

High Availability & Load Balancing Clusters



Mark Drago
Long Island Linux Users Group
December 9th, 2008

Heartbeat, LVS, Mon, MySQL, cssh, etc. on Linux



Mark Drago
Long Island Linux Users Group
December 9th, 2008

Making 3 Slow and Old Computers Act Like 1 Highly Available, Not Quite As Slow, But Still Old Computer



Mark Drago
Long Island Linux Users Group
December 9th, 2008

Outline

- Clustering in general
 - Types of clusters
 - High availability clusters
 - Load balancing
- Specific software
 - Heartbeat (Linux-HA)
 - Ldirectord / IPVS / LVS
- Demo

HPC Clusters

- HPC (High Performance Computing) clusters
 - Beowulf
 - MPI (OpenMPI)
 - Mosix (OpenMOSIX/LinuxPMI)
 - Grid computing (SETI@Home, Distributed.net, Folding@Home)
 - Not covered tonight – Jeff covered, ask Ilya

High Availability Clusters

- Goals:
 - Increase reliability
 - Survive failure of components
- Strategy:
 - Duplicate as much infrastructure as possible
 - Remove single-points-of-failure
 - Detect failures
 - Initiate failover to work around failure
- Software:
 - Generic: Heartbeat
 - Specialized: Samba, DHCPD

Load Balancing / Load Sharing

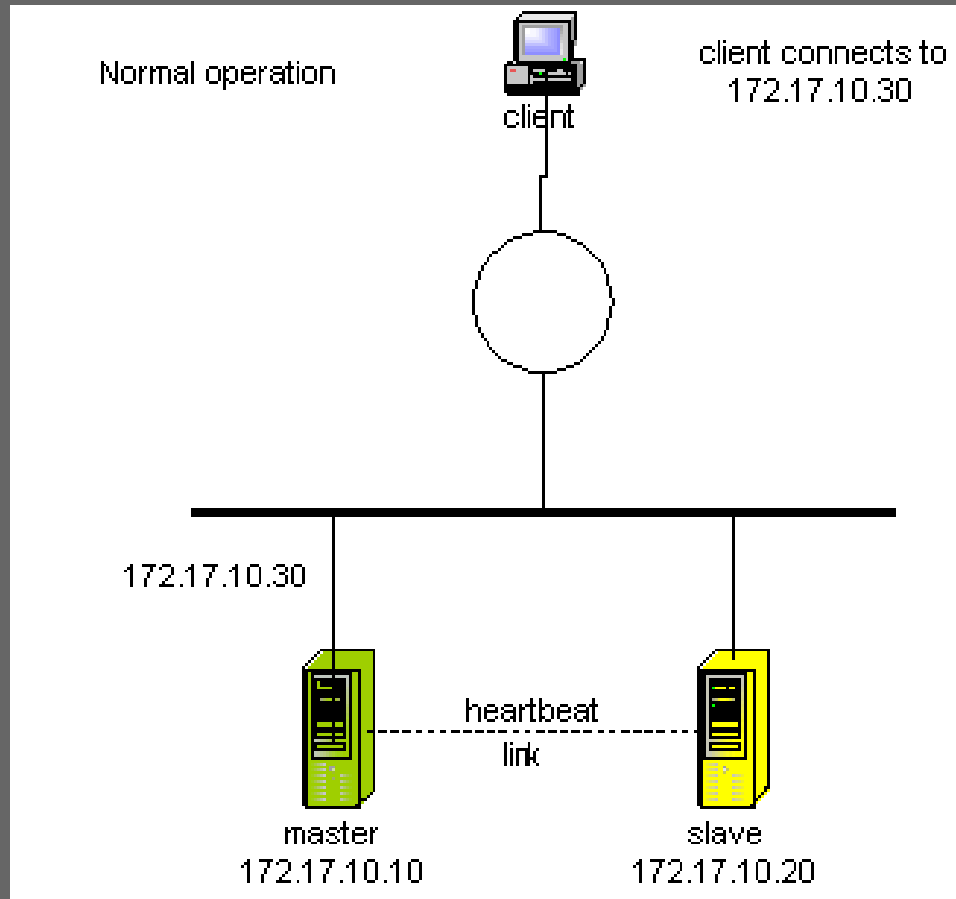
- Goals:
 - Increase speed (latency & throughput)
 - Meet larger demand
- Strategy:
 - Split load across many servers without changing the user-experience
- Software:
 - Generic: Ldirectord/IPVS/LVS
 - Specialized: Apache, Squid

