Minimale Downtime beim Patchen von Failover (Flying) Oracle Solaris Containern

Hartmut Streppel , Detlef Drewanz
Oracle Deutschland BV & Co. KG
München, Berlin

Schlüsselworte:

Oracle Solaris Container, Oracle Solaris Zonen, Oracle Solaris Patching, Oracle Solaris Cluster, Update-on-attach, Live Upgrade

Einleitung

Oracle Solaris Container, die oft auch Zonen genannt werden, ermöglichen die Konsolidierung einer großen Zahl von Anwendungen. Die Anwendungen werden dann jeweils in Oracle Solaris Containern installiert, die alle in einer Oracle Solaris Instanz angelegt und betrieben werden. Solche Umgebungen müssen sehr sorgfältig geplant und betrieben werden, da sich Fehler sofort auf eine Vielzahl von Oracle Containern, und damit Anwendungen und Kunden, auswirken können. Einer der kritischen Punkte im Betrieb ist das Thema Patch Management, das auch das Installieren von Updates mit einschließt. (Im Folgenden wird der Begriff Update für beide Aufgaben verwendet.)
Eines der Probleme in hoch konsolidierten Umgebungen ist es, gemeinsame Downtimes, z.B. für das Patch Management zu finden. Traditionelle Verfahren wie z.B. das Durchführen des Updates während einer Downtime und auch fortgeschrittenere wie z.B. Live Upgrade, erfordern mindestens eine gemeinsame Downtime für das Durchführen eines Reboots, um den Update zu aktivieren.
Daraus ergeben sich folgende Fragen, die wir in diesem Papier kurz anreißen wollen:

- Wie können Updates für Zonen mit minimaler Downtime durchgeführt werden?
- Welche Möglichkeiten gibt es, Updates für einzelne Zonen durchzuführen?
- Wie können Daten beim Update von Zonen behandelt werden?
- Gibt es eine Fallback-Möglichkeit, falls ein Update fehlschlagen sollte.

Wir werden zunächst ein entwickeltes Verfahren für nicht-geclusterte Systeme vorstellen und am Schluß die notwendigen Erweiterungen in einer Oracle Solaris Cluster Umgebung erklären.

Updates für Oracle Solaris Systeme, auf denen Oracle Container installiert sind

Softwarestände in der globalen und allen lokalen Zone auf einem System müssen so weit es geht, identisch sein. Diese Anforderung erfüllen die in Oracle Solaris enthaltenen Werkzeuge, die z.B. beim Update einer globalen Oracle Solaris Zone auch alle nicht-globalen Zonen mit auf den neuen Softwarestand bringen.
Wenn nun eine Zone von einem System mit einem alten Softwarestand auf ein System mit einem neuen Softwarestand migriert werden soll, muss vor dem “Anhängen” an das neue System ein Software-Update durchgeführt werden. Hierzu bietet sich das Update-on-attach Verfahren an, das dies automatisch als Teil des Prozess des Anhängens durchführt.
An dieser Stelle ist es sinnvoll, die unterschiedlichen Verfahren, einzelne Zonen auf einen neuen Softwarestand zu bringen, miteinander zu vergleichen (das Standardverfahren mit langer Downtime für alle Zonen wird dabei außer Acht gelassen).

Prinzipiell können zwei Arten von Verfahren unterschieden werden:

- Update einer Kopie einer Zone und
- Anlegen einer neuen Zone

Beide Verfahren können noch einmal unterteilt werden. Beim Arbeiten mit einer Kopie der aktiven Zone können verwendet werden:

- Live Upgrade, welches eine Kopie erzeugt und dann den Software-Update auf dieser Kopie durchführt
- Manuelles Duplizieren der Zone und Durchführen des Software Updates auf dem Duplikat.

Beim Erzeugen einer neuen Zone gibt es ebenfalls zwei Verfahren:

- Neuanlegen einer Zone auf der Basis einer Referenzzone, die bereits alle notwendigen Anpassungen enthält.
- Neuanlegen der Zone mit Hilfe eines Skripts, das sämtliche Anpassungen durchführt

Vergleicht man diese 4 Ansätze im Detail ergibt sich folgendes Bild:

- **Oracle Solaris Live Upgrade (LU)**: LU legt eine Kopie der Root-Dateisysteme aller