

(Information Network Center, University of Jinan, Jinan 250014, China)

**Abstract:** LVS is an excellent open source software for cluster and IP Load balancing. Squid is also the most popular system for proxy service. This paper discusses how to set up efficient and reliable system based on LVS and Squid.

**Key words:** load balance; proxy; LVS; Squid

目前园区网 (campus) 出口大都采用防火墙活代理服务器作为互联网的出口设备, 代理服务器由于具有防火墙所不具备的缓存 (cache) 功能, 在许多大型园区网的出口得到了广泛的应用. 代理服务器 (Proxy Server) 是介于客户端和 Web 服务器之间的一台服务器, 使用代理服务器的内网计算机与 Internet 交换信息时都先将信息发送给代理服务器, 由其转发, 并且将收到的应答回送给该计算机.

作者工作单位的校园网是一个规模比较大的园区网, 不仅为校内的办公、教学用计算机提供互联网出口, 还为教工宿舍提供上网服务, 上网终端 5000 台左右 (不包括学生宿舍网的计算机), 对互联网访问的稳定性和速度具有很高的要求. 作者根据上述要求, 经过不同集群软件和代理服务器软件的调研分析, 经过长期测试, 成功配置出基于 LVS 和 Squid 的代理服务器集群系统, 大大降低了互联网出口的平均无故障时间, 网络服务质量得以提升.

## 1 IVS工作原理、架构及工作方式

### 1.1 LVS 简介

LVS 是 Linux Virtual Server 的简写, 意即 Linux 虚

拟服务器, 是一个服务器集群系统软件. LVS 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.

单服务器不能处理不断增长的负载, 并且无法提供针对单点故障的容错功能. 通过高性能网络互联的服务器集群正成为实现高可伸缩的、高可用网络服务的有效结构. 组成集群的 PC 服务器和标准网络设备因为大规模生产, 价格低, 具有很高的性能价格比. 针对上述需求, LVS 给出了基于 IP 层和基于内容请求分发的负载均衡调度解决方法, 并在 Linux 内核中实现了这些方法, 将一组服务器构成一个实现可伸缩的、高可用网络服务的服务器集群, 称之为 Linux 虚拟服务器 (Linux Virtual Server).

在 LVS 集群中, 服务器集群的结构对客户是透明的, 客户访问集群提供的网络服务就像访问一台高性能、高可用的服务器一样, 客户程序不需作任何修改. 系统的伸缩性通过在服务机群中透明地加入和删除一个节点来达到, 通过检测节点或服务进程故障和正确地重置系统达到高可用性.

### 1.2 LVS 工作原理

LVS 的典型结构如图 1 所示:

<sup>①</sup> 收稿时间:2012-03-19;收到修改稿时间:2012-04-17

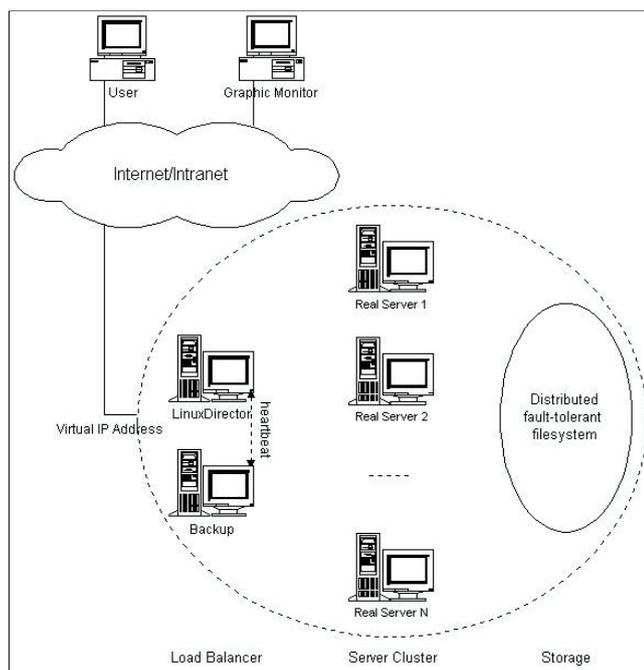


图 1 系统总体框图

LVS 系统通常由调度机(Director)和多台真实服务器 (Real Server) 构成, 调度机通过虚拟 IP (vertual IP Address) 接受客户端的请求后按调度算法将其转发到不同的真实服务器上, 真实服务器响应请求并将结果返回, 根据 LVS 的工作方式不同, 返回结果将直接或通过调度机返回客户端.

