

Hochverfügbarkeit und Lastverteilung am Beispiel von Apache

Stephan Wienczny

Inhalt

- Motivation
- Hochverfügbarkeit
- Aufbau
- Virtualbox
- DRDB als verteiltes Blockdevice
- OCFS2 Clusterdateisystem
- Apache Webserver
- LVS und Keepalived zur Lastverteilung
- Probleme
- Fazit

Motivation

- Serversysteme können ausfallen und so ihre Dienste nicht mehr anbieten
- Der Ausfall eines Dienstes kann sehr teuer werden
- Negativbeispiele
 - Ticketsystem der Deutschen Bahn (15.1.2009)
 - Netzausfall T-Mobile (21.4.2009)

Hochverfügbarkeit

- Ein System gegen Ausfall von einzelnen Komponenten absichern
- Möglich durch redundante Auslegung aller Systeme und Dienste
- Fällt ein System oder Dienst aus wird dessen Funktion von anderen übernommen

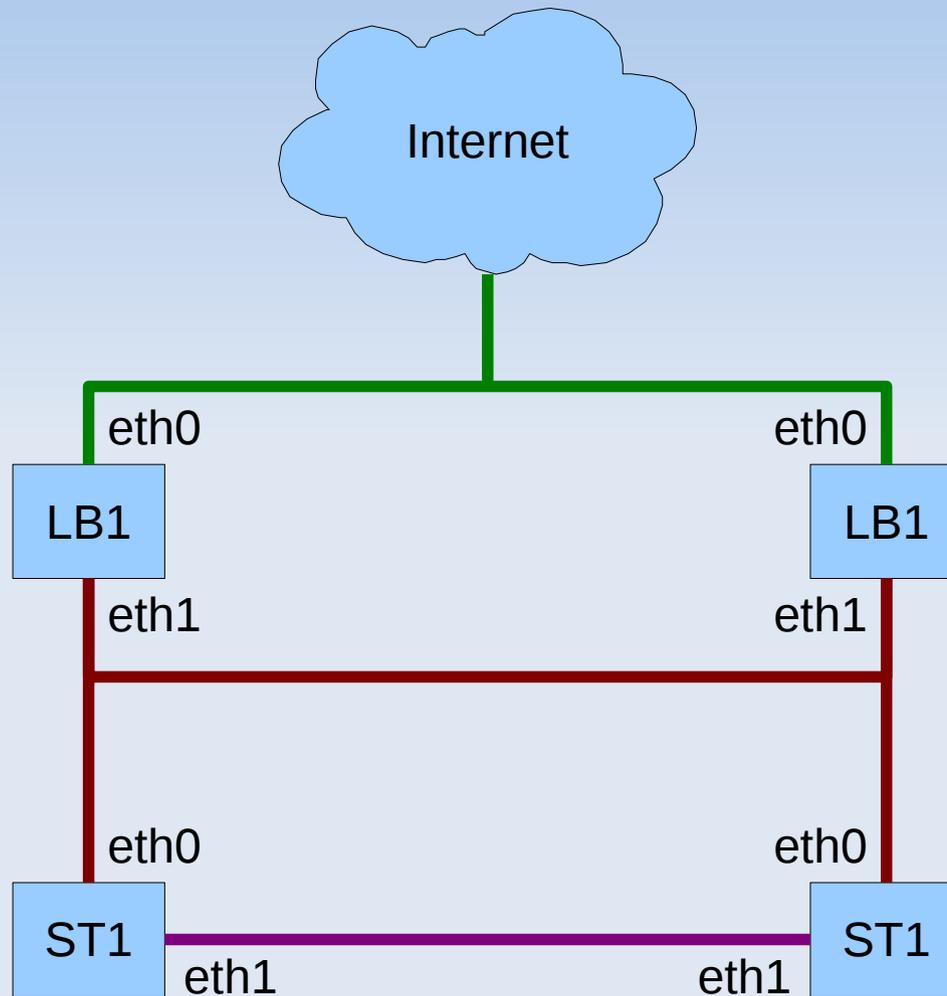
Hochverfügbarkeit

- Verschiedene Arten von Hochverfügbaren Systemen möglich
 - Cold Standby
 - Hot Standby
 - Active-Active Lastverteilung (Überlastung möglich)
- Übernahmestrategien
 - Stateless
 - Stateful

