

Bei der Diskussion von Server-Virtualisierung steht häufig die Konsolidierung von Hardware mittels virtueller Maschinen im Vordergrund. Dabei gibt es gute Gründe, andere Virtualisierungstechnologien alternativ oder ergänzend dazu einzusetzen, und es gibt auch weitere Zielsetzungen neben einer höheren Auslastung der Hardware. Dieser Artikel gibt einen Überblick über Virtualisierungstechnologien und ihre Einsatzbereiche.

Technologien zur Server-Virtualisierung im Überblick

Franz Haberhauer, ORACLE Deutschland B.V. & Co. KG

Betrachtet man die Wurzeln der Server-Virtualisierung, so gibt es zwei Triebfedern: Einerseits den Wunsch, mehrere logische Server auf einen physischen Server abzubilden – und zwar mit ihrem jeweils spezifischen Software-Stack. Insbesondere wenn – wie etwa in der Windows-Welt üblich – eine Anwendung jeweils auf einen eigenen Server abgebildet wird, ist in vielen Fällen selbst ein kleiner physischer Server bei Weitem nicht ausgelastet. Hier ist es na-

heliegend, physische Hardware partitioniert oder quasiparallel durch einen sogenannten „Hypervisor“ oder „Virtual Machine Monitor“ als virtualisierte Hardware mehreren Betriebssystem-Instanzen, den sogenannten „Gästen“, zur Verfügung zu stellen.

Dieser Hypervisor kann auf der Hardware selbst implementiert sein (Typ 1, unter anderem Xen-basierte Lösungen wie Oracle VM Server for x86, VMWare ESXi, Microsoft Hyper-V) oder in-

nerhalb eines eigenständigen Betriebssystems laufen (Typ 2, unter anderem Oracle VM VirtualBox oder KVM). Ein Typ-1-Hypervisor muss damit selbst die Gerätetreiber für die physische Hardware mitbringen, sodass ein Anwender darauf angewiesen ist, dass der Hypervisor-Anbieter für die einzusetzende Hardware diese integriert und gegebenenfalls auch qualifiziert.

Technisch gesehen ist das dann einfach, wenn als Basis des Hypervisors

PROLICENSE[®]
OPTIMIZING SOFTWARE ASSETS
Kompetent – Unabhängig – Erfolgsbasiert

SO RICHTIG UNTERLIZENSIERT?

Sprechen Sie mit uns!

Wir sind nur unseren Mandanten verpflichtet.

- > **Compliance sichern**
- > **Audit vermeiden**
- > **Kosten senken**

ProLicense GmbH
Friedrichstraße 191 | 10117 Berlin
Tel: +49 (0)30 60 98 19 230 | www.prolicense.com