### 1.3 LVS 的三种工作方式及本项目的选择

LVS 有三种架构, 分别是 NAT、TUN 和 DR 方式. 这三种工作方式适用于不同的应用环境, 以下我们分析这三种工作方式的原理及优缺点, 并给出我们选择 DR 方式作为代理服务器集群实现机制的原因.

#### 1.3.1 NAT 方式

NAT (Network Address Translation) 方式的转发流程是: Director 服务器收到外界请求, 改写数据包的目标地址, 按相应的调度算法将其发送到相应 Real Server 上, Real Server 处理完该请求后, 将结果数据包返回到其默认网关, 即 Director 机器上, Director 机器再改写数据包的源地址, 最后将其返回给外界. 这样就完成一次负载调度.

因为执行 NAT 需要重写流经 Director 的数据包, 在速度上有一定延迟; 另外, 当用户的请求集中, 而服务器的回应数据量又很大的情况下, 会对 Director 形成很大压力, 成为新的瓶颈, 从而使整个系统的性

能受到限制.

#### 1.3.2 TUN 方式

TUN (IP Tunneling) 的转发流程是: Director 机器收到外界请求, 按相应的调度算法将其通过 IP 隧道发送到相应 Real Server, Real Server 处理完该请求后, 将结果数据包直接返回给客户. 至此完成一次负载调度.

该方式中 Real Server 处理请求后直接将结果回应用户, 这样 Director 就只处理客户机与服务器的一半连接, 极大地提高了 Director 的调度处理能力, 使集群系统能容纳更多的节点数.

另外, TUN 方式中的 Real Server 可以分布在不同的地理区域, 系统可以跨广域网运行, 这样可以构筑跨地域的集群, 其应对灾难的能力也更强. 如果要构建跨地域跨广域网的集群系统, 可以考虑采用 TUN 方式.

TUN 方式中, 服务器需要为 IP 封装付出一定的资源开销, 而且后端的 Real Server 必须是支持 IP Tunneling 的操作系统.

#### 1.3.3 DR 方式

DR (Direct Routing) 的转发流程是: Director 机器收到外界请求, 按相应的调度算法将其直接发送到相应 Real Server, Real Server 处理完该请求后, 将结果数据包直接返回给客户, 完成一次负载调度.

DR 方式与前面的 TUN 方式有些类似, 前台的 Director 机器也是只需要接收和调度外界的请求, 而不需要负责返回这些请求的反馈结果, 所以能够负载更多的 Real Server, 提高 Director 的调度处理能力, 使集群系统容纳更多的 Real Server. 另外, 由于 DR 方式不需要封装和解封装 IP 隧道, 所以效率比 TUN 方式更高. DR 需要改写请求报文的 MAC 地址, 所以所有服务器必须在同一物理网段(VLAN)内.

#### 1.3.4 工作方式比较及选型

三种工作方式总结比较表 1 所示:

表 1 LVS 三种工作方式总结比较

	扩展性	转发效率	对网络环境要求	对真实服务器要求
NAT	低	低	同一局域网	无
TUN	一般	较高	可跨广域网	必须支持 IP Tunneling
DR	高	高	必须在同一 VLAN	无

可以看出, DR 和 TUN 方式的吞吐量和转发效率要大大高于 NAT 方式, TUN 方式对真实服务器有要求, 必须支持 IP Tunneling, 且 TUN 方式调度机和真实服务器可以不再一个网段的优势在局域网部署中没有任何意义. DR 方式性能最优, 且我们构建的代理服务器集群系统中的所有服务器恰好都位于同一物理网段 (VLAN) 所以, 我们选择 DR 工作方式.

## 2 代理服务器集群的网络结构

我们搭建的代理服务器集群如图 2 所示, 包括一台调度服务器和 3 台真实服务器, 采用 LVS 的 DR 工作方式.

