Oracle VM 3.0 Was nicht im Handbuch steht...

DOAG Konferenz 2012

Martin Bracher

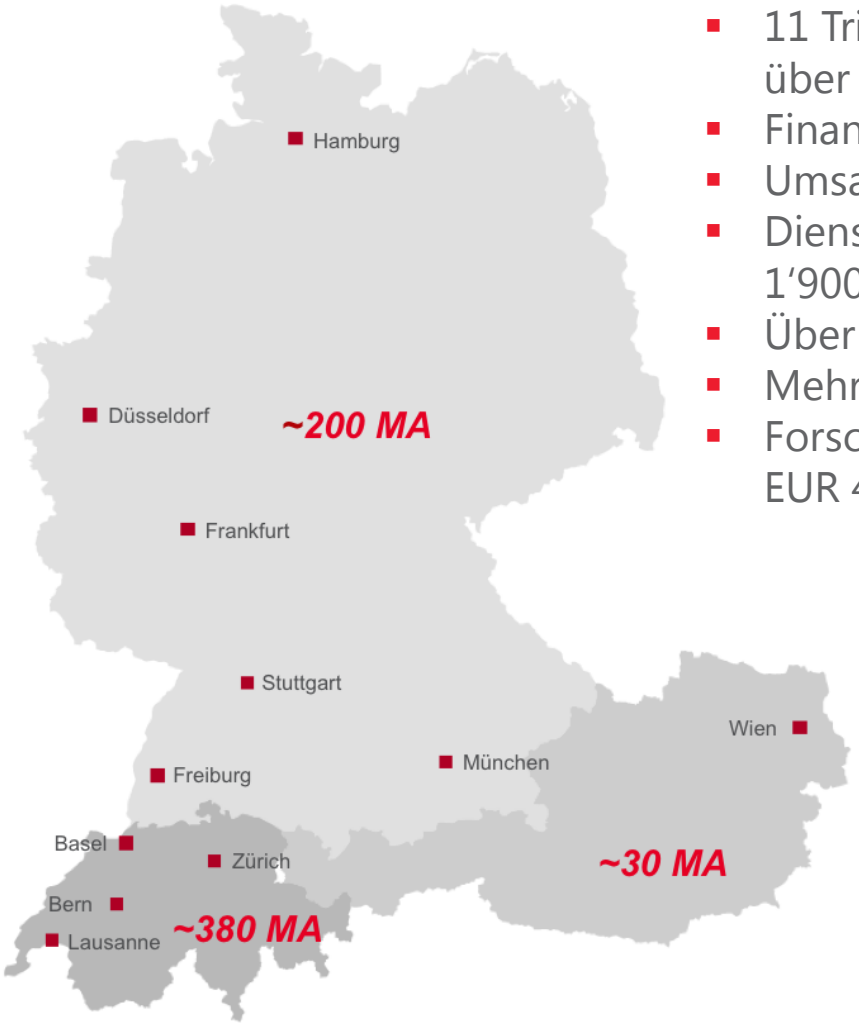
Senior Consultant
Trivadis AG

21. November 2012

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.

Trivadis makes IT easier.



- 11 Trivadis Niederlassungen in CH, DE und AT mit über 650 Mitarbeitenden
- Finanziell unabhängig und nachhaltig profitabel
- Umsatz CHF 104 / EUR 84 Mio.
- Dienstleistungen für über 800 Kunden in mehr als 1‘900 Projekten
- Über 200 Service Level Agreements
- Mehr als 4'000 Trainingsteilnehmer
- Forschungs- und Entwicklungsbudget: CHF 5.0 / EUR 4 Mio.

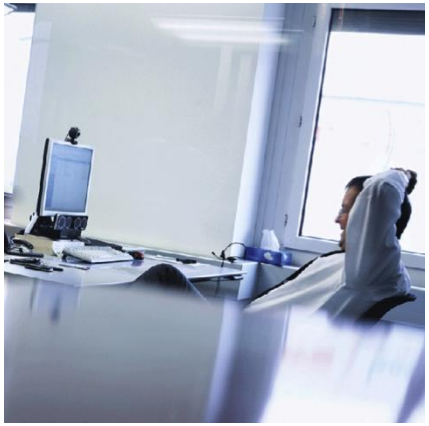
ORACLE Platinum Partner

Microsoft Partner
Gold Application Lifecycle Management
Gold Business Intelligence
Silver Customer Relationship Management
Silver Learning

TOP JOB Zertifizierter Arbeitgeber 08|2012

trivadis makes IT easier.

Application Development
Business Intelligence
Business Integration Services
Infrastructure Engineering
Managed Services
Training



Compliance

- übersetzt
- optimiert
- umfassend

biGenius

- schnell
- einfach
- umfassend

Toolbox

- standardisiert
- generiert
- automatisiert

Infrastructure Care

- optimiert
- nachhaltig
- modular

Application Care

- planbar
- effizient
- nachhaltig

Comprehensive Application Development

- unabhängig
- kompetent
- vollständig

AGENDA

1. Working on Commandline
2. Snapshots
3. High available database connect
4. Hard-partitioning

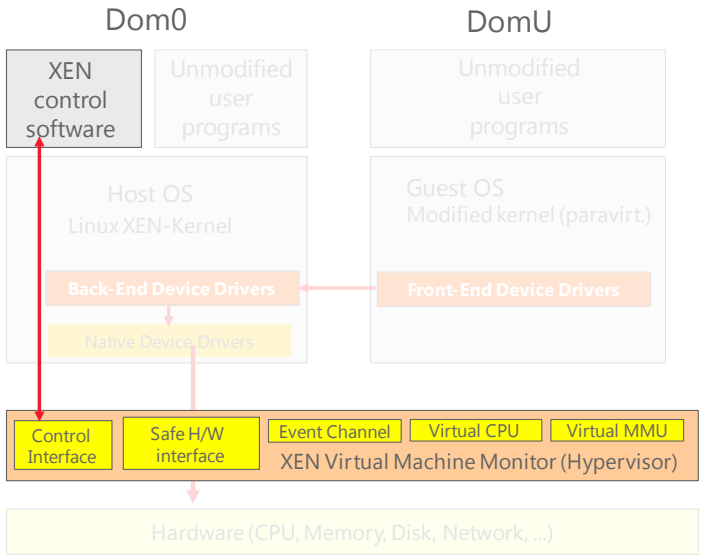
XEN commands

- OracleVM is based on XEN
  - "xm" is the tool to access control interface
    - Start a DomU

