

Oracle VM – die Virtualisierungslösung von Oracle – besteht aus zwei Komponenten. Einerseits dem OVM Server, dem Host für die virtuellen Maschinen, und andererseits dem OVM Manager, der Komponente zur Verwaltung des OVM Servers.

Backup einer Oracle-VM3-Umgebung

Martin Bracher, Trivadis AG

Das Thema „Backup und Recovery“ ist in den Oracle-Dokumentationen nur sehr rudimentär beschrieben. Dieser Artikel betrachtet das Thema etwas genauer und zeigt, was man wo und wie sichern muss. Die zu sichernden Komponenten sind zum einen die Infrastruktur, auf der die VMs laufen (OVM Server Installation (Disk), Pool Filesystem, Repository Filesystem und optional die in VMs verwendeten LUNs). Zum anderen der OVM Manager (lokale Disks (OS/Software) und die Repository-Datenbank) sowie die virtuellen Maschinen (VM) selbst (Definition, Disk-Files und die Daten in der VM, siehe Abbildung 1).

Der OVM Server

Hier muss beim Backup der Host selbst gesichert werden, also das Betriebssystem, das sich normaler-

weise über „ssh“ starten, da auf dem OVM Server kein „cron“ installiert ist. Man kann aber auch die Strategie verfolgen, kein Backup des Servers zu machen, sondern ihn im Problemfall einfach aus der Konfiguration zu löschen, danach neu zu installieren und wieder in den Pool einzufügen.

Pool Filesystem

Das Pool Filesystem ist das gemeinsame Filesystem eines Cluster-Pools. Es ist somit ein Shared Filesystem mit OCFS2. Es befindet sich auf einem SAN, iSCSI oder auf NFS. Wenn es auf NFS liegt, ist es eine etwa 12 GB große Datei, die ein OCFS2-Filesystem enthält und auf den Servern

sdb bs=1M | gzip >pool_fs.gz“ wegkopiert und ebenfalls komprimiert. Es bleiben erfahrungsgemäß 10 bis 15 MB übrig. Danach kann man im regulären Betrieb ein regelmäßiges Backup des Filesystems etwa mit „tar“ erstellen.

Repository Filesystem

Auf dem Repository Filesystem sind die virtuellen Maschinen abgelegt (Definition und virtuelle Disks). Es ist ebenfalls „shared“ zwischen den Servern und entweder ein NFS Share oder ein OCFS2-Filesystem. Zu Beginn sollte man die Verzeichnisstruktur sowie die Datei „ovsrepo“ sichern.