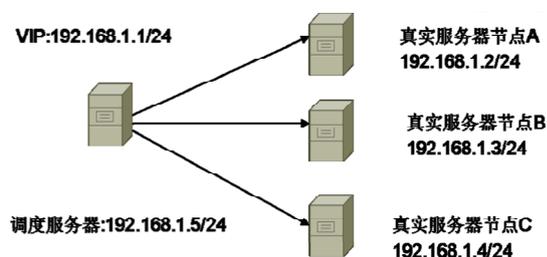


图 2 代理服务器集群系统

设备配置如下:

1 台 Linux 服务器做 Director server (192.168.8.2), 操作系统 CentOS5.

3 台做 realserver (192.168.1.2, 192.168.1.3, 192.168.1.4) 操作系统 Debian6.

Virtual IP: 192.168.1.1

Director: 192.168.1.5

RealServer1: 192.168.1.2

RealServer2: 192.168.1.3

RealServer3: 192.168.1.4

其中 Virtual IP : 192.168.1.1 就是对外提供服务的 IP, 在代理服务器集群中就是代理服务器的地址.

构建代理服务器集群主要工作包括配置调度服务器和配置 Squid 代理服务器. 其中调度服务器配置就是安装 LVS 软件及相关管理工具如 ldirectord 和 ipvsadm 等. 而配置代理服务器主要包括安装和优化 Squid 服务器.

## 3 LVS 调度服务器的安装与配置

LVS 调度服务器的安装主要包括软件安装、网络

参数修改、配置文件修改几个步骤.

LVS 在主要的 Linux 发行版下都可以实现, 下面以 Centos 环境为例, 讲述 LVS 的安装和配置过程:

### 3.1 调度服务器软件安装

```
#yum install ipvsadm //ipvsadm 用于管理 LVS 调度服务器
```

```
#yum install heartbeat //heartbeat 用于双调度机容错
```

```
#yum install heartbeat-lldirectord //lldirectord 是实现 LVS 调度功能的进程
```

### 3.2 系统环境参数调整

Linux 下 IP 转发 (IP Forward) 默认是禁止的, 需要打开打开调度机器的 IP 转发, 方法是:

```
vi /etc/sysctl.conf
```

将这条改成 net.ipv4.ip\_forward=1

执行 sysctl -p 使配置生效.

还需要在 Linux 操作系统下使能 lldirectord, 方法如下:

```
# chkconfig --add lldirectord
```

```
# chkconfig lldirectord --list
```

```
# chkconfig lldirectord on
```

### 3.3 编辑 lldirectord.cf 配置文件

Lldirectord 实际上是实现 LVS 调度功能的一个进程, 其配置文件是 lldirectord.cf, 需要根据自己实际的网络环境进行修改.

将 /usr/share/doc/heartbeat-lldirectord-2.1.3/lldirectord.cf 拷贝到/etc/ha.d/lldirectord.cf 并编辑, 修改完成后如下所示:

```
virtual=192.168.0.1:8000
```

```
real=192.168.0.2:8000 gate
```

```
real=192.168.0.3:8000 gate
```

```
real=192.168.0.4:8000 gate
```

```
scheduler=wlc
```

```
persistent=600
```

```
protocol=tcp
```

```
checktype=connect
```

```
checkport=8000
```

```
.....
```

其中 192.168.0.1 是提供服务的虚拟地址, 客户端浏览器的代理服务器都设置成这个地址; 192.168.0.2-192.168.0.4 是 3 台 Real Server 的地址; 负载均衡算法

为 wlc, LVS 支持多种调度算法, 具体参阅 LVS 官方网站; 采用协议是 TCP, ldirectord 的检测方法是 connect, persistent=600 的含义是将在 600 秒之内来自同一个 PC 的连接请求都发送到一台服务器. 这一条很有意义, 可避免将同一台 PC 在一段时间内的访问请求调度到不同服务器上, 从而会避免这给诸如网银、视频点播许多应用带来的问题等.

## 4 Squid代理服务器配置与优化

目前企业级的代理服务器软件主要有开源的 Squid 和微软的 ISA server 两个产品.