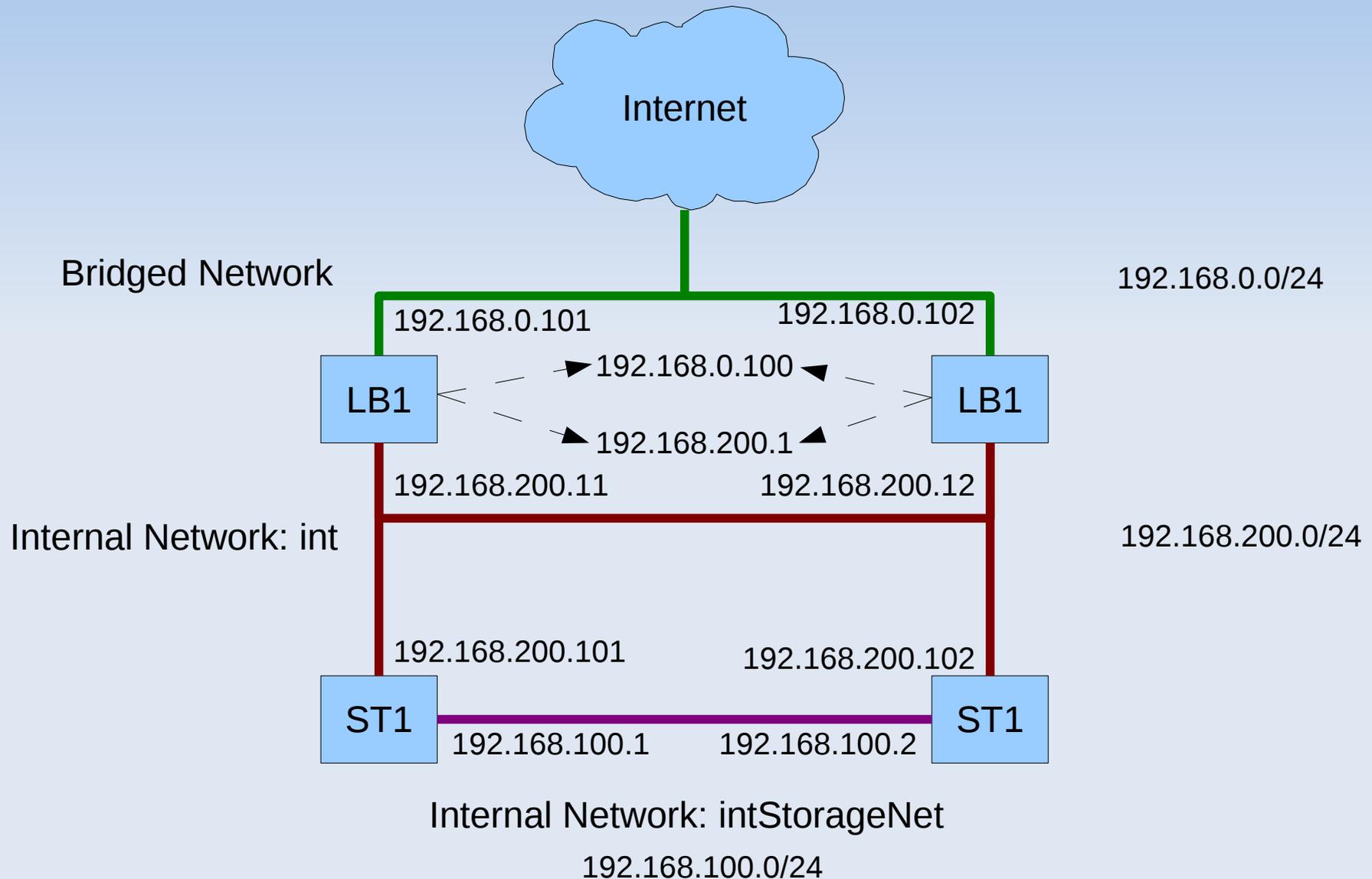
Hochverfügbarkeit

- Wichtige Komponenten eines hochverfügbaren Dienstes
 - Redundante gepufferte Stromversorgung und Klimaanlage
 - Redundante Verbindungen für
 - Netzwerk
 - Strom
 - Storage (SAN)
 - Regelmäßige vorbeugende Wartung (ahead-of-failure)
 - Vorbereitete Pläne für kritische Situationen
 - Change Management

