Introduction to Linux Virtual Server and High Availability

Chen Kaiwang
kaiwang.chen@gmail.com

December 5, 2011

If you don't know the theory, you don't have a way to be rigorous.

Robert J. Shiller

http://www.econ.yale.edu/~shiller/

Misery stories

- Jul 2011 Too many connections at zongheng.com
- Aug 2011 Realserver maintenance at 173.com
  quiescent persistent connections
- Nov 2011 Health check at 173.com
- Nov 2011 Virtual service configuration at 173.com
  persistent session data

Outline of Part I

Outline of Part II

HA Basics

- LVS High Avaliablity
- Realserver Failover
- Director Failover

Solutions

- Heartbeat
- Keepalived

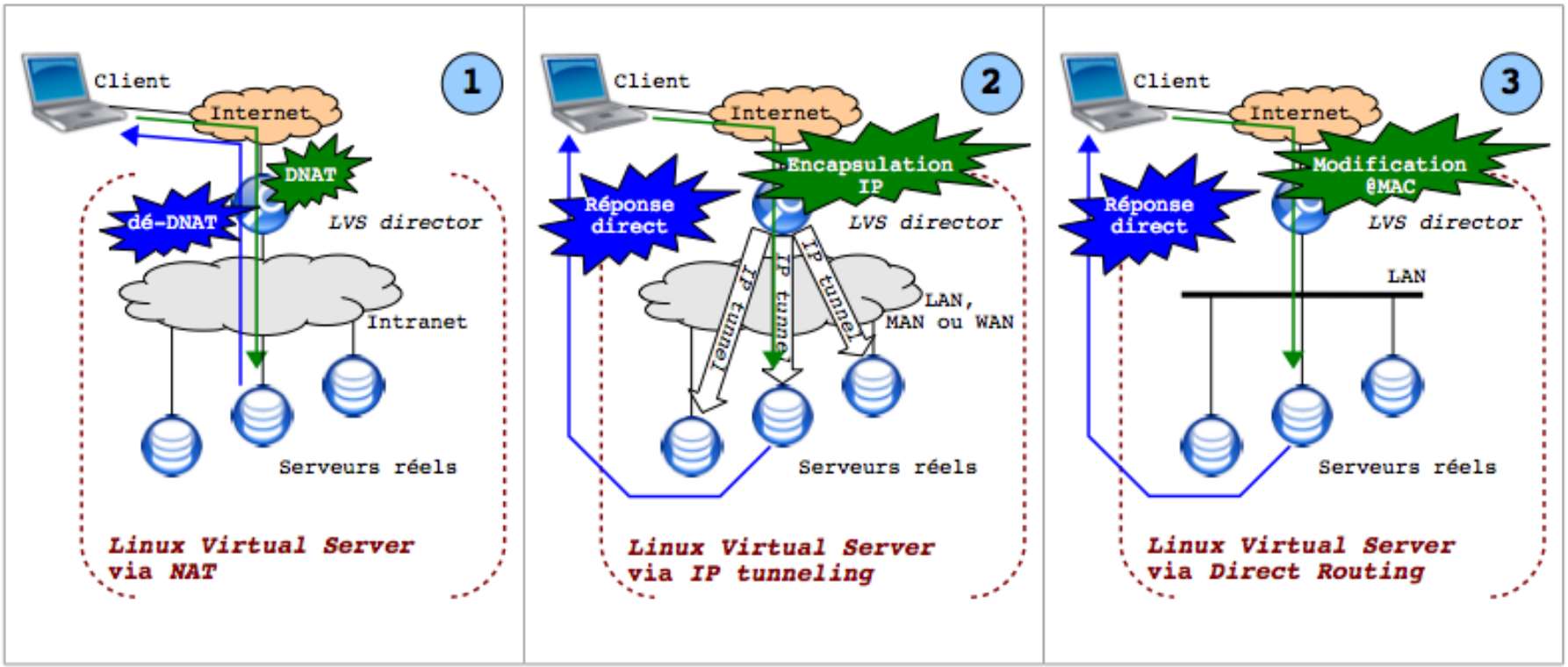
Part I

Introduction to Linux Virtual Server

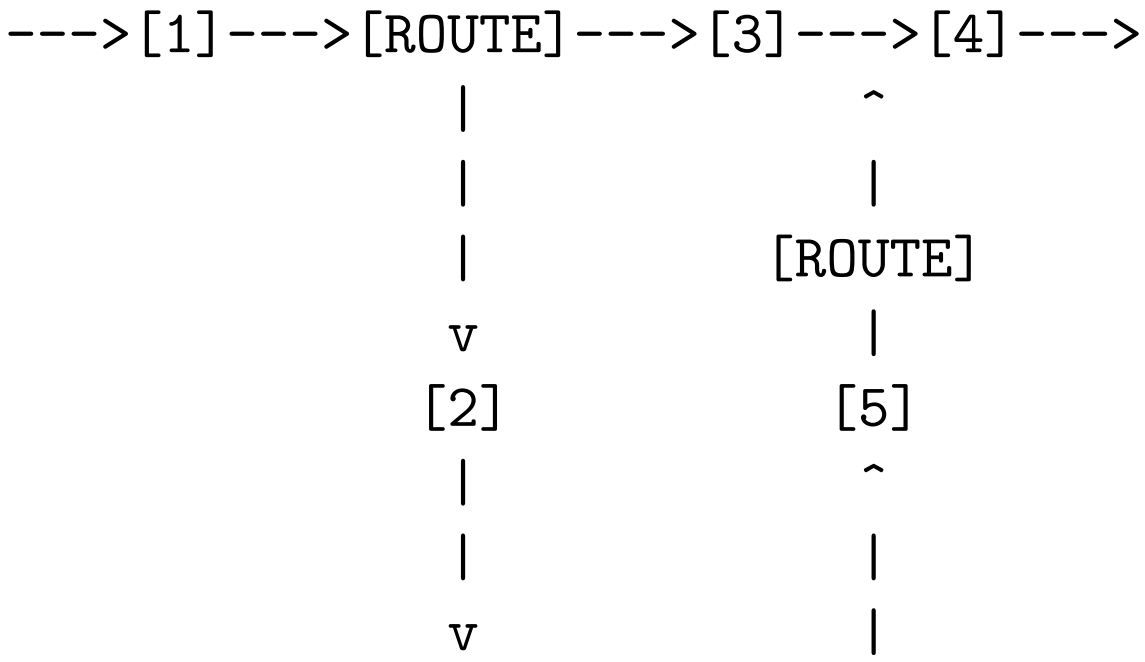
A Linux Virtual Serverr (LVS) is a group of servers that appear to the client as one large, fast, reliable (highly available) server. The core of the project is the ip_vs code, which runs on the LVS director.

- Layer 4 switch (Director)
- Backend servers (Realservers)

