Research on the Construction of High Available Load Balancing Cluster Based on LVS

Yun Zhang[1, a], Yongqi Han[2, b,*] and Guo Sun[1,c]

[1] Changchun Sci-Tech University of Information Engineering, Changchun, 130600, China

[2] Jilin Agricultural University College of Information Technology, Changchun, 130118, China

[a] 250814909@qq.com , [b] yongqihan@163.com, [c] 252918463@qq.com

* Corresponding author

Keywords: LVS; load balancing; VS/DR; keepalived; Web server

Abstract. Facing the network of high-speed development, people through the network were daily work, learning and shopping activities increased significantly, attendant is server receives the user request number is exponential growth. In this context, Linux virtual server (LVS) cluster provides a better solution, both at home and abroad of LVS cluster has been widely used. In view of the superiority of cluster system performance and price, the LVS cluster platform to build the VS/DR forwarding way to achieve load balancing test, to achieve the design of LVS+keepalived high availability cluster architecture, to achieve the Web server cluster load balancing and high availability.

Introduction

With the rapid development of Internet, providing network services are also changing, for people's life, work and study and provides convenient channels. At the same time, submitted by the user demand for network and server traffic was exponential growth.Such as large e-commerce site, every day to accept tens of millions or even billions of customer access request, and often have to face tremendous pressure.Especially when there are large amounts of data burst arrives, a single server often don't have the ability to finish all requests, which caused the request data loss, serious when may cause the server to crash, the collapse of service will to the site to bring huge losses and thus provides a reliable network data is very necessary.In view of the superiority of the performance and price of the cluster system, and with the application of Linux operating system more and more widely, Linux cluster has been greatly developed.

This article is through the LVS Cluster in the load balancing scheduling algorithm research, after a detailed analysis of the LVS cluster load balancing mechanism and several existing scheduling algorithm, the realization of the architecture of LVS + keepalived, the cluster can shorten the service response time, improve the overall performance of the cluster.

LVS cluster technology

In the design of the system needs to consider the transparency, scalability, high availability and easy to manage.Generally speaking, the LVS cluster adopts three layer structure.

Load balancer (load balancer), it is the whole cluster on the outside of the front end of the machine, will be responsible for the customer please send to a group of servers to perform, and customers think service is from an IP address (we can call it as a virtual IP address).

Server pool (pool server), is a group of real implementation of client requests the server, the implementation of the services are WEB, MAIL, FTP and DNS and so on.

Shared memory (storage shared), it provides a shared storage area for the server pool, so it is easy to make the server pool has the same content, providing the same service.

At present, LVS mainly has three kinds of request forwarding methods and eight scheduling algorithms.According to the different request forwarding mode, the network topology, installation method, performance and performance of the architecture cluster are also different.With LVS can be the main framework of the three forms of cluster, respectively, LVS/NAT, LVS/TUN and LVS/DR, according to the need to choose one of them.Under the selected forwarding method, the scheduling algorithm will determine the load balance of performance, different algorithms applicable to different applications, may sometimes need for special occasions, self-designed scheduling algorithm.

LVS cluster structures

Configuration environment

System OS：CentOS 5.4,Software：ipvsadm-1.24.tar.gz, keepalived-1.1.19.tar.gz

Table 1 list of information

Name	IP
LVS-Master	192.168.0.212
LVS-BACKUP	192.168.0.213
LVS-VIP	192.168.0.215
Realserver1	192.168.0.205
Realserver2	192.168.0.203
Realserver3	192.168.0.204

(1) Install LVS backup and LVS master respectively.

(2) Install keepalived

Install keepalived on LVS backup and LVS master, respectively.It runs on top of LVS, the main function is to achieve the failure of the real machine isolation and load balancing between the switch (failover), detect the state of the Web server.

(3) install Apache server

Install the three RealServer servers.Open httpd service, after the success of the installation, Apache will be installed below the /usr/local/web/apache.

Then enter the IP address of the Apache server in the IE host windows.

Configuration keepalived

Keepalived is an excellent implementation of high availability software, it runs on top of LVS, the main function is to detect the state of the Web server, to achieve the failure of the real machine isolation and load equalizer switch. (1) set the configuration file for LVS-Master.(2) set the configuration file for LVS-backup.

Start #/etc/init.d/keepalived start keepalived services, keepalived will be able to use the keepalived.conf configuration file to achieve load balancing and high availability.