Squid 是 Duane Wessles 等开发并发行在 GPL 版权下, 允许用户自由使用、修改的免费代理服务软件, 具有功能强大、配置简单、文档丰富, 支持 HTTP、FTP、GOPHER、SSL、WAIS 等协议.

Squid 可以在代理服务器上配置缓存, 将已经获取的数据放在硬盘的缓存目录里, 数据的摘要表缓存在内存中, 并且以后客户要获取相同并且未改变的数据时, 就通过内存中的数据摘要表查找定位到硬盘的缓存目录中的缓存对象, 直接从缓存目录里获取数据. 加快了内部网浏览互联网网站的速度, 提高客户机的访问命中率.

Squid 的安装大概可分为源码安装和二进制安装两种方式. 目前在大多数 Linux 的发行版本中都提供了二进制的安装方式, 就是直接安装编译完成的 Squid 程序, 比如在 debian 下安装命令是:

```
apt-get install squid
```

在 Radhat、CentOS 下是:

```
yum install squid.
```

但是由于二进制的程序由于在编译时的硬件环境和您的服务器的硬件环境的不同, 并且二进制安装方式使得好多 Squid 的配置调优无法进行, 所以不推荐这种安装方式.

源码的安装方式就是从 Squid 的官方网站下载源程序然后编译安装的方法, 下面以 Debian 环境为例, 讲述 Squid 的安装过程.

### 4.1 Linux 环境准备

Squid2.7.x 源程序用 C 语言写成, 所以要在 Linux 中安装 C 语言编译程序:

```
apt-get install gcc
```

```
apt-get install build-essential
```

增加最大文件描述符数量, 以避免文件描述符在大量用户并发访问时文件描述符 (file-descriptor) 的耗尽:

```
echo 8192 > /proc/sys/fs/file-max
```

```
ulimit -Hn 8192
```

### 4.2 下载、编译并安装 Squid 程序

从 Squid 的官方网站下载 Squid 的源程序, 需要说明的是, 现在 Squid 的最新版本是 3.1.x, 但许多基于 Squid 的系统都仍在采用 2.7.x 的版本, 测试证实, 2.7.x 的版本比 3.1.x 的版本效率更高, 原因是 3.1.x 使用 C++ 重写, 增加了一些功能, 而这些功能对大多数用户来说是没有意义的, 所以建议采用 2.7.x 的版本.

Squid 官方网站的地址是 <http://www.squid-cache.org/>, 使用 wget 命令直接将 Squid 源程序下载到服务器本地硬盘.

```
wget http://www.squid-cache.org/Versions/v2/2.7/squid-2.7.STABLE9.tar.gz
```

```
解压 tar -xzvf squid-2.7.STABLE9.tar.gz
```

```
进入目录 cd squid-2.7.STABLE9
```

```
./configure--enable-storeio=aufs, ufs --disable-internal-dns
```

象许多其他 Unix 软件一样, Squid 在开始编译之前使用 ./configure 脚本来发现关于库, 函数, 类型, 参数等操作系统的信息.

Configure 脚本有大量的选项, 最重要的是安装 prefix. 读者可参考 Squid 权威指南, 从中选择适合自己环境和功能要求的选项.

本文使用了选项 enable-storeio=aufs, ufs 和 disable-internal-dns, 其中 enable-storeio=aufs, ufs 是配置 squid 使用 aufs 文件系统, 以提高 Squid 读写硬盘 cache 时的速度和吞吐率. Disable-internal-dns 是禁用 Linux 内置的 DNS 解析功能, 而改由 Squid 指定外部的 DNS 服务器, 并可提供多个 DNS 解析的进程, 以提高 Squid 服务器的 DNS 解析能力.