      `xm create /path-to/vm.cfg`
    - show DomU's

      `xm list; xm list -l`
    - Power-off a DomU

      `xm destroy <name_of_vm | number>`



XEN commands

- Show the running VM's
  - Unfortunately, Oracle uses an UUID name instead of name displayed in GUI

```
#   xm list
Name                                        ID   Mem VCPUs      State   Time(s)
0004fb0000060000059ec32f98918b07            20  2048     2     -b----  51543.1
0004fb00000600004c953ac4246637eb            19   512     1     -b----    756.0
0004fb0000060000008219e1e797e01b0d          29  2048     2     -b----    998.7
0004fb000006000088cde8cce067a63b            24  2048     2     -b----  60768.4
Domain-0                                     0  1134    16     r----- 474653.3
```

  - Fortunately, with a wrapper-script we can adapt the output

```
#  xm2 list
Shortname   Name                                ID  Mem VCPUs State   Time(s)
slot031     0004fb0000060000059ec32f98918b07  20 2048     2 -b----  51600.7
ovminfra    0004fb00000600004c953ac4246637eb  19  512     1 -b----    756.4
master249   0004fb0000060000008219e1e797e01b0d  29 2048     2 -b----   1003.8
slot001     0004fb000006000088cde8cce067a63b  24 2048     2 r-----  60829.3
Domain-0                                       0 1134    16 r----- 474750.4
```

XEN commands

- How to get the mapping from UUID to well-known name?
  - It is stored in the file /OVS/Repositories/*/VirtualMachines/*/vm.cfg

```
OVM_simple_name = 'slot031'
disk = …
...
uuid = '0004fb00-0006-0000-059e-c32f98918b07'
...
name = '0004fb0000060000059ec32f98918b07'
```

  - First, get the name of the configuration file for the given uuid (in $1):

```
vm_cfgfile=$( grep -l "name = '$1'" /OVS/Repositories/*/*/*/vm.cfg )
```

  - Afterwards, read the human-readable name from this file

```
vm_name=$( sed -n -e "s/^OVM_simple_name = '\(.*\)'/\1/p"  ${vm_cfgfile} )
```

  - Then, print the original "xm list" output, prefixed by the $vm_name

XEN commands

- "xentop": shows usage of hardware by VM's (xm top will do the same)
  - "top" only shows usage of curent domain

```
xentop - 10:20:36   Xen 4.0.2-OVM
3 domains: 1 running, 2 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown
Mem: 12581880k total, 5088772k used, 7493108k free    CPUs: 8 @ 2327MHz
      NAME  STATE   CPU(sec) CPU(%)     MEM(k) MEM(%)  MAXMEM(k) MAXMEM(%) VCPUS NETS NETTX(k) NETRX(k) VBDS   VBD_OO   VBD_RD   VBD_WR VBD_RSECT VBD_WSECT SSID
0004fb0000 --b---        223    0.2    2097152   16.7    2097152      16.7     2    1     3954       23    3        0    28983     6658    577707     94634    0
0004fb0000 --b---          0    0.1    2097152   16.7    2097152      16.7     2    1      132        0    1        0        0       28         0       224    0
  Domain-0 -----r       1183    4.8     752128    6.0  no limit       n/a     8    0        0        0    0        0        0        0         0         0    0

  Delay  Networks  vBds  Tmem  VCPUs  Repeat header  Sort order  Quit
```

  - Good to have a wide-screen ☺
  - Very bad! Output uses UUID-name of VM and cuts the name
    - you only see the first 10 chars, usually identical for all VM's

XEN commands

- Starting and stopping a VM on the server
  - Get the location of the vm.cfg file, then start it with xm commands

```
xm create /OVS/Repositories/.../vm.cfg
xm2 create slot002
```

  - OVM Manager detects change of state

  - VM can be handled by OVM Manager afterwards

XEN commands

- What is different to start via OVMM?
  - If the VM is configured "Enable High Availability" and the vm crashes