Heartbeat

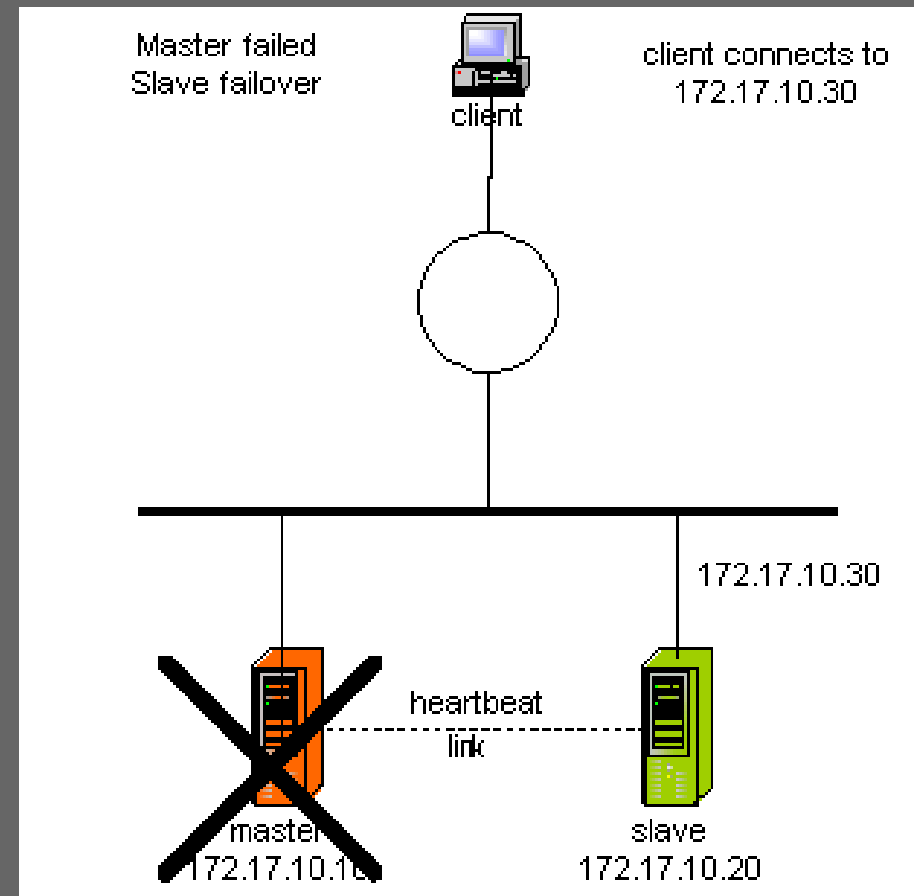
- Remote control for 'init'
 - init(8) is responsible for starting & stopping processes
 - Heartbeat starts & stops processes on networked machines
- HB monitors machines to make sure they are up and functioning properly
- Once a machine/service is down it follows rules to start those services on other machines
- Their tagline: “Add a 9 to your uptime percentage.”

Heartbeat

Normal



After Failure of Primary Node



Heartbeat

- Define nodes, resources, resource rules
- Nodes are self-explanatory
- Resources:
 - Any resource that can be moved b/w nodes
 - IP address (probably most common)
 - Anything in /etc/init.d
 - Mount remote storage
- Resource Rules:
 - Define preferred nodes for resources
 - Define preference for inter-resource location
- Main configuration is annoyingly XML
- [example configuration]

Heartbeat Split Brain

- Split Brain – Cluster is split, one part can not communicate with the other, but they're both up
- At this point you essentially have 2 smaller subclusters
- You can't distinguish downed nodes from nodes that are incommunicado
- Dunn's Law: “What you don't know, you don't know – and you can't make it up.”

