

# Load Balancing und Load Sharing mit freier Software

Volker Dormeyer  
<volker@ixolution.de>

im Mai 2007

# Agenda

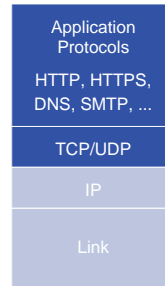
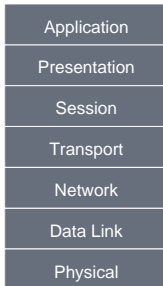
- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary

# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary

# Which OSI-Layers are used by Load Balancers?

- OSI Layers 4-7



# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 **Load Balancing**
  - **Concepts**
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary

## What it is...

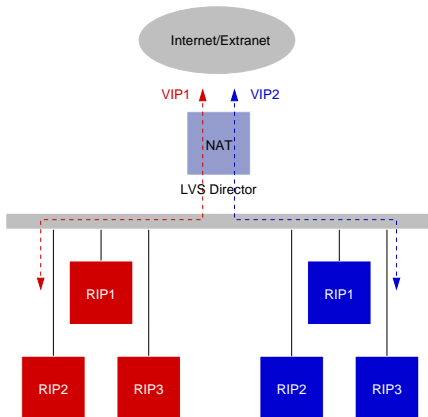
- Dynamic distribution of incoming requests across real servers in the background
- Layer 4-7 Switch

# Forwarding Methods

- NAT (Network Address Translation)
- DR/DSR (Direct Routing / Direct Server Return)
- TUN (Tunneling)

# NAT

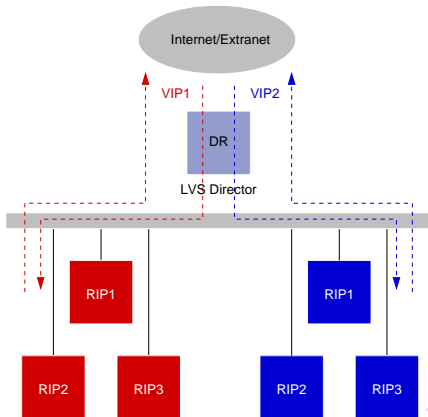
- Request and answer traverse the Load Balancer





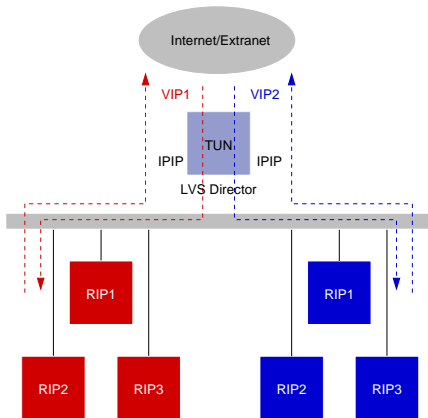
# DR/DSR

- Requests traverse the Load Balancer
- Answers do not



# TUN

- IP IP Encapsulation



# NAT

+

- Independent from real server
- Simple to understand and implement

-

- It is not possible to NAT every protocol
- Relative low throughput compared to DR/DSR

## DR/DSR

+

- Application independent
- High throughput

-

- Additional configuration on real servers necessary
- Load Balancer has to be located in the real server network

# TUN

+

- Application independent
- High throughput
- Real servers can be located in foreign networks

-

- Additional configuration on real servers necessary

# Scheduling

- Determine the right real server for a single request

## Usual scheduler types (weighted/unweighted)

- Round Robin
- Least Connection
- Locality Based Least Connection (replication)
- Hashing (source/destination)
- Shortest Expected Delay
- Never Queue
- Weighted Least Incoming Bps

# Connection Persistence

- Also known as “Stickyness” or “Server Affinity”
- Layer 4 Persistence
  - Based on source IP-address
  - Based on source IP-address and source/destination port
  - Based on source network
- “Connection Template” expires after user defined time interval
- Layer 7 Persistence

# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing
  - Concepts
  - **Health Checking**
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary



# Health Checking (1)

- Check whether real servers are still alive
- Methods in Layers 3-7
  - ICMP Pings
  - Port Availability
  - Application Availability (HTTP GET/HEAD, etc.)
  - Analysation of application check result
- In Band / Out of Band checks
- GNU/Linux based real servers
  - Feedback Agent
  - Dynamic modification of scheduler weights depending on load of real servers

