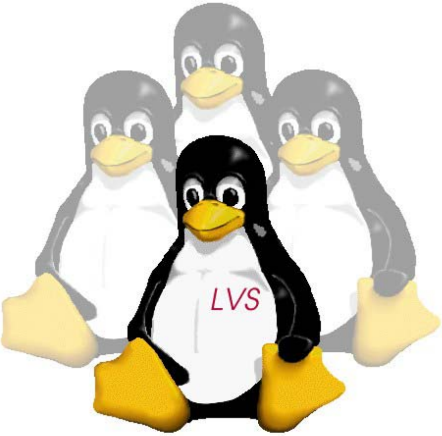
Linux Virtual Server



Miroslav Siket
FIO-FS

Outline

- Solutions and comparison
- LVS (Linux Virtual Server)
- LVS solutions, high availability
- Scheduling possibilities
- Performance
- Deployment

Definition of the problem

- Multiple clients (>100) accessing the same resource
- Effort to parallelize requests in a distributed system (clusters, computer farms,…)
- Create transparent environment for the users
- Enable load balancing to increase efficiency and speed
- Reliability

Solutions

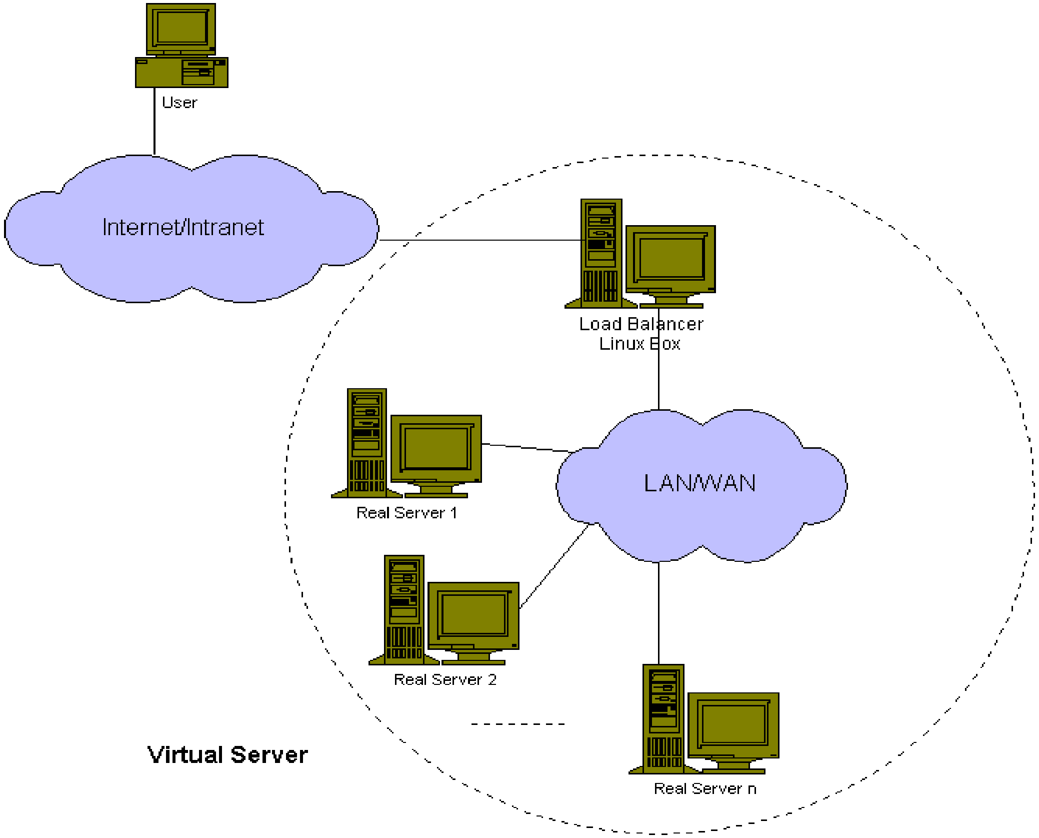
- DNS load balancing (BIND RR)
- Proxy servers (Squid, MS, Netscape, Apache)
- Reverse proxy (Apache)
- Custom built application balancing
- Hardware based (CISCO Load Director, intelligent routers,...)
- Linux Virtual Server (LVS)

