



# High Availability Services

Arjan Dekker en Remco Hobo

06-07-2005

# Inhoud in 20 seconden

- Uitleg High Availability
- Proprietary en Open source systemen
- Ons project
- Conclusie

# Wat is HA?

- High Availability
- Makkelijker beheer(s)baar door redundantie  
-Ieder systeem is meervoudig uitgevoerd
- Meer points of failure, minder/geen single points of failure
- Het geheel is transparant en zal zich als 1 systeem gedragen
- High Availability kan op vele lagen voorkomen

# Aandachtspunten HA

- Kosten
- Sessies
- Encryptie
- Meer componenten, dus meer problemen?
- Installatie / Configuratie
- Flexibiliteit
- Services

# Methoden

- DNS
- ARP-spoofing
- HA-cluster

# DNS

- Meerdere IP-adressen per DNS-naam
- Round-Robin
- Gevaren
- Onderhoud
- Geen echte HA-oplossing

# ARP-spoofing

- Twee identieke systemen
- Heartbeat
- Fake
- Gratuitous ARP Broadcast

# ARP-spoofing 2

- Goedkoop / Eenvoudig
- Tijdelijke downtime
- Connecties verdwenen
- Niet flexibel
- Management



# HA-Cluster

- Load balancer
- Twee of meer servers
- ARP-spoofing bruikbaar voor Load balancers

# HA-Cluster 2

## Load Balancer

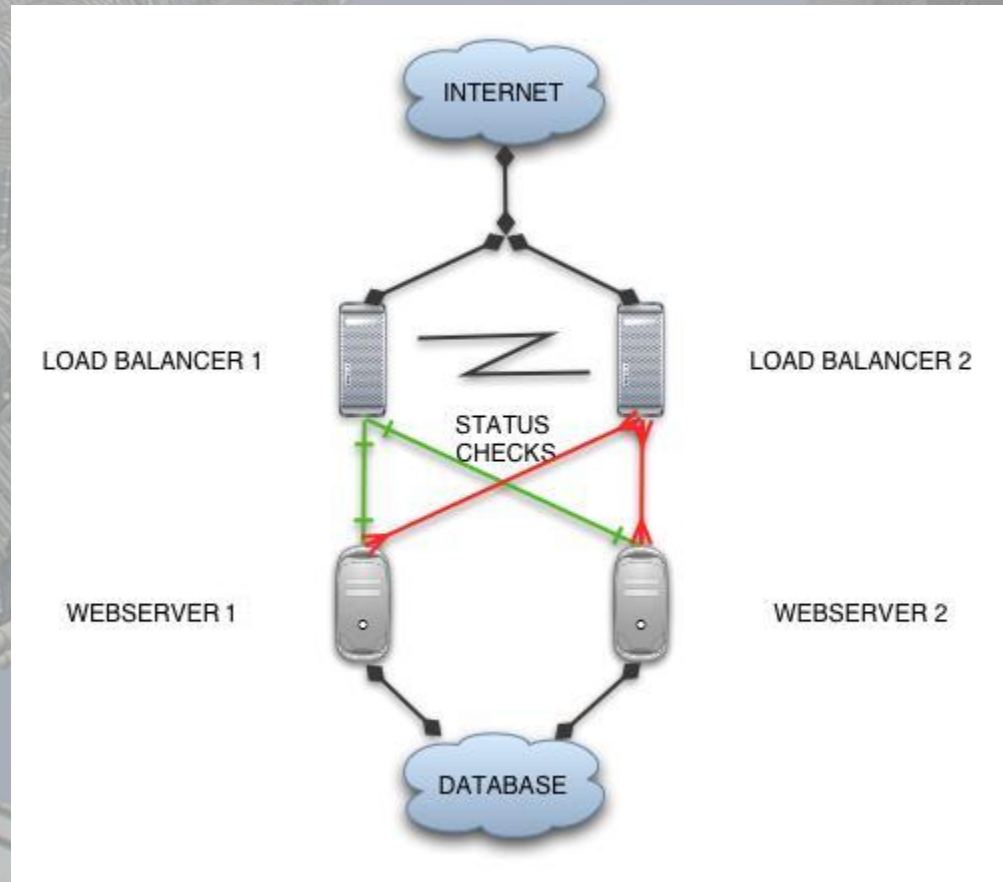
- Verschillende algoritmes
- Drie manieren van packet forwarding
  - NAT
  - Directe routing
  - IP-IP encapsulatie