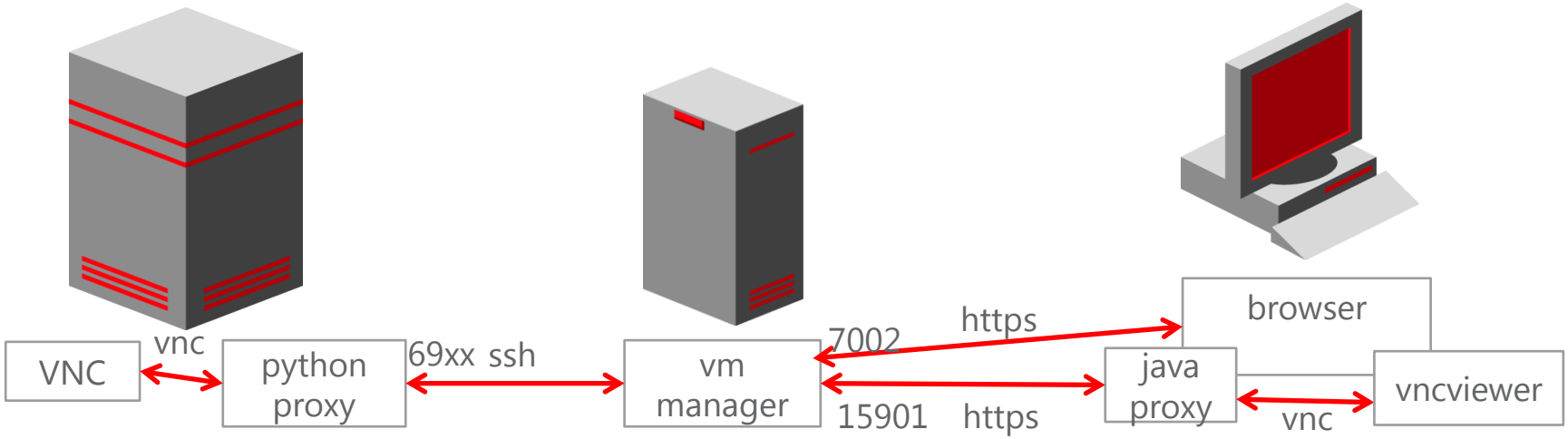
    - If started on server: the VM will not be automatically restarted
    - If started via OVMM: It will be monitored and restarted by a server agent
      - HA also works if the manager is down
      - "xm shutdown" on server works without automatic restart
  - → no problem to start on server as long as we do not use HA

  - Alternative 1: **ovm_vmcontrol** (from Oracle VM utilities)
    - but does not run on OVM Server (requires java), OK on OVM Manager
  - Alternative 2: OVM cli with the latest patch of OVM Manager 3.1.1
    - `ssh -p 10000 admin@ovmmanager "start vm name=slot002"`

Virtual Console

Access to the console of the VM

- VNC protocol ist tunneled via http(s) to the Manager
  - On OVM Server the console is started
    - /usr/bin/python /opt/ovm-console/vncViewer/vnc.py 127.0.0.1 34171
  - On client, via browser, a Java proxy is downloaded



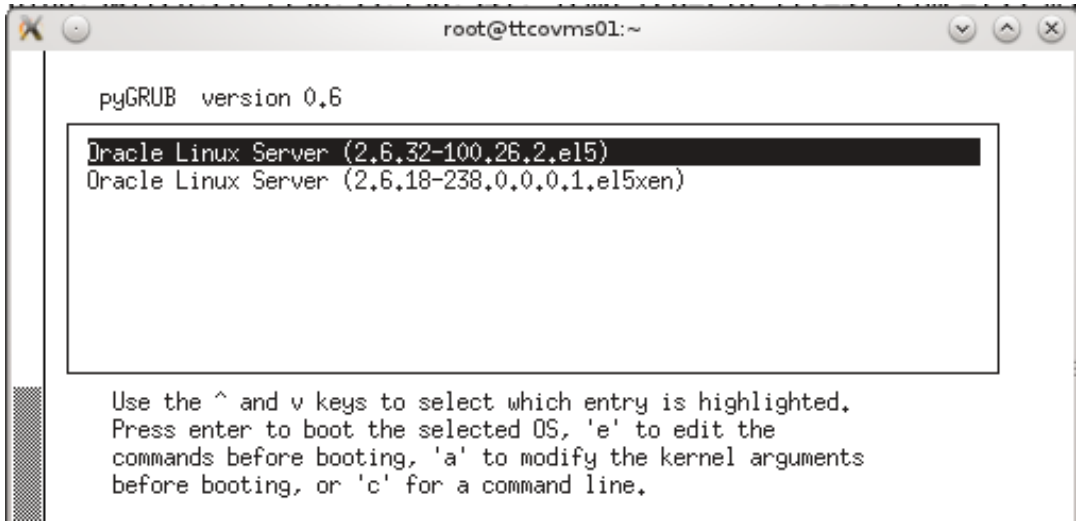
Virtual Console in text-mode

- Why do so?
  - VNC console does not work (wrong Java-Version on client)
  - Troubleshooting boot-procedure
    - If it does not boot (no output on vnc-console), use the textmode console
      - See note ID 579413.1
      - Maybe the wrong kernel is set as default
      - Leave the console with Ctrl-] or Ctrl-5
      - To see the whole boot-process in textmode console:
        > remove temporarily "vfb=" entry in vm.cfg

Virtual Console in text-mode

- Start the console
  - Login to OVM server and start it manually
    - in this textmode-console, the emulated GRUB bootloader is accessible
      - Not displayed in VNC console (emulation running outside of VM)

```
xm create -c /<path_to>/vm.cfg extra="console=xvc0"
```



    - Leave the console with Ctrl-] or Ctrl-5
    -

Virtual Console in text-mode (permanent setting)

- Modify the VM for textmode-console
  - Inside the VM:
    - /etc/inittab

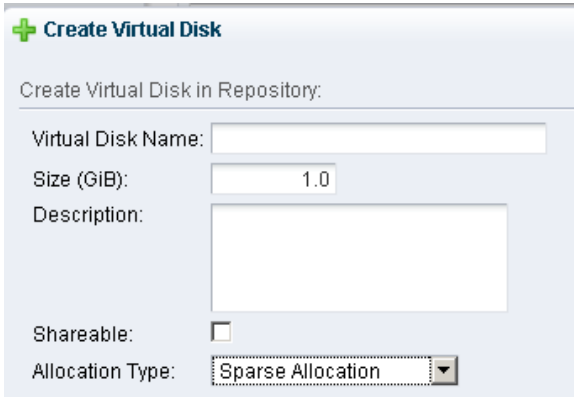
```
# Run a getty on the virtual console
co:2345:respawn:/sbin/agetty xvc0 9600 vt100-nav
```

