

# Budowanie tanich, wysoko wydajnych i wysoko dostępnych systemów pod Linuksem

**Mariusz Drożdziel**  
Październik 2009



# Klasyfikacja klastrów

- klastry obliczeniowe (compute cluster)
  - „Beowulf”
  - grid / rozproszone (distributed)
- klastry usługowe
  - wysoka dostępność / high-availability
  - wysoka wydajność / load balancing



# Cele budowania klastrów

## Klastry HA:

- unikanie SPOF
- ułatwienie zmian konfiguracji
- backup
- klimatyzacja :-)
- minimalizacja ilości okien serwisowych

## Klastry LB:

- zwiększenie wydajności, skalowanie w poziomie
- zapewnienie HA wykorzystując sprzęt



# SPOF

Single Point of Failure – Pojedynczy Punkt Awarii

- połączenia heartbeat (split brain)
- interfejsy IP
- storage
- sieć LAN (STP, OSPF)
- autorskie aplikacje



# HA / LB na poziomie aplikacji

Przykłady aplikacji, których klastrowanie na niższej warstwie bywa dyskusyjne:

- DNS
- SMTP
- DHCP
- SQL



# Load Balancing / Sharing z DNS

- + przejrzystość rozwiązania
- + prosta i nieskomplikowana konfiguracja
- brak odporności na awarie węzłów
- zawodność przy odpytaniach z dużych sieci
- utrudnienia zmian w konfiguracji

Obecnie stosowane głównie w połączeniu z innymi rozwiązaniami na innych warstwach.



# Virtual Router Redundancy Protocol

Zaprojektowany w celu przenoszenia adresów IP

- + łatwa konfiguracja
- + proste zasady działania i implementacja
- inne przeznaczenie



# Virtual Router Redundancy Protocol

Istniejące implementacje:

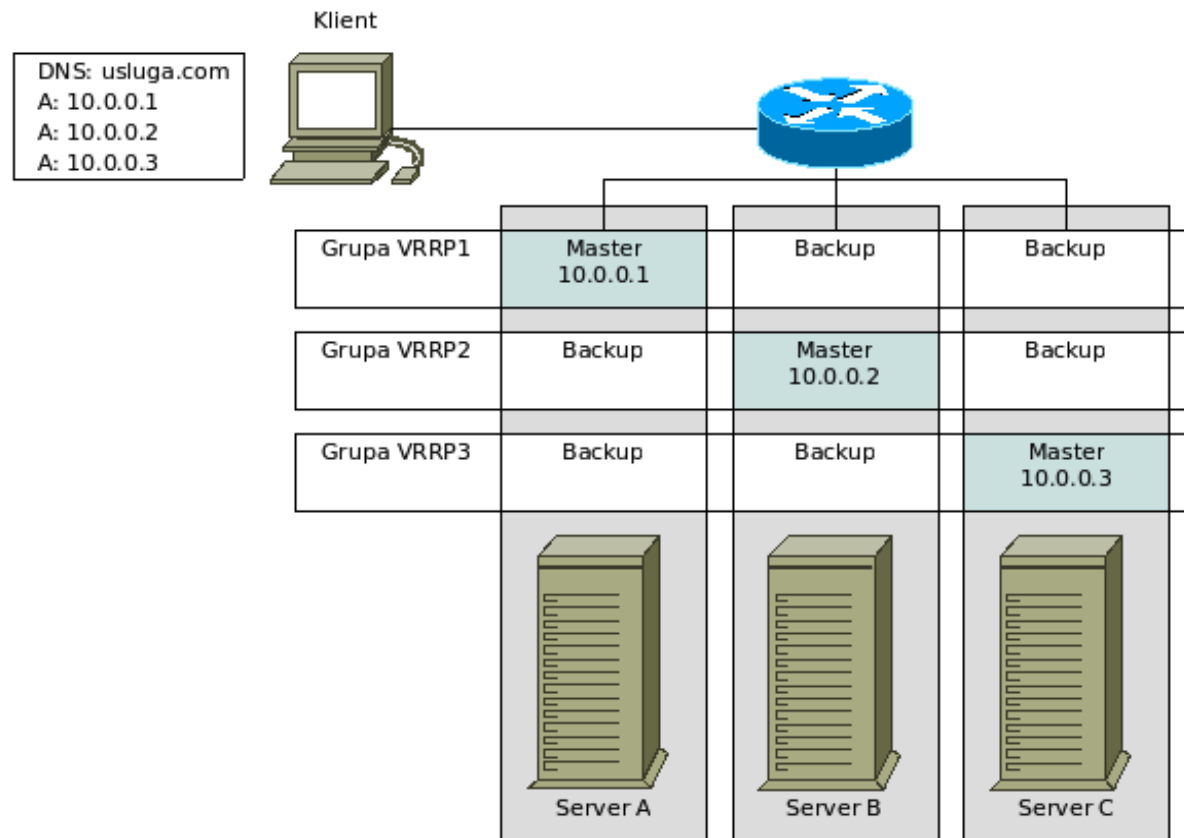
- vrrpd
- Keepalived

Funkcjonalne odpowiedniki:

- ucarp
- heartbeat + crm



# Virtual Router Redundancy Protocol



Konfiguracja Load Sharing via DNS  
i przenoszenie adresów IP przez VRRP



# Linux Virtual Server

Load Balancing protokołów TCP, UDP, AH, ESP (lub dowolny payload IP na podstawie fwmark)

- rozwiązanie rozwijane od ponad 10 lat, stabilne i przetestowane,
- łatwa konfiguracja
- wydajność