LVS Configurations



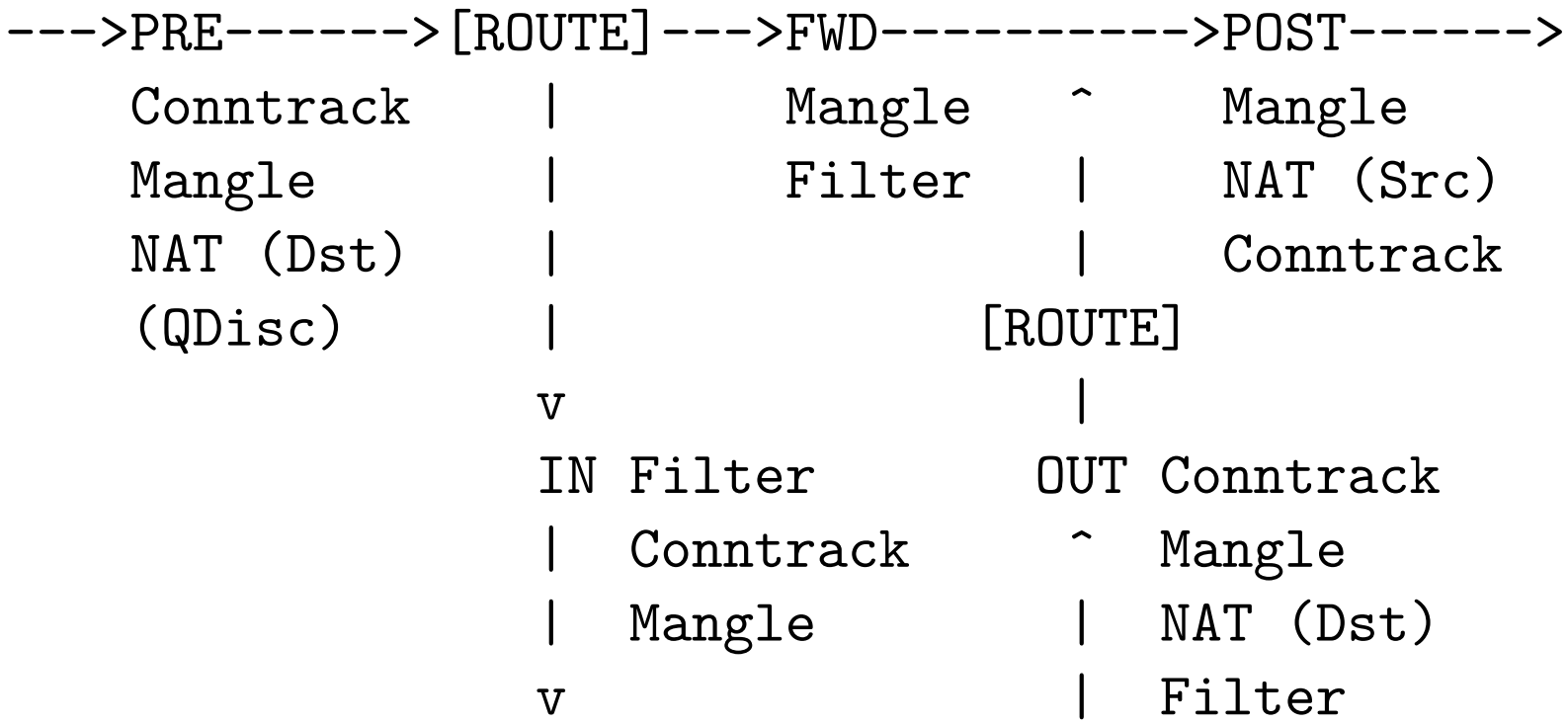
A Packet Traversing the Netfilter System