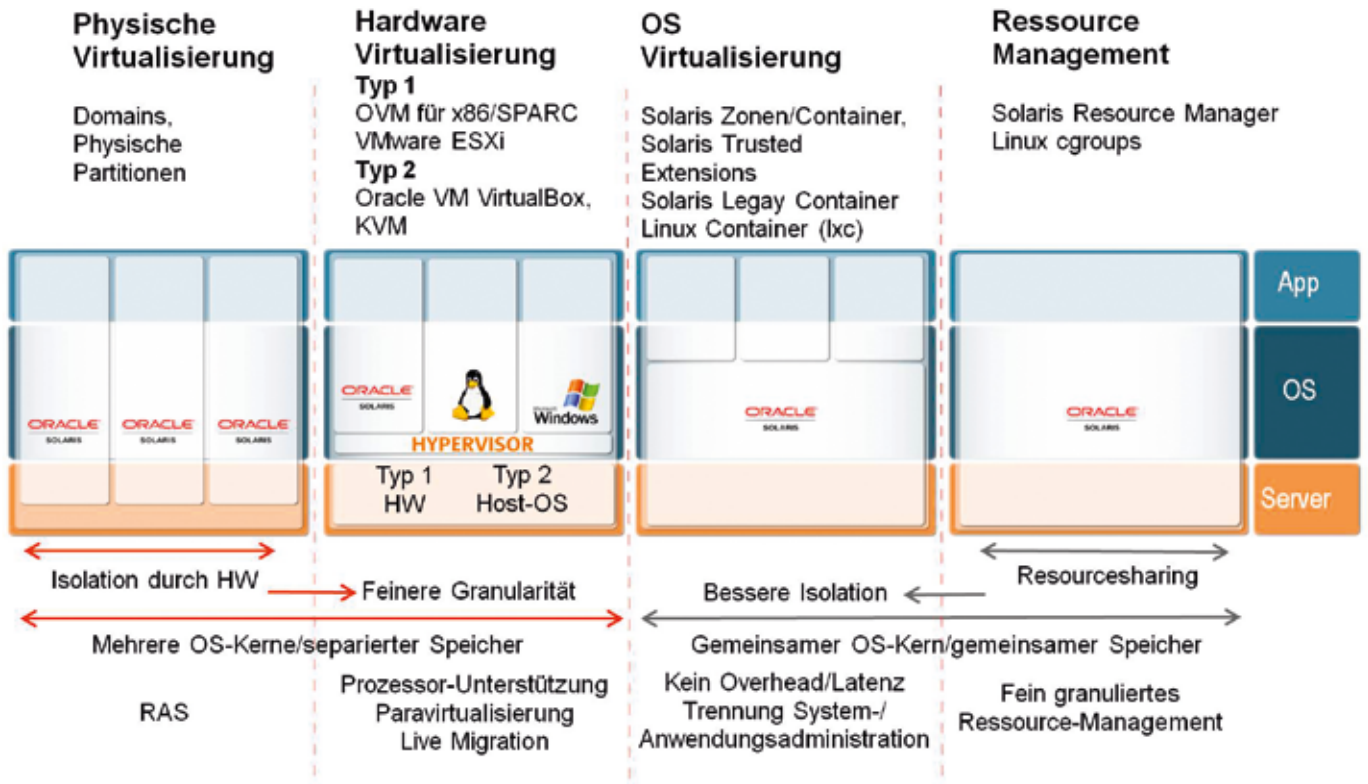


Abbildung 1: Technologien zur Server-Virtualisierung

ein verbreitetes Betriebssystem mit einem breiten Spektrum an Treibern dient – wie im Falle von OVM Server for x86 Linux oder bei OVM Server for SPARC Solaris. Den Gast-Betriebssystemen werden virtualisierte Geräte in der Regel als gängige „Standardgeräte“ bereitgestellt, was es einerseits ermöglicht, ältere Gast-Betriebssystem-Instanzen etwa auf neuerer Hardware zu nutzen, ohne in den Gästen neue Treiber zu installieren. Andererseits werden eventuell spezielle Hardware-Funktionalitäten wegabstrahiert und die Behandlung von Hardware-Problemen in den Hypervisor verlagert. Zudem kostet die Virtualisierungsschicht Performance und führt zu zusätzlicher Latenz.

Um diese Effekte zu reduzieren, wurde das Konzept der Paravirtualisierung eingeführt, bei dem die Gast-Betriebssysteme an die virtualisierte Umgebung angepasst werden – was allerdings der jeweilige Anbieter leisten muss. Die Integration spezieller Technologien zur Unterstützung von Virtualisierung in neueren Prozessorarchitekturen (unter anderem Intel VT, AMD-V, SPARC v9) ermöglichte auch eine effiziente, Hardware-unterstützte Virtualisierung (HVM-Modus) nicht paravirtualisierter Betriebssysteme, insbesondere wenn spezielle paravirtualisierte Treiber genutzt werden, um den Bereich zu optimieren, in dem Virtualisierung den stärksten Einfluss auf die Performance hat: die I/Os (siehe Abbildung 1).

Leichtgewichtige Virtualisierung mit Solaris-Zonen und Linux-Containern

Neben der Virtualisierung der Hardware-Schnittstelle gibt es schon seit Langem auch Virtualisierungslösungen an der Schnittstelle zwischen Anwendungen und dem Betriebssystem – die virtualisierten Ausführungs-Umgebungen. Die Wurzeln dieser Ansätze liegen in den traditionellen Mehrbenutzer-Umgebungen, für die eine stärkere Isolation der einzelnen Anwendungen benötigt wurde – im Hinblick auf Sicherheit wie auf die Eingrenzung von Fehlersituationen sowie im Hinblick auf ein einfaches Ressourcen-Management.

