Während wir es gewohnt sind, bei Virtualisierungslösungen wie VirtualBox oder VMware mal schnell einen Snapshot einer virtuellen Maschine zu erstellen und bei Nicht-Gefallen des aktuellen Zustands wieder auf den Snapshot zurückzuwechseln, finden wir im Oracle VM Manager keine solche Funktion. Es gibt jedoch eine Cloning-Funktion, die hier möglicherweise weiterhelfen kann.

# Snapshot einer VM mit Oracle VM 3

Martin Bracher, Trivadis AG

Ein Snapshot speichert den Zustand der Disks einer virtuellen Maschine (VM). Diese Disks sind in der Regel durch Dateien auf dem Host abgebildet. Produkte wie VirtualBox oder VMware erzeugen Snapshots applikatorisch. Nach dem Erstellen wird die Original-Disk-Datei nicht mehr verändert – Änderungen sind in einer Differenzdatei gespeichert. Es ist auch möglich, von einem Snapshot wieder einen Snapshot zu erstellen, wobei sich die Virtualisierungslösung die Blöcke aus der eigenen und der vorangegangenen Differenzdatei sowie aus der Originaldatei zusammensuchen muss. Vorteil dieser Lösung ist, dass sie unabhängig vom Betriebs- oder Dateisystem funktioniert. Nachteil ist jedoch, dass solche Snapshots durch den Verwaltungs-Overhead eine schlechtere Performance haben. Wenn man einen Snapshot löscht, müssen die geänderten Blöcke aufwändig wieder in die Originaldatei zurückgeschrieben werden.

### Vom OVM3-Cloning zum Snapshot

Der aktuelle OVM3-Manager ist nicht in der Lage, den Zustand einer VM zwischenzuspeichern. Stattdessen gibt es die Möglichkeit, die gesamte VM zu klonen. Dabei werden die Disk-Files kopiert und ein neues Konfigurationsfile (die Definition der virtuellen Maschine) erzeugt. Wir halten also nicht den Zustand unserer VM fest, sondern erzeugen eine neue VM (sogenanntes „Template"), die aktuell den identischen Inhalt wie das Original hat und beim Starten neu konfiguriert werden muss (Anpassen von Netzwerk-Konfiguration (neue MAC-Adresse), Hostname etc.).

Falls unsere VMs auf einem lokalen LUN-Storage (SAN, iSCSI) liegen, verwendet OVM ein „ocfs2"-Clusterfile-System. Dieses bietet seit Kurzem eine sehr praktische Funktion: Es können von Files schreibbare Snapshots erstellt werden, in der „ocfs2"-Terminologie „Reflinks" genannt. Wenn man eine Datei auf diese Weise kopiert, dann werden nicht der Inhalt, sondern lediglich die Inodes (die Zuordnungstabelle von den Blöcken zur Datei) kopiert (siehe Abbildung 1).

Die neue Datei benötigt also anfänglich keinen zusätzlichen Platz (siehe Listing 1), da sie auf dieselben Blöcke verweist wie die Ursprungsdatei. Im weiteren Verlauf werden bei Änderungen an einer der Dateien dann nicht mehr die Originalblöcke verän-

```
# df -k .
  1K-blocks   Used Available Use%
  976563200 63370240 913192960  7%
# reflink file1 file2
# df -k .
  1K-blocks   Used Available Use%
  976563200 63370240 913192960  7%
```

*Listing 1*

```
# dd if=/dev/zero of=file1 bs=1M
count=1000 seek=500 conv=notrunc
# df -k .
  1K-blocks   Used Available Use%
  976563200 64418816 912144384  7%
# Diff: 1048576 blocks
```