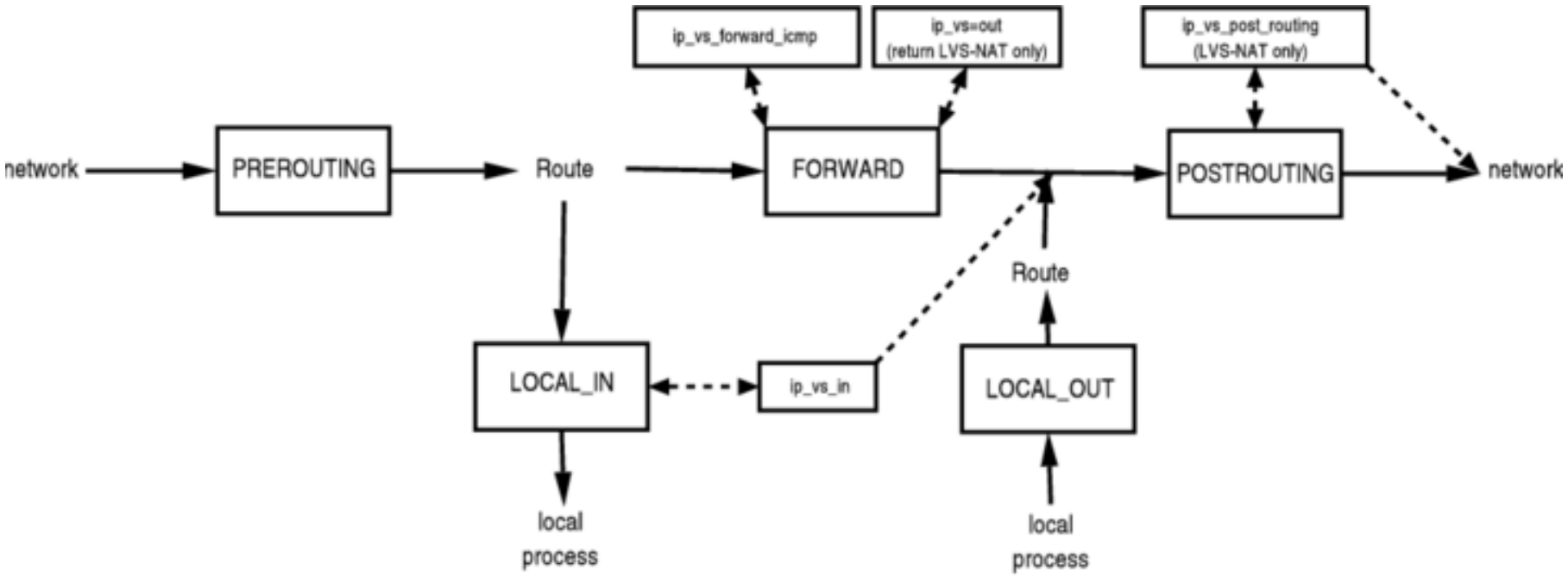

```
NF_IP_PRE_ROUTING [1]    NF_IP_POST_ROUTING [4]
NF_IP_LOCAL_IN    [2]    NF_IP_LOCAL_OUT    [5]
NF_IP_FORWARD     [3]

NF_ACCEPT, NF_DROP, NF_STOLEN, NF_QUEUE, NF_REPEAT
```

Packet Selection: IP Tables



Interaction of LVS with Netfilter



Linux Kernel Netfilter Hooks and LVS

Horms <horms@verge.net.au>, v0.1.9-1, October 2003

Modules register with a priority, the lowest priority getting to look at the packets first. LVS registers itself with a higher priority than iptables rules, and thus iptables will get the packet first and then LVS.

Job Scheduling

Which realserver to service a new connection request

Virtual service, and server pool

VIP:PORT, or fwmark

Scheduling granularity

LVS - network connection-based
DNS - host-based, TTL

Quiescent feature

stop the realserver from being selected by the scheduler

Schedulers

- RR, WRR,
  weight and scheduling sequence
- LC, WLC,
  estimating realserver connections? TIME_WAIT
- DH, LBLC, LBLCR,
  by dest ip, aplicable to transparent proxy where the dest ip could be variable.
- SH,
  by client ip.
- Persistent connection

If the realservers are offering different services and some have clients connected for a long time while others are connected for a short time, or some are compute bound, while others are network bound, then _none_ of the schedulers will do a good job of distributing the load between the realservers. LVS doesn't have any load monitoring of the realservers.

Ratz Nov 2006 After almost 10 years of my involvement with load balancers, I have to admit that no customer _ever_ truly asked or cared about the scheduling algorithm :). This is academia for the rest of the world.

The two meanings of persistence

- Keep-Alive
  used for clients connecting to webservers and databases
- Affinity (LVS persistence)
  used by cisco.

Persistance opperates independant of the scheduler

It looks up a persistance template and if it finds one, then it uses it, else it asks the scheduler what to do.