接下来就是编译和安装:

```
Make
```

```
make install
```

### 4.3 Squid 配置及优化

Squid 安装完成后配置文件默认是位于 /usr/local/squid/etc/ 下的 squid.conf, 主要配置项包括: 访问列表、认证方法、cache 目录及大小、日志存放等. 关于 Squid 的配置及性能优化方法, 网上有大量文档, 其

随机文档亦解释特别详细,在此不再赘述。

#### 4.4 测试并启动 Squid

```
#/usr/local/squid/sbin/squid -z
```

这是初始化 Squid 的 cache 磁盘空间

```
#/usr/local/squid/sbin/squid -NCd1
```

这是测试 squid 配置文件是否有问题,如果没问题的话将显示:"Ready to serve requests", Squid 就可以正常运行了,在/etc/rc.local 里加入/usr/local/squid/sbin/squid -s,这样可以使得每次重启服务器, Squid 都能自动启动。

### 5 代理服务器集群功能测试

代理服务器集群的 LVS 调度服务器和 Squid Real Server 安装完成后可以做测试,在调度机上,用命令 ipvsadm 查看:

```
IP Virtual Server version 1.2.1 (size=4096)
```

```
Prot LocalAddress:Port Scheduler Flags
```

```
-> RemoteAddress:Port
```

```
Forward Weight ActiveConn InActConn
```

```
TCP 202.194.64.200:irdmi wlc persistent 600
```

```
-> 192.168.1.2:irdmi Route 1 1673 19874
```

```
-> 192.168.1.3:irdmi Route 1 1313 6620
```

```
-> 192.168.1.4:irdmi Route 1 1440 11010
```

显示的是目前 3 台 Real Server 的连接信息,我们可以做测试,拔掉其中一台 Real Server 的网线,或者用 kill 杀掉 Squid 进程,甚至关掉服务器, LVS 都能检测到,因为 LVS 检测的是 Real Server 的 TCP 端口(本项目是 TCP 端口 8000),任何导致 TCP 端口失效的故障(包括进程死掉或失效) LVS 都能检测到。LVS 检测

到某台 Real Server 的故障后,将其 Weight 设置为 0,同时将不再向其转发请求,但当这台服务器的 Squid 进程回复正常以后, LVS 检测到以后,重新将其 Weight 设置为 1,重新向其转发需求,从而起到了负载分担和避免单点故障影响业务的状况。

### 6 结语

基于 LVS 和 Squid 的代理服务器集群系统目前在作者所在单位承担了大部分出口代理服务,我们为该系统配置了认证系统,利用 Squid 的认证功能和后台数据库记录,根据源 IP 地址决定是否为一个 IP 地址提供代理服务;为了提高 Squid 系统的工作效率和稳定性,对 Linux 操作系统和 Squid 的配置文件进行了许多优化。由于 LVS 集群提供了极高的可靠性,所以极少发生因设备或系统故障导致的服务中断。长期运行的结果显示,该系统提供优质的代理服务的同时,又极大地降低了系统维护的工作量。

#### 参考文献

- 1 LVS 中文站点.<http://zh.linuxvirtualserver.org>
- 2 Linux Virtual Server Project. <http://www.LinuxVirtualServer.org>
- 3 Squid 英文站点. <http://www.squid-cache.org>
- 4 姜文颖.网络中几种负载均衡实现技术的探讨.中国数据通信,2004,(1):61-62.
- 5 罗忠石,刘虹霞.用 Squid 架构本地信息代理服务器.计算机科学,2005,31(6):95-97.
- 6 章文嵩.Linux 服务器集群系统.<http://www.ibm.com/developerworks/cn/linux/cluster/lvs/part1/index.html>
- 7 彭永华.Squid 权威指南.<http://blog.s135.com/book/squid/>

(上接第 23 页)

#### 参考文献

- 1 王淑营,赵慧娟.基于 Web 的动态信息发布系统技术研究.计算机应用研究,2004,4:189-190.
- 2 王红萍,王黎明,梁民赞,等.基于 Facade 模式的移动定位服务系统平台的设计.计算机工程与科学,2010,32(7):151-153.
- 3 胡东锋.微博是这样炼成的.北京:人民邮电出版社,2010. 46-50.
- 4 Haustein S, Slominski A. XMLpull Parsing. [2012-02-21]. <http://www.xmlpull.org/>
- 5 APACHE. htmlparser.[2012-03-24]. <http://htmlparser.sourceforge.net/>.
- 6 罗刚,王振东.自己动手写网络爬虫.北京:清华大学出版社,2010.6-21.
- 7 Google. Androiddevelopers. [2012-03-24]. <http://developer.android.com/>.
- 8 杨丰盛.android 应用开发揭秘.北京:机械工业出版社,2010. 28-109.