# Linux Virtual Server for Scalable Network Services

Wensong Zhang
wensong@gnuchina.org

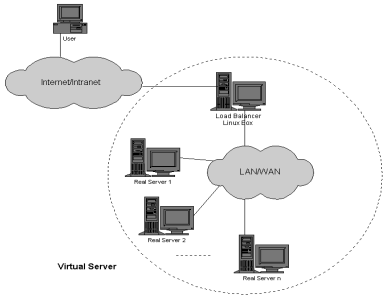
Ottawa Linux Symposium 2000
July 22th, 2000



## Introduction

- Explosive growth of the Internet
- The requirements for servers
  - Increamental scalability
  - 24x7 availability
  - Manageability
  - Cost-effectiveness
- The single server solution
- The cluster of servers solution

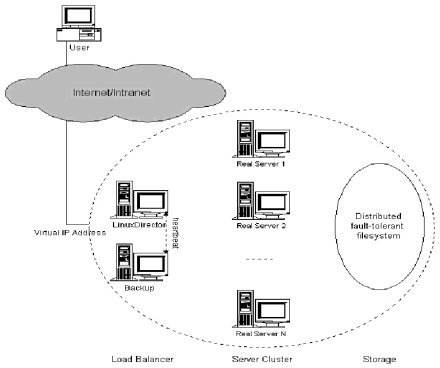
## Linux Virtual Server




Linux Virtual Server is a software tool that supports load balancing among multiple Internet servers that share their workload. It can be used to build scalable network services.



## 3-tier architecture of LVS

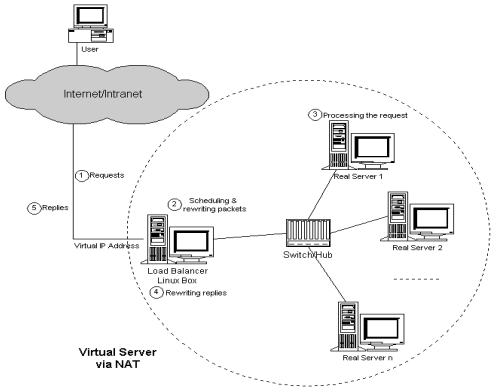



## IP Load Balancing Techniques

- Virtual Server via NAT (Network Address Translation)
- Virtual Server via IP Tunneling
- Virtual Server via Direct Routing



## Virtual Server via NAT



## An example of virtual server via NAT

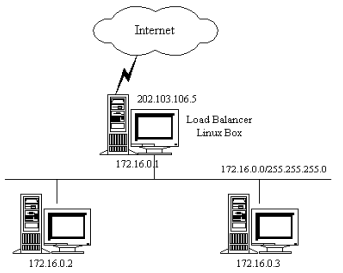



**Table 1: an example of virtual server rules**

| Protocol | Virtual IP Address | Port | Real IP Address | Port | Weight |
|---|---|---|---|---|---|
| TCP | 202.103.106.5 | 80 | 172.16.0.2 | 80 | 1 |
| | | | 172.16.0.3 | 8000 | 2 |
| TCP | 202.103.106.5 | 21 | 172.16.0.3 | 21 | 1 |



## Packet rewriting flow:

The incoming packet for web service would has source and destination addresses as:

| SOURCE | 202.100.1.2:3456 | DEST | 202.103.106.5:80 |
|---|---|---|---|

The load balancer will choose a real server, e.g. 172.16.0.3:8000. The packet would be rewritten and forwarded to the server as:

| SOURCE | 202.100.1.2:3456 | DEST | 172.16.0.3:8000 |
|---|---|---|---|

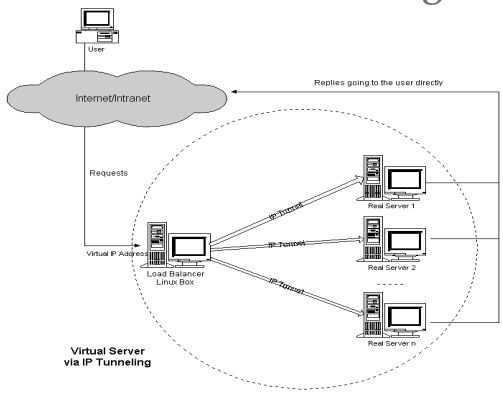
Replies get back to the load balancer as:

| SOURCE | 172.16.0.3:8000 | DEST | 202.100.1.2:3456 |
|---|---|---|---|

The packets would be written back to the virtual server address and returned to the client as:

