

Exalogic-Virtualisierung durch Exa-Templates

Matthias Fuchs, ISE Information Systems Engineering GmbH, und Andreas Chatziantoniou, Foxglove-IT BV

Dieser Artikel zeigt, was es bedeutet, eine Exalogic für den Betrieb in einem Unternehmen vorzubereiten. Im Mittelpunkt steht die einfache und schnelle Bereitstellung von Oracle-Fusion-Middleware-Applikationen auf der Exalogic Virtual Appliance. Darüber hinaus werden Ansätze zur Vereinfachung des Rollouts vorgestellt.

Das Engineered System „Exalogic“ kann in zwei Varianten beim Kunden zum Einsatz kommen. Es besteht die Möglichkeit, entweder mit Oracle VM zu virtualisieren oder die Rechner auf Basis von Oracle Enterprise Linux zu betreiben. Im weiteren Verlauf wird auf die Exalogic-Virtualisierung eingegangen. Sie ist im Rahmen einer Konsolidierung von Plattformen ein sehr beliebter Ansatz. Trotz Virtualisierung kann nur Oracle Enterprise Linux als Gast-Betriebssystem zum Einsatz kommen.

Ausgangssituation

Das gelieferte Exalogic-System ist durch den Advanced Customer Support (ACS) mit der Virtualisierung (OVM) installiert und das Basis-Template für die Erstellung von virtuellen Maschinen (vServer) eingespielt. Eine Integration in Cloud Control fehlt, Middleware-Software ist ebenfalls nicht vorhanden. Die Netzwerk-Konfigurationen sind für das Exalogic Control abgeschlossen, vServer-Client-Netzwerke sind nicht eingerichtet und der Software-Load-Balancer, Oracle Traffic Director (OTD), ist nicht vorhanden.

Das geplante Einsatzszenario

Ziel ist es, die Oracle Fusion Middleware nach den Prinzipien des Enterprise Deployment Guides (EDG) und der Maximum Availability Architecture (MAA) aufzubauen. Ebenso soll es möglich sein, Development, Test und Produktion nach Software identisch, aber in unterschiedlichen Ausprägungen zu installieren. Das bedeutet, dass je nach Umgebung die gleiche Software, jedoch mit unterschiedlichen

Ressourcen vorhanden sein muss. Die virtuellen Maschinen sind somit unterschiedlich groß (CPU, Memory) und die Applikation Cluster sind flexibel von einem bis n Knoten zu skalieren. Dies ergibt eine große Anzahl von möglichen Konfigurationen.

Nach der Installation durch Oracle erfolgt die Integration ins eigene Unternehmens-Netzwerk. Es werden die Netzwerke für den Clientzugriff eingerichtet und die Verbindung mit Cloud Control (CC) hergestellt. Das Cloud Control stellt den zentralen Administrations- und Überwachungs-Punkt dar. Es ist darauf zu achten, dass dort alle Plugins aktuell sind und die letzte Version 12c R2 vorhanden ist. Ältere Versionen haben erhebliche Nachteile in der Administration der Engineered Systems.

Das Exalogic-Basis-Linux-Template ist den Erfordernissen der Fusion Middleware entsprechend einzurichten, so sind beispielsweise die internen Plattengrößen der virtuellen Server anzupassen. Jeder vServer – der Oracle-Sprachgebrauch für virtuelle Maschine – muss mit einem CC-Agent versorgt sein. Netzwerk-Anpassungen wie NTP-Server, Mountpoints für gemeinsame Laufwerke oder Änderungen an den Linux-Paketen sind entsprechend den Unternehmens-Vorgaben manuell durchzuführen. Alternativ können auch eigene Templates erstellt werden. Der Einsatz von Templates aus einer bestehenden X86-Oracle-Virtualisierungs-Lösung ist nicht möglich, da die notwendigen Anpassungen für die Exalogic nicht nachinstalliert werden können.

Nachdem der vServer im CC konfiguriert ist, müssen die Provisioning-

Skripte für das Ausrollen von Node-manager und WebLogic Server angepasst oder erstellt werden. Zudem sind die Applikationen inklusive WebLogic Server in die Software-Library aufzunehmen. Dies ist notwendig, da eine identische Installation über alle Stages (Dev, Test, Prod) hinweg ein primäres Ziel ist. Dies kann nur durch automatisierte Installation sichergestellt werden.