Heartbeat Split Brain (Fencing)

- Fencing – If you can't figure out the answer to a question, force an answer to be correct.
 - Resource fencing
 - Block access to SAN, etc.
 - Create firewall rules to block off node
 - Node fencing – STONITH
- Ensure that errant node does not have access to shared resources
- Do not rely on errant node to give up access

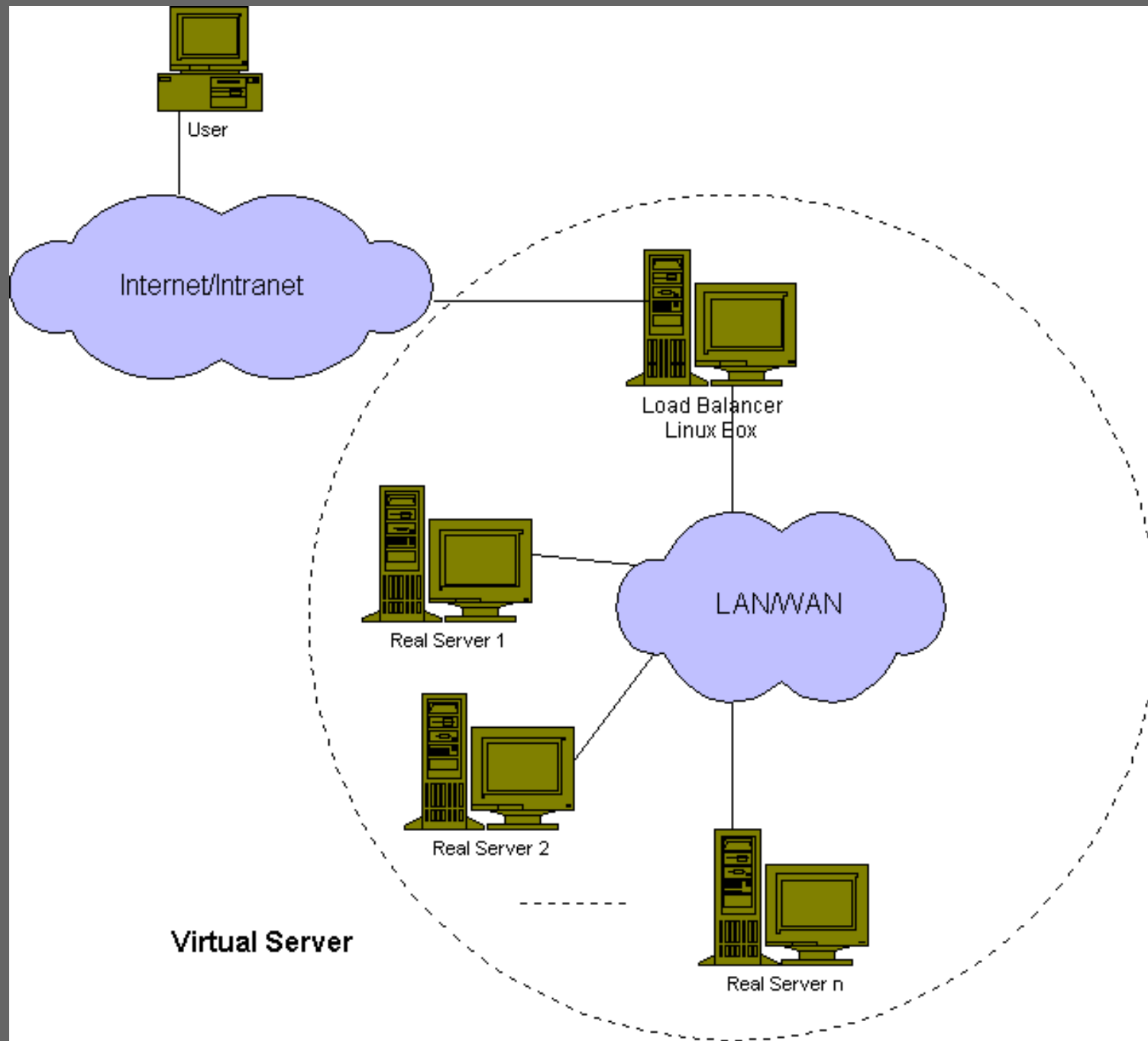
Heartbeat Split Brain (Quorum)

- But, what if the errant node is really up and thinks that the rest of the cluster is errant?
- We need a way to pick a winning subcluster
- Quorum:
 - SCSI Reserve (Quorum Disk)
 - Both nodes try and reserve a disk partition and only one can succeed.
 - Will not work over geographic distances
 - Quorumd
 - Third-party software that decides which subcluster wins
 - Will work over geographic distances

Linux Virtual Server (LVS)

- Layer-4 switch
- Splits TCP connections up amongst a number of real servers
- All packets from a given connection are sent to the same real server
- Components:
 - IPVS – IP Virtual Server, lives in kernel
 - Ldirectord can monitor real servers, can change weight to exclude real servers if they are down
 - ipvsadm – control / monitor real & virt servers
- [example configuration]