Realserver configuration

The three client scripts are the same.

```
cat /etc/rc.d/init.d/realserver.sh
#!/bin/bash
# description: Config realserver lo and apply noarp
SNS_VIP=192.168.0.215
/etc/rc.d/init.d/functions
case "$1" in
start)
        ifconfig lo:0 $SNS_VIP netmask 255.255.255.255 broadcast $SNS_VIP
        /sbin/route add -host $SNS_VIP dev lo:0
        echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
        echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
        echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
        echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce
        sysctl -p >/dev/null 2>&1
        echo "RealServer Start OK" ;;
stop)
```

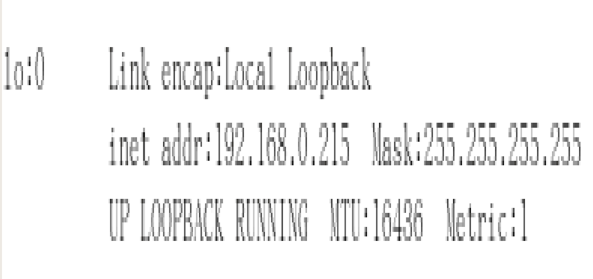
```
        ifconfig lo:0 down
        route del $SNS_VIP >/dev/null 2>&1
        echo "0" >/proc/sys/net/ipv4/conf/lo/arp_ignore
        echo "0" >/proc/sys/net/ipv4/conf/lo/arp_announce
        echo "0" >/proc/sys/net/ipv4/conf/all/arp_ignore
        echo "0" >/proc/sys/net/ipv4/conf/all/arp_announce
        echo "RealServer Stoped";;*)
        echo "Usage: $0 {start|stop}"
        exit 1
  esac
  exit 0
```

Start script command:/etc/rc.d/init.d/realserver.sh start

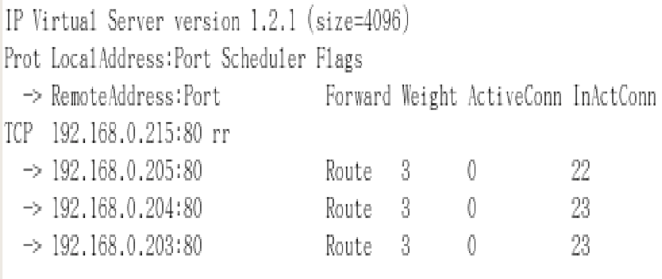
Run script prompt permission is not enough, enter the command: chmod a+x /etc/rc.d/init.d/realserver.sh

Test

(1) the first test of each readserver, to determine the various readserver can normally visit, test realserver.sh script, look at the start can not bind VIP LVS, after stopping can not remove the binding vip.Fig 1. Binding VIP.



```
lo:0      Link encap:Local Loopback
          inet addr:192.168.0.215  Mask:255.255.255.255
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
```



```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.0.215:80 rr
  -> 192.168.0.205:80             Route   3      0          22
  -> 192.168.0.204:80             Route   3      0          23
  -> 192.168.0.203:80             Route   3      0          23
```

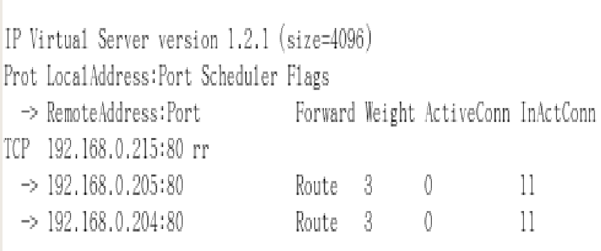
Fig1. Binding VIP

Fig2. Test LVS

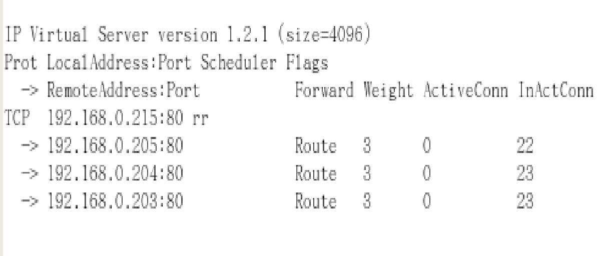
(2) test LVS can carry on the load balance forward, using the ipvsadm command to view, order:watch ipvsadm -ln.Fig 2. Test LVS.

(3) to test whether a server after a failure to shield the machine.

To shut down a server, the test results are as follows:



```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.0.215:80 rr
  -> 192.168.0.205:80             Route   3      0          11
  -> 192.168.0.204:80             Route   3      0          11
```



```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.0.215:80 rr
  -> 192.168.0.205:80             Route   3      0          22
  -> 192.168.0.204:80             Route   3      0          23
  -> 192.168.0.203:80             Route   3      0          23
```