# HA-Cluster 3

## Soorten HA-clusters

- High Availability IP Cluster
  - IP niveau
  - Ping
- High Availability Application Cluster
  - Applicatie niveau
  - Ldirectord
  - “Out of Order”-pagina

# HA-Cluster 4 - HA IP + Application



# HA-Cluster 5

- Duurder
- Sessie continuïteit afhankelijk van service
- Servers eenvoudig te upgraden
- Zeer flexibel
- Stabiël en robuust
- Hogere performance haalbaar

# Proprietary en Open source systemen

Voordelen proprietary:

- Out-of-the-box oplossingen
- Vaak te configureren met 1 script
- Geen harde schijf/onnodige componenten
- Support!

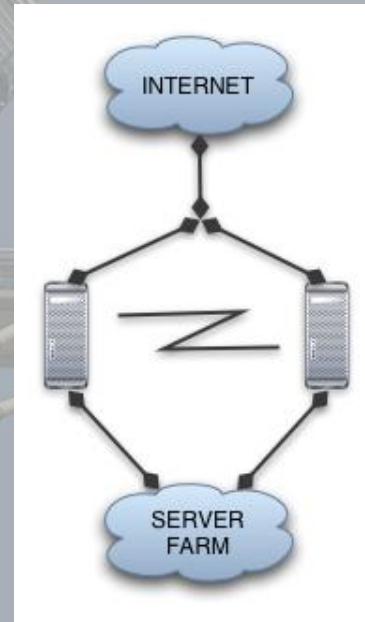
# Proprietary en Open source systemen

## Voordelen Open Source:

- Aanpasbaarheid door de open source
- Werkt op vele soorten hardware
- Community
- Hardware kan weer worden ingezet voor andere doeleinden
- Kosten(?)

# Het High Availability Project

Setup 1: Twee load balancers die 1 gedeeld IP gebruiken:  
HA-IP

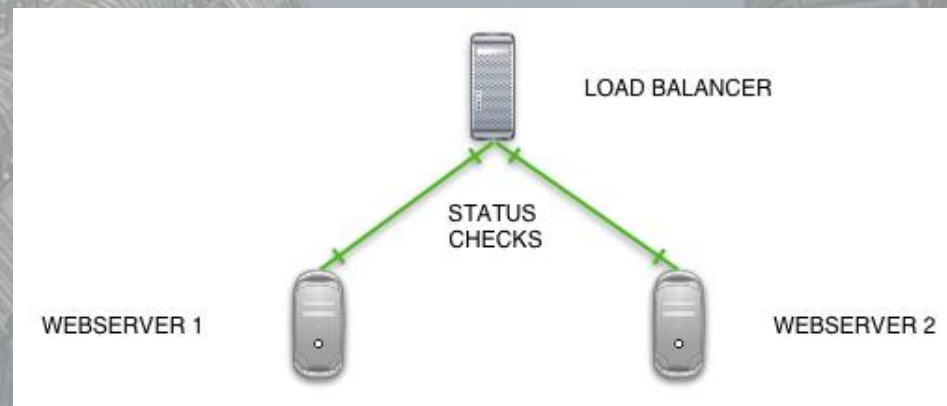


-2xDebian Sarge met Heartbeat



# Het High Availability Project

Setup 2: Load balancing over meerdere servers, van buitenaf ziet dit er als 1 server uit (HA-Application):



