HA clustering made simple
with OpenVZ

Werner Fischer, Thomas-Krenn.AG
(wfischer@thomas-krenn.com)

Profoss Virtualisation event 2008
Brussels, 23rd January 2008
**Short Bio**

- **Werner Fischer**
  - 2000-2004: Computer- and Media Security (Upper Austria University of Applied Sciences, Hagenberg Campus)
  - 2004-2005: IBM Mainz, Linz, San Jose/CA, Raleigh/NC
  - redbooks covering HA Clustering and Storage
  - since 9/2005: Thomas-Krenn.AG, R&D (HA-Clustering, Virtualisation)

- **relationship to OpenVZ project**
  - using OpenVZ for over two years
  - focusing on OpenVZ clustering, written HOWTO
    http://wiki.openvz.org/HA_cluster_with_DRBD_and_Heartbeat
Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
1) Cluster Technologies Overview

- term *clustering*
  - High Availability (HA) cluster
  - Load Balancing cluster
  - High Performance Computing (HPC) cluster
  - Grid computing
Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
2) **HA clustering best practices**

- High Availability (HA) cluster
  - goal: increase availability of services
  - elimination of all SPOFs (single points of failure)
  - failover / switchover
  - 2-node-clusters widely-used

<table>
<thead>
<tr>
<th>Uptime [%]</th>
<th>Downtime per year</th>
<th>Downtime per week</th>
</tr>
</thead>
<tbody>
<tr>
<td>98 %</td>
<td>7,3 days</td>
<td>3 h 22 min</td>
</tr>
<tr>
<td>99 %</td>
<td>3,65 days</td>
<td>1 h 41 min</td>
</tr>
<tr>
<td>99,8 %</td>
<td>17 h 30 min</td>
<td>20 min 10 sec</td>
</tr>
<tr>
<td>99,9 %</td>
<td>8 h 45 min</td>
<td>10 min 5 sec</td>
</tr>
<tr>
<td>99,99 %</td>
<td>52,5 min</td>
<td>1 min</td>
</tr>
<tr>
<td>99,999 %</td>
<td>5,25 min</td>
<td>6 sec</td>
</tr>
<tr>
<td>99,9999 %</td>
<td>31,5 sec</td>
<td>0,5 sec</td>
</tr>
</tbody>
</table>
2) HA clustering best practices

active/passive vs. active/active with 2-node-clusters

- when would active/active bring advantages
  - mainly when each of the two servers exceed an utilisation of 50%

- what would be the consequence in case of an outage?
  - the remaining node does not have enough free resources, services cannot be provided reliably
2) HA clustering best practices

- cluster tests:
  - manual switchover tests (2)
  - power outage tests (7)
  - serial connection tests (4)
  - crossover network connection tests (4)
  - public network connection tests (9)
  - shutdown tests (2)
  - reboot tests (2)
  - hard drive outage tests (2)
2) HA clustering best practices

- Shared Storage (SAN) vs. Replicated Storage
  - Shared Storage
    - Shared SCSI, Fibre Channel SAN, iSCSI SAN
    - Storage system can be SPOF
    - Shared Resource Protection (Node/Resource Level Fencing (STONITH, SCSI Locking), Quorum)
  - Replicated Storage
    - eg. DRBD (Distributed Replicated Block Device)
    - No dedicated storage system (no SPOF)
    - Cost-effective
    - Shared Resource Protection less critical
Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
3) Concept of HA cluster with OpenVZ

- challenges of traditional HA cluster systems
  - most applications need to be customised
  - config files (/etc) must be synchronised manually (or be replaced by symbolic links to /data/...)
  - keeping system config files like /etc/passwd in sync is complex
  - time-consuming and error-prone
    -> causes additional costs
3) Concept of HA cluster with OpenVZ

- clustering of entire virtual machines
- whole file system of a virtual machine is mirrored
- applications are only installed once (within the virtual machine), not twice (on each node)
3) Concept of HA cluster with OpenVZ

- **Virtual Environment**: VE 101, VE 102, VE 103...
- **OS virtualisation OpenVZ**
- **Data Replication**:
  - DRBD
- **Cluster Manager Heartbeat**
- **Operating System CentOS 4**

Managed by Heartbeat

**Failover**

**Switchover**

NODE1

NODE2

OpenVZ server virtualization

DRBD®

High Availability

CentOS

The Community ENTERprise Operating System
3) Concept of HA cluster with OpenVZ

- Operating System
  - Community ENTerprise Operating System
  - based on Red Hat Enterprise Linux
  - strives to be 100% binary compatible with the upstream product
  - www.centos.org
3) Concept of HA cluster with OpenVZ

- Cluster Manager Heartbeat
  - cluster nodes communicate via three paths (eth0, eth1, ttyS0)
  - connectivity from outside is monitored via pingnode
3) Concept of HA cluster with OpenVZ

- Data replication **DRBD** „RAID1 over network“
  - write operation (on both nodes)
  - read operation (on primary node)
3) Concept of HA cluster with OpenVZ

- OS virtualisation
  - containers-type virtualisation on Linux
  - creates multiple secure, isolated containers (VEs, VPSs)
  - single-kernel technology
  - enables better server utilisation
  - allows resource configuration
  - http://openvz.org
  - (other OS virtualisation tech.: VServer, FreeBSD Jails, Solaris Containers)
Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
4) **OpenVZ details**

OpenVZ components:

- **Kernel**
  - Virtualization and Isolation
  - Resource Management
  - Checkpointing

- **Tools**
  - vzctl: Virtual Environment (VE) control utility
  - vzpkg: VE software package management

- **Templates**
  - precreated VE images for fast VE creation
4) OpenVZ details

Each virtual environment has its own:

- **Files**
  System libraries, applications, virtualized /proc and /sys, virtualized locks etc.

