

Praxistauglich: Oracle auf VMware

Patrick Schwanke
Quest Software GmbH
Im Mediapark 4e, 50670 Köln

Schlüsselworte:

VMware, vSphere, Virtualisierung, Private Cloud, Templates, Datenbank, RAC, SAP, ASM, NFS

Einleitung

Die Virtualisierungswelle erfasst zunehmend auch Oracle-Datenbanken. Nicht für alle, aber für viele Datenbanken bietet der Betrieb in VMware interessante Vorteile.

Selbst produktive RAC- und SAP-Datenbanken sind auf dieser Infrastruktur mittlerweile unterstützt.

Der Autor berichtet aus seinen Praxiserfahrungen, wann Virtualisierung in diesem Umfeld sinnvoll ist und wann nicht, was speziell für Oracle-Datenbanken beachtet werden sollte und wie man virtualisiert und gleichzeitig eine Brücke zurück zu physischen Umgebungen offen hält.

Wann lohnt sich Virtualisierung?

Hierzu sollte man sich die Vor- und Nachteile der Virtualisierung, speziell im Oracle-Datenbankumfeld, genau anschauen und dann abwägen, ob es sich im konkreten Fall lohnt.

Pro Virtualisierung sprechen ganz allgemein (auch außerhalb der Oracle-Welt) sicherlich diese Punkte:

- Die verfügbaren (und budgetierbaren) Ressourcen können besser ausgenutzt werden, da Server stärker ausgelastet sind und auf Lastschwankungen einfach reagiert werden kann (Stichwort: vMotion, DRS siehe unten)
- Ein virtueller Datenbankserver mitsamt Datenbank kann erheblich schneller erstellt, angepasst und auch wieder abgebaut werden als ein entsprechender physischer Server. Die Reaktionszeiten und damit auch die Time-to-Market sind deutlich reduziert.
- Einzelne VMs sind untereinander vollkommen isoliert, auch die Ressourcenzuordnungen lassen sich sehr fein steuern.
- Muss die Hardware irgendwann getauscht werden, fallen keine Auszeiten dafür an. Das wirkt sich positiv auf Verfügbarkeit und Servicequalität aus.
- Außerdem bietet sich eine Umstellung auf virtuelle Umgebungen als erster Schritt in Richtung Private Cloud (Infrastructure-as-a-Service) an. Ein ganz wichtiger Aspekt hierbei ist gerade die Standardisierung, also die Beschränkung auf wenige VM-Typen mit bestimmten Standard-Ausstattungen.

Dass im Oracle-Umfeld bisher nicht flächendeckend virtualisiert wurde, hat natürlich seine Gründe, und damit wären wir bei den Argumenten kontra Virtualisierung:

- Bis einschließlich VMware 3.5 erzeugt jeder I/O-Request in einer VM einen CPU-Overhead. Hohe I/O-Last, die für Datenbanken typisch ist, führte zu hohem CPU-Overhead. Für Hochlast-Datenbanken potenziell ein KO-Kriterium, ist dieser Aspekt aber seit VMware vSphere 4 entschärft.
- Es gibt harte Obergrenzen für die Anzahl CPU-Kerne pro VM. Diese haben sich bisher mit jeder VMware-Version nach oben verschoben, so dass dies mittlerweile nur für wenige Datenbanken eine Einschränkung darstellt. Man sollte aber auch zukünftige Skalierungsanforderungen der Datenbank im Auge behalten.
- Support und Zertifizierung seitens Oracle, VMware und Dritten, z.B. SAP (hierzu unten mehr)
- Neue Schnittstellen zwischen Oracle-Datenbankteam sowie den für VMware und Storage zuständigen Teams.
- Es entstehen natürlich auch VMware-seitige Lizenzkosten

Was ist unterstützt, was ist zertifiziert?

Grundsätzlich unterstützt Oracle die Ausführung von Oracle-Software auf VMware-Umgebungen. Nachdem Oracle RAC für längere Zeit eine Ausnahme von dieser Regel darstellte, hat sich dies mit dem Erscheinen des Release 11.2.0.2 geändert. RAC-Datenbanken in VMware sind ab dem Release 11.2.0.2 unterstützt. Wichtigste Referenz hierfür ist [1].