Darüber hinaus werden die Informationen zum OCFS2-Filesystem ge-

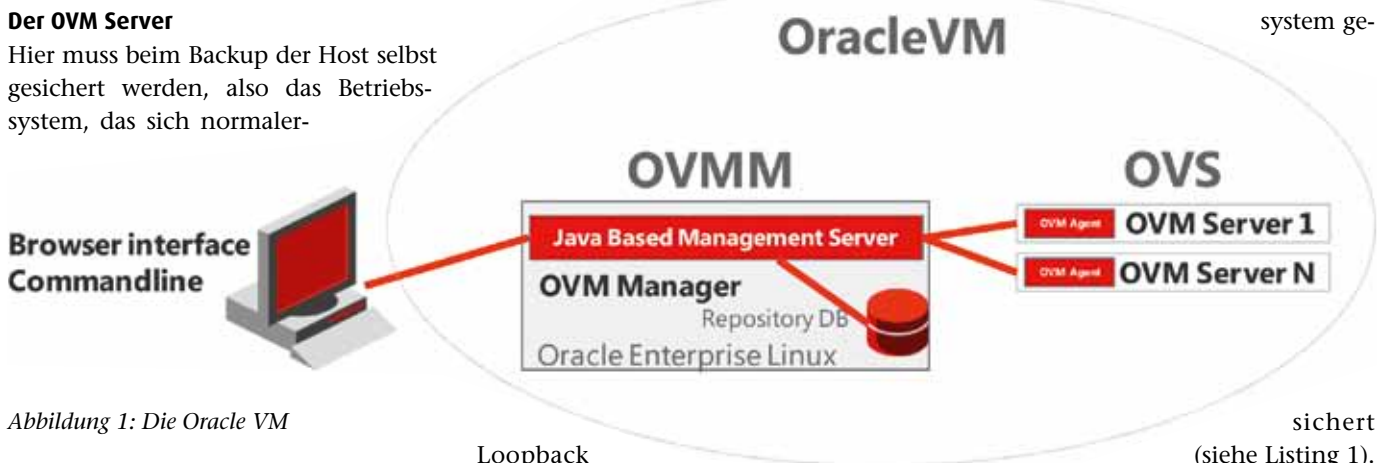


Abbildung 1: Die Oracle VM

weise auf lokalen Disks befindet. Es ist jedoch nicht möglich, zusätzliche Software auf einem OVM Server zu installieren. Man kann aber lokal vorhandene Tools („tar“, „gzip“, „cp“, „scp“ etc.) verwenden. Dieses Backup kann man per „scp“ wegekopieren oder auf dem Repository Filesystem ablegen und dort via NFS abholen: „tar --one-file-system -zcvf /OVS/repository/<uuid>/root_<HOSTNAME>.tgz /“.

Ein zeitgesteuertes Backup muss man beispielsweise vom OVM Ser-

ver aus über „ssh“ starten, da auf dem OVM Server kein „cron“ installiert ist.

Dieses Filesystem beinhaltet die Informationen zum Cluster und wird zum Handling der Node-Membership benötigt. In der Terminologie von RAC hat es die Funktionalität der Cluster-Registry und der Voting Disks.

Dieses Filesystem ist initial nach der Installation bei gestopptem Cluster zu sichern. Bei NFS kopiert und komprimiert man die erwähnten Datei, ansonsten wird mit dem „dd“-Befehl die entsprechende LUN mit „dd if=/dev/

sichert (siehe Listing 1).

Mit diesen Informationen lässt sich dann das Filesystem notfalls wieder neu erstellen (siehe Listing 2).

Regelmäßig sollten nun die Dateien gesichert werden – sowohl die Definitions-Files in den Verzeichnissen „VirtualMachines“, „Templates“ und „Assemblies“ als auch die Dateien in ISOs. Die Dateien in VirtualDisks können nur gesichert werden, wenn sie nicht in Verwendung sind. Wie man geöffnete Files sichern kann, folgt später.

Bei Verlust des Filesystems muss die-

```
# tuneefs.ocfs2 -Q „B=%B, T=%T, N=%N V=%V U=%U\n“ /dev/mapper/1IET_00010001
b=4096, T=131072, N=32 V=0VS56e8973883d87 U=0004FB00000500002DC56E8973883D87
```

Listing 1

ses also neu erzeugt werden, danach ist das Backup der Directory-Struktur sowie der (statischen) Files durchzuführen. Zum Schluss sind die Snapshots (siehe später) wieder an den Platz der Original-Diskdateien zu kopieren.

OVM Manager

Der OVM Manager wird auf einem Red-Hat- oder Oracle-Enterprise-Linux installiert. Es sollte ein Backup der ganzen Filesysteme erstellt werden, wie für jeden produktiven Server auch. Bei Verlust können diese Dateien vom Backup zurückgespielt werden. Der Manager kann eine existierende Oracle-Standard- oder Enterprise-Edition-Datenbank als Repository-Datenbank verwenden. Es wird nur ein zusätzliches Schema (OVS) installiert. Diese Datenbank kann sich lokal auf dem Manager Host oder auf einem anderen Server befinden.

Bis OVM3.1 wurde für Testzwecke eine Oracle XE Edition mitgeliefert (für produktive Systeme nicht unterstützt, läuft aber perfekt). Ab Version 3.2 wurde diese leider durch MySQL ersetzt. Wieso leider? Die meisten Benutzer verwenden OVM für Oracle-Datenbanken, also was soll man mit so einem Exoten, den kaum ein Oracle-DBA bedienen kann? Für den Autor ein absolutes No-Go.

Die Oracle-Datenbank kann wie gewohnt mit „rman“ oder Export gesichert werden. Falls die Datenbank verloren geht, kann sie mit den dem Oracle-DBA bekannten Verfahren wiederhergestellt werden. Wenn man die letzten Transaktionen verloren hat, ist das nicht schlimm. Der Manager ist tolerant genug, auch mit einem etwas älteren Backup-Stand weiterarbeiten zu können. Er kann die geänderten Daten von den OVM Servern zurücksynchronisieren. Vor dem Restore sollte man den

Manager stoppen („/etc/init.d/ovmm stop“) und nach dem Restore wieder starten („/etc/init.d/ovmm start“).

Virtuelle Maschinen

Eine VM besteht aus zwei Teilen: Erstens dem Storage, also (virtuellen) Disks. Dies können eine Datei oder ein Block-Device auf dem OVM Server sein, per Netzwerk ein externer NFS Share oder eine iSCSI LUN. Zweitens aus der virtuellen „Hardware“ ein Konfigurationsfile, in dem die CPU, Netzwerkkarten, Memory etc. definiert sind.