- **Process tree**
  Featuring virtualized PIDs, so that the init PID is 1

- **Network**
  Virtual network device, its own IP addresses, set of netfilter and routing rules

- **Devices**
  Plus if needed, any VE can be granted access to real devices like network interfaces, serial ports, disk partitions, etc.

- **IPC objects**
  shared memory, semaphores, messages
4) **OpenVZ details**

OpenVZ Resource Management:

- **User BeanCounters** is a set of per-VE resource counters, limits, and guarantees
  (kernel memory, network buffers, phys pages, etc.)

- **Fair CPU scheduler**
  (with shares and hard limits)

- **Two-level disk quota**
  (first-level: per-VE quota; second-level: ordinary user/group quota inside a VE)

- **I/O scheduler**
  (two-level, based on CFQ)
4) **OpenVZ details**

OpenVZ Kernel Checkpointing/Migration:

- Complete VE state can be saved in a file
  - running processes
  - opened files
  - network connections, buffers, backlogs, etc.
  - memory segments
- VE state can be restored later
- VE can be restored on a different server
4) OpenVZ details

OpenVZ Tools:

```
# vzctl create 101 --ostemplate fedora-core-5
# vzctl set 101 --ipadd 192.168.4.45 --save
# vzctl start 101
# vzctl exec 101 ps ax
```

```
PID   TTY     STAT   TIME COMMAND
 1     ?       Ss     0:00 init
11830 ?       Ss     0:00 syslogd -m 0
11897 ?       Ss     0:00 /usr/sbin/sshd
11943 ?       Ss     0:00 xinetd -stayalive -pidfile ...
12218 ?       Ss     0:00 sendmail: accepting connections
12265 ?       Ss     0:00 sendmail: Queue runner@01:00:00
13362 ?       Ss     0:00 /usr/sbin/httpd
13363 ?       S       0:00  \\ /usr/sbin/httpd
13373 ?       S       0:00  \\ /usr/sbin/httpd
6416 ?         Rs     0:00 ps axf
# vzctl enter 101
bash# logout
# vzctl stop 101
# vzctl destroy 101
```
4) OpenVZ details

OpenVZ Tools:

# vzpkgls
fedora-core-5-i386-default
centos-4-x86_64-minimal

# vzpkg gcache
(creates templates from metadata(updates existing templates)

# vzyum 101 install gcc
(installs gcc and its deps to VE 101)
4) **OpenVZ details**


„For all the cases tested, the virtualization overhead observed in OpenVZ is limited, and can be neglected in many scenarios.“

Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
5) **live switchover enhancement**

- uses OpenVZ's checkpointing feature
- allows rolling kernel-upgrades without shutting down virtual environments

- the following scripts are necessary:
  - `cluster_freeze.sh`
  - `cluster_unfreeze.sh`
  - `live_switchover.sh`
  - an adjusted init script for openvz
Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
6) outlook: LBVM

- LBVM (load balancing of virtual machines)
  - allows sharing virtual machines among physical servers in a predefined cluster
  - LB MONITOR: load balancer itself
    (uses different algorithms to decide which virtual machines should be moved or reported)
  - LBM script: management interface to the load balancer
    (is used to view all balanced virtual machines, review log files and reports, manually migrate)
  - LB LOG: small cronjob which runs regularly on each server to monitor predefined resources
    (the resource logs are stored on a shared storage and are evaluated by the load balancer)
6) outlook: LBVM
Agenda

1. Cluster Technologies Overview
2. HA clustering best practices
3. Concept of HA cluster with OpenVZ
4. OpenVZ details
5. Live Switchover enhancement
6. Outlook: LBVM (load balancing of virtual machines)
7. Conclusion
## 7) Conclusion

<table>
<thead>
<tr>
<th>What is it?</th>
<th>Linux High Availability Cluster with OS-level virtualisation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What does it do?</strong></td>
<td>● mirrors whole virtual environments on two cluster nodes</td>
</tr>
<tr>
<td></td>
<td>● restarts virtual environments in case of an outage on</td>
</tr>
<tr>
<td></td>
<td>the second (remaining) node</td>
</tr>
<tr>
<td>Who can use it?</td>
<td>Linux administators</td>
</tr>
<tr>
<td><strong>What are typical usage szenarios?</strong></td>
<td>Misson-Critical database server, mail server, web server,</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Resources

- http://openvz.org/
- http://wiki.openvz.org/HA_cluster_with_DRBD_and_Heartbeat
- http://www.centos.org/
- http://www.linux-ha.org/
- http://www.drbd.org/
- http://lbvm.sourceforge.net/

Thanks for your attention!