What is LVS

- Scalable, highly available server built on cluster of real servers
- Offers load balancing
- Transparent for users
- Can be used for WWW, ftp, ssh or any other TCP/IP, UDP/IP protocol based services

LVS - general schema

- Idea is to have transparent service available to users
- Should be scalable
- Should be dynamical
- Should provide fail over mechanism



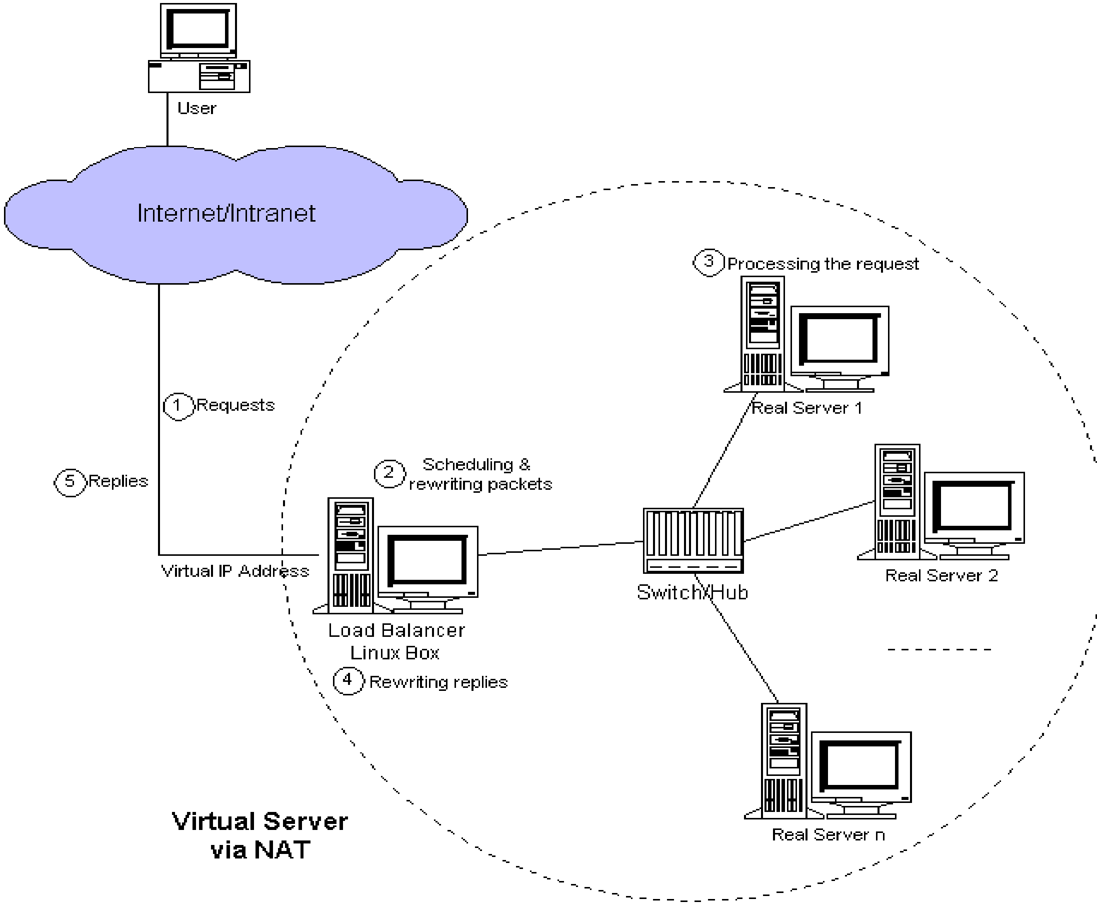
Solutions with LVS

- Is based on "smart" kernel routing of packets over IP

	LVS-NAT	LVS-TUN	LVS-DR
Network	Any	LAN/WAN	LAN
# of RealS	Low(10-20)	High	High
Gateway	Load Balancer	Own router	Own router