    - /boot/grub/menu.lst

```
kernel /vmlinuz ... console=xvc0
```

    - Without modifying menu.lst
      - Boot-messages are displayed in VNC console
      - But afterwards, login is possible in text-mode **and** VNC console

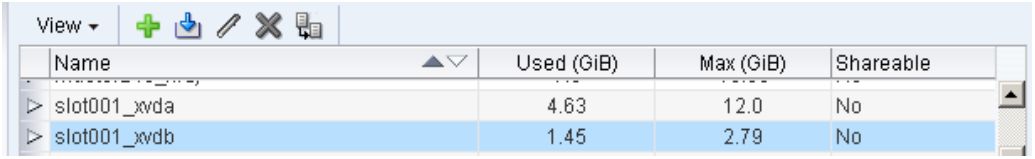
Creating virtual disks on commandline

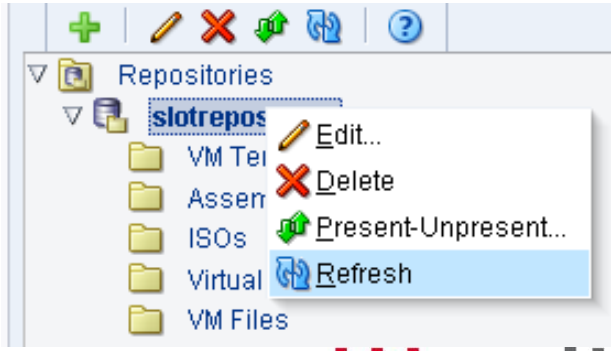


- Can be done via GUI
  - But not very efficient to create 100 similar disks...
- Solution: Can be done on commandline and OVM Manger recognizes it
  - Example: creating sparse-disks on the server

```
cd /OVS/Repositories/0004fb0000030000078778888fdcdb96e/VirtualDisks
for i in {01..30}; do
  nr=$(printf "%02d" $i)
  dd if=/dev/zero of=slot0${nr}_xvda bs=1k count=0 seek=12582912
  dd if=/dev/zero of=slot0${nr}_xvdb bs=1k count=0 seek=2929688
done
```

  - Afterwards, re-scan the repository
    - Found disks will be registered in OVMM
    - With the name of the file (no UUID)





Add missing commands - BusyBox

- OVM server only contains minimal packages
  - It is possible to install additional RPM packages, but not supported
  - Essential tools for troubleshooting are missing
    - traceroute, nslookup, telnet, ...
  - Solution without installing additional RPM's:
    - **BusyBox** - The Swiss Army Knife of Embedded Linux
    - Static binary, containing many of these small tools
    - Get busybox binary from a Linux installation and copy it to /usr/local/bin

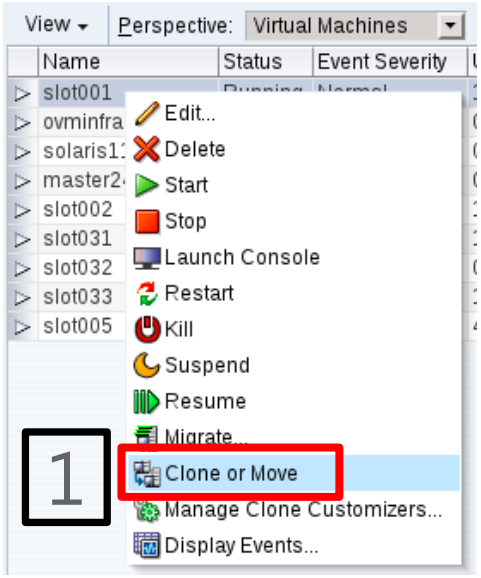
```
# busybox <command>  <arguments>
busybox nslookup ttcovms02.ttc.trivadis.com
busybox traceroute ttcovms02.ttc.trivadis.com
busybox telnet ttcovms02.ttc.trivadis.com 22
# if symlinked to <command>, it can be called directly
ln -s /usr/local/bin/busybox /usr/local/bin/telnet
telnet ttcovms02.ttc.trivadis.com 22
```