Ausgehend vom klassischen Unix-Konzept einer alternativen, typischerweise stark eingeschränkten „chroot“-Ausführungsumgebung, wurden über das einfache Sandbox-Konzept der Jails

im FreeBSD sehr leichtgewichtige Virtualisierungslösungen auf Unix und Linux entwickelt. Beispiele sind unter Linux Linux-VServer oder OpenVZ / Virtuozzo mit einem Fokus auf Hosting sowie das neuere Projekt der Linux Containers (LXC), die jetzt im Oracle Linux Unbreakable Enterprise Kernel Release 2 (UEK2) als Technology Preview enthalten sind. Einen sehr breiten Einsatz fanden die Solaris-Container, die 2005 mit Solaris 10 eingeführt und seither in Solaris 10 Updates weiterentwickelt wurden [1]. Der Begriff „Solaris-Container“ wurde als Oberbegriff für die beiden wesentlichen Komponenten dieser Technologie eingeführt: Solaris-Zonen als Isolationsmechanismus und das Resource-Management, das bereits in Solaris 9 wesentlich erweitert worden war. Auch die Linux-Container folgen diesem Konzept: Integration von Funktionalitäten zur Isolation durch eine Trennung von Namensräumen mit Resource-Management. Bei Linux sind diese Technologien im Control-Groups-Projekt (cgroups) zusammengefasst, das die Basis für die Linux Containers bildet [2].

Die Isolation der Namensräume bewirkt, dass aus der initialen Betriebssystem-Instanz heraus – der sogenannten „globalen Zone“ in Solaris – zwar alle Prozesse gesehen werden, umgekehrt in den nicht-globalen Zonen aber nur die dieser Zone zugehörigen Prozesse. Um das sicher zu gewährleisten, fehlen den Prozessen in nicht-globalen Zonen Privilegien, die sie auch nicht erwerben können. Insofern ist der Superuser jeweils auf seine nicht-globale Zone beschränkt, was praktische administrative Ansätze ermöglicht: Häufig besteht seitens der Anwendungs-Administration der Wunsch, für bestimmte administrative Tätigkeiten als Superuser arbeiten zu können. Andererseits gibt es dagegen oft Bedenken seitens der Plattform-Administration, da damit etwa Konfigurationsänderungen möglich wären, die in deren Hoheit liegen. In Solaris-Zonen entfallen solche Bedenken, da aus einer nicht-globalen Zone heraus auch für den Superuser Änderungen der Gerätekonfiguration nicht oder nur eingeschränkt möglich sind.

Die Solaris-Container sind die Grundlage der Solaris Trusted Extensions, die in Solaris 10 Labeled Security implementieren. Solaris 10 wurde gegen das Labeled Security Protection Profile (LSPP) auf dem Assurance Level EAL4+ zertifiziert, was auch die sicherheitstechnische Isolation der Container dokumentiert. Für Solaris 11 wurden zusätzliche Sicherheitskonzepte eingeführt, insbesondere die „Immutable Zones“ mit einem Read-Only Root-Verzeichnis, das aus der Zone heraus im Normalbetrieb nicht geändert werden kann [3].

Da Prozesse in Zonen Systemaufrufe und I/Os genauso gegen den Betriebssystemkern des Gesamtsystems absetzen wie Prozesse in der globalen Zone, ist nicht nur die Performance gleich – auch das Leistungsprofil ändert sich nicht. In virtuellen Maschinen haben I/O-intensive Lasten einen höheren Overhead als rechenintensive Lasten. Solche Änderungen des Leistungsprofils einer Anwendung sollte man bedenken, wenn man etwa für Test und Produktion unterschiedliche Umgebungen einsetzt – statt für Test eine

virtualisierte und für Produktion physische Hardware.

Auch im Hinblick auf das Ressourcen-Management ist eine Virtualisierung auf Betriebssystemebene sehr leichtgewichtig. In nicht-globalen Zonen laufen vor allem Anwendungsprozesse, Infrastruktur-Prozesse dagegen in der globalen Zone. Dadurch ist der Start oder Neustart einer nicht-globalen Zone sehr schnell und benötigt nur wenig Sekunden: Es sind nur wenige Prozesse zu starten und die Initialisierung von Geräten oder großen Hauptspeicherbereichen, die ja auch beim Start von Betriebssystemen in virtuellen Maschinen nötig ist, entfällt, da Dateisystempuffer und andere Datenstrukturen des Betriebssystemkerns von allen Zonen gemeinsam genutzt werden. Insofern sind Zonen ein interessantes Instrument, um schnelle Neustarts komplexer Anwendungen zu realisieren. Zudem unterliegen die Prozesse aller Zonen der globalen Hauptspeicherverwaltung. Es ist also nicht nötig, einzelnen Zonen explizit Hauptspeicher zuzuweisen und dadurch den insgesamt verfügbaren Speicher zu partitionieren und damit potenziell zu fragmentieren.

Es besteht jedoch die Möglichkeit, für einzelne Zonen die Menge Hauptspeicherresidenter Seiten zu limitieren oder die Größe des Working Sets zu definieren, bei deren Überschreiten gezielt Seiten der Prozesse der betreffenden Zone ausgelagert werden. Darüber hinaus können weitere Ressourcenlimits für einzelne Zonen konfiguriert werden. Über das in Solaris 11 neu eingeführte Utility „zonestat (1)“ können elegant aggregierte Werte für die Ressourcen-Nutzung einzelner Zonen gesammelt werden.