Persistence Template

IP_VS connection table

<CIP, VIRTUAL_SERVICE, RIP:PORT, FLAGS>

- VIRTUAL_SERVICE is either VIP:PORT or fwmark
- the NONE flag is the trick

  # ipvsadm -L -n -c | grep NONE

The two kinds of conneciton tracking

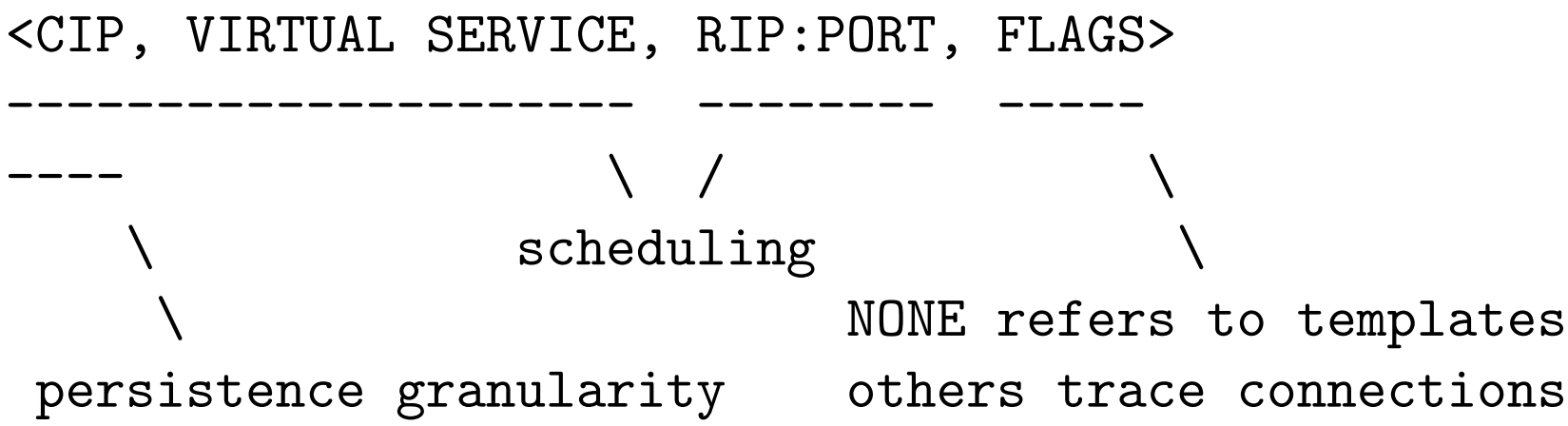
- IP_VS ip_vs
- Netfilter ip_conntrack

Persistence Granularity

Loadbalance all clients from a netmask as one group.

- Applied to the CIP
- Works the same whether you are using fwmark or the VIP to setup the LVS

IP_VS table entry revisited



Clear the table

ipvsadm -C

expire_nodest_conn (destined for a server no longer in the pool)

- 1: expire entry and reset client
- 0: keep entry and drop packet

Clear quiescent persistent connections

expire_quiescent_template (timeout persistent template when server goes down)

ARP problem with LVS-DR

Part II

LVS High Availability

A design principle of HA systems is three distinct paths to the servers: resource-path (or public path), heartbeat, and administrative.

- Single point of failure
  you can't protect against everything
- Health check accuracy
- Stateful failover
  - connection table
  - persistent session data

Realserver Failover

Service check

tcp, http, https, smtp, ...

Server operations

- Realserver pool
  - Added to pool
  - Quiescent (weight=0)
  - Removed from pool
- Sorry server