Fig 3.Test server failure

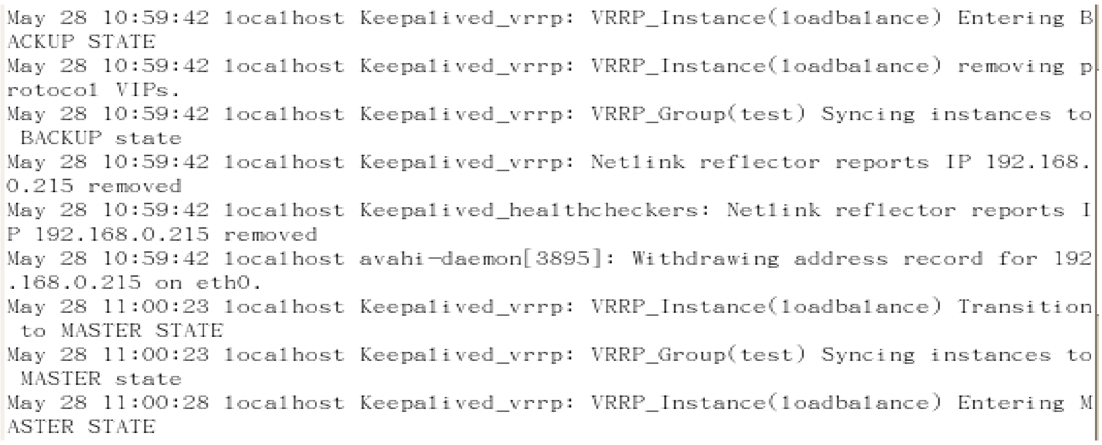
Fig 4. After the server failure repair

Through the experiment, we can know, when the RealServer fault or is not available, load balancers through health checks automatically delete failure of the machine from the forwarding queue, realize the fault isolation, ensure that the user's access is not affected.

The restart is closed by the RealServer, the test results are show in Fig. 3 and Fig. 4.

When the RealServer fault recovery, the load equalizer automatically added to the forwarding queue after the recovery of the restored machine through health checks.

(4) stop the main LVS to see if the backup LVS is taking over



```
May 28 10:59:42 localhost Keepalived_vrrp: VRRP_Instance(loadbalance) Entering B
ACKUP STATE
May 28 10:59:42 localhost Keepalived_vrrp: VRRP_Instance(loadbalance) removing p
rotocol VIPs.
May 28 10:59:42 localhost Keepalived_vrrp: VRRP_Group(test) Syncing instances to
 BACKUP state
May 28 10:59:42 localhost Keepalived_vrrp: Netlink reflector reports IP 192.168.
0.215 removed
May 28 10:59:42 localhost Keepalived_healthcheckers: Netlink reflector reports I
P 192.168.0.215 removed
May 28 10:59:42 localhost avahi-daemon[3895]: Withdrawing address record for 192
.168.0.215 on eth0.
May 28 11:00:23 localhost Keepalived_vrrp: VRRP_Instance(loadbalance) Transition
 to MASTER STATE
May 28 11:00:23 localhost Keepalived_vrrp: VRRP_Group(test) Syncing instances to
 MASTER state
May 28 11:00:28 localhost Keepalived_vrrp: VRRP_Instance(loadbalance) Entering M
ASTER STATE
```

Fig 5. Test LVS preparation machines

For more details, please check the log, -f /var/log/messages tail.From the log such as Fig. 5 test LVS preparation machines, we can know that the host appeared after the failure, the machine immediately detected, this time the machine into a MASTER role, and took over the host of the virtual IP resources, and finally the virtual IP binding on the eth0 device.

Summary

Scheduler has a good throughput, can be very good to the request of equalizing transfers to a different server execution and scheduler was able to automatically shield off the server fault, which will be a group of servers constitute a high performance and high availability of the virtual server.

Acknowledgments

This work was funded by Jilin provincial agricultural science and Technology Innovation Center,National College Students Innovation and entrepreneurship training program in 2015(Multifunctional Family Assistant).Project support of Jilin Provincial Department of Education:(624th[2014]) (175th [2015]) (184th[2015]), Project support of Changchun Municipal Science and technology bureau:(13KG71)(14NK029).

References

[1] Yuo, Zehua, et al. "Improving the performance of load balancing in software-defined networks through load variance-based synchronization." Computer Networks 68.11(2014):95-109.

[2] Han, Yongqi, et al. "Research on Real-time Data Push Technology of Web Server." 5th International Conference on Information Engineering for Mechanics and Materials Atlantis Press, 2015.

[3] Wei, L. I., L. U. Jie, and M. A. Jun. "Implementation and Research for High-availability Cluster System of Load Balance Based on Linux." Journal of Hebei United University (2013).

[4] Zhang, Yongqiang, and W. Li. "Research and Implement of the High-Performance Website Architecture Based on the Cluster Technology." Proceedings of the 2012 Second International Conference on Electric Information and Control Engineering - Volume 02 2012:1208-1211.

[5] Yang, Shun Tao. "The Design about the High-performance Load Balancing Cluster System based on LVS." Journal of Guangxi University for Nationalities (2012).