*Listing 2*

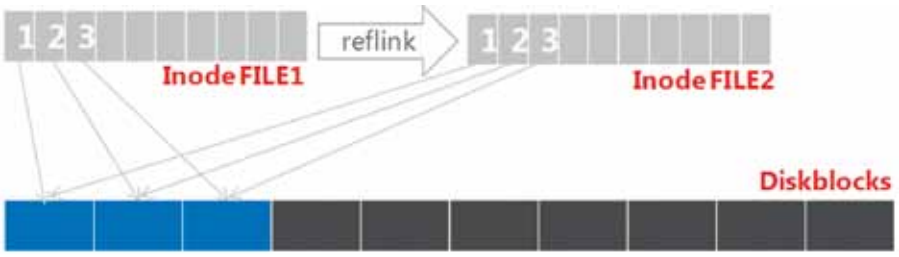


*Abbildung 1: Reflink-Kopie*

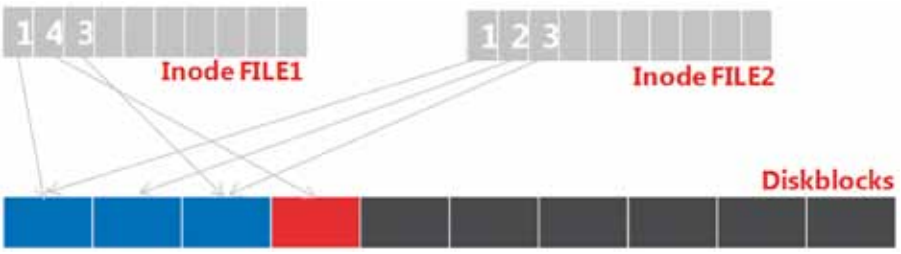


*Abbildung 2: Nach Änderung eines gemeinsamen Blocks*

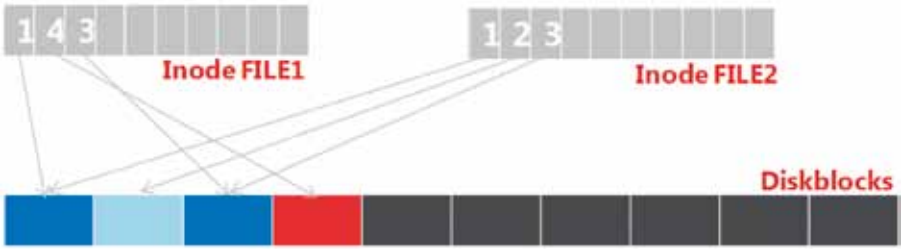


*Abbildung 3: Nach Änderung eines nicht mehr gemeinsamen Blocks*

```
# dd if=/dev/zero of=file2 bs=1M count=1000 seek=500
conv=notrunc
# df -k .
    1K-blocks      Used Available Use%
    976563200  64418816 912144384   7%
```

*Listing 3*

```
#!/bin/bash
orig=$1
clone=$2
origcfg=$(grep -l «OVM_simple_name = <$orig>» \
/OVS/Repositories/*/VirtualMachines/*/vm.cfg)
clonecfg=$(grep -l «OVM_simple_name = <$clone>» \
/OVS/Repositories/*/*/*/vm.cfg )
```

*Listing 4*

```
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")
```

*Listing 5*

```
for i in $origdisks; do
   #extract the filename
   origfile=${i#*:}
   origfile=${origfile%%,*}
   #extract the VM devicename
   origvmdevice=${i%,*}
   origvmdevice=${origvmdevice#*,}
   for j in $clonedisks; do
     #if «,$origvmdevice,» found, then …
     if [ -z ${j/*,$origvmdevice,*} ]; then
       #extract the filename
       clonefile=${j#*:}
       clonefile=${clonefile%%,*}
       <WhatToDo>
     fi
   done
 done
```

*Listing 6*

dert (siehe Abbildung 2), sondern für diese Datei neue Blöcke alloziert (Copy-On-Write, siehe Listing 2, Seite 29).

Erst wenn nur noch eine Datei (Inode) auf den Originalblock verweist, kann dieser überschrieben werden (siehe Abbildung 3). Der Platzverbrauch auf der Disk verändert sich dabei nicht (siehe Listing 3).

Diese effiziente Möglichkeit steht uns nur zur Verfügung, wenn wir mit „ocfs2" arbeiten oder ein vom Storage-Hersteller geliefertes Storage-Plug-in installiert haben, das in der Lage ist, ebenfalls Snapshots zu erzeugen. Bei einem Repository mit dem generischen NFS-Plug-in müssen die Dateien ganz normal kopiert werden.

Solange wir die geklonte VM nicht starten, entsprechen die Disks genau dem Zustand des Originals zum Zeitpunkt des Klonens. Wenn wir den Klon als Snapshot verwenden wollen, dürfen wir ihn daher niemals starten. Aus diesem Grund empfiehlt es sich, den Klon als Template und nicht als neue VM anzulegen.

Um zu einem Snapshot-Zeitpunkt zurückzukehren, sind folgende Schritte selbst zu implementieren: Die VM stoppen, die geklonten Files wieder an den Ursprungsort zurückkopieren und die VM neu starten. Die größte Herausforderung besteht darin, die Dateinamen der zu tauschenden Diskfiles zu finden. Aber dies lässt sich mit Shell-Scripts einfach automatisieren.

In einem ersten Schritt müssen die Konfigurationsfiles (die Definition der virtuellen Maschine) vom Original und vom Klon gefunden werden. OVM3 arbeitet mit einer Unique ID der VM und nicht mit dem von uns vergebenen und im OVM-Manager angezeigten Namen. Dieser befindet sich als „OVM_simple_name" im Konfigurationsfile (siehe Listing 4).

Die Konfigurationsfiles liegen unter „/OVS/Repositories/<repository-uuid>/VirtualMachines/*/" (beziehungsweise „Templates" statt „VirtualMachines", falls es sich um ein Template handelt). In diesen Konfigurationsfiles finden wir dann die Disk-Definition, beispielsweise „file:/OVS/Repositories/…/VirtualDisks/<uuid>.img,xvda,w" (siehe Listing 5).

**Newsticker**

**Datenbank 11*g* R2 und Fusion Middleware 11*g* für Oracle Linux 6 zertifiziert**

Mit der Zertifizierung der Datenbank 11*g* R2 sowie der Fusion Middleware 11*g* für das hauseigene Oracle Linux 6 übernimmt Oracle den Support für Installationen auf dem Unbreakable Enterprise Kernel. Dazu Wim Coekaerts, Senior Vice President of Linux and Virtualization Engineering: „Die Zertifizierung ist das Ergebnis von stringenten Tests auf Oracle Linux mit dem Unbreakable Enterprise Kernel."