Linux Virtual Server (LVS)

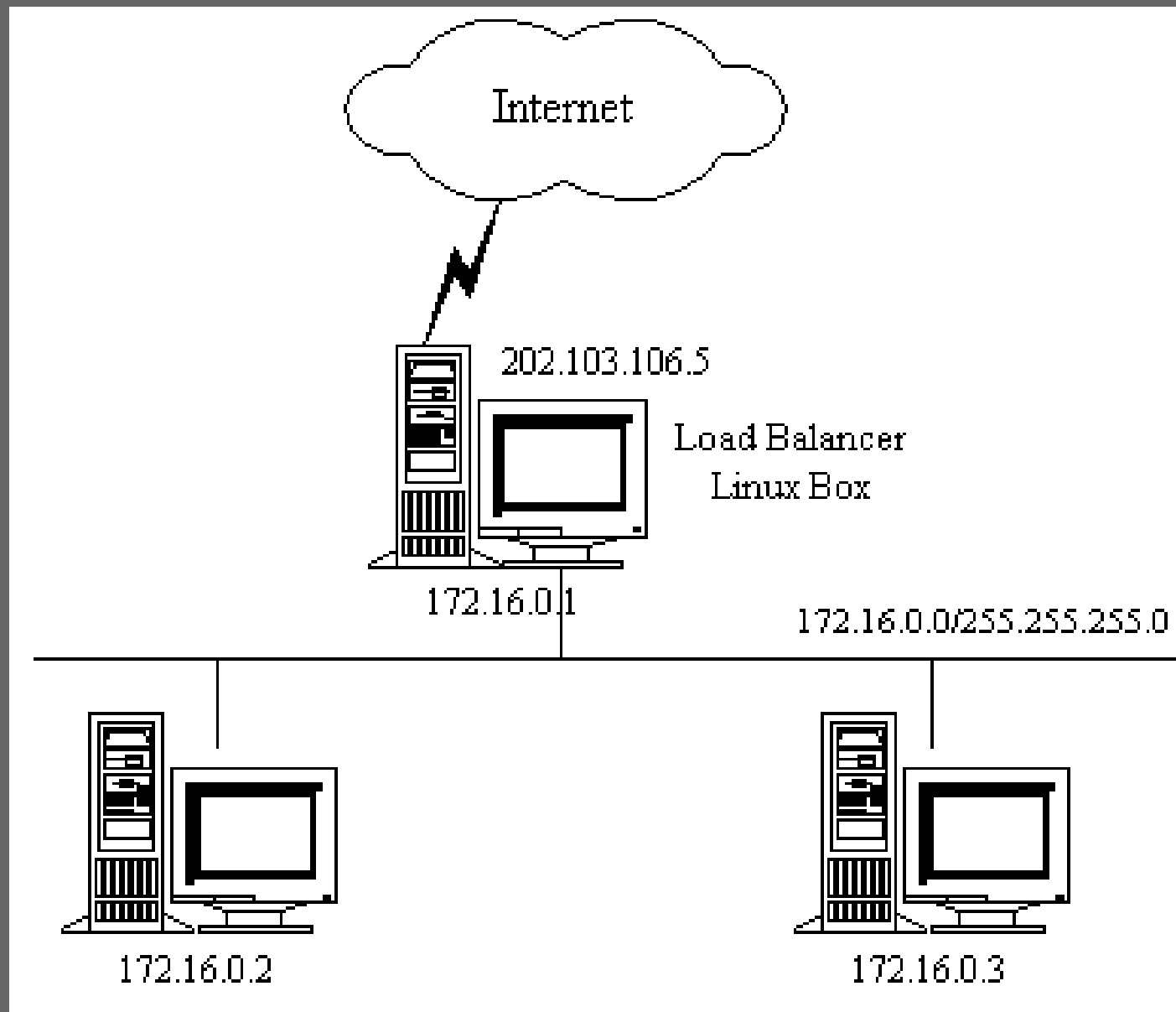


Linux Virtual Server (LVS)

- Of course it's not that simple
- There are 3 ways to connect virtual servers with real servers (VS/NAT, VS/Tun, VS/DR)
- VS/NAT – Virtual server NATs packets to the IP of the chosen real server
- VS/Tun – Virtual server sends packets to real server through an IPIP tunnel
- VS/DR – (Direct Routing) Virtual server simply changes the destination MAC address on the packet and sends it along