Applikationen, die Dienste für die Infrastruktur darstellen, wie OTD, Oracle Webservice Manager (OWSM), Oracle Access Management (OAM) oder SOA Suite sind aufwändig per Hand pro vServer zu installieren und zu konfigurieren, da sie nicht in der Software-Library vorhanden sind beziehungsweise keine automatisierte Installation beim Kunden verfügbar ist. Das Erstellen von automatisierten Skripten lohnt im Normalfall nicht, da diese nur einmal pro Umgebung notwendig sind. Das Erstellen und Testen der Skripte würde mehrere Wochen dauern, was nur praktikabel ist, wenn viele gleiche Installationen auszurollen sind.

Der Lebenszyklus einer Installation ist ebenfalls vorzubereiten, weil damit zu rechnen ist, dass neue Major-Releases der Oracle-Software während der Lebensdauer einer Exalogic erscheinen. Diese müssen angefangen bei der Entwicklung über den Test bis hin zur Produktion neu ausgerollt werden. Die Art des Rollouts unterscheidet sich in den meisten Fällen, so wird zum Beispiel in der Entwicklung kein Applikations-Cluster verwendet – dieser ist aber im Test und in der Produktion zwingend erforderlich. Dies bedeutet viel zusätzlichen Aufwand.

Die Idee: Exa-Templates

Um die beschriebenen Punkte zu behandeln, ist es wichtig, die Abhängigkeiten der Komponenten zu inventarisieren. Da die Cloud-Control-Umgebung zentrale Anlaufstelle eines Exa-Stacks ist, sollten die ersten Schritte dahin gehen, diese Cloud-Control-Umgebung aufzubauen. Eine weitere wichtige Zielsetzung ist im Rahmen der Lifecycle-Support-Unterstützung die Reproduzierbarkeit der Ausführung. Dies führt natürlich zu einer Lösung, die auf der Basis von Skripten funktioniert. Diese Skripte werden dann aus dem Ops Center (beziehungsweise EM-CLI) zur Erstellung der vServer ausgeführt, die dann aus dem Cloud Control ergänzt werden, um die Fusion-Middleware-Komponenten (FMW) zu konfigurieren.

Es bietet sich an, verschiedene vServer für die diversen FMW-Komponenten anzulegen, also unter anderem vServer für SOA Suite, WebCenter oder Identity Management (IDM). Der Grund für diese Vorgehensweise liegt in der Skalierung der Umgebungen und in der Reproduzierbarkeit. Eine Entwicklungsumgebung kann mit einem kleinen Cluster der SOA Suite seinen Zweck erfüllen, aber für andere Lebenszyklen sind massive Systeme erforderlich. Auf Basis der vServer-Templates lassen sich diese schnell erstellen. Um die Skalierung zu fördern, wird im Endeffekt dieselbe Konfiguration eingesetzt, der verfügbare Speicher jedoch per vServer angepasst.

Ein Beispiel für Exa-Templates

Nehmen wir eine typische technische Umgebung, die auf der SOA Suite basiert. Natürlich wird ein Großteil der Verarbeitung innerhalb des BPEL stattfinden, die Integration mit anderen Anwendungen erfolgt über den Oracle-Service-Bus (OSB), zudem kommen Business Activity Monitoring (BAM) und Human Task Engine (HTE) zum Einsatz. Diese Umgebung benötigt einen BPEL-Cluster mit vier Knoten, zwei OSB-Maschinen und jeweils einer BAM- und einer HTE-Maschine. Deshalb werden mit dem Ops-Center-Cloud-Infrastructure-API und dem Command-Line-Interface die Maschi-

nen (vServer) auf Basis der vorbereiteten Templates (siehe Listing 1) erzeugt.