Da ein Betriebssystem darauf ausgelegt ist, verfügbaren Hauptspeicher zu nutzen – falls kein anderweitiger Bedarf besteht, werden Datei-Inhalte gepuffert –, ist dagegen die Planung und Verteilung des Hauptspeichers für virtuelle Maschinen nicht ganz so offensichtlich, wenn man Hauptspeicher minimieren will und nicht großzügig konfiguriert.

Durch neue Technologien wie Memory Ballooning und Transcendent

Memory (als Technology Preview im UEK2 [4]) wird die Zuweisung von Hauptspeicher zu virtuellen Maschinen flexibilisiert, indem ein gewisses Memory Overcommit ermöglicht wird, also den virtuellen Maschinen insgesamt mehr Hauptspeicher zugewiesen werden kann, als physisch tatsächlich vorhanden ist. Auf plötzliche, große Speicheranforderungen reagieren solche Mechanismen allerdings träge.

Flexibilität mit virtuellen Maschinen

Die Solaris-Container, Linux-Container und andere virtualisierte Ausführungsumgebungen sind nicht zuletzt deswegen so leichtgewichtig und beim Ressourcen-Management so flexibel, weil sie auf einem System den zentralen Betriebssystemkern nutzen. Das beschränkt diese Ausführungsumgebungen allerdings auch auf solche, die mit diesem Kern kompatibel sind oder einfach gemacht werden können. Im ersten Fall der nativen Zonen sind der Kern und die systemnahen Pakete in der globalen Zone und den nicht-globalen Zonen auf dem gleichen Stand.

Nachdem über Solaris-Versionen hinweg eine Binärkompatibilität gilt, können in sogenannten „Branded Zones“ Umgebungen für ältere Solaris-Versionen angeboten werden: auf Solaris 10 für Solaris 8 und Solaris 9 die sogenannten „Solaris Legacy Container“ als eigenständiges Produkt und auf Solaris 11 die Solaris-10-Zonen als integraler Bestandteil. Weshalb Binärkompatibilität allein nicht ausreicht, lässt sich an einem einfachen Beispiel zeigen: an einer Anwendung, die die Versionsnummer des Betriebssystems abfragt. Branded Zones mit Ausführungsumgebungen für ältere Betriebssystem-Versionen sind ein elegantes und effizientes Mittel für eine schrittweise Migration von einer Betriebssystem-Version auf die nächste.

Die Umsetzung von einer physischen Installation in eine virtualisierte auf einer neueren Plattform wird von sogenannten „P2V-Utility“ (physical-to-virtual) unterstützt oder im Falle von Solaris-10-Containern, die auf ein Solaris-11-System gebracht werden sollen, durch ein V2V-Utility (virtual-to-virtual). Virtuelle Maschi-

nen sind noch flexibler. Da sie lediglich eine virtualisierte Hardware bereitstellen, kann in jeder virtuellen Maschine auf derselben physischen Hardware ein anderes Betriebssystem laufen – verschieden auch im Sinne unterschiedlicher Patch-Stände. Zudem ist es an dieser Schnittstelle vergleichsweise leicht möglich, eine laufende Instanz von einem physischen Server auf einen anderen zu verschieben, was als „Live Migration“ bezeichnet wird.

Technisch wird dabei der Hauptspeicher der virtuellen Maschine, in dem sich ja der gesamte Zustand des darin laufenden Betriebssystems und aller Anwendungen widerspiegelt, vom einen auf das andere System kopiert. Da dieser Vorgang einige Zeit dauert, erfolgt er iterativ. Die virtuelle Maschine wird schließlich auf dem ersten System angehalten, noch fehlende geänderte Speicherinhalte werden transferiert und die Instanz dann auf dem zweiten System wieder aktiviert, insbesondere dadurch, dass die Netzwerkadressen nun hier aktiviert werden. Externe Speicher müssen natürlich von beiden Systemen aus gleichermaßen erreichbar sein. Weniger offensichtlich ist, dass die CPUs der Systeme identisch oder nahe verwandt sein müssen. Diese Anforderung resultiert daraus, dass sich der Systemzustand auch im Prozessorstatus widerspiegelt, der von einem auf das andere System abgebildet werden muss.