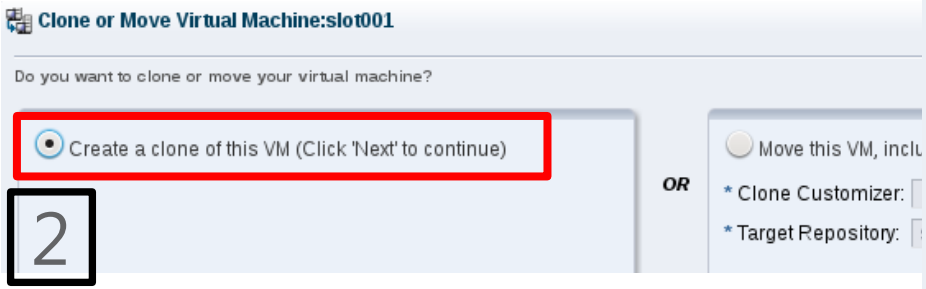
AGENDA

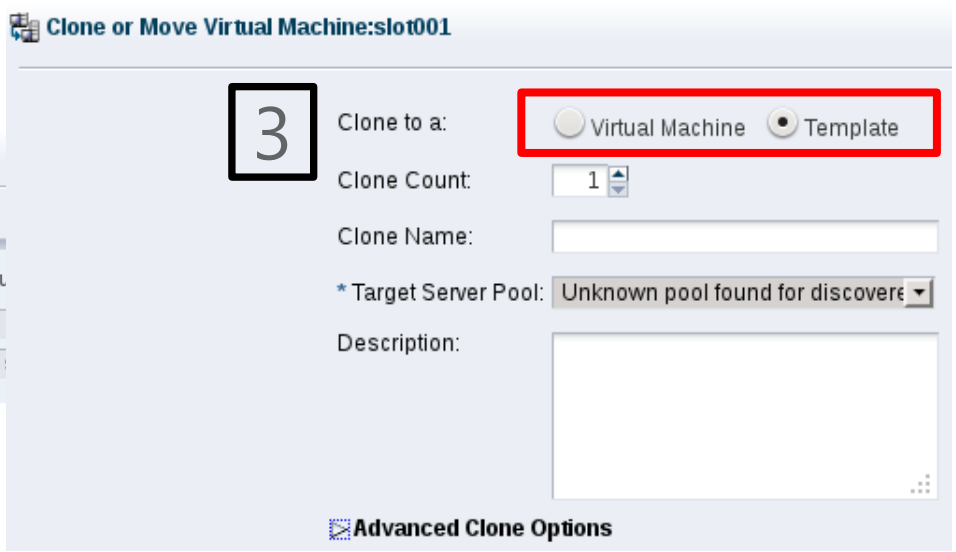
1. Working on Commandline
2. Snapshots
3. High available database connect
4. Hard-partitioning

Clone, Snapshot

- What is aSnapshot?
  - Saved state of a VM
- There are no snapshots like in VMware or VirtualBox
  - Take a snapshot and revert to it at a later time
- But: you can clone a VM
  - Create a template (recommended)
  - Create a new VM







Clone

- What cloning will do:
  - Copy files (if possible, a snapshot copy is created (on ocfs2))
  - Create new vm.cfg file
  - Register it as a template or new VM
- How to "revert to snapshot", the function that does not exist?
  - Create a clone (template) instead of a "snapshot"
  - "revert to snapshot" means, exchange the disks from the template with the disks of the VM
  - That is supported. We **do not change the definition of the VM**, we **exchange the contents of the disks**

Snapshots of a VM: Restore from a clone

- Restore of a clone
  - Get the disks of original and clone VM (vm.cfg)

    'file:/OVS/Repositories/…/VirtualDisks/<uuid>.img,xvda,w'
    - Filename
    - Devicename (inside VM)

```
origcfg=$(grep -l "OVM_simple_name = '$orig'" /OVS/Repositories/*/VirtualMachines/*/vm.cfg)
clonecfg=$(grep -l "OVM_simple_name = '$clone'" /OVS/Repositories/*/*/*/vm.cfg )

origdisks=$( grep "^disk *= *\[" $origcfg | \
             sed -e "s/^disk.*\[\(.*\)\].*/\1/g" | sed -e "s/', *'/' '/g")
clonedisks=$(grep "^disk *= *\[" $clonecfg | \
             sed -e "s/^disk.*\[\(.*\)\].*/\1/g" | sed -e "s/', *'/' '/g")
```

  - Stop the VM before restoring the clone
    - Caution! Check if it is not running on another node

```
xm destroy $(basename `dirname $origcfg` )
```

    - Better: with OVM cli

```
ssh -p 10000 admin@ovmmanager "kill vm name=$orig"
```

Snapshots of a VM: Restore from a clone

- Map the cloned disks to the original disks (same device-name)

```
for i in $origdisks; do
   #extract the filename
   origfile=${i#*:}
   origfile=${origfile%%,*}
   #extract the VM devicename
   origvmdevice=${i%,*}
   origvmdevice=${origvmdevice#*,}

   for j in $clonedisks; do

     if [ -z ${j/*,$origvmdevice,*} ]; then #if ",$origvmdevice," found, then …
       #extract the filename
       clonefile=${j#*:}
       clonefile=${clonefile%%,*}
       <WhatToDo>
     fi

   done
 done
```

Snapshots of a VM: Restore from a clone

- <WhatToDo>
  - Replace the original files with the files of the clone

```
rm –f $origfile
reflink $clonefile $origfile
#or "mv $clonefile $origfile"  if snapshot is no longer needed
```

  - Exchange the files of original and clone (current state is the new clone)

```
mv $origfile $origfile.tmp
reflink $clonefile $origfile
mv $origfile.tmp $clonefile
```

- Afterwards, restart the VM
  - Can be done on server, OVM Manager detects it

```
xm create $origfile
```

  - Better (supported): with OVM cli

```
ssh –p 10000 admin@ovmmanager "start vm name=$orig"
```

Snapshots of a VM: Without OVM Manager

- Implementation of Snapshots without Clones via OVM Manager
  - With scripts on server
    - Create a directory "snapshots" on the same level as "VirtualDisks"
    - Get filename of VM and create reflink to snapshot

```
reflink   VirtualDisks/disk1.img   snapshots/disk1.img.$(date +%Y%m%d-%H%M%S)
```

    - Restore is similar to "clone" description above
      - Identification of required snapshot files via naming-convention, not via config-file
  - Possibility to store the memory-content of the VM in the snapshot
    - Suspend the VM and make a snapshot of the "state" file (and diskfiles)

```
reflink $(dirname $origfile)/state  snapshots/state.$(date +%Y%m%d-%H%M%S)
```

AGENDA

1. Working on Commandline
2. Snapshots
3. High available database connect
4. Hard-partitioning

Repository Database with RAC, DataGuard or Failover Cluster

- OVM Manager connects to repository-DB via jdbc thin-driver
  - SQL*Net configuration (tnsnames.ora, ldap) is NOT used
- Initial setup for database connect
  - jdbc:oracle:thin:@Host:Port:SID
  - Failover Cluster
    - OK if "Host" is a VIP address
  - RAC
    - If "Host" is the SCAN-address, good idea
    - but it does not work with SID → SERVICE_NAME is required
  - DataGuard
    - Does not work with "Host", databases on different hosts (no VIP addresses)
    - Use SERVICE, not SID; otherwise the standby could be connected
- What we need is a failover connect

Repository Database with RAC, DataGuard or Failover Cluster

- Failover connect
  - Jdbc-thin driver allows connect-time failover, but not transparent appl. failover