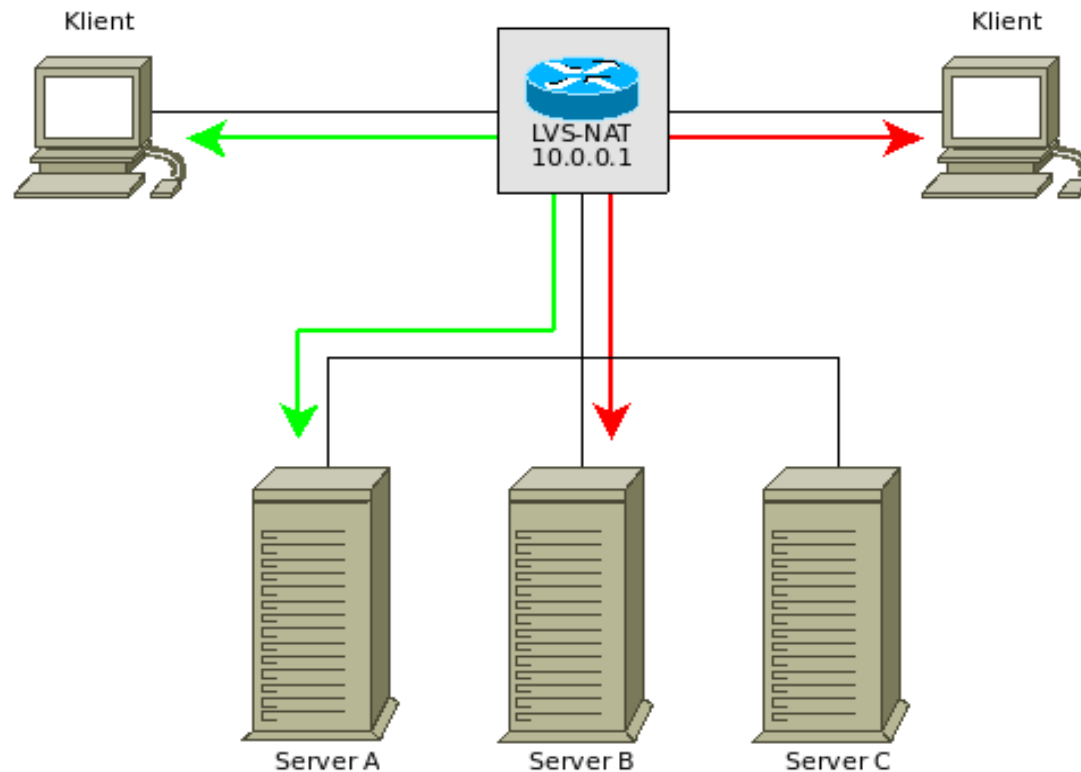
# Linux Virtual Server LVS-NAT

## LVS-NAT

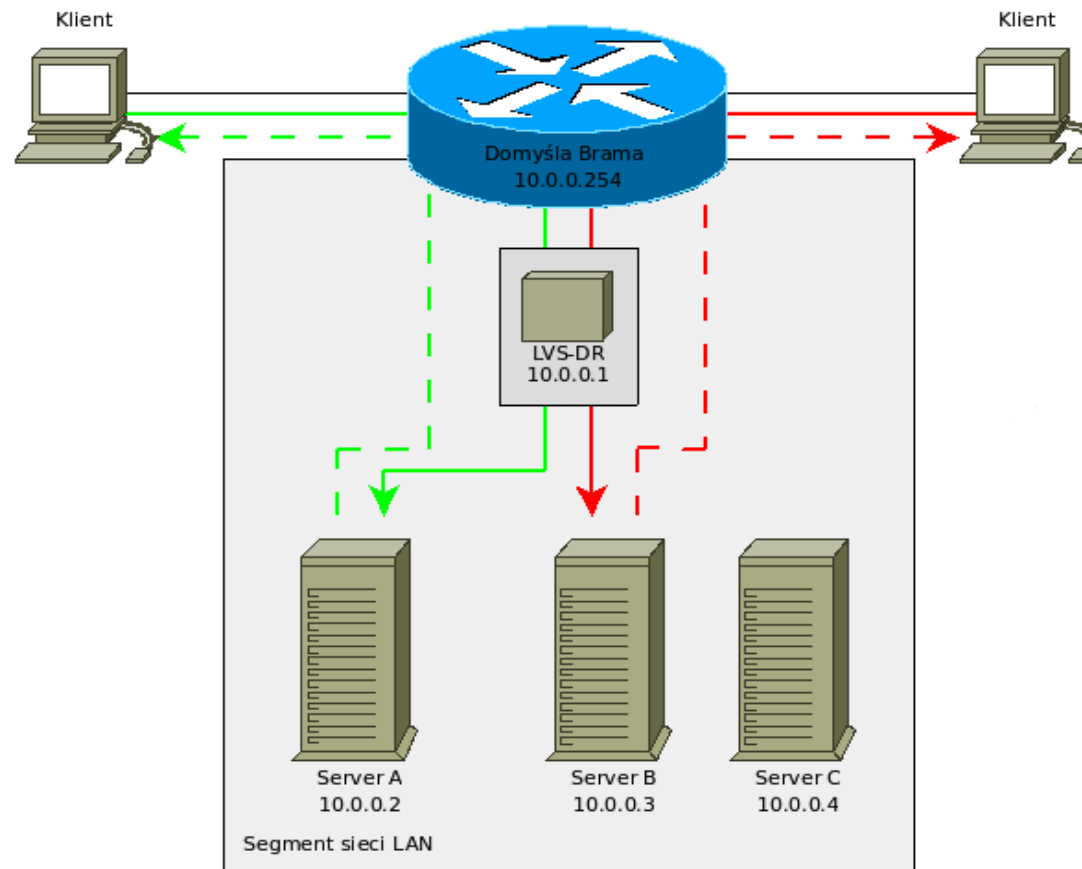
- proste zasady działania
- łatwe rozwiązywanie problemów
- większe zapotrzebowanie na CPU i większe opóźnienia



# Linux Virtual Server LVS-NAT



# Linux Virtual Server LVS-DR



# Linux Virtual Server

Brak funkcjonalności zapewniającej redundancję / high-availability.

Brak natywnej metody redundancji Directorów LVS.

Konieczność zastosowania wraz z dodatkowymi rozwiązaniami rozszerzającymi funkcjonalność.



# Keepalived

Głównym celem projektu jest rozszerzenie funkcjonalności LVS.