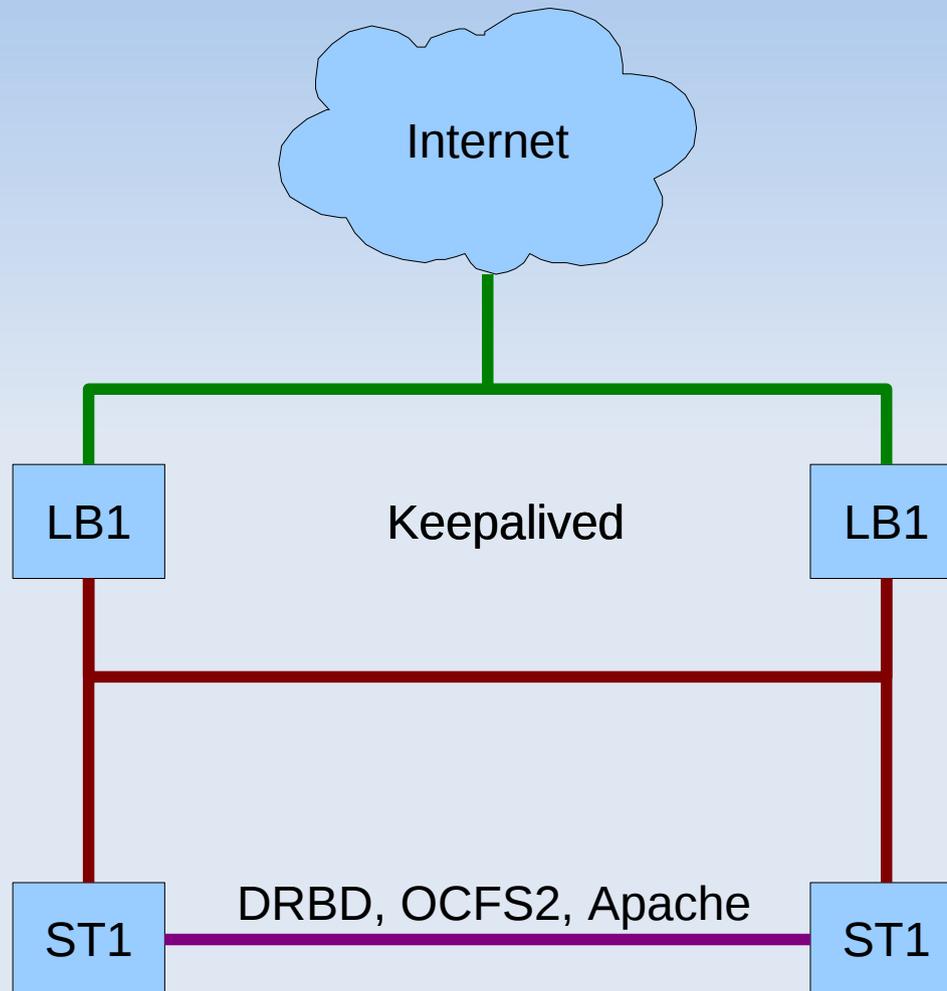
Aufbau: Topologie



Aufbau: IP Adressen



Aufbau: Software





Virtualbox: Geschichte

- Wurde Ursprünglich von der Firma Innotek entwickelt
- Wurde im Februar 2008 durch Sun Microsystems gekauft. Eingliederung in xVM
- Kauf von Sun Microsystems durch Oracle im April 2009