Beim Transfer der Hauptspeicher-Inhalte werden unter Umständen höchst sensible Daten transferiert – etwa Passwörter, die während der Verarbeitung im Klartext vorliegen. Daher werden beim Oracle VM Server for SPARC die Daten verschlüsselt transferiert, was sehr performant möglich ist, da die CPUs der SPARC-T-Serie Verschlüsselung Hardware-seitig unterstützen [5].

Live Migration ist insbesondere dann sehr hilfreich, wenn etwa für Wartungszwecke ein Hardware-System freigeräumt werden muss, auf das Anwendungen mehrerer Nutzer konsolidiert sind. Damit lässt sich eine ansonsten nötige gemeinsame Downtime vermeiden.

Laufende virtualisierte Anwendungs-umgebungen lassen sich nicht so ein-

fach aus der Betriebssysteminstanz, in die sie eingebettet sind, herauslösen. Eine Migration angehaltener Umgebungen ist dagegen möglich. Für Solaris-Container haben sich dafür Best Practices unter dem Schlagwort „Flying Zones“ [6] etabliert. Das Abkoppeln einer Zone von einem System („detach“) und das Andocken an einen anderen wird dabei direkt von Solaris unterstützt. Dabei kann der Patch-Stand der Zone gegebenenfalls automatisch an das Zielsystem angepasst werden („Update on Attach“) [7].

Netzwerk-Virtualisierung

Die Möglichkeit, auf einem physischen System mehrere Gäste zu betreiben, liefert im Hinblick auf deren Vernetzung Herausforderungen, aber auch interessante Anwendungen. Ist das physische System über ein einziges Netzwerk-Interface angebunden, muss ein Virtual Machine Manager offensichtlich als Router oder Bridge fungieren, um Netzwerkpakete an die Gast-Betriebssysteme zu vermitteln. Hierbei kommt es naturgemäß zu Performance-Einbußen, insbesondere zu zusätzlicher Latenz.

Das Netzwerk ist der Bereich, in dem sich die Virtualisierung am stärksten auf das Leistungsverhalten auswirkt. Bei der Virtualisierung über Anwendungsumgebungen ist dieser Effekt signifikant geringer, da hier zum Beispiel bei den Solaris-10-Zonen in einer Shared-IP-Konfiguration Teile des Netzwerk-Stacks gemeinsam genutzt werden. Beim Verkehr zwischen Zonen auf einem System müssen die Stacks noch nicht einmal bis in die physische Schicht durchlaufen werden. Da damit gewisse Konfigurationseinschränkungen etwa beim Routing verbunden sind, gibt es bei Solaris 10 alternativ das Konzept der „Dedicated IP Stacks“ mit vollständig virtualisierten IP-Stacks auf eigenen Interfaces (NICs).

Für Solaris 11 wurde im Projekt „Crossbow“ der gesamte Netzwerk-Stack grundlegend überarbeitet – nicht zuletzt im Hinblick auf Virtualisierungslösungen. Moderne, intelligente NICs können Netzwerkpakete unterschiedlicher Verbindungen bereits in der Hardware trennen (Hardware Rx/Tx-Rings). Die Netzwerk-Virtualisie-

rung in Solaris 11 ermöglicht es, direkt auf NICs mehrere sogenannte „Virtuelle NICs“ (VNICs) zu konfigurieren. Sind keine oder nicht ausreichend Hardware-Ringe vorhanden, werden sie als Software bereitgestellt. Diese VNICs können dann in Solaris-11-Zonen als hoch performante Netzwerk-Schnittstellen konfiguriert werden. Es ist auch möglich, VNICs als rein virtuelle NICs, mit denen keine physischen NICs assoziiert sind (sogenannte „Etherstubs“), und damit virtuelle Netzwerk-Switches aufzubauen.

Da für VNICs auch Bandbreiten-Management möglich ist, können über die Netzwerk-Virtualisierung in Solaris 11 in Verbindung mit den Solaris-11-Containern ganze Netzwerk-Topologien sehr effizient virtualisiert werden. Ein interessantes Anwendungsszenarium ergibt sich aus der Möglichkeit, über einen speziellen Treiber gezielt in Verbindungen Latenz oder durch das kontrollierte Verwerfen von Paketen Fehler einzustreuen. Damit können dann in einer lokalen virtualisierten Umgebung fehlerbehaftete Weitverkehrs-Verbindungen simuliert und Anwendungen in solchen Umgebungen reproduzierbar getestet werden [8].