Speziell für produktive SAP-Oracle-Datenbanken gibt [2] Auskunft darüber, welche Versionen unterstützt sind. Stand August 2011 kommen die Datenbankversionen 10.2.0.4 und höher bzw. 11.2.0.1 und höher in Frage, für produktive SAP RAC-Datenbanken wiederum alles ab 11.2.0.2 aufwärts.

Keine Brücken einreißen

Außer bei sehr kleinen und unkritischen Datenbanken ist es empfehlenswert, sich immer eine Brücke zurück in die nicht-virtuelle Welt stehen zu lassen. Dafür ist es entscheidend, dass die Datenbankdateien – also die Daten-, Temp-, Control-, Redolog-Dateien etc. – nicht in der VM selbst, in den sogenannten vmdk-Dateien liegen, sondern:

- Entweder in Oracle ASM, indem sogenannte VMware Raw Device Mappings anstelle normaler virtueller Platten verwendet werden und aus der VM heraus eine ASM-Instanz auf diese „rohen Platten“ zugreift;
- oder auf NFS, wobei direkt aus der VM heraus ein NFS-Export gemountet wird. Seit Oracle 11g empfiehlt sich hierfür Oracle Direct NFS. Außerdem sollte ein 10Gbit-Netzwerk Pflicht sein.

Datenbanken sind I/O-lastig

Oben wurde bereits angedeutet, dass in älteren VMware-Versionen jeder I/O-Request zu CPU-Overhead führte. Mit vSphere 4 kann man dies umgehen, indem man die sogenannten paravirtualisierten Treiber nutzt. Diese Treiber wissen, dass sie in einer VM laufen, und reichen den I/O-Request direkt an den Hypervisor durch, ohne dass dieser mit CPU-Zyklen eingreifen muss.

Hat man nun die Datenbankdateien wie oben beschrieben in ASM abgelegt (oder bei einer kleinen Datenbank als Dateien auf einer normalen virtuellen Platte abgelegt), empfiehlt es sich, bei der Erstellung der VM als SCSI-Controllertyp „VMware Paravirtual“ auszuwählen. Außerdem müssen die VMware-Tools in die VM installiert werden, da diese den entsprechenden Treiber (pvSCSI) mitbringen.

Liegen die Datenbankdateien auf einem NFS-Server, der von der VM direkt gemountet wird, ist nicht SCSI-I/O, sondern Netzwerk-I/O entscheidend für die I/O-Performance. Hier sollte der VMXNET3-Netzwerktreiber ausgewählt werden. Damit Netzwerk grundsätzlich mit Fibre-Channel konkurrieren kann, sollte es ein 10Gbit-Netzwerk sein.

VMware-Clustering

vSphere bietet zwei Clustering-Möglichkeiten an, die einzeln oder auch in Kombination genutzt werden können. Diese ersetzen zwar kein Oracle RAC, wohl aber einen Active/Passive-Cluster, und sorgen außerdem für eine automatische Lastverteilung über mehrere physische Server. Beim Clustering wird eine Gruppe von vSphere-Servern zu einer gemeinsamen Ressource zusammengezogen.

Ein HA-Cluster (High Availability) bietet zwei wichtige Vorteile, um ungewollte Auszeiten zu reduzieren:

- Nach dem Crash eines physischen Servers werden alle VMs, die auf dem betroffenen Server liefen, auf anderen Servern neu gestartet, vorzugsweise auf solchen mit geringer Auslastung. Es handelt sich also um ein klassisches Cold-Failover, aber unabhängig von Betriebssystem bzw. Anwendung.
- Optional kann man auch bei einem Windows Blue Screen bzw. Linux Kernel Panic einen automatischen Reboot erzwingen. Dies setzt die Installation der VMware-Tools voraus. Durch einen Heartbeat zwischen VMware-Tools und HA-Cluster-Software wird der Status der VM geprüft und bei Ausbleiben des Heartbeats diese durchgestartet.

Ein DRS-Cluster (Dynamic Resource Scheduling) sorgt dagegen für eine möglichst gleichmäßige Lastverteilung über die physischen Server, mit dem Ziel, Antwortzeiten und damit die Servicequalität einer VM nicht durch Lastspitzen einer anderen VM zu beeinträchtigen.

Im Wesentlichen handelt es sich bei DRS um ein automatisches Live-Verschieben (vMotion) von VMs, getriggert durch die hohe Auslastung eines physischen Servers. Für einzelne VMs können Ausnahmen definiert werden. In der Praxis hat sich dies als sehr nützlich und robust herausgestellt.