Nachdem die vServer auf diese Art und Weise entstanden sind, besteht zwar eine Infrastruktur von Maschinen, aber noch keine Oracle-FMW-Software. Für die Bereitstellung der Exa-Templates gibt es nun verschiedene Möglichkeiten: entweder die Installation der Oracle-Software auf den Templates der vServer oder die Installation auf Basis von Skripten und Provisioning der Software durch Cloud Control. Bei der ersten Option ist mit dem Erzeugen des vServers schon die Software vorhanden. Dieser Vorteil der schnellen Bereitstellung wird aber durch einen Nachteil eingeschränkt: Bei Änderungen des Betriebssystems und/oder der Software ist ein neues vServer-Template notwendig. Angesichts der Tatsache, dass regelmäßig Patches der Oracle-Software beziehungsweise des Betriebssystems erscheinen, ein nicht zu vernachlässigender Aufwand. Die Bereitstellung der Software durch Cloud Control ist eine valide Option, die zwar einigen Aufwand erfordert, da die Provisioning-Skripte eine gewisse Lernkurve haben, jedoch den Vorteil mit sich bringen, dass die Administration der Software innerhalb des Cloud Control schon erfolgt ist. Diese Option basiert auf den Silent-Install-Möglichkeiten der Oracle-Software. Da die Software-Installation ein generischer Prozess ist, der eventuell mehrmals auf einem vServer stattfinden muss (etwa bei parallelen Installationen zum Zweck der Unabhängigkeit der Komponenten), erfolgt innerhalb der Exa-Templates die Installation auf Basis von Silent Installations. Die Ansteuerung ist in der Regel durch ein einfaches XML-Konfigurationsfile denkbar einfach. Der Aufruf aus dem Cloud Control erfolgt über

das Enterprise-Manager-Command-Line-Interface (EMCLI). Hierbei werden einige Befehle abgesetzt, die das betreffende Skript starten.

Nach dem Aufbau des Cloud-Control-vServers und der Installation der Cloud-Control-Umgebung ist die Software-Library des Cloud Control zu füllen. Dies kann manuell erfolgen (Download der Software von <http://otn.oracle.com>). Für eine Exalogic, die von mehreren Kunden benutzt wird (Multi-Tenant), ist die Bereitstellung auf einem Shared-Filesystem zu empfehlen, das dann die jeweilige Cloud-Control-Software-Library befüllt, beispielsweise ein vServer mit Samba/NFS.

Abhängig vom Einsatzzweck der Exalogic-Umgebung wird nun das Ziel aus den Exa-Templates gewählt. Diese sind Abbildungen der Enterprise Deployment Guides (EDG) von Oracle. Darin wird eine System-Landschaft dargestellt. Diese technische Architektur hat als Merkmal, dass die verschiedenen Oracle-Fusion-Middleware-Komponenten auf dedizierten Rechnern installiert sind und dort die jeweiligen Dienste anbieten. So sind beispielsweise die Identity-Management-Komponenten über einzelne Rechner zu erreichen. Die Namensgebung innerhalb solcher EDGs zeigt dann auch immer die Funktion des Rechners, zum Beispiel „oam.mydomain.local“, wenn der Rechner den Oracle Access Manager beherbergt.

Der Rollout dieser EDG-basierten Umgebungen findet über das Ops Center statt. Hierbei kommt derselbe Mechanismus wie beim Aufbau des Cloud Control vServers zum Einsatz. Auf der Ebene des Ops-Center-Clients (webbasiert) sind durch die Autoren Skripte verfügbar, die ein XML-Konfigurationsfile lesen. Dieses XML-File enthält

```
iaas-run-vserver --base-url https://exactrl/ --access-key-
file /home/oracle/exatemplate/accesskey.file --vnets VNET-
74aba322-2c13-4c86-4465-1dd7f9c8aadf --key-name myKeyPair
--name cloudctrlvserver --server-template-id TMPL-4b4a8ed3-
a665-45a1-9c75-b11c25a55632 --ip-addresses 192.168.0.2
--vserver-type 457
```

Listing 1

die Beschreibung des EDG. Neben dem Aufbau der Rechner-Landschaft (im Grunde die vServer-Landschaft) enthält das XML-File auch die zugeordneten Parameter der vServer wie den verfügbaren Speicher, Disks und Netzwerk-Anschlüsse.

Diese EDG-XML-Konfiguration ist in den Ausprägungen „single instance“, „high availability“ und „heavy duty“ verfügbar. Diese geben an, mit welcher Verfügbarkeit und Kapazität die Umgebungen aufgebaut werden sollen. Bei „heavy duty“ ist zum Beispiel an ein massives System mit wenigstens vier Managed Servern pro Komponente zu denken. Der Ansatz, die EDGs in XML-Konfigurationsfiles zu beschreiben, ermöglicht eine schnelle Anpassung an die Kundenanforderungen.

Nach Bereitstellung der vServer in der EDG-Ausprägung installiert Cloud Control die Oracle Software, anschließend kann die Erstellung der Domänen-Konfiguration erfolgen. Vergleichbar mit den Deployment Plans der vServer umfassen die Exa-Templates auch den Bereich der vollständigen Erstellung von Domains per Skript. Natürlich sind diese aufeinander abgestimmt, so gibt es in der „Single Instance“-Version kein Clustering von Managed Servern, während dies in der „High Availability“-Umgebung auf die jeweiligen Maschinen (vServer) verteilt wird. Entsprechend der Deployment Plans der vServer ist auch hier eine Anpassung einfach zu erreichen. Wenn für bestimmte Kunden eine andere Namenskonvention erforderlich ist oder entsprechende WebLogic-Server-Einstellungen gewünscht sind, lassen sich diese einfach anpassen.

Die Vorteile der Exa-Templates

Nach Ansicht der Autoren liegt der größte Vorteil der Exa-Templates darin, dass die Unterstützung des Lifecycle eines Projekts und dessen Systeme gegeben ist. Oft genug findet sich in IT-Projekten eine ansteigende Linie, was die Infrastruktur-Komponenten betrifft. Die Entwicklung findet oft auf Laptops (gegebenenfalls sogar von externen Mitarbeitern) statt; der Test erfolgt auf einer einfachen Umgebung, die meistens nur einen Teil der Kom-

ponenten umfasst; die Integration ähnelt dann schon dem späteren System; die Produktion ist dann nicht selten die erste und einzige Umgebung, in der alle NFRs wie Hochverfügbarkeit und Twin Data Center vorhanden sind. Grund dafür sind Kostengründe (Hardware) und die Komplexität der Bereitstellung.

Mit Nutzung der Exa-Templates ist dies einfach zu umgehen, da die Erstellung nur einen Bruchteil der Zeit in Anspruch nimmt. Darüber hinaus sind die Umgebungen der verschiedenen Lebenszyklen so aufgebaut, dass sie zumindest logisch vergleichbar sind (wenn etwa Clustering erforderlich ist, dann wird dies in allen Umgebungen angeboten). Hiermit kann zu Recht von einer Turn-Key-Lösung gesprochen werden, die konsistent und schnell die Infrastruktur eines Projekts liefert.

In Bereich des Backup und Recovery liegt mit dieser Lösung ein weiterer Erfolgsfaktor. Es müssen nicht alle Systeme gesichert werden – es reicht, wenn die Installations- und Konfigurationskripte vorhanden sind. Ein Restore ist damit schnell möglich.

Zudem besitzen alle Umgebungen eine eindeutige Namenskonvention. Dies erleichtert dem Betrieb die Ausführung der operationalen Tätigkeiten, da immer deutlich ist, welchen Einsatzzweck ein vServer hat. Hier zeigt sich die Bedeutung der anfänglichen Installation des Cloud Controls. Alle vServer und die darauf installierte und konfigurierte Oracle-Software werden selbstverständlich dort administriert.

Exa-Templates sind zurzeit für die SOA Suite, Identity Management und WebCenter verfügbar beziehungsweise in Vorbereitung. Eine BI-Version ist noch in Arbeit, wobei der Nachdruck auf dem Einsatz der Exalytics-Maschine liegt. Noch ein Wort zur Datenbank: In den Exa-Templates wird davon ausgegangen, dass eine Datenbank zur Verfügung steht. Bei den Autoren war dies eine Exadata, aber dies ist nicht zwingend notwendig – eventuelle Anpassungen der Exa-Templates bei Datenbanken auf externen Systemen sind schnell realisiert.

Matthias Fuchs

matthias.fuchs@ise-informatik.de



Andreas Chatziantoniou

andreas@foxglove-it.nl



Unsere Inserenten

DOAG Deutsche ORACLE-Anwendergruppe e.V.	U2
DOAG Deutsche ORACLE-Anwendergruppe e.V.	U3
Hunkler GmbH & Co. KG www.hunkler.de	S. 3
its-people/enterpriser GmbH & Co. KG www.its-people.de	S. 53
Krankenhausverbund gemeinnützige GmbH www.clinotel.de	S.43
Libelle AG www.libelle.com	S. 15
MuniQsoft GmbH www.muniqsoft.de	S. 35
Universitätsmedizin Göttingen	S.63
Trivadis GmbH www.trivadis.com	U4