Stateful failover

- conntrackd
- persistent session data

Director Failover

VRRP, Linux-HA

- VIP takeover
  unsolicited ARP
- Server state sync daemon

Heartbeat

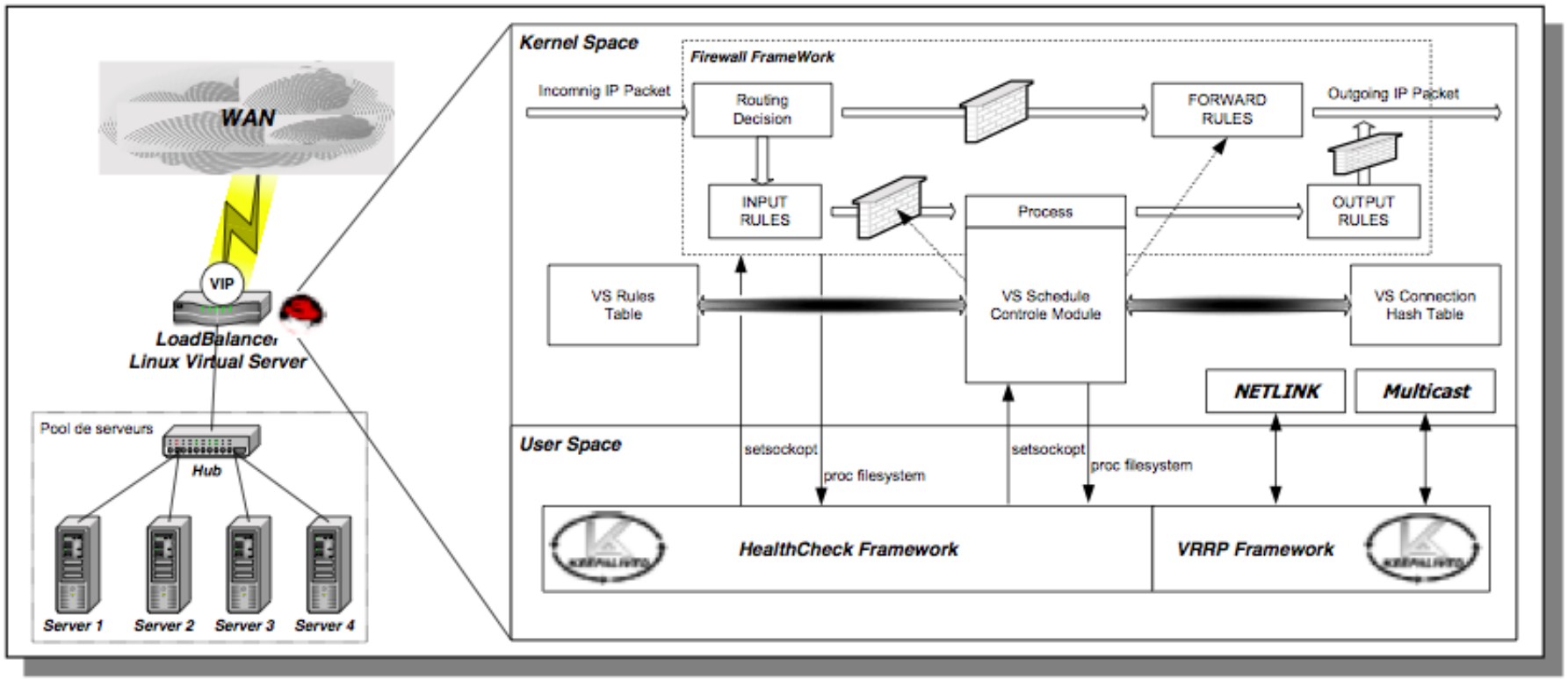
- The heartbeat cluster messaging layer
- Resource agents
- Local resource manager, and STONITH