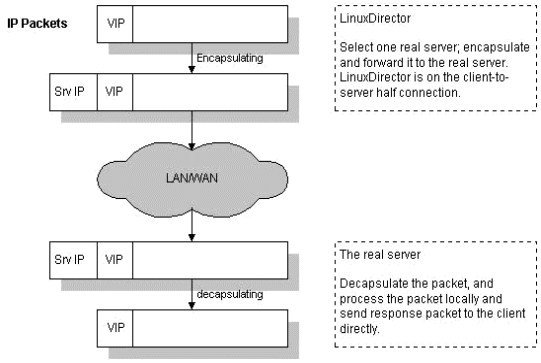
| SOURCE | 202.103.106.5:80 | DEST | 202.100.1.2:3456 |
|---|---|---|---|

## VS via IP Tunneling

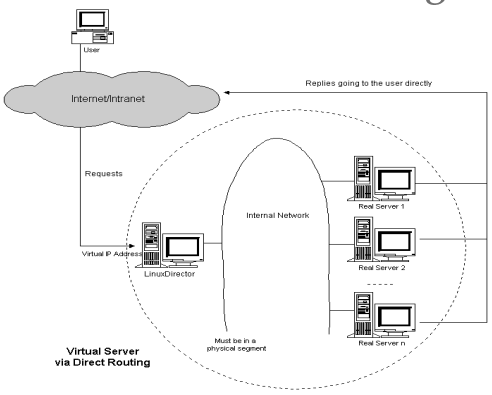





## VS-Tunneling Workflow

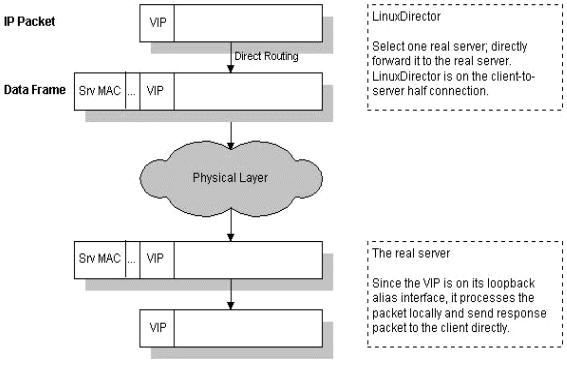



## VS via Direct Routing






## VS-DRouting Workflow



## Advantages and Disadvantages

- Virtual Server via NAT
- Virtual Server via IP Tunneling
- Virtual Server via Direct Routing

13

## Virtual Server via NAT

- Advantages:
  - real servers can run any OS that supports TCP/IP
  - only an IP address is needed for the load balancer, real servers can use private IP addresses.
- Disadvantages:
  - the maximum number of server nodes is limited, because both request and reponse packets are rewritten by the load balancer. When the number of server nodes increase up to 20, the load balancer will probably become a new bottleneck.

## Virtual Server via IP Tunneling

- Advantages:
  - real servers send response packets to clients directly, which can follow different network routes
  - real servers can be in different networks, LAN/WAN.
  - greatly increasing the scalability of Virtual Server.
- Disadvantages:
  - real servers must support IP tunneling protocol.

15

## Virtual Server via Direct Routing

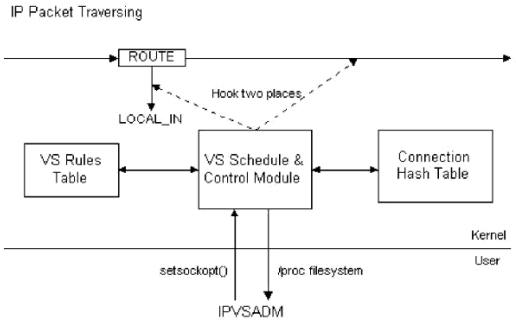
- Advantages:
  - real servers send response packets to clients directly, which can follow different network routes
  - no tunneling overhead
- Disadvantages:
  - servers must have non-arp alias interface; or servers can be configured to redirect some packets to local port.
  - the load balancer and servers must have one of their interfaces in the same LAN segment.