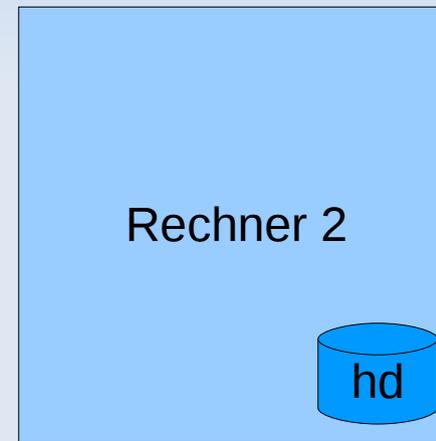
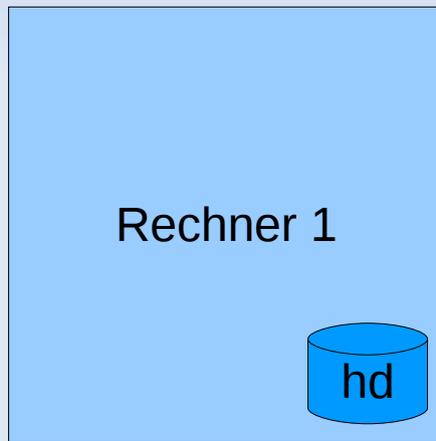
Virtualbox: Vorteile

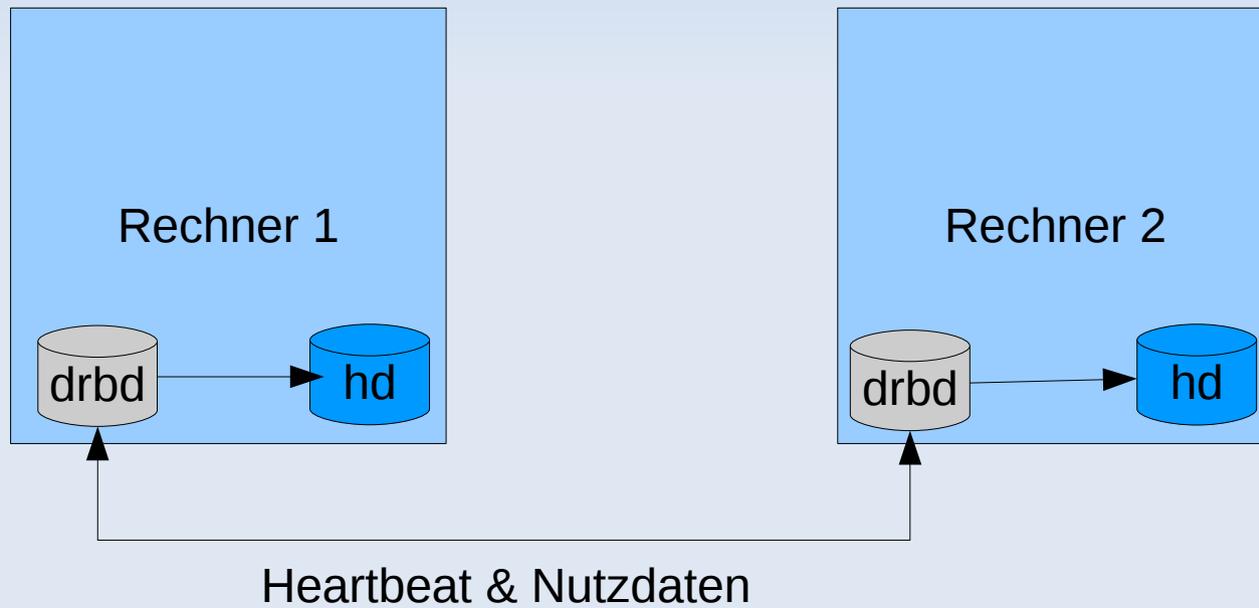
- Unterstützung für SATA, Intel e1000, OpenGL, Serielle Schnittstellen, iSCSI, Remote Desktop Protocol, VMDK (VMware), VHD (Microsoft), OVF (Open Virtualization Format)
- Kostenlose Versionen verfügbar
 - xVM Virtualbox unter der Personal Use and Evaluation License kostenlos nutzbar für persönliche Zwecke und Nutzung in Lehreinrichtungen
 - xVM Virtualbox OSE unter der GPL