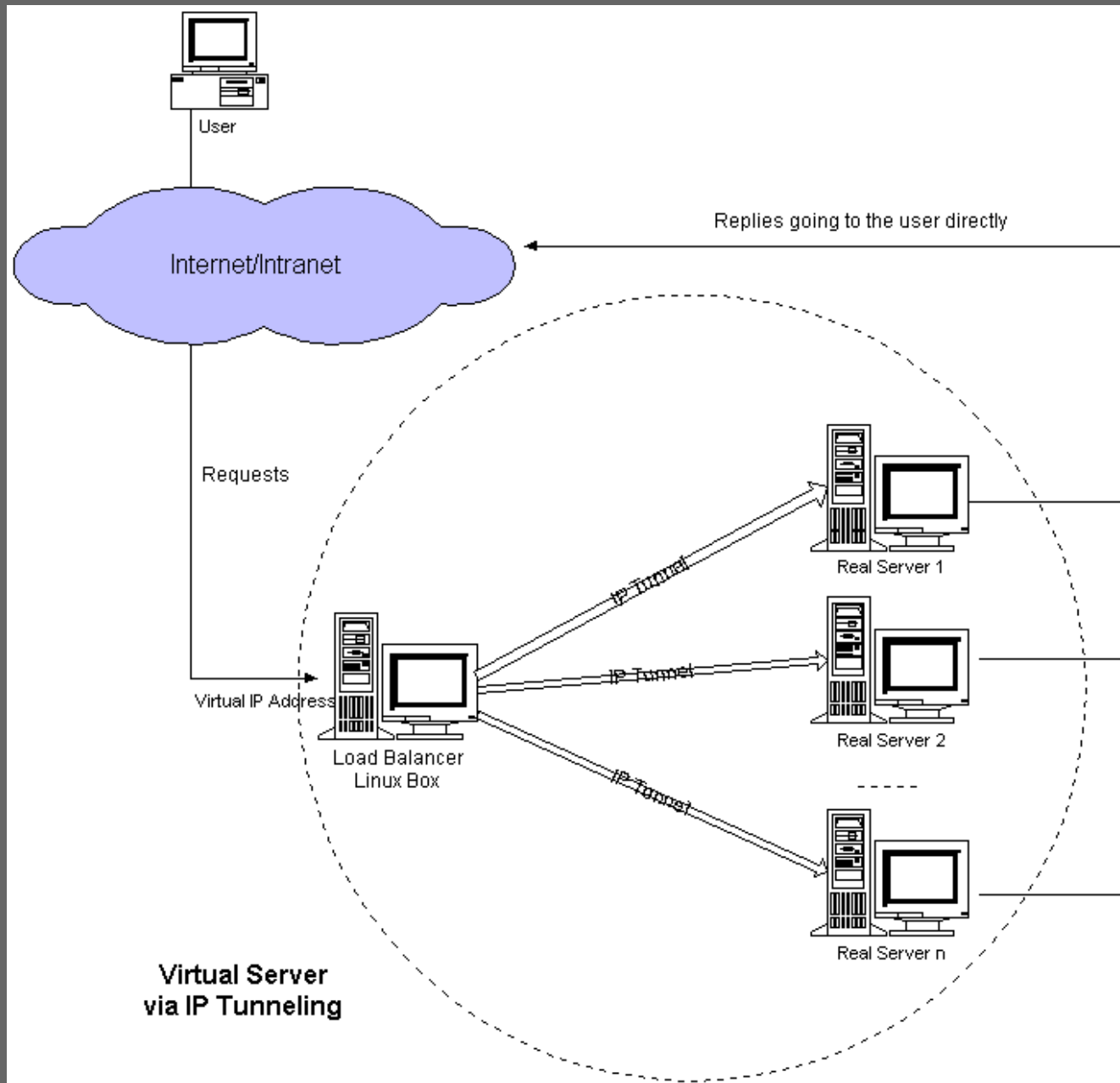
Linux Virtual Server (LVS)