## The Comparison Table of VS-NAT, VS-Tunneling and VS-DRouting

| | VS-NAT | VS-Tunneling | VS-DRouting |
|---|---|---|---|
| Server OS | any | tunneling | non-arp device |
| Server network | private | LAN/WAN | LAN |
| Server number | low (10~20) | high (>100) | high (>100) |
| Server gateway | load balancer | own router | own router |

17

## Implemention Issues



## Implemention Issues (cont')

- Each connection entry uses 128 bytes effective memory.
- Connection hash table using clients' <protocol, address, port> as hash key.
- Slow timer to collect stale connection.
- ICMP handling
- Three packet forwarding methods can be used together in a single load balancer.



## Connection Scheduling

Fine scheduling granularity:

Network connection

The scheduling algorithms:

- Round-Robin Scheduling
- Weighted Round-Robin Scheduling
- Least-Connection Scheduling
- Weighted Least-Connection Scheduling

## Connection Affinity

**Sometimes the connections from the same client must be assigned to the same server either for functional or for performace reasons, such as FTP, SSL, http cookies.**

**Use the persistent template to handle connection affinity.**
**<cip, 0, vip, 0, sip, 0> for FTP**
**<cip, 0, vip, vport, sip, sport> for persistent services except FTP.**



## The LocalNode Feature

- In a virtual server of only a few nodes(2,3 or more), it is a resource waste if the load balancer is only used to direct packets.
- The LocalNode feature enable that the load balancer not only can redirect packets, but also can processe some packets locally.

## FWMARK-based services

- Use a firewall-mark to denote a virtual service instead of <protocol, address, port>
- It can be flexibly used to build a virtual services associated to different IP addresses and port numbers.



## LVS Cluster Management Software

- RedHat Cluster Server / Piranha
  - LVS+Piranha Cluster Management tools.
- UltraMoney: Open-Source Server Farm
  - LVS+lvs-gui+heartbeat+ldirectord
- heartbeat+ldirectord
- heartbeat+mon
- ...

## Some sites using LVS

- UK National JANET Cache (wwwcache.ja.net)
- www.linux.com
- sourceforge.net
- One of largest PC manufacturers
- www.netwalk.com
- ...



## Related Works

- The client-side approach
- The server-side Round-Robin DNS approach
- The server-side application-level scheduling approach
  - EDDIE
  - pWEB
  - Reverse-proxy (Apache)
  - SWEB

## Related Works (cont')

- The server-side IP-level scheduling approach.
  - Berkeley's MagicRouter , Cisco's LocalDirector, Alteon's ACEDirector, F5 Big/IP
  - IBM's TCP router
  - ONE-IP
  - IBM's NetDispatcher



## Conclusion

- LVS has patched Linux kernel 2.0 and kernel 2.2 to support three IP load balancing techniques:
  - VS-NAT, VS-Tunneling, VS-DRouting
- Four scheduling algorithms
  - RR, WRR, LC, WLC
- High scalability (up to 100 nodes)
- High availability
- Supporting most of TCP and UDP services, no modification to either clients or servers.

## Compared to Other Commercial Products

- Three IP load balancing technologies
- Multiple scheduling algorithms
- A robust and stable code base, a large user and developer base.
- Reliability proven in big real world applications
- Free to everyone



## Future Work

- Making the LVS netfilter module for kernel 2.4 stable in the following month
- Implementing application-based (layer-7) load balancing inside the kernel.
- More load-balancing algorithms or load-sharing algorithms
- Exploring higher degrees of high availability (or even fault-tolerance)

## Acknowledgements


- Thanks to Julian Anastasov, Joseph Mack, Peter Kese, Horms, Lars Marowsky-Bree, Roberto Nibali and others.
- Thanks to Red Hat, OLS.