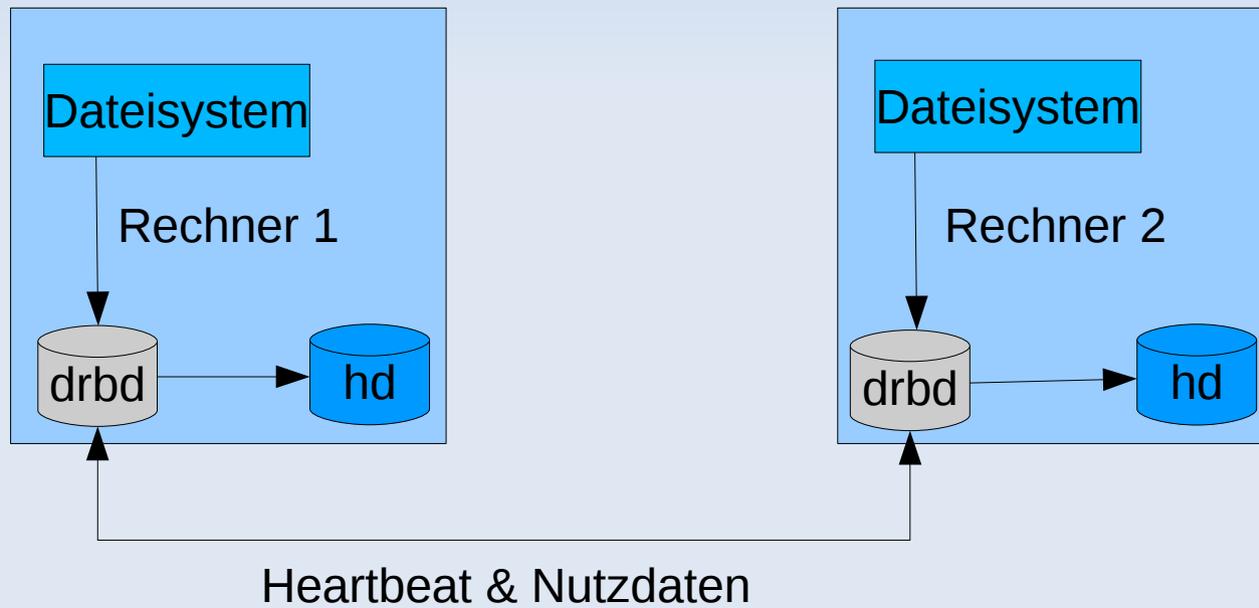


Virtualbox: Vorführung

- Distributed Replicated Block Device
- Wurde ursprünglich von Linbit entwickelt und steht inzwischen unter GPL
- Bietet die Möglichkeit Blockgeräte (z.B. Festplatten oder Partitionen) auf mehreren Rechnern zu spiegeln







- Benutzerstatistik abschalten
- Geschwindigkeit der Synchronisation erhöhen
- Protokoll C für sichere Schreiboperationen auf beiden Hosts
- 2 primäre Server erlauben und Verhalten bei Splitbrain konfigurieren
- Beim Start beide Server primär

```
Global { usage-count no; }  
  
common { Syncer { rate 10M; } }  
  
resource web {  
    protocol C;  
  
    net {  
        allow-two-primaries;  
        after-sb-0pri discard-zero-changes;  
        after-sb-1pri discard-secondary;  
        after-sb-2pri disconnect;  
    }  
  
    Startup { become-primary-on both; }  
...  
}
```

- Die beiden Server konfigurieren
 - **address** ist die Adresse des Dienstes auf dem Knoten
 - **device** ist das virtuelle Blockgerät
 - **disk** ist das real existierende Blockgerät
 - Metadaten werden in der Partition abgelegt

```
on StorageNode1 {  
    address 192.168.100.1:7788;  
    device /dev/drbd0;  
    disk /dev/sdb1;  
    meta-disk internal;  
}
```

```
on StorageNode2 {  
    address 192.168.100.2:7788;  
    device /dev/drbd0;  
    disk /dev/sdb1;  
    meta-disk internal;  
}}
```

- Initialisierung des Blockgerätes mit Metadaten
 - `drbdadm create-md resource`
- Ressource starten
 - `drbdadm up resource`
- Auf dem Masterknoten starten der ersten Synchronisation
 - `drbdadm --overwrite-data-of-peer primary resource`



OCFS2

- Oracle Cluster Filesystem 2
- Erlaubt konkurrierenden Zugriff auf einen geteilten Festplattenspeicher
- Ist seit 2.6.16 Bestandteil des Linux Kernels
- Einfache Konfiguration
 - Konfigurationsdatei
 - `mkfs.ocfs2`



OCFS2: Konfiguration

- Definition eines Clusters
- Definition eines Knoten

cluster:

node_count = 2

name = web

node:

ip_port = 7777

ip_address = 192.168.100.1

number = 0

name = StorageNode1

cluster = web

node:

...