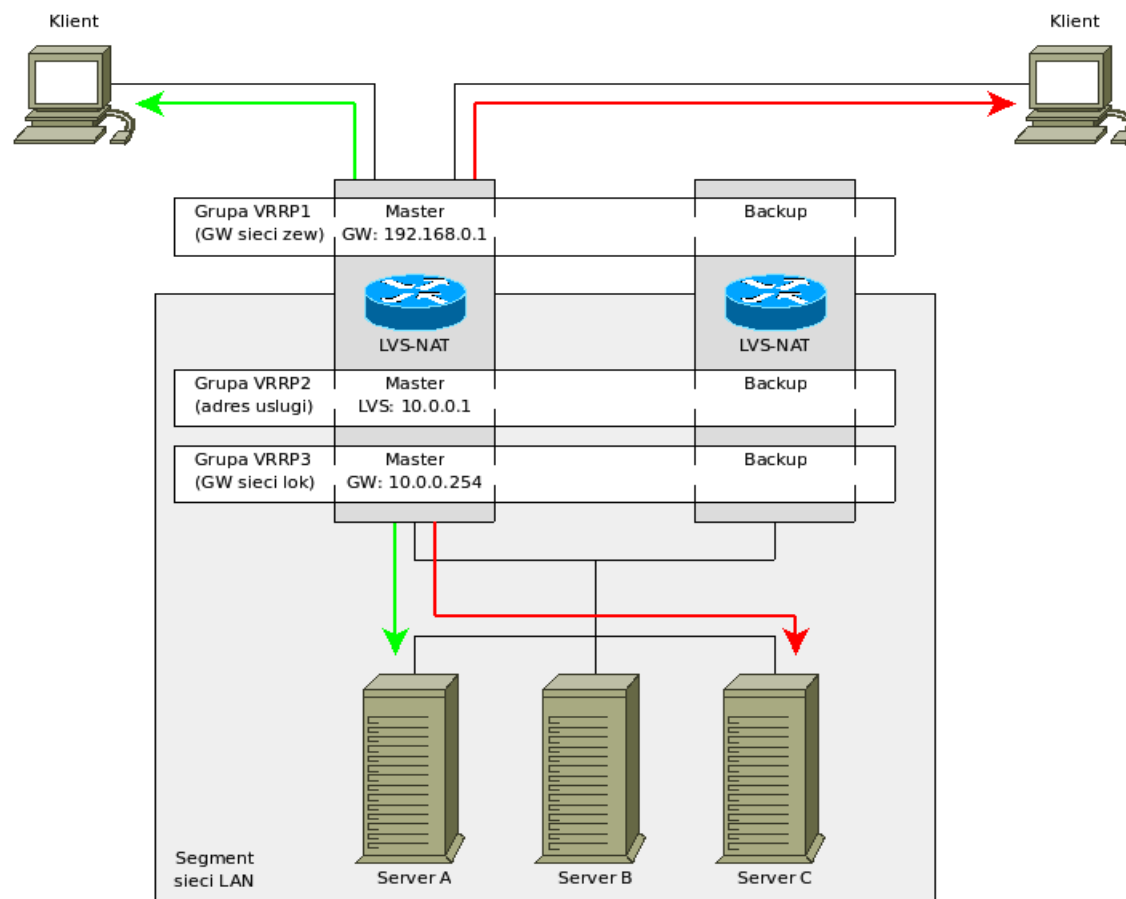
- sprawdzanie dostępności usług na real-servers
- redundancja directorów za pomocą VRRP
- dodawanie tras, skrypty, sorry\_server

Problemy:

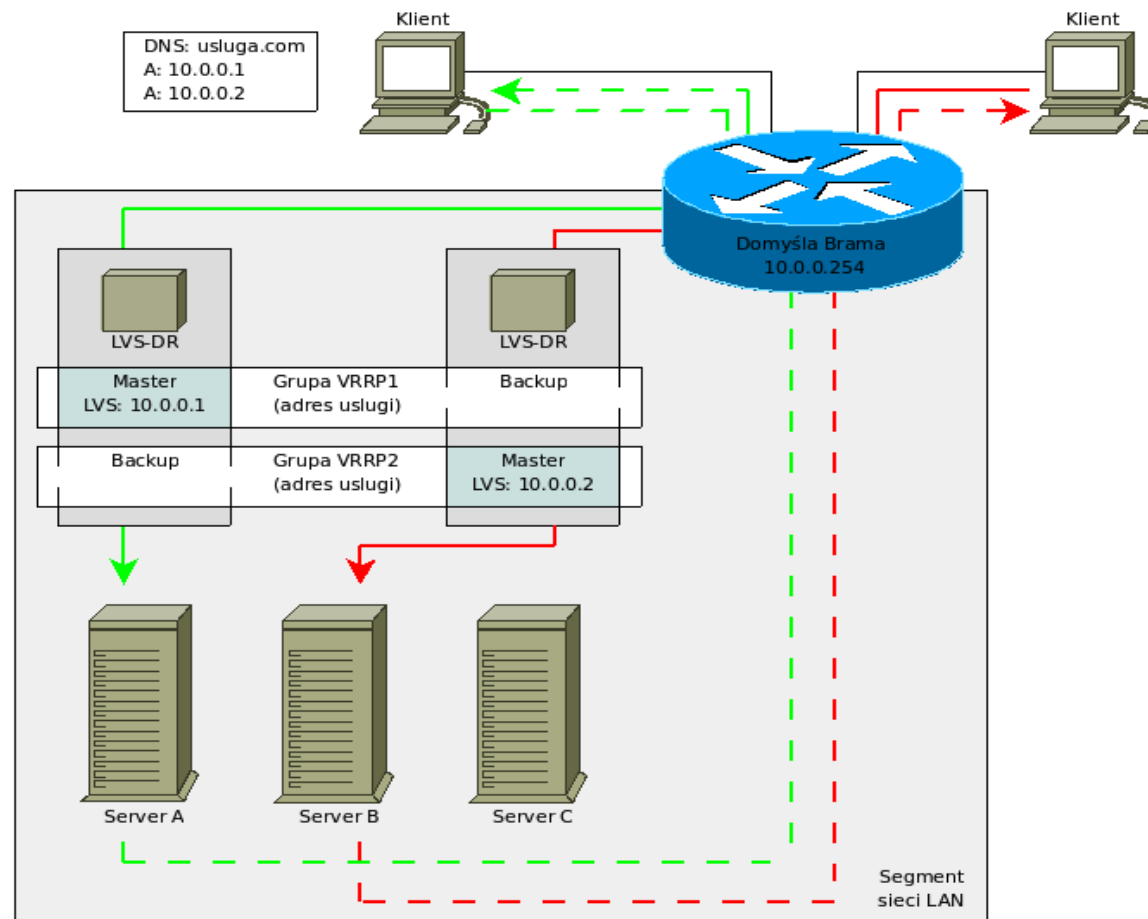
- rozwój aplikacji
- poprawianie błędów
- zmiana konfiguracji i problemy z reloadem
- „alpha-omega” patch



# Keepalived + LVS Active-Passive



# Keepalived + LVS Active-Active



# Linux-HA Heartbeat

Kompletna infrastruktura do zarządzania zestawem usług pracujących na wielu węzłach w klastrze.

- zarządzanie infrastrukturą węzłów w klastrze
- komunikacja między węzłami
- wymienianie informacji o stanie procesów i usług
- przełączanie aktywnych usług
- odcinanie zasobów



# Fencing / STONITH

Odgradzanie krytycznych współdzielonych zasobów od źle funkcjonujących węzłów klastra. Zapobieganie utracie / uszkodzeniu danych.

STONITH - shoot the other node / machine in the head

Metody fencingu:

- power fencing
- LUN fabric fencing
- ILO / DRAC (HP / DELL)
- VM fencing
- software fencing



# Współdzielenie danych

Zunifikowany interfejs dostępu do danych z zewnętrznymi źródłami (SQL)

Współdzielenie urządzeń blokowych:

- SAN
- iSCSI
- NFS
- DRBD



# Urządzenia blokowe

Metody dostępu do danych na współdzielonych urządzeniach blokowych.