Allerdings gibt es hier zwei verschiedene Modi: halbautomatisch und vollautomatisch. Beim halbautomatischen Modus werden entsprechende vMotion-Empfehlungen lediglich protokolliert und können dann per Knopfdruck umgesetzt werden. Häufig ist es dann aber schon wieder zu spät bzw. die Empfehlungslage hat sich bereits wieder geändert. Dieser Modus ist geeignet, um erstmal mit DRS insgesamt „warm zu werden“ und ein Gefühl dafür zu bekommen. Auf Dauer sollte man aber den vollautomatischen Modus verwenden.

Lizenzierungsaspekte

Oracle wertet Virtualisierung mit VMware als sogenanntes Soft-Partitioning, d.h. im Gegensatz zu Hard-Partitioning ist die CPU-Ausstattung der virtuellen Maschinen nicht fixiert [3]. Lizenzrechtlich wird daher erwartet, dass bei einem vSphere-Server oder einem VMware-Cluster, auf dem eine oder mehrere VMs mit Oracle-Datenbank laufen, genügend Datenbanklizenzen erworben werden, um den gesamten Server bzw. Cluster abzudecken.

Es empfiehlt sich daher, dedizierte Server bzw. Cluster aufzubauen, auf denen ausschließlich VMs mit Oracle-Datenbanken laufen, um nicht unnötig viele Oracle-Lizenzen einsetzen zu müssen.

Bezüglich der VMware-Lizenzierung seien hier einige für Datenbanken wichtige Aspekte als Entscheidungshilfen genannt:

- vMotion, die Möglichkeit HA-Cluster zu bilden sowie die unten besprochenen VM-Templates sind bereits in der vSphere Standard Edition enthalten, setzen aber einen vCenter Server voraus.
- Die Möglichkeit, im laufenden Betrieb CPU- und Memory-Ressourcen eines virtuellen Oracle-Server aufzustocken („Hot-Add“), besteht nur bei der Advanced Edition (vSphere 4.0), der Enterprise Edition und der Enterprise Plus Edition.
- DRS-Cluster und damit automatische Lastverteilung via vMotion ist ausschließlich in der Enterprise- und Enterprise-Plus-Edition möglich.
- Standardvorgaben für die Konfiguration von physischen Hosts und virtuellen Netzwerken („Host Profiles“ und „dvSwitches“) sind für große Umgebungen, d.h. einige Dutzend vSphere-Server, sehr nützlich, benötigen aber die Enterprise-Plus-Edition.
- Die maximale Anzahl Kerne pro VM und damit pro virtuellem Oracle-Server beträgt bei vSphere 4.0 jeweils vier für alle Editionen außer der Enterprise Plus Edition. Letztere erlaubt acht vCPUs. In vSphere 5.0 bilden acht vCPUs die Obergrenze für alle Editionen, wobei die Enterprise Plus Edition wiederum mit bis zu 32 vCPUs die Ausnahme bildet.

Details und weitere Aspekte hierzu finden sich unter [4].

Tipps zur konkreten Umsetzung

Ein wichtiges Hilfsmittel für den Betrieb virtueller Datenbanken sind VM-Templates. Zwar ist ein virtueller Server grundsätzlich schnell erstellt, aber es müssen ja auch noch passende OS-Versionen und Datenbanksoftware aufgespielt und die Datenbank aufgebaut sowie Konfigurationen vorgenommen werden. Mithilfe von Templates lässt sich dies das alles abkürzen.

So könnte es z.B. ein Template für Oracle 11.2.0.2 Standard Edition auf Oracle Enterprise Linux (OEL) 5.6 geben. Ein solches Template sollte bereits eine ganze Reihe von Komponenten, Software und Konfiguration beinhalten, insbesondere:

- Standardwerte für die Anzahl virtueller CPUs sowie die Größe des virtuellen Memory. Beim Erstellen einer VM aus dem Template kann man diese Werte aber abändern, wodurch sich verschiedene Instanztypen realisieren lassen (z.B. kleiner, größer, sehr großer Server).