  - RAC

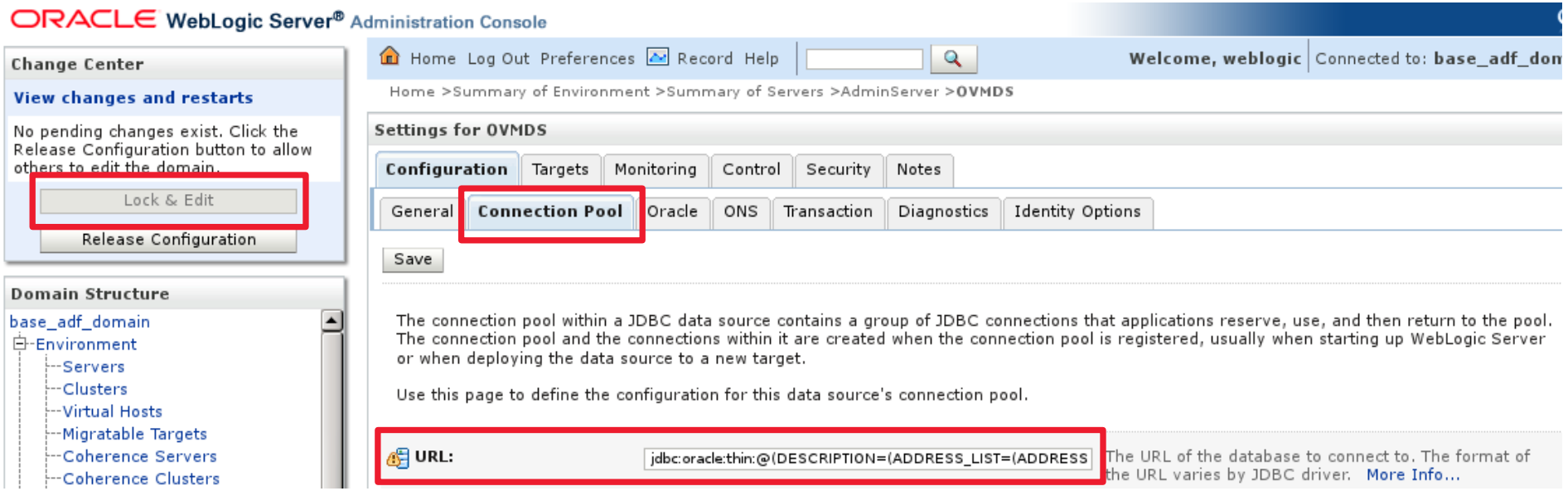
```
jdbc:oracle:thin@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=SCAN_IP)(PORT
=1521))(CONNECT_DATA=(SERVICE_NAME=OVMSERVICE.domain))
```

  - DataGuard

```
jdbc:oracle:thin@(DESCRIPTION=
(FAILOVER=ON)(LOAD_BALANCE=OFF)(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=TCP)(HOST=site1)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=site2)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=OVMSERVICE.domain)))
```

Repository Database with RAC, DataGuard or Failover Cluster

- Configure Failover connect
  - Login to Weblogic (url like OVMM, but /console instead of /ovm/console)
  - Domain Structure: base_adf_domain → Services→ Data Sources:
  - click "OVMDS", then "Connection Pool" tab



Repository Database with RAC, DataGuard or Failover Cluster

- Automatic re-connect
  - From the page before, scroll down to "Advanced" and expand it
  - Check "Test Connections On Reserve"
  - Test Frequency: e.g. 5
  - Seconds to Trust an Idle Pool Connection: e.g. 10
  - Connection Creation Retry Frequency: e.g. 5

  - Usually it takes some minutes until the connection is re-established

AGENDA

1. Working on Commandline
2. Snapshots
3. High available database connect
4. Hard-partitioning

Hard partitioning

Why Oracle VM for virtualization

- Oracle accepts hard-partitioning for licensing (ID 466538.1)
  - Hard-partitioning
    - explicitly assign physical CPU-cores to a VM
    - The VM can only use these cores (the default is all cores)
    - On top of the physical cores: virtual CPU's can be configured
  - You only have to pay licenses for the assigned cores
  - On VMWare there is no hard-partitioning accepted
    - All available cores have to be licensed (in a cluster all cores on all nodes!)

Hard-partitioning – What is a CPU in OracleVM?

- A physical core is a CPU in OVM

Socket 0		Socket 1	
Core 0	Core 1	Core 0	Core 1
CPU0	CPU1	CPU2	CPU3

physical hardware on server

In ovm presented CPUs

  - If using hyperthreading (HT), a core is visible as multiple CPU's
    - Carefully test, if it is better for your system
    - often it is better to deactivate HT

Socket 0				Socket 1			
Core 0		Core 1		Core 0		Core 1	
HT 0	HT 1	HT 0	HT 1	HT 0	HT 1	HT 0	HT 1
CPU0	CPU1	CPU2	CPU3	CPU4	CPU5	CPU6	CPU7

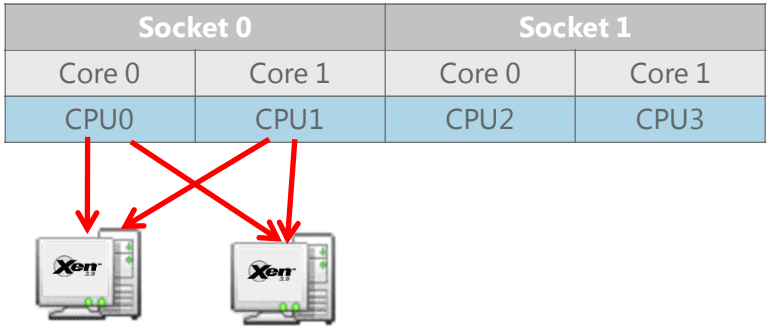
physical hardware on server

Threads presented as CPUs

In ovm presented CPUs

Hard-partitioning

- The same physical core can be assigned to more than one VM



- You only have to pay Oracle licenses for assigned cores (ID 466538.1)
  - But: live-migration is not permitted
    - no Dynamic Power Management / Dynamic Resource Scheduler
    - no server-maintenance without VM downtime
- Setup is not possible via GUI, edit vm.cfg
  - or use ovm_vmcontrol from the OracleVM3 utilities (patch 13602094)
  - www.oracle.com/technetwork/server-storage/vm/ovm-hardpart-168217.pdf