Test und Debugging verteilter Systeme sind ein Szenarium, in dem virtualisierte Ausführungsumgebungen einen großen Vorteil bieten: die zeitgleiche Sicht in separate Instanzen. Aus der globalen Zone können ja die Prozesse aller nicht-globalen Zonen gesehen, beobachtet und „debugged“ werden – falls erforderlich über ein einziges Kommando oder ein DTrace-Skript. Ereignisse in virtuellen Maschinen lassen sich nicht so einfach in hoher zeitlicher Auslösung korrelieren.

Verwendet man virtuelle Maschinen für Entwicklung und Test auf Desktops, etwa um Hardware-Landschaften auf einem Laptop nachzustellen, ist dafür ein Typ-2-Hypervisor wie Oracle VM VirtualBox oft einfacher wegen der engeren Integration der Benutzeroberflächen der Gäste mit der des Hostsystems. In [9] wird gezeigt, wie ein Solaris-Hochverfügbarkeits-Cluster mit VirtualBox in drei virtuellen Maschinen auf einem Laptop abgebildet werden kann.

VirtualBox ist auch eine interessante Lösung beim Einsatz einer Virtual Desktop Infrastructure (VDI). Dabei ist nicht nur die Konsolidierung von Desktop-Hardware das einzige Motiv; eine verbesserte Sicherheit insbesondere auch bei der Einbindung mobiler Endgeräte ist ein weiterer wesentlicher Vorteil [10].

Provisionierung

Oft ist die auch die Motivation zur Virtualisierung nicht primär die Konsolidierung von Hardware, sondern es sind Aspekte der Provisionierung, also der Bereitstellung von Anwendungen. Insbesondere virtuelle Maschinen entkoppeln den vollständigen Software-Stack von der Anwendung über die Middleware bis zum Betriebssystem der konkreten Hardware dadurch, dass der Hypervisor dem Betriebssystem in den Gast-Instanzen aktuellste Hardware, für die es unter Umständen gar keinen Treiber gibt, virtualisiert als gängigen Hardware-Typ vorgaukelt, der vielleicht nicht einmal mehr produziert wird. Damit ist es möglich, Software-Stände zu archivieren und für funktionale Regressionstests schnell wieder verfügbar zu machen, indem dieser Stand wieder in eine virtuelle Maschine geladen wird. Die Konfiguration der virtuellen Maschine wird ebenfalls gespeichert. So ist es etwa möglich, für Schulungs- oder Demozwecke die Management-Oberfläche einer Appliance als virtuelle Maschine mit entsprechender virtueller Hardware zur Verfügung zu stellen. Realisiert wurde das zum Beispiel im Oracle ZFS Storage Appliance Simulator [11].

Interessant ist auch die Möglichkeit, über Snapshot-Funktionalitäten

in den virtuellen Maschinen Stände zu sichern, auf die man später einfach wieder zurückkehren kann. Solche Snapshots oder Clones können auch mit Mitteln der Plattform angelegt werden – sehr elegant etwa mit Solaris ZFS – wobei die virtuelle Maschine für einen konsistenten Zustand auf der virtuellen Platte angehalten sein sollte.

Oracle stellt in Form sogenannter „Oracle VM Templates“ ein breites Spektrum an Software zur Verfügung, die sofort in einer virtuellen Maschine genutzt werden kann, ohne dass sie erst über einen Installer auf einem Betriebssystem installiert und dann konfiguriert werden muss. Parameter, die spezifisch für die Ziel-Umgebung sind, wie Netzwerk-Adressen, werden in einem Skript konfiguriert, das beim ersten Start ausgeführt wird. Auch die Konfiguration der virtuellen Maschine selbst kommt mit dem Template mit, sodass dieses aus folgenden drei Komponenten besteht: dem Inhalt der virtuellen Platte(n) einer Installation, einem Konfigurationsskript sowie der Konfiguration der virtuellen Maschine.

Die meisten Implementierungen virtueller Maschinen unterstützen solche, auch als Virtual oder Software Appliances bezeichnete Konzepte. Mit dem Open Virtualization Format (OVF) gibt es dafür ein standardisiertes Format, das unter anderem vom Oracle VM Server for x86 3.0, von Oracle VM VirtualBox und VMware unterstützt wird.

Oracle VM Templates können auch mehrere Maschinen enthalten, etwa für Multi-Tier-Umgebungen. Einen Schritt weiter geht hier der Oracle Virtual Assembly Builder, der insbesondere für WebLogic-Umgebungen eine flexib-

le Abbildung von Multi-Tier-Landschaften erlaubt. Während die Oracle VM Templates traditionell Oracle Linux insbesondere in einer minimierten Form als Oracle Linux JeOS („Just enough Operating System“) nutzen, wurde der Begriff „Oracle VM Template“ auch schon für Software Appliances auf der Basis von Solaris-Containern verwendet [12], was zeigt, dass das Konzept vorinstallierter Software in Form von Software Appliances auch mit dieser Technologie realisiert werden kann.