## Health Checking (2)

- Connection Limits
  - Limit number of connections to selected real server systems
  - Incl. automatic correction of scheduler weights

# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 **Load Balancing**
  - Concepts
  - Health Checking
  - **Layer 7 Switching**
  - Director-HA
  - Products
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary

# Layer 4 Switch Problems

- Megaproxies

# Layer 7 Switching

- Parsing of application protocols

## Methods

- URL-Switching (hash + pattern comparison)
- Session-ID parsing (HTTPS)
- Cookie parsing

# Layer 7 Persistency

- URL
- Session-ID (HTTPS)
- Cookie Data

# Agenda

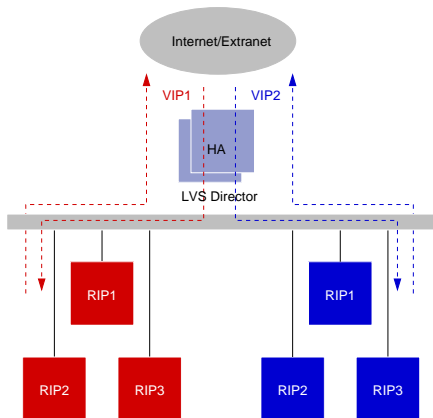
- 1 Introduction
  - OSI- and IP-Layers
- 2 **Load Balancing**
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - **Director-HA**
  - Products
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary

# Redundant Directors

- Active/Passive
- Active/Active
- Connection synchronisation (saru)

## Linux-HA or Keepalived

- STONITH
- ipfail / pingd
- VRRP





# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing**
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products**
- 3 Load Sharing
  - Concept
  - Health Checking
  - Products
- 4 Summary

# Load Balancing Products

## Load Balancing

- LVS (Layer 4, combined with KTCPVS/l7sw also Layer 7)
- Pen (Proxy, Layer 4, TCP only)
- Pound (Proxy, Layer 7)
- HAProxy (Proxy, Layer 7)

## Health Checking

- Keepalived
- Ldirectord
- Mon

## Director-HA

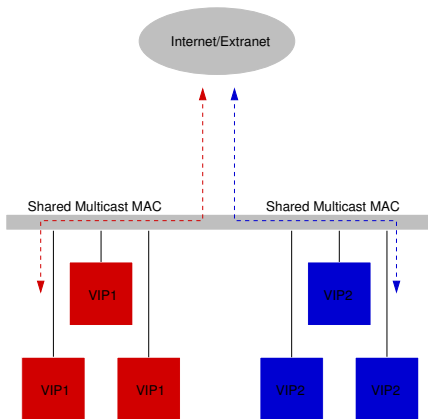
- Linux-HA
- Keepalived

# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing**
  - Concept**
  - Health Checking
  - Products
- 4 Summary

# Load Sharing

- Requests are directly processed by real servers



# Load Sharing

+

- High Throughput

-

- Requires additional software/logic on real servers
- Dynamic load balancing impossible
- Health Checking difficult

# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing**
  - Concept
  - Health Checking**
  - Products
- 4 Summary

# Health Checking

- Mutual node check
- Mutual application check
- Coordinating node
- ClusterManager integration

# Agenda

- 1 Introduction
  - OSI- and IP-Layers
- 2 Load Balancing
  - Concepts
  - Health Checking
  - Layer 7 Switching
  - Director-HA
  - Products
- 3 Load Sharing**
  - Concept
  - Health Checking
  - Products**
- 4 Summary



# Load Sharing Products

## CLUSTERIP

- Netfilter Target
- Integrates with Linux-HA
- Health Checking by Linux-HA

## Appreciated Side-Effects

### of Load Balancing and Load Sharing

- Simple/flexible maintenance possibilities without service interruption
- Automatic “Fallback”

# Literature



Chandra Kopparapu  
*Load Balancing Servers, Firewalls and Caches*  
Wiley Computer Publishing, 2002.



Matthew Syme, Philip Goldie  
*Optimizing Web Performance With Content Switching*  
Prentice Hall, 2003.



Karl Kopper  
*The Linux Enterprise Cluster*  
*Build a HA Cluster with Commodity Hardware and Free Software*  
New Starch Press, 2005.



Evan Marcus, Hal Stern  
*Blueprints for High Availability*  
Wiley Publishing, 2003.

# Questions

Questions?