Hard-partitioning

- Per default, only virtual CPUs are allocated
  - Can run on any physical core (Benefit: using the least used core)

```
vcpus = 2
```

```
[root@elektra ~]# xm vcpu-list 3
Name                                ID  VCPU   CPU State   Time(s) CPU Affinity
0004fb00000600000a44c1da6a887b00     3     0     6   -b-     560.9 any cpu
0004fb00000600000a44c1da6a887b00     3     1     7   -b-     161.9 any cpu
```

- Edit the vm.cfg file and add "cpus" to limit the VM to certain cores

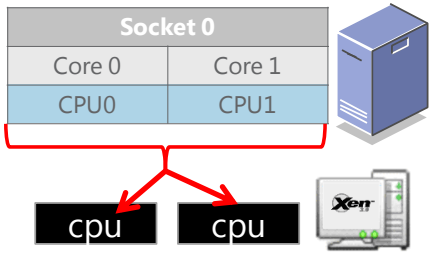
```
vcpus = 4
cpus = "1,3"
```

```
Name                                ID  VCPU   CPU State   Time(s) CPU Affinity
0004fb0000060000a980f351496991cc     4     0     1   -b-       1.0 1,3
0004fb0000060000a980f351496991cc     4     1     1   -b-       0.7 1,3
0004fb0000060000a980f351496991cc     4     2     3   r--      20.5 1,3
0004fb0000060000a980f351496991cc     4     3     1   ---       0.9 1,3
```

Hard-partitioning

- But can we assign virtual CPUs explicitely to a CPU Core?
  - Yes, we can. This feature is not well documented...
  - Instead of

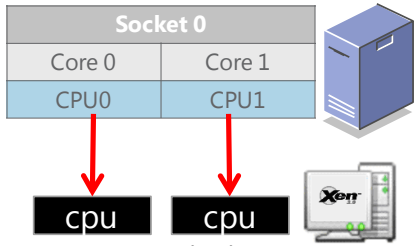
```
vcpus = 2
cpus = "0,1"
```




```
Name                                ID  VCPU   CPU State   Time(s) CPU Affinity
0004fb0000060000a980f351496991cc     4     0     1   -b-       1.0 0,1
0004fb0000060000a980f351496991cc     4     1     1   -b-       0.7 0,1
```

  - Use a slightly different syntax

```
vcpus = 2
cpus = ["0","1"]
```




```
Name                                ID  VCPU   CPU State   Time(s) CPU Affinity
0004fb0000060000a980f351496991cc     4     0     0   -b-       1.0 0
0004fb0000060000a980f351496991cc     4     1     1   -b-       0.7 1
```

Hard-partitioning

- Is explicit mapping Core-CPU better? It depends...
  - No overhead to switch tasks between physical cores
  - In OVM 2, substantial performance gain
    - In OVM 3, only slight difference

  - If Cores are assigned to only 1 VM, it is better to do this mapping
  - If Cores are assigned to more than 1 VM, it is better to use dynamic mapping
    - If 1 Core is used 100% by a process in a VM, it would probably be better if another VM can map 2 CPUs to the less used Core
    - The CPU scheduler inside the VM does not know that 1 CPU is slower because it is running on a overloaded core
  - → That's only theory! Test carefully what is better ☺

THANK YOU.