VS/NAT



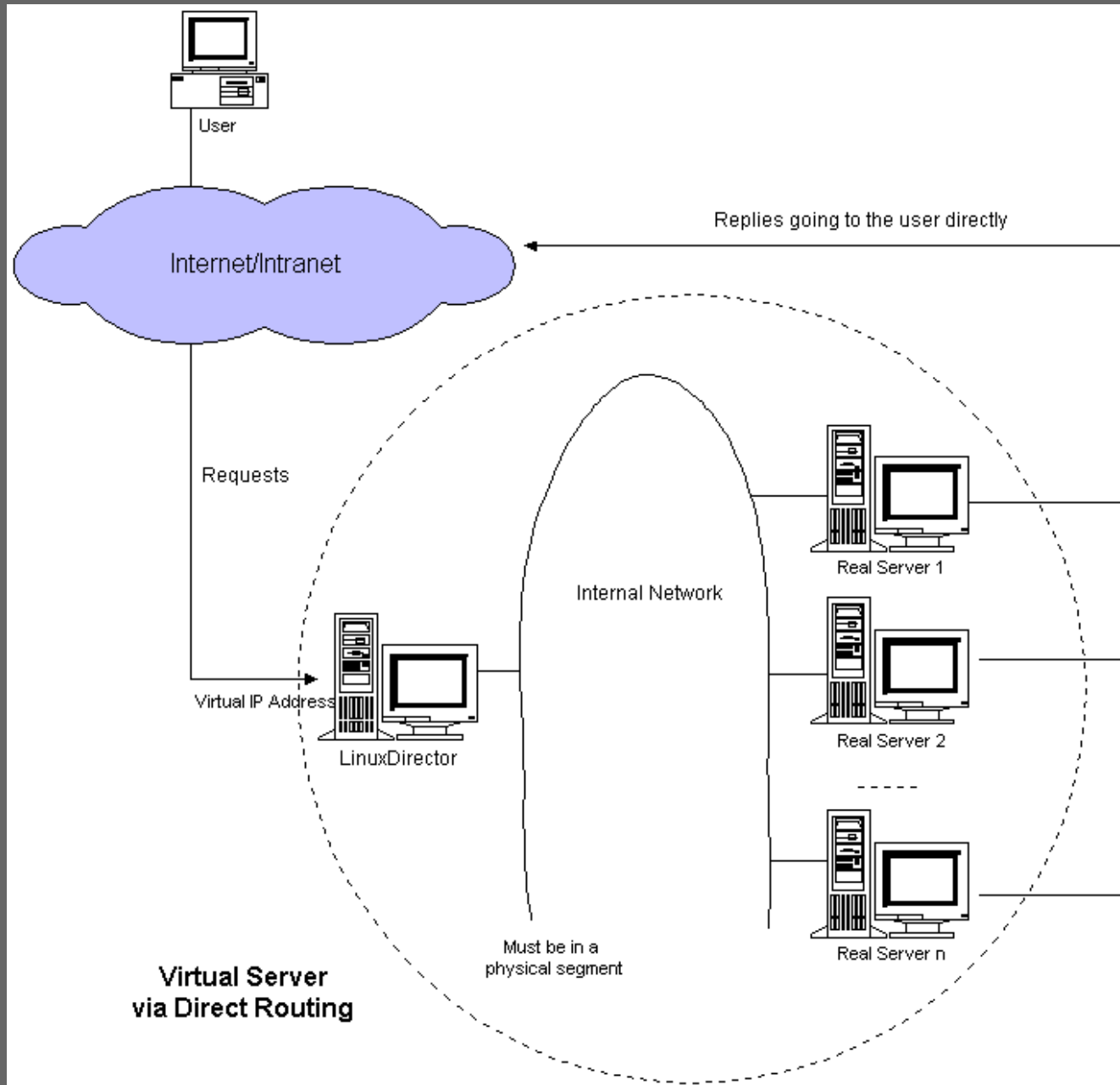
Linux Virtual Server (LVS)

VS/Tun



Linux Virtual Server (LVS)