Wyłączny dostęp dla jednego węzła (HA):

- Filesystem bezpośrednio na urządzeniu
- CLVM

Wiele urządzeń posiada dostęp jednocześnie (LB):

- GFS / GFS2
- OCFS / OCFS2



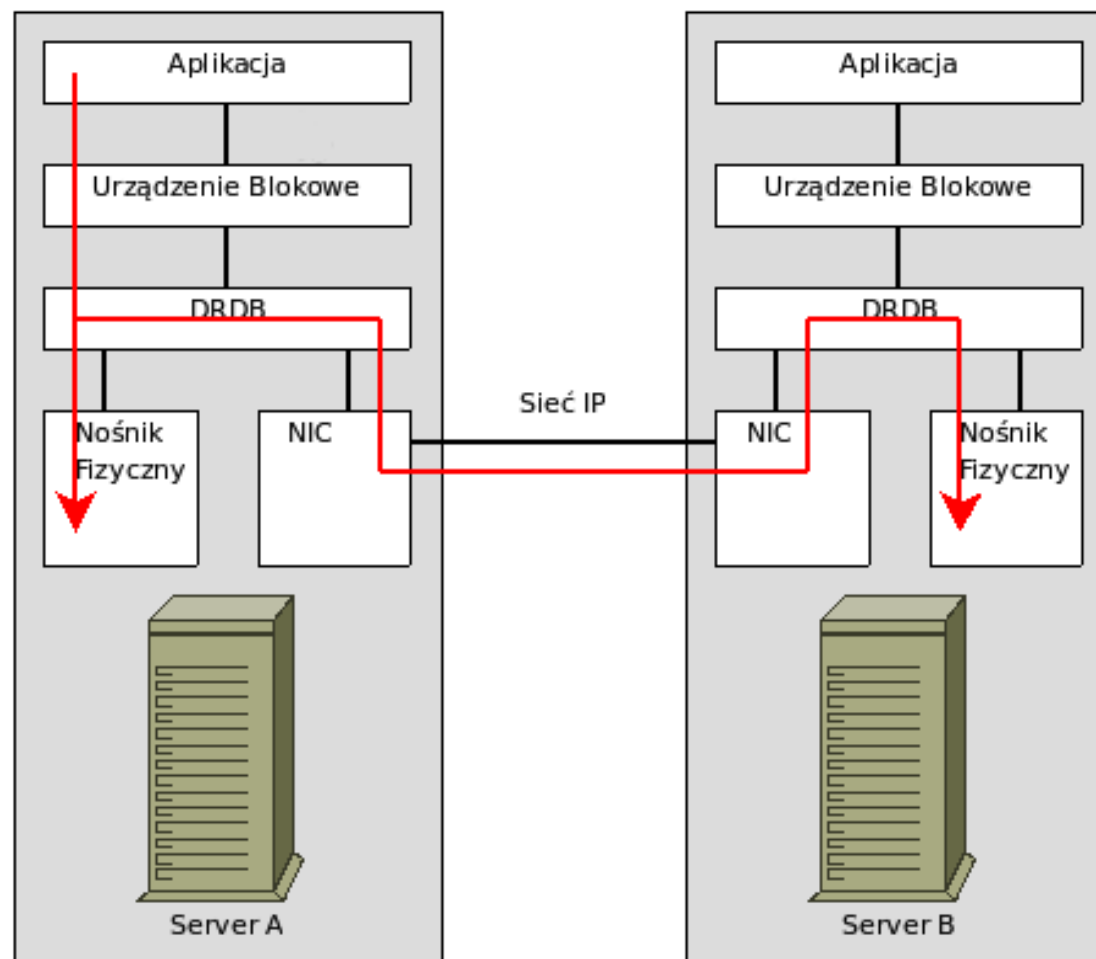
# DRBD

DRBD - Distributed Replicated Block Device – RAID1  
przez sieć IP

- tryby pracy active/active i active/passive
- praca synchroniczna i asynchroniczna
- nadchodzące włączenie do oficjalnej gałęzi kernela
- łatwa konfiguracja



# DRBD



# Rodzaje klastrów

## Active-Active, N+N:

- problem z konfiguracją usług bez quorum
- konieczność posiadania dodatkowych zasobów