Trivadis AG

Martin Bracher

Europa-Strasse 5
8152 Glattbrugg

Tel. +41 31 928 09 60 / +41 44 808 70 20
Fax +41 44 808 70 21

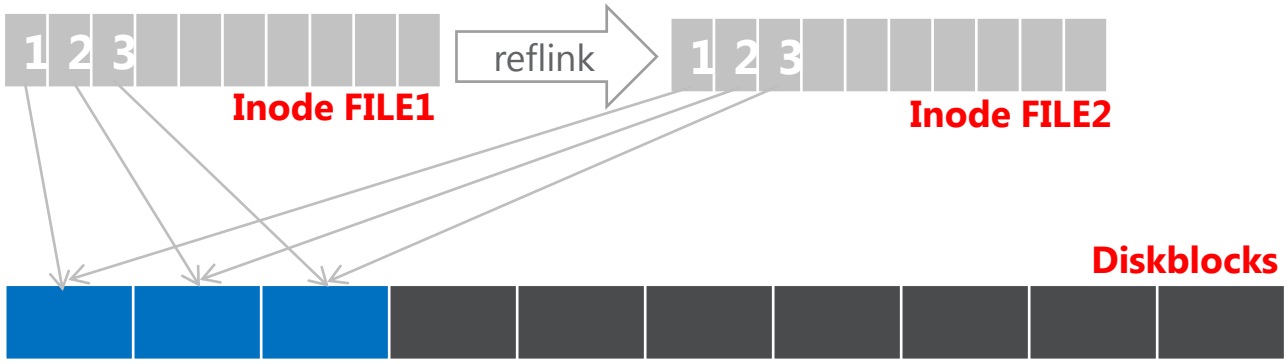
info@trivadis.com
www.trivadis.com

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

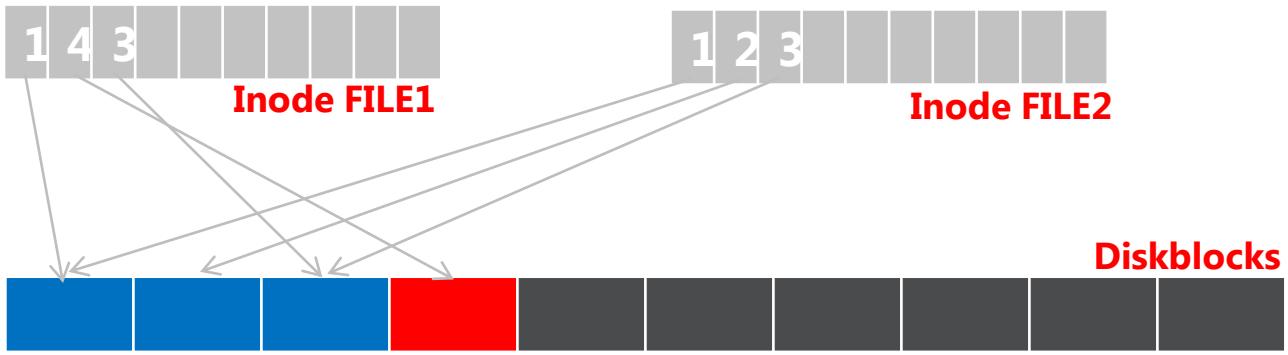
ANNEX

Snapshot of diskfiles with OCFS2: Reflink

reflink FILE1 FILE2

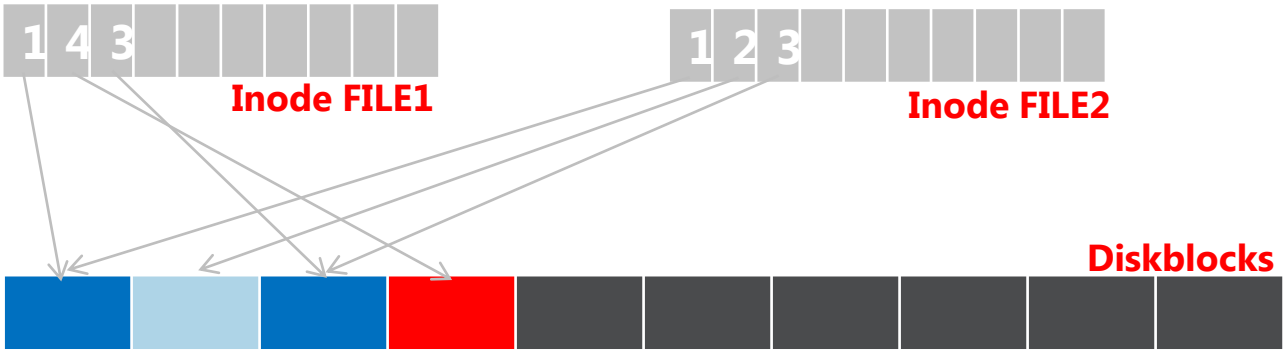


Change block 2 in FILE1

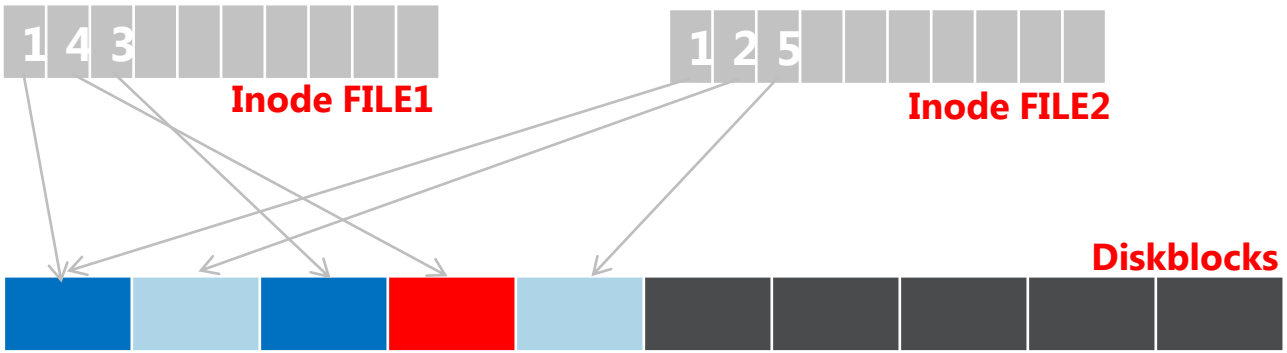


Snapshot of diskfiles with OCFS2: Reflink

Change block 2 in FILE2



Change block 3 in FILE2



Snapshot of diskfiles with OCFS2: Reflink

```
# df -k .
Filesystem           1K-blocks      Used Available Use% Mounted on …
                     976563200  63370240 913192960   7% …
# reflink file1 file2
# df -k .
Filesystem           1K-blocks      Used Available Use% Mounted on …
                     976563200  63370240 913192960   7% …
# ls -l file?
-rw-r--r-- 1 root root 2000000000 Mar 27 17:09 file1
-rw-r--r-- 1 root root 2000000000 Mar 27 17:11 file2
# dd if=/dev/zero of=file1 bs=1M count=1000 seek=500 conv=notrunc
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 17.5098 seconds, 59.9 MB/s
# df -k .
Filesystem           1K-blocks      Used Available Use% Mounted on …
                     976563200  64418816 912144384   7% …
# dd if=/dev/zero of=file2 bs=1M count=1000 seek=500 conv=notrunc
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 13.5592 seconds, 77.3 MB/s
# df -k .
Filesystem           1K-blocks      Used Available Use% Mounted on …
                     976563200  64418816 912144384   7% …
```