- Virtuelle Netzwerkkarten und deren Mapping auf physische Portgruppen. Außerdem den verwendeten Netzwerkadapter, möglichst VMXNET3
- Platten für OS und Datenbanksoftware
- Verwendeter virtueller SCSI-Controller: Soll die Datenbank in ASM liegen, empfiehlt sich der Controllertyp „VMware Paravirtual“. Liegt die Datenbank auf NFS, kann man auch den Controllertyp „LSI Logic Parallel“ oder „LSI Logic SAS“ verwenden.
- Evtl. eine Memory Reservation in Höhe mindestens der Standard-SGA-Größe. Bekommt die Datenbank eine größere SGA, sollte diese entweder beim Erstellen der VM aus dem Template oder nachträglich über die VM Settings angepasst werden!
- Hot-Add-Fähigkeit für CPU und Memory (in den VM Settings, dort im Reiter „Options“)
- OS auf aktuellem Patchlevel
- installierte VMware-Tools (hierfür ist auf Linux zwar ein GCC-Compiler erforderlich, aber der muss für die Datenbanksoftware ohnehin vorhanden sein)
- Datenbanksoftware auf aktuellem Patchlevel. Hier ist ggf. ein Template für Standard Edition und ein anderes für Enterprise Edition notwendig.
- Falls die Datenbanken auf NFS liegen: Vorbereitete Pfade für Mountpoints
- Ausgeschaltete Gast-Host-Zeitsynchronisation der VMware Tools
- NTP-Konfiguration in der VM (siehe [5])
- Eine Dummy-Datenbank der passenden Datenbankversion mit einem Dummy-Namen und den Standard-Tablespaces SYSTEM, SYSAUX, TEMP, UNDOTBS (siehe unten)
- Ein Anpassungsskript, das nach der Erstellung einer VM aus dem Template notwendige Konfigurationsschritte durchführt (siehe unten).

Die vorhandene Dummy-Datenbank erspart das Erstellen einer Datenbank mittels DBCA oder Skripten, was ansonsten auch wieder Zeit kosten würde.

Das genannte Anpassungsskript kümmert sich darum, die Dummy-Datenbank auf das endgültige Storage (ASM oder NFS) zu verschieben und umzubenennen sowie Oracle-Konfigurationsdateien wie listener.ora, tnsnames.ora und SPFile anzupassen. Auch Themen wie das Setzen der gewünschten IP-Adresse und des Hostnamens oder das Einbinden der zu benutzenden NFS-Exports in die /etc/fstab-Datei gehören hierhin.

Wichtig ist natürlich ein regelmäßiges Aktualisieren der verwendeten Templates, indem zunächst aus dem Template eine VM erzeugt wird, Datenbank- und OS-Patches in diese VM eingespielt werden, die Dummy-Datenbank entsprechend gepatcht und anschließend diese VM wieder zu einem Template konvertiert wird. Letztlich hängt es von den Update-Zyklen des Betriebssystems und der Datenbanksoftware (Patchsets und CPUs) ab, wie häufig das geschehen muss.

Noch ein Tipp: Möchte man das Ganze als ersten Schritt in Richtung einer Private Cloud begreifen, sollte die Anzahl verschiedener Templates und damit Instanztypen möglichst im einstelligen Bereich bleiben. Die Verwirklichung jedes Sonderwunsches in Form einer speziellen Anpassung bringt jedes Cloud-Konzept und insbesondere die Kostenvorteile eines solchen Konzepts schnell zum Erliegen. Standardisierung und Datenbanken von der Stange sind hier entscheidende Werte!

Wer macht was?

Oftmals gibt es getrennte Teams für Oracle-Datenbanken und Infrastruktur, sprich Hardware und in diesem Falle auch virtuelle Hardware. Naturgemäß benötigt ein DBA nicht sämtliche Rechte auf der virtuellen Infrastruktur. Einige Zugriffsmöglichkeiten sind aber dennoch wichtig, um beispielsweise ein schnelles Troubleshooting durchführen oder Performancewerte aus VMware-Sicht abgreifen zu können, beispielsweise wie viel physische CPUs und Memory der VM überhaupt zugeordnet sind.

Hierfür ist in jedem Falle ein Read-Only-Login an den vCenter Server (bei sehr kleinen und statischen Umgebungen alternativ an den betreffenden vSphere-Server) notwendig, um Performance-Statistiken abgreifen zu können. Werkzeuge wie der vSphere Client oder auch Quest Spotlight on Oracle können diese Daten abgreifen und – im Falle von Spotlight – gemeinsam mit den Oracle-Perfomancedaten darstellen.