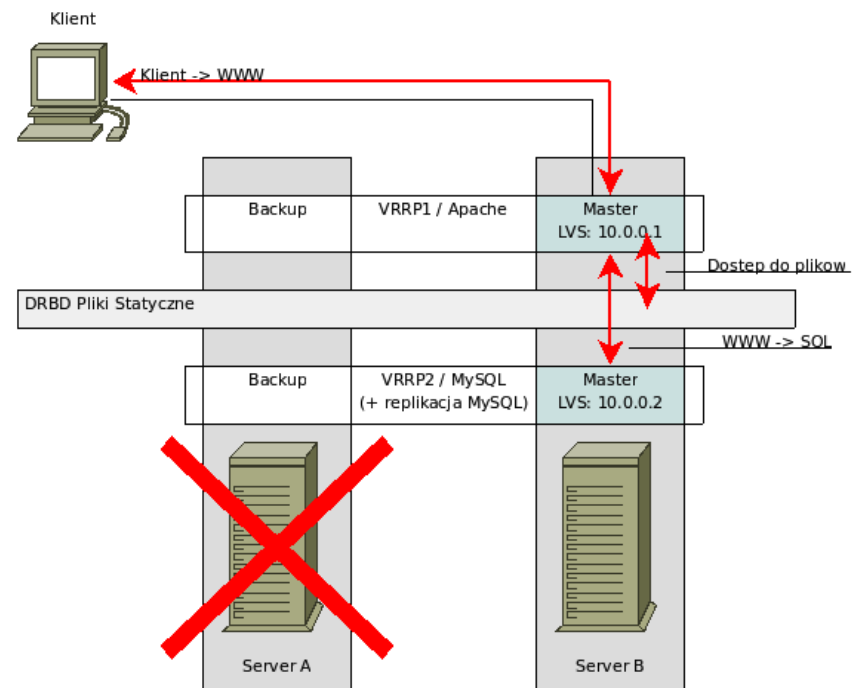
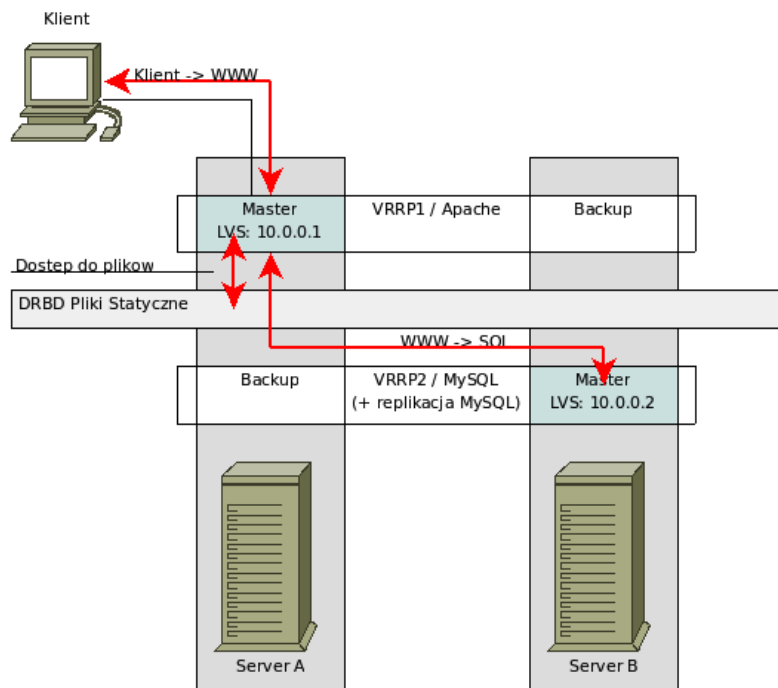
## Active-Passive i N+1:

- „marnowanie” zasobów
- konieczność testowania funkcjonalności klastra
- brak konsystencji konfiguracji

Problem z modyfikacją i konfiguracją oprogramowania zapewniającego usługi warstwy klastra.



# Przykład konfiguracji



# Wirtualizacja

Xen / KVM

Łatwiejsze zarządzanie i migrowanie usług poprzez stworzenie oddzielnych, zunifikowanych warstw dla poszczególnych usług.

Awaryjność poszczególnych elementów ma mniejszy wpływ na inne komponenty klastrów.

Nowa warstwa komplikuje konfigurację i wprowadza kolejny punkt awarii.



# Inne oprogramowanie

- Piranha – LVS + HA + Healthcheck + GUI
- GNBD - Global Network Block Device
- OpenAIS



# Dziękuję za uwagę

**Mariusz Drożdziel**  
Październik 2009