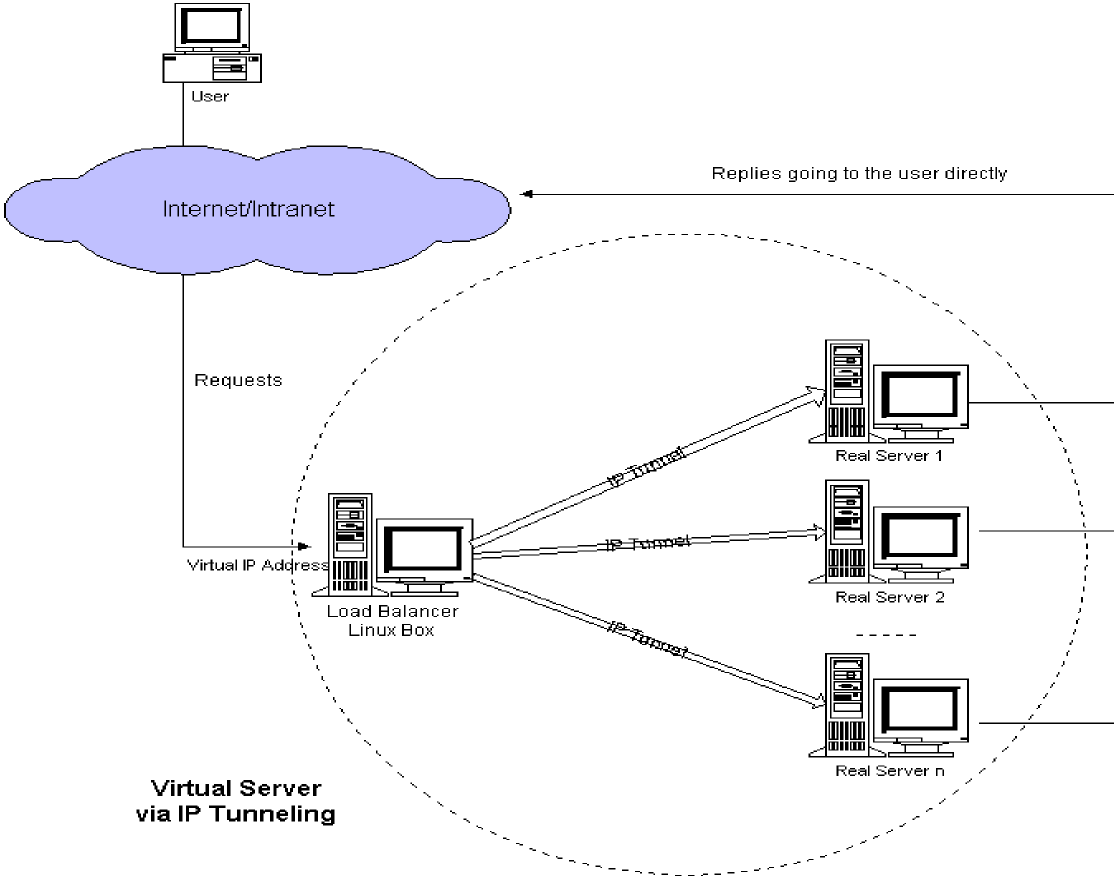
LVS-NAT

- Requests translated by load balancer
- □ Allows internal network
- □ Limited by the load balancer
- □ Limited by network connection
- □ Up to 10-20 real servers
- □ NAT (network address translation)



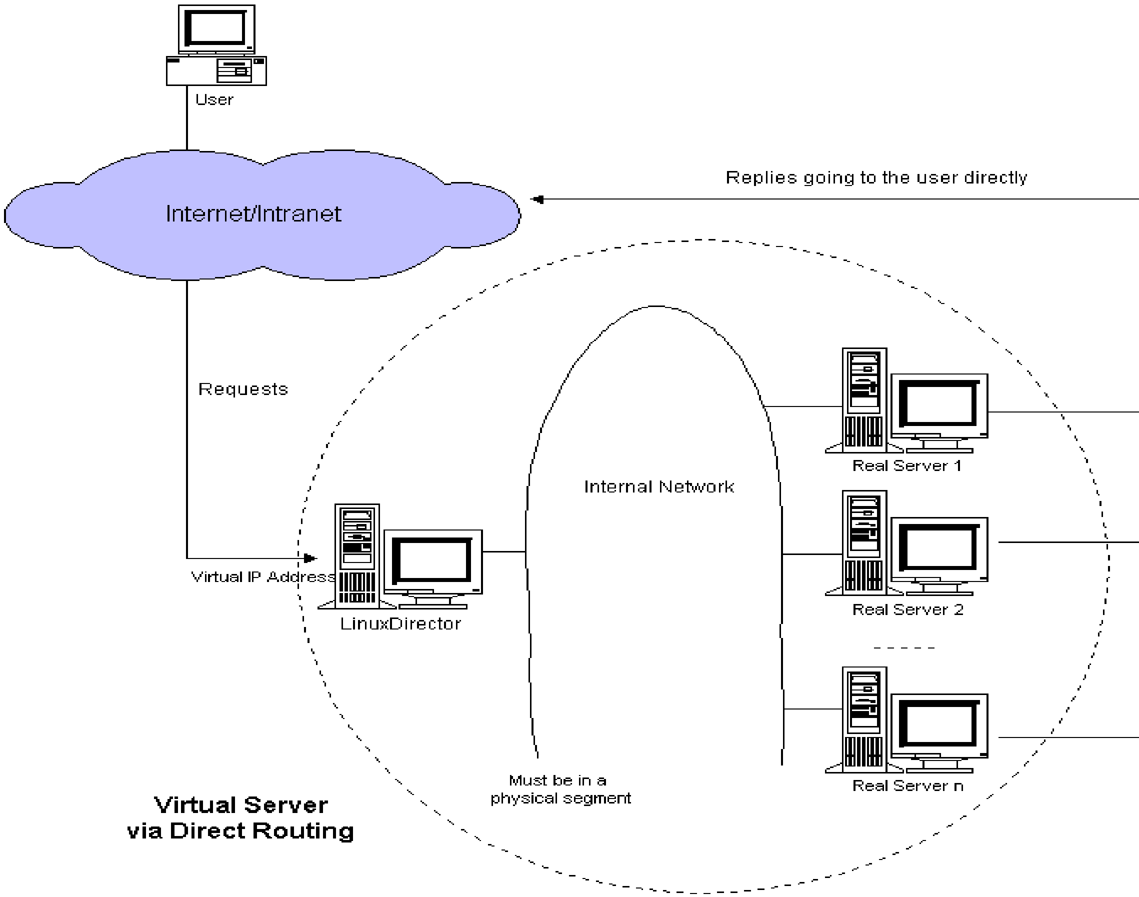
LVS-TUN

- Load balancer encapsulates IP packet, sends it to real server
- Real server replies directly to client
- Works on LAN/WAN
- Scalable
- TUN (Network Tunneling) – using tunl devices



LVS-DR

- Load balancer changes MAC address and broadcasts it to the real server
  - Limited to LAN
  - Scalable
  - Replies go directly to user
  - DR (direct routing) using lo devices



LVS Scheduling

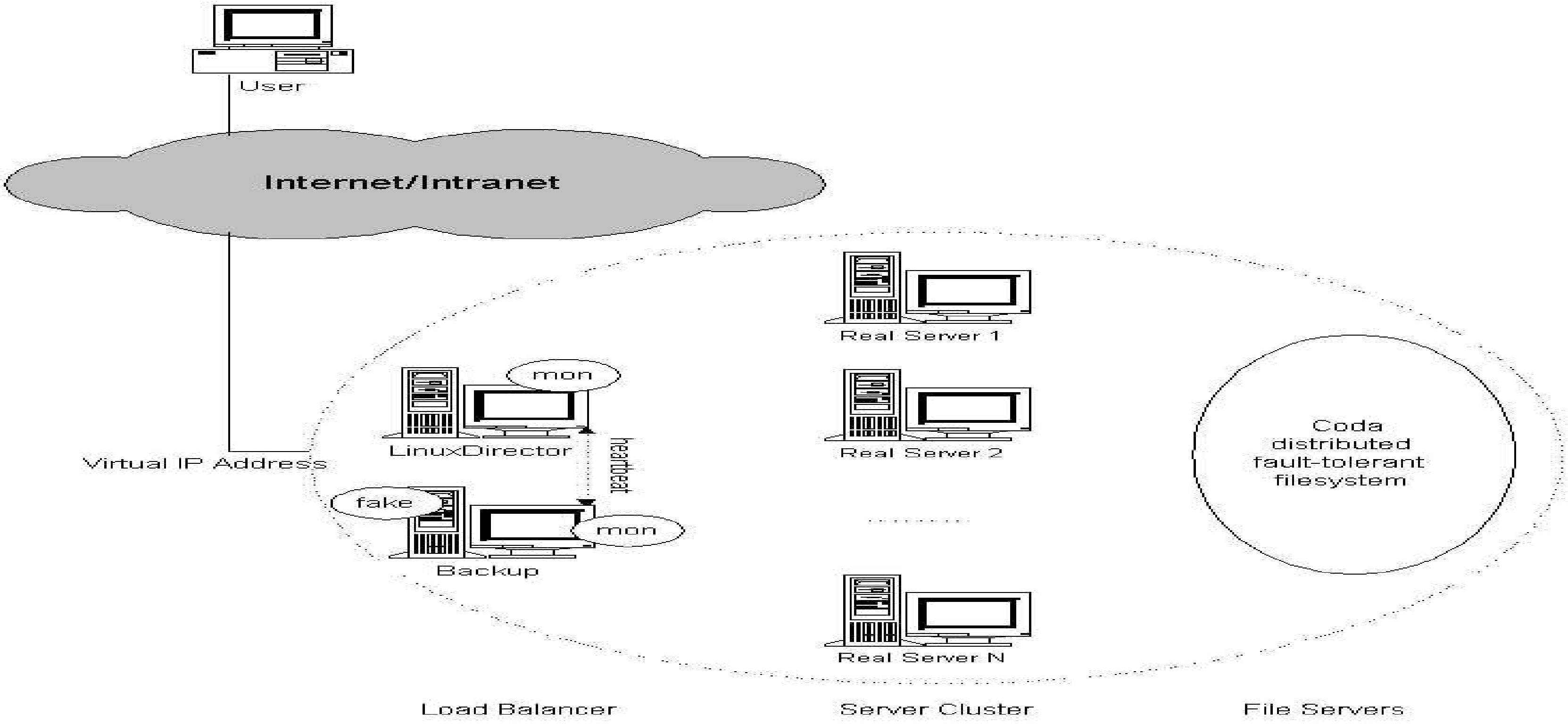
- Round Robin
- Wighted RR
- Least-Connection
- Weighted LC
- Locality-Based LC
- LBLC with Replication
- Destination Hashing
- Source hashing

LVS also provides support for persistency of connections.

High availability

- Real servers are added/removed to/from the pool by a monitoring daemon
- Load balancer can be replicated to allow fail over to another (backup) load balancer (spoofing master's IP address)
- Provides notification tools

Fail over solution



High Availability of Linux Virtual Server

Requirements

- Kernel 2.4.23+ or patched previous kernels (part of the 2.6) for load balancer
- In case of LVS-TUN and LVS-DR kernel patch for ARP-hidden for real servers
- Standard Apache listening on Virtual IP and real IP
- Clients configured for Virtual IP
- Mon (standard monitoring server) and Heartbeat
- Fake (IP spoofing)

Performance expectations

- I have tested 2 real servers + 1 load balancer (LVS TUN and DR), with fail over – all OK
- 20MB/s performance - below 10% CPU utilization on LoadBalancer
- 1Gbps+ performance for 100Mbit/s LB
- See http://www.linuxvirtualserver.org

Deployment

- Possible deployment for the SWREP web servers
- Together with head nodes (Apache Proxy or Squid Proxy) should provide stable, highly available and scalable solution with desired throughput
- Possible other deployments where high availability is required
- Used by linux.com, sourceforge.net,...
- Base for the RedHat's Piranha