Beim Backup müssen wir zwischen einem Backup der gesamten VM (zwecks „Bare-Metal“-Recovery) und einem Backup des Inhalts der (virtuellen) Disks (zwecks Restore einzelner Filesysteme/Dateien) unterscheiden. Für die virtuelle Hardware ist die Datei „<repo-fs>/VirtualMachines/<vm-uuid>/vm.cfg“ zu sichern. Sie wird zwar aus den Informationen im Repository erzeugt, jedoch zum Restore des Repository Filesystems benötigt. Seit OVM ein vernünftiges Commandline-Interface zur Verfügung stellt, sollten wir die Befehle zur Erstellung der VM ebenfalls speichern (siehe Listing 3). Dieses Script würde man brauchen, wenn man die gesamte Umgebung neu aufbauen müsste.

Falls die Disks eine LUN auf dem OVM Server sind, können sie bei gestoppter VM mit dem „dd“-Befehl gesichert werden. Falls der Storage eine Möglichkeit für Snapshot bietet, könnte dies sogar online erfolgen.

Wenn die virtuellen Disks ein File auf dem Storage-Repository sind, kann dieses gesichert werden. Falls es sich um SAN-/iSCSI-Storage mit einem OCFS2-Filesystem handelt, können diese Files auch durch einen Snapshot gesichert werden, der anschließend auf ein an-

deres Medium wegkopiert wird. Der Snapshot selbst ist noch kein Backup, denn er befindet sich immer noch auf demselben Storage.

Um einen solchen Snapshot zu erstellen, kann man aus der VM ein Template klonen und danach die (nicht geöffneten) Files dieses Klons wegsichern. OVM bietet die Möglichkeit, das Storage-Repository per NFS zu exportieren. Man erinnere sich: Es ist nicht zertifiziert, Backup-Software auf dem OVM Server zu installieren. Aber wir können die Dateien via NFS sichern.

Der Restore einer VM erfolgt dann, indem man diese stoppt und danach die Diskfiles der VM durch die gesicherten Snapshot-Files des Klons ersetzt. Danach ist die VM neu zu starten und sie enthält wieder den Zustand vom Zeitpunkt des Snapshots.

Zum Sichern einzelner Filesysteme oder einzelner Files sowie von Datenbanken innerhalb der VM unterscheidet sich das Backup-Konzept nicht von dem eines physischen Hosts. Man installiert innerhalb der VM entsprechende Backup-Software und sichert die Dateien auf Storage oder direkt auf ein Tape.

Ein Problem ist, dass Backup-Clients häufig pro Hostname lizenziert sind. Mit etwas Scripting-Aufwand lassen sich diese Kosten umgehen: Man erstellt einen Snapshot der Disk-Datei, hängt diesen dann an eine VM mit lizenziertem Backup-Client und macht den Backup von dort aus. Wenn man, wie oben beschrieben, jeweils die gesamte VM mit Snapshots sichert, kann man sich den Filesystem-Backup auch sparen, bei Bedarf den Snapshot „read-only“ hinzufügen und die Dateien von dort zurückkopieren.

Spezielle Recovery-Situationen

- *Verlust der Repository-Datenbank (ohne Backup)*

In dem Fall muss eine neue Datenbank ohne „OVS“-Schema erzeugt werden. Ebenso ist der OVM Manager neu zu installieren, und zwar per „--uuid 0004EC00000100001C87-

```
mkfs.ocfs2 -J block64 -b 4096 -L 0VS56e8973883d87 \
-U 0004fb00000500002dc56e8973883d87 -T vmstore \
-N 32 /dev/mapper/1IET_00010002
```

Listing 2

```

create Vm name=slot011 repository=slotreposan01 domainType=XEN_PVM osType=0L_5 bootOrder=DISK cpuCount=2
highAvailability=yes memory=2048 on ServerPool name=slot

create Vnic name=00:21:f6:01:00:0b network=vlanpublic

add Vnic name=00:21:f6:01:00:0b to Vm name=slot011

create VmDiskMapping slot=0 storageDevice=slot011_xvda name=xvda on Vm name=slot011

```

Listing 3

C2AEE23B7“ mit der bisherigen UUID „./runInstaller“. Diese ist beispielsweise in „./etc/sysconfig/ovmm“ hinterlegt. Nach dieser Neu-Installation können nun die bestehenden OVM Server „rediscovered“ werden. Da wir immer noch mit derselben UUID arbeiten, beginnt der Manager, sein Repository aufgrund der auf dem Server gespeicherten Informationen zu rekonstruieren.