Lifecycle-Management

Neben den CLI-Schnittstellen zur Administration gibt es für die verschiedenen Virtualisierungslösungen auch grafische Administrationswerkzeuge, wobei es hier im letzten Jahr einige Neuerungen gab: Für den Oracle VM Server for x86 3.0 wurde der OVM Manager komplett überarbeitet und kann sich jetzt in den Oracle Enterprise Manager 12c Cloud Control integrieren. Für den Oracle VM Server for SPARC (LDOMs) sowie für die Solaris-10-Container gab es bereits seit Längerem das Virtualization Pack für das Oracle Enterprise Manager Ops Center, das bis zur Oracle OpenWorld 2011 kostenpflichtig war. Seither ist es beim Einsatz von Oracle-Produkten frei verfügbar. In der Version Ops Center 11g ist das Management von nicht damit erzeugten Instanzen nur eingeschränkt möglich. Die aktuelle, neue Version zeigt sich hier flexibler.

Während bei den virtuellen Maschinen die Unabhängigkeit der Versionsstände einerseits von Vorteil sein kann, ergibt sich daraus andererseits aber auch kein Effizienzgewinn bei ei-

Impressum

Herausgeber:
DOAG Deutsche ORACLE-
Anwendergruppe e.V.
Tempelhofer Weg 64, 12347 Berlin
Tel.: 0700 11 36 24 38
www.doag.org

Verlag:
DOAG Dienstleistungen GmbH
Fried Saacke, Geschäftsführer
info@doag-dienstleistungen.de

Chefredakteur (ViSDP):
Wolfgang Taschner
redaktion@doag.org

Chefin von Dienst (CvD):
Carmen Al-Youssef
office@doag.org

Titel, Gestaltung und Satz:
Claudia Wagner, Fana-Lamielle Samatin
DOAG Dienstleistungen GmbH

Titelfoto: censhare AG

Anzeigen:
CrossMarketteam Doris Budwill
www.crossmarketteam.de
Mediadaten und Preise finden Sie unter:
www.doag.org/go/mediadaten

Druck:
adame Advertising and Media
GmbH Berlin, www.adame.de

ner Konsolidierung. Die Zahl der zu verwaltenden – insbesondere der unabhängig zu patchenden – Instanzen bleibt dieselbe. Will man im Rahmen einer Konsolidierung auch die Betriebs-Umgebungen standardisieren, zum Beispiel durch die Einführung von Baselines für Patches, dann bietet die engere Kopplung der leichtgewichtigen Virtualisierung mittels Solaris-Containern den Vorteil, dass mit dem Patchen der globalen Zone installierte, nicht-globale Zonen automatisiert mit aktualisiert werden.

Anfangs bei Solaris 10 ein Kritikpunkt an den Zonen – Zonen wurden beispielsweise seriell während einer Downtime aktualisiert – wurden mit Solaris-10-Updates die Mechanismen zum Patchen von Zonen massiv verbessert und insbesondere die Zeitdauer bei Systemen mit zahlreichen Zonen durch eine Parallelisierung sowie die Möglichkeit zum Live Upgrade drastisch reduziert. Auch durch den Mechanismus des „Update on Attach“ beim Verschwenken von Zonen zwischen Rechnern lassen sich elegante Update-Konzepte realisieren [7].

Lizenzierung in virtualisierten Umgebungen

Lizenzierungsschemata orientieren sich oft an der Hardware-Konfiguration, insbesondere an der CPU-Kapazität, wobei sich hier als Metrik die Zahl der CPU-Kerne etabliert hat. Dabei sind CPU-Kerne in Betriebssystemen oft gar nicht so ohne Weiteres erkennbar, da hier oft CPUs als Abstraktion von Strands („Hardware-Threads“) gesehen werden. Diese Strands teilen sich gewisse Ressourcen wie L3-Caches auf den physischen CPUs oder L2-Caches und Verarbeitungseinheiten in CPU-Kernen, was aber für den Anwender transparent ist.