VMware bietet außerdem eine fein gegliederte Rechtestruktur, wobei für einen DBA insbesondere folgende Rechte auf Ebene des VMware Data Centers oder Clusters relevant sind:

- Virtual Machine / Configuration / Change resource: VM Memory Reservation ändern, z.B. nach Anpassung von `sga_target`, `pga_aggregate_target` oder `memory_target`
- Virtual Machine / Configuration / Change CPU count: Anzahl vCPUs ändern (auch online)
- Virtual Machine / Configuration / Memory: VM Memory ändern (auch online)
- Virtual Machine / Interaction / Console interaction: Konsole an die VM öffnen (Troubleshooting, wenn kein normaler SSH-Zugang mehr möglich ist)
- Virtual Machine / Interaction / Power Off: VM stoppen
- Virtual Machine / Interaction / Power On: VM starten
- Virtual Machine / Provisioning: Arbeiten mit VMs und Templates
- Resource / Apply recommendation: Manuelles Umsetzen von DRS-Empfehlungen für VMs ohne Full-Automation

Migrationspfade

Wie läuft nun die Migration der Oracle-Datenbanken aus der physischen in eine virtuelle Infrastruktur ab? Dabei kommt es neben der Datenbankgröße und möglichen Auszeit zunächst darauf an, ob der Übergang von physisch nach virtuell die einzige Änderung ist oder ob zusätzlich ein Betriebssystemwechsel, eine Reorganisation oder ähnliche Änderungen stattfinden sollen.

Je kleiner die Datenbank und je länger eine mögliche Auszeit, desto eher kommen natürlich einfache Verfahren wie Datapump, exp/imp oder Tablespace-Transport in Frage.

Bei einem reinen Wechsel von physisch nach virtuell ist der einfachste Weg, die alte Datenbank zu stoppen und aus der VM heraus wieder zu starten. Dies funktioniert, wenn die Datenbank vorher bereits ASM- oder NFS-basiert war. War die Datenbank vorher dateibasiert, sollte man überlegen, eine physische Standby-Datenbank (mit oder ohne Dataguard) aufzubauen, mit der VM als Zielmaschine. Steht die Replikation, öffnet man die Standby-Datenbank und macht sie zur normalen Datenbank. Mit Dataguard ist sogar ein Rollentausch denkbar, um die alte Datenbank noch als Rückfallmöglichkeit mitzupflegen.

Ist der Umzug mit weiteren Änderungen wie z.B. einem Plattformwechsel zu kombinieren, kommt ein plattformübergreifender Tablespace-Transport (ab Oracle 10g und Enterprise Edition) sowie logische Replikationen in Frage. Unter letztere Kategorie fällt Quest Shareplex for Oracle, womit auch

wiederum eine Rückfallmöglichkeit gegeben ist, für den Fall, dass in der VM doch nicht alles glatt läuft.

Resümee

Dieselben Gründe, die ganz allgemein für die Virtualisierung von Servern sprechen, lassen sich auch auf Datenbanken anwenden. Auch hier kann Virtualisierung als erster Schritt in Richtung einer Private Cloud angesehen werden.

In der Vergangenheit gab es aber auch ein paar handfeste Argumente dagegen. Tatsächlich sind aber mit den vSphere-Versionen 4 und 5 sowie der klareren Supportpolitik seitens VMware, Oracle und auch SAP einige dieser Hürden gefallen.

Der Autor hat mehrfach die Erfahrung gemacht, dass die Kombination aus Oracle-Datenbank und VMware vSphere eine sehr robuste und leistungsfähige Plattform darstellt. Beim richtigen Ausbau und insbesondere der Verwendung von Templates lassen sich komplette Datenbankserver inklusive Datenbank auf dem neusten Patchstand in Minuten bereitstellen.

Referenzen

- [1]: My Oracle Support Doc-ID 249212.1
- [2]: SAP Note 1173954 - Support of Oracle for VMWare
- [3]: <http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf>
- [4]: http://www.vmware.com/files/pdf/vsphere_pricing.pdf
- [5]: <http://kb.vmware.com/kb/1006427>

Bitte fügen Sie Ihre Kontaktadresse hinzu.

Kontaktadresse:

Patrick Schwanke
Quest Software GmbH
Im Mediapark 4e
D-50670 Köln

Telefon: +49 (0) 173 733 8577
E-Mail patrick.schwanke@quest.com
Internet: www.quest.com

Xing: https://www.xing.com/profile/Patrick_Schwanke

LinkedIn: www.linkedin.com/in/ps000

Twitter: @pschwanke