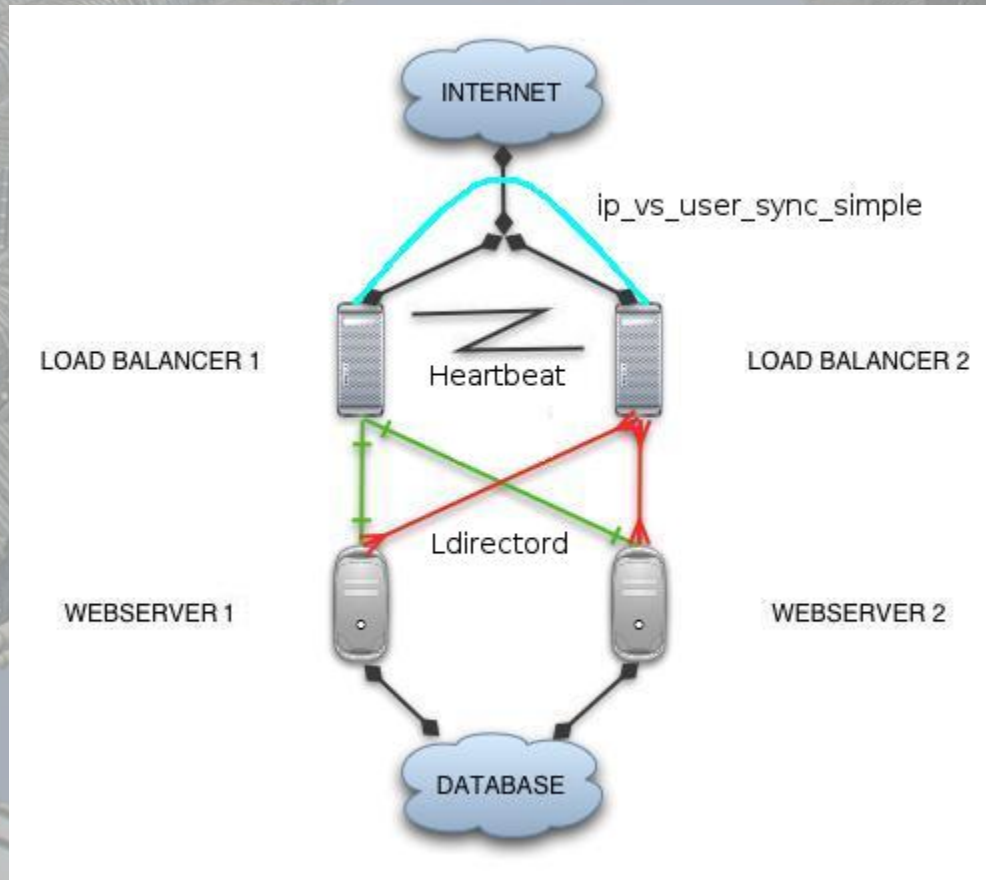
-2xDebian Sarge met Ldirectord (Onderdeel van UltraMonkey)

# Het High Availability Project

Setup 3: Loadbalancing over meerdere servers met meerdere load balancers, van buitenaf ziet dit er als 1 server uit. Deze setup bestaat uit:

- 2x Debian Sarge als Load Balancers
- 2x Debian Sarge als Webservers
- UltraMonkey:
  - Heartbeat
  - Ldirectord
  - ip\_vs\_user\_sync\_simple

# Het High Availability Project



# Bevindingen

- Incomplete documentatie
- ip\_vs\_user\_sync\_simple is work in progress: Geen documentatie
- Geen IPv6 ondersteuning (er is 1 verlaten project)
- Heartbeat controleert zijn eigen state niet
- Het systeem draait wel stabiel

# Conclusie

- HA is niet compatible met het end-to-end principe
- UltraMonkey maakt beheer van een service veel makkelijker
- Alleen bruikbaar met uitgebreide \*nix kennis.
- Zonder IPv6 is de toekomst somber

# Aanbevelingen

- UltraMonkey is te gebruiken onder IPv4, het wordt al gebruikt door onder andere:
  - real.com
  - SourceForge
- We hebben geen harde cijfers over UltraMonkey's stabiliteit kunnen vinden
- Voordat UltraMonkey live kan worden ingezet, zou er eerst een duurtest moeten worden gedaan om de stabiliteit te kunnen waarborgen.

# Vragen?

?