- *Verlust des OVM Manager (ohne Backup), Datenbank noch vorhanden*
Wenn der Backup fehlen sollte, kann man den Manager neu installieren. Zu diesem Zweck muss man aber zuerst das noch existierende OVS-Schema exportieren, dieses dann löschen und danach den Manager mit der alten UUID neu installieren. Nach der Installation stoppt man den Manager nochmals, löscht das neu erstellte OVS-Schema und importiert den Backup des alten OVS-Schemas wieder. Nach dem Start des Managers steht der bisherige Zustand wieder zur Verfügung.
- *Verlust von OVM Manager und Datenbank (ohne Backup)*
Wenn beide Komponenten nicht gesichert wurden, installiert man eine neue Datenbank und einen neuen Manager mit der alten UUID. Diese findet man beispielsweise auf dem Repository Filesystem in der „.ovsrepo“-Datei. Nach einem Re-Discover der bestehenden Server sollte alles wieder weitgehend in Ordnung sein.
- *Desaster-Szenario: Verlust der ganzen Umgebung*
Mit den Backups der Snapshot-Files und den Create-Scripts für die VM lässt sich eine neue Umgebung aufbauen. Dazu eine Neu-Installation/-Konfiguration von OVM Manager und Servern durchführen. Danach Kopieren der Diskfiles vom Back-

up ins Repository unter VirtualDisks und Durchführen eines Re-Scans des Repository. Diese Disks sind dem Manager danach wieder bekannt. Jetzt müssen nur noch die VMs neu definiert und die Diskfiles zugewiesen werden.

Fazit

Im Prinzip müssen sich die Backup-Konzepte von physischen Servern und OVM praktisch nicht unterscheiden. Der (virtuelle) Server wird mit klassischen Backup-Clients gesichert. Wenn man nun einen physischen Server oder eine VM verliert, besorgt man sich einen neuen physischen/virtuellen Server und installiert diesen neu beziehungsweise macht einen Restore des Backups. OVM bietet uns aber einige zusätzliche Möglichkeiten, um das Backup-Konzept zu optimieren oder zu erweitern, wie beispielsweise die Möglichkeit für Snapshots ganzer Diskfiles.

Der Verlust des OVM Manager ist kein großes Problem. Die Server und VMs laufen unabhängig von diesem weiter und der Manager kann aufgrund der Informationen auf den Servern weitgehend wieder neu aufgebaut werden.

Martin Bracher
martin.bracher@trivadis.com



■ Neu: ADF Mobile 1.1 ist verfügbar

Mit dem neuen JDeveloper-Release 11.1.2.4 hat Oracle nicht nur zahlreiche Bugs behoben, sondern auch die Version 1.1 des ADF Mobile Frameworks mit neuen Features veröffentlicht. Neben der Unterstützung der aktuellen Phone- und Tablet-Versionen gestaltet sich die Entwicklung mobiler Unternehmensanwendungen einfacher. Die ersten Testdrives weisen zudem eine verbesserte Performance und Stabilität auf. Das ADF Mobile Framework ist als JDeveloper-Extension verfügbar und unterstützt mittels Single-Source-Ansatz die Entwicklung von iPhone und Android Apps auf Basis von HTML5 und Java. Durch die Verwendung des Cordova-Frameworks (ehemals PhoneGap) wird zudem eine Vielzahl nativer Geräte-Services unterstützt.

Zu den Neuheiten zählen Push-Benachrichtigungen und die sogenannten „Badges“. Dabei handelt es sich um Benachrichtigungssymbole, die die oftmals in Kombination mit Push-Benachrichtigungen zum Einsatz kommen und die Anzeige der Anzahl neuer Anwendungsevents ermöglicht. Zudem lassen sich nun Dateien über eine Dateivorschau mit der nativen Vorschau-App anzeigen.

Eine neue Funktionalität der Version 1.1 ist die Bereitstellung vom Mobile Application Archive (MAA). damit ist es möglich, eine Basis-Applikation in Form einer Vorlage anderen ADF-Mobile-Projekten zur Verfügung zu stellen. Das konsumierende Projekt kann damit bestimmte Elemente anpassen und als App-Rahmen für neue Apps mit den eigenen Zertifikaten als Herausgeber verwenden.

Zudem hat Oracle die technische Infrastruktur geändert. Neben einer verbesserten Performance ist nun auch eine iPhone5- und iPad-Mini-Unterstützung möglich.

Weitere Informationen unter <http://www.doag.org/home/aktuelle-news/article/oracle-adf-mobile-11-ist-verfuegbar.html>