Apache HTTP Server



- Opensource Webserver der Apache Software Foundation
- Ist laut Netcraft der meistbenutzte Webserver im Internet (Stand Juni 2009)
- Extrem vielseitig und durch Module gut erweiterbar
- Trotz seiner Verbreitung sicher
- Performant

Apache HTTP Server: Konfiguration



- Verzeichnis für Webserver definieren
- Einstellungen für das Verzeichnis vornehmen

- Webdav Lockfile definieren
- Webdav-Bereich aktivieren

```
DocumentRoot /srv/web/
```

```
<Directory /srv/web/>
```

```
Options Indexes FollowSym LinksMultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
allow from all
```

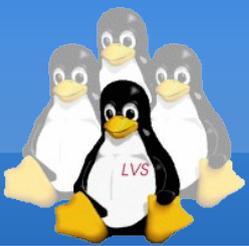
```
</Directory>
```

```
DavLockDB /srv/web/DavLock
```

```
<Location /webdav>
```

```
Dav On
```

```
</Location>
```



LVS

- Linux Virtual Server ist eine Software zur Lastverteilung in Serverfarmen
- Ist als Linux-Kernel-Modul implementiert
- Ermöglicht das Weiterleiten von Anfragen an andere Server



Keepalived

- Keepalived erweitert LVS um Tests auf Funktion von Servern
- Nicht funktionierende Server werden aus dem Pool genommen
- Bietet auch Hochverfügbarkeit der LVS-Server über VRRPv2 (Virtual Router Redundancy Protocol)



LVS & Keepalived

- Definition des Namens des Routers
- Definition der VRRP Synchronisationsgruppen
- Definition einer VRRP-Instanz mit verwendetem Netzwerkinterface eth0 und und geteilter Netzerwerkadresse
- VRRP-Instanz INT ist ähnlich hat nur eine andere Netzwerkadresse

```
global_defs { router_id LB1 }  
vrrp_sync_group VG1 { group { INT EXT }}  
vrrp_instance EXT {  
    state MASTER  
    interface eth0  
    lvs_sync_daemon_interface eth0  
    virtual_router_id 51  
    priority 150  
    advert_int 1  
    virtual_ipaddress { 192.168.0.100 }  
}  
...
```



LVS & Keepalived

- Definition eines virtuellen Server
- Definition der Art der Lastverteilung
- Definition der Server auf die die Last verteilt wird

```
virtual_server 192.168.0.100 80 {  
    delay_loop 6  
  
    lb_algo wrr ! weighted round robin  
  
    lb_kind NAT  
  
    nat_mask 255.255.255.0  
  
    protocol TCP  
  
    real_server 192.168.200.101 80 {  
        weight 1  
  
        TCP_CHECK { connect_timeout 3  
                    connect_port 80 }}  
  
    ... ! Definition weiterer real_server  
}
```

Beispielanwendungen

- cms -db
 - Ein Datenbankloses CMS
- Toenda CMS
 - Ein XML-Datei basierendes CMS
- Webdav

Probleme

- Kein "Stateful Failover" bei Ausfall eines der Webserver
- Session gehen verloren, wenn sie nicht auf beiden Webservern existieren

Fazit

- Webserver lassen sich hochverfügbar auslegen
- Komplette Netzwerke sind virtualisierbar
 - Storage (DRBD)
 - Rechner und Netzwerk (Virtualbox)
 - Dienste (LVS + Keepalived)
- Virtualisierung ermöglicht Aufbau einer Test- und Abnahmeumgebung auch für hochverfügbare Anwendungen
- Alternative Heartbeat

Quellen

- <http://www.virtualbox.org>
- <http://www.drbd.org>
- <http://oss.oracle.com/projects/ocfs2>
- <http://httpd.apache.org>
- <http://www.linuxvirtualserver.org>
- <http://www.keepalived.org>
- <http://www.linux-ha.org>
- <http://www.toendacms.org>
- <http://cms-db.de>
- http://news.netcraft.com/archives/web_server_survey.html

Vielen Dank für Ihre Aufmerksamkeit!