Die zu einem Kern gehörigen Strands lassen sich je nach Betriebssystem etwa über ihre Nummerierung als virtuelle CPUs identifizieren. Wird nun noch eine Virtualisierungsschicht dazwischen gelegt, ist die Zuordnung von in Gästen gesehenen CPUs zu physischen CPU-Kernen noch weniger offensichtlich. Insofern ist, selbst wenn wie bei Oracle Virtualisierungslösungen unter dem Stichwort „Hard Partitioning“ an-

erkannt werden [13], um Hardware nur teilweise für bestimmte Software zu lizenzieren, die regelkonforme Umsetzung manchmal nicht trivial. [14] beschreibt die regelgerechte Umsetzung für Oracle VM for SPARC (LDMs), der erst seit Ende letzten Jahres als Hard-Partitioning-Lösung anerkannt ist. Für die Solaris-Container hatte Oracle vor Jahren den Begriff „Capped Containers“ als Begriff für eine spezifische, als Hard Partitioning anerkannte Konfiguration unter Nutzung von Solaris-Processor-Sets eingeführt [15].

Fazit

Es wurde gezeigt, dass es neben dem Ansatz der virtuellen Maschinen weitere Lösungen zur Server-Virtualisierung gibt, insbesondere die leichtgewichtige Virtualisierung in Form von virtualisierten Anwendungsumgebungen auf Betriebssystemebene. Diese Ansätze haben jeweils ihre spezifischen Stärken, weshalb sie mittlerweile auch auf den meisten Plattformen verfügbar sind – als Solaris-Container seit Jahren im breiten, produktiven Einsatz oder mit dem Oracle Linux Unbreakable Enterprise Kernel 2 als Linux-Container im Technology Preview. Zudem schließen sich diese Technologien nicht gegenseitig aus, da unterschiedliche Schnittstellen virtualisiert werden: zum einen die Schnittstelle zwischen Hardware und Betriebssystem, zum anderen die zwischen Anwendung und Betriebssystem. Es ist technisch möglich und kann sinnvoll sein, Virtualisierungslösungen auf diesen Ebenen miteinander zu kombinieren.

Weitere Informationen

Die Links dazu finden Sie auf der Seite <http://www.doag.org/go/doagnews/virtualisierung> oder beim Scannen des abgebildeten QR-Codes.



- [1] Detlef Drewanz, Ulrich Gräf: Solaris Container Leitfaden, November 2009: https://blogs.oracle.com/solarium/entry/neue_version_container_leitfaden_3
- [2] Wim Coerkaerts: Containers on Linux, 16.10.2011: https://blogs.oracle.com/wim/entry/containers_on_linux

- [3] Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management: http://docs.oracle.com/cd/E23824_01/html/821-1460/glhep.html
- [4] Unbreakable Enterprise Kernel R.2 for Oracle Linux: <http://www.oracle.com/us/technologies/linux/uek-r2-features-and-benefits-1555063.pdf>
- [5] Increasing Application Availability by Using the Oracle VM Server for SPARC Live Migration Feature: An Oracle Database Example, An Oracle White Paper, February 2012: <http://www.oracle.com/technetwork/server-storage/vm/ovm-sparc-livemigration-1522412.pdf>
- [6] Hartmut Streppel: Flying Container mit Oracle Solaris Container und Oracle Solaris Cluster, DOAG 2011 Konferenz + Ausstellung
- [7] Detlef Drewanz, Hartmut Streppel: Minimale Downtime beim Patchen von Failover (Flying) Oracle Solaris Containern, DOAG 2011 Konferenz + Ausstellung
- [8] Jörg Möllenkamp: Simulation von Netzwerken in virtualisierten Umgebungen, DOAG 2011 Konferenz + Ausstellung
- [9] Heiko Stein: Testumgebungen für Oracle Solaris Cluster mit Oracle VM Virtual-Box, DOAG 2011 Konferenz + Ausstellung
- [10] Rolf-Per Thulin: Oracle VDI, DOAG 2011 Konferenz + Ausstellung
- [11] Sun ZFS Storage Appliance Simulator: http://www.oracle.com/webapps/dialogue/ns/dlgwelcome.jsp?p_ext=Y&p_dlg_id=10521841&src=7299332&Act=45
- [12] Oracle VM Template: <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/virtual-machines-1355605.html>
- [13] Server/Hardware Partitioning: <http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf>
- [14] Hard Partitioning With Oracle VM Server for SPARC, An Oracle White Paper, December 2011: <http://www.oracle.com/technetwork/server-storage/vm/ovm-sparc-hard-partitioning-1403135.pdf>
- [15] Best Practices for Running Oracle Databases in Oracle Solaris Containers, An Oracle White Paper, April 2010: <http://www.oracle.com/technetwork/server-storage/solaris10/solaris-oracle-db-wp-168019.pdf>

Franz Haberhauer
franz.haberhauer@oracle.com