Keepalived

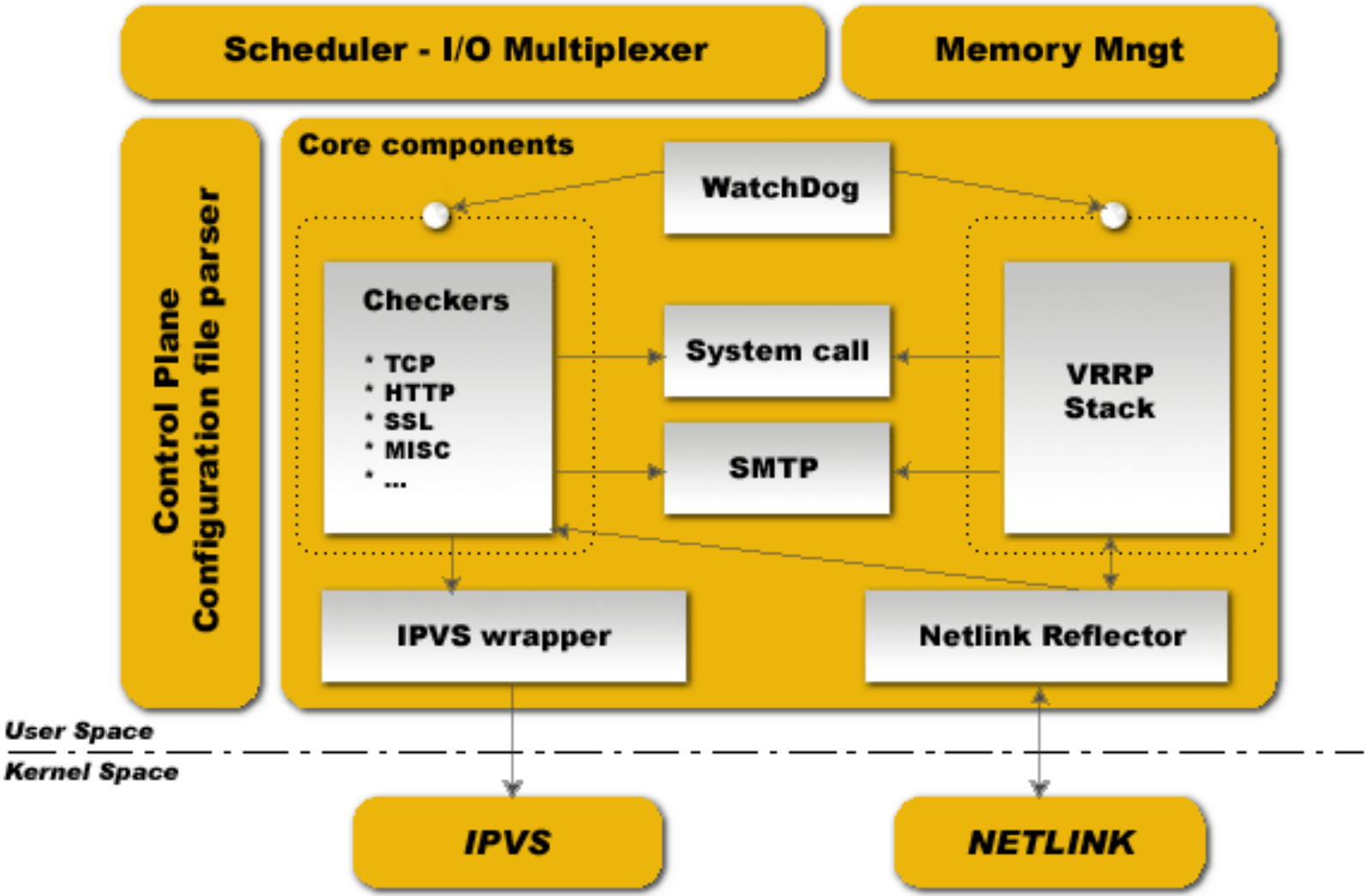
Keepalived is a userspace daemon for LVS cluster nodes healthchecks and LVS directors failover.

```
PID
111 Keepalived    <-- Parent process monitoring childs
112 \_ Keepalived <-- VRRP children
113 \_ Keepalived <-- Healthchecking children
```

Keepalived Global View



Keepalived Design



Thanks

References

[1] LVS-HOWTO http://www.austintek.com/LVS/LVS-HOWTO/

[2] The Linux Virtual Server Project http://www.linuxvirtualserver.org/

[3] Netfilter workshops http://workshop.netfilter.org

[4] Linux netfilter Hacking HOWTO
http://netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO.html

[5] Linux Advanced Routing & Traffic Control HOWTO http://lartc.org/howto/lartc.netfilter.html

[6] Recent and Future Developments in IPVS
http://workshop.netfilter.org/2010/wiki/images/6/6a/Lvs.en.pdf

[7] Conntrackd: High Availability for stateful Linux firewalls
http://workshop.netfilter.org/2007/presentations/conntrackd-nfws.odp

[8] Linux-HA project http://linux-ha.org

[9] Keepalived for Linux http://www.keepalived.org

[10] Alexandre Simon at JRES'2011 http://www.keepalived.org/pdf/asimon-jres-paper.pdf