Laut Oracle soll auch eine eine Zertifizierung der Datenbank 11*g* R2 und der Fusion Middleware 11g unter Red Hat Enterprise Linux (RHEL) 6 folgen. Gleichzeitig kündigt Oracle an, die Kompatibilität von Oracle Linux mit Red Hat Linux in Zukunft aufrechterhalten zu wollen.

Bevor wir nun den Snapshot wiederherstellen, muss die VM gestoppt sein. Dies kann mit dem Xen-Tool „xm" erfolgen: „xm destroy $(basename $origfile.cfg)". Der OVM-Manager zeigt dann innerhalb kurzer Zeit den gestoppten Zustand an.

Vorsicht: Vielleicht läuft die VM auch auf einem anderen Knoten im Cluster. Mit „xm" lassen sich nur VMs auf dem lokalen Host steuern. Ebenfalls zu berücksichtigen ist, dass wir bei Shared Storage (zu erkennen an „w!", zum Beispiel RAC-Umgebung) alle darauf zugreifenden VMs stoppen müssen.

In einem Loop über alle Diskfiles können wir dann die zueinander passenden finden. „Zueinander passend" heißt, dass sie innerhalb der VM gleiche Device-Namen besitzen, etwa „xvda" (Variable „origvmdevice", „clonevmdevice", siehe Listing 6).

Im Bereich „<WhatToDo>" können wir dann die Art unseres Snapshots definieren. Hier: Ersetzen des Originals durch den Klon und der Klon soll unverändert erhalten bleiben (siehe Listing 7) – oder Tauschen der Disk von Original und Klon (siehe Listing 8). Danach kann die VM entweder mit „xm create $origfile" oder über den OVM-Manager wieder gestartet werden.

Es ist durchaus möglich, dass wir solche Snapshots ohne einen über OVM-Manager erstellten Klon implementieren können. Wir erstellen uns auf dem Storage-Repository ein Verzeichnis „snapshots" und erzeugen darin unsere Disk-Snapshots (siehe Listing 9). Für deren Konsistenz sind wir selbst verantwortlich (Stoppen der VM, Freeze). Wir müssen uns auch selbst merken, welche Snapshot-Kopie zu welcher Ursprungsdatei gehört. Aber mit einer geeigneten Namenskonvention dürfte dies kein Problem sein.

Mit der selbst implementierten Variante ist es nun auch möglich, den Zustand einer laufenden VM, also inklusive Memory, zu speichern und als Snapshot abzulegen. Dazu müssen wir die VM in den „Suspend"-Zustand bringen. Via OVM-Manager liegt danach der Memory-Inhalt in „VirtualMachines/<uuid_of_vm>/state", von dem wir nun ebenfalls eine Reflink-Kopie erstellen können, zusammen mit den Diskfiles. Der Betrieb der VM kann nun mit „resume" wieder fortgesetzt werden. Um zu einem solchen Snapshot zurückzukehren, müssen wir natürlich die zum Snapshot passenden VM-Diskfiles wieder an die Ursprungsposition kopieren und danach die VM mit dem gespeicherten Memory-Inhalt wieder starten (siehe Listing 10). Auch dies lässt sich natürlich über ein Script automatisieren.

```
rm -f $origfile
reflink $clonefile $origfile
```

*Listing 7*

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

*Listing 8*

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

*Listing 9*

```
vm=0004fb0000060000a5d437f721235948
disk=0004fb0000120000e2a9e7b
9a0252323.img
snapprefix=snapshots/oel2clo-
ne.20110331-141832
xm destroy ${vm}
rm -f VirtualDisks/${disk}
reflink  ${snapprefix}.${disk}
VirtualDisks/${disk}
xm restore ${snapprefix}.state
```

*Listing 10*

## Fazit

OVM bietet zwar bezüglich Snapshots auf den ersten Blick weniger als Mitbewerber, auf den zweiten Blick haben wir dafür jedoch eine sehr flexible Lösung. Im Gegensatz zu anderen Lösungen, die nach einem Snapshot geänderte Blöcke in speziellen Differenzdateien speichern und beim Entfernen des Snapshots die geänderten Blöcke aufwändig wieder in die Ursprungsdatei zurückkopieren müssen, haben wir Dank „ocfs2" zwei aus Sicht des Betriebssystems und der Virtualisierung unabhängige Dateien. Dies wirkt sich positiv auf die Performance aus. Wir können auch problemlos solche Dateien in ein Backup einbeziehen, ohne Abhängigkeiten zur Ursprungsversion berücksichtigen zu müssen. Mit etwas Handarbeit oder einem Script haben wir also dieselben Funktionalitäten zur Verfügung, wie wir sie auch bei anderen Virtualisierungslösungen haben.

**Weitere Informationen**

- http://www.trivadis.com/technologie/download-area.html
- https://edelivery.oracle.com/linux

Martin Bracher
info@trivadis.com