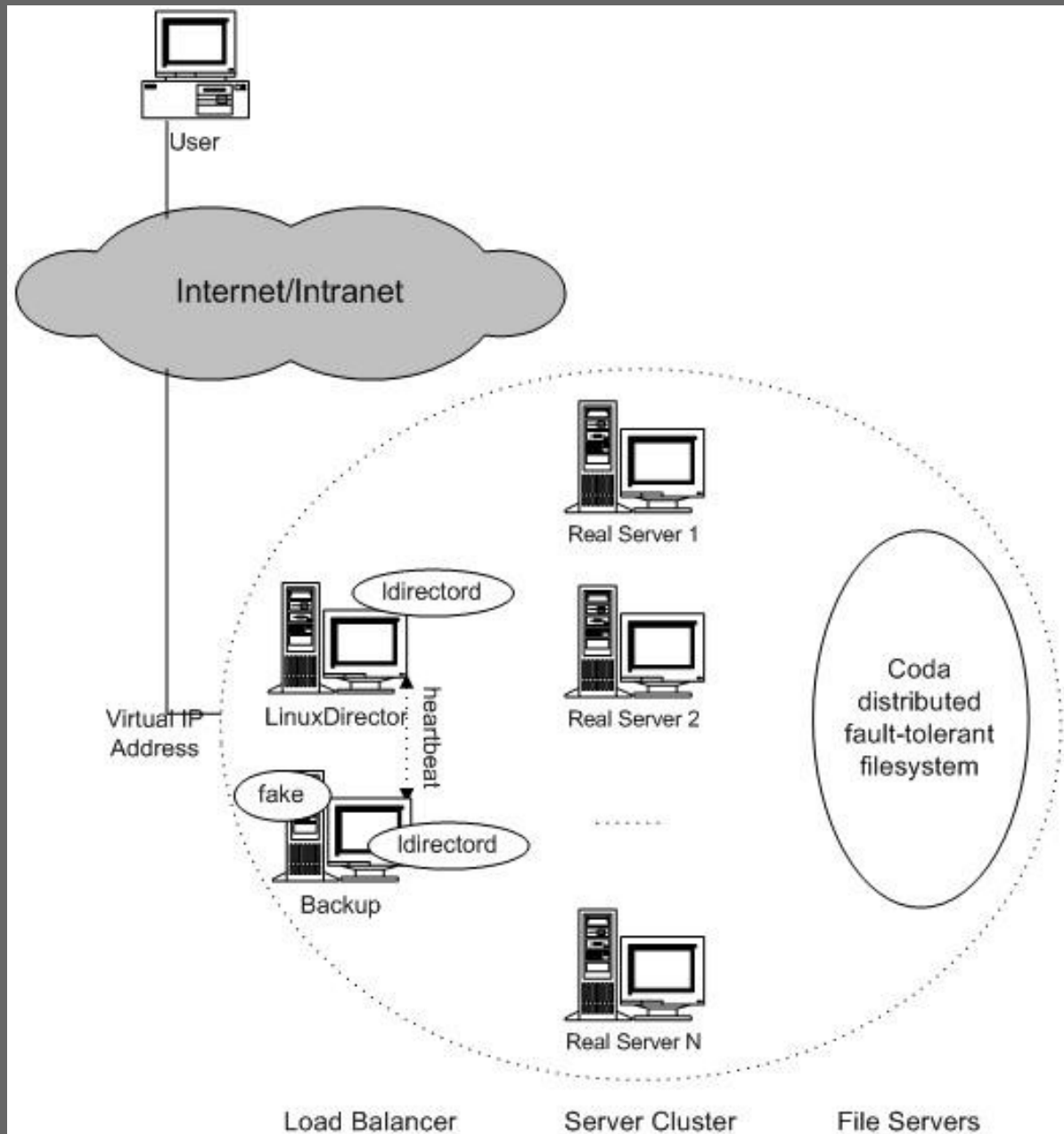
VS/DR



VS/Tun and VS/DR ARP Problem

- VS/Tun and VS/DR require the real server to have the virtual server IP configured somewhere or the real servers won't pickup the packet
- But, that would create an IP conflict, thus, the ARP problem
- ...brief explanation of ARP...
- So, we must configure the real servers to handle ARP in a special way
- ignore arp requests for IPs on lo or eth0:x
 - `net.ipv4.conf.{all,eth0}.arp_ignore = 1`
- use main IP (eth0) for outbound arp requests
 - `net.ipv4.conf.{all,eth0}.arp_announce = 2`

LVS + Heartbeat



High Availability of Linux Virtual Server

Demo

Thanks & Credits

- Thanks
 - LILUG
 - Bascom
 - My lovely and understanding wife Jen
- Credits
 - www.linux-ha.org
 - Quorum & Fencing: tinyurl.com/4ss35d
 - www.linuxvirtualsever.org
 - LVS mini-mini-howto: tinyurl.com/6cpw2a
 - www.phorum.org
 - www.mysql.org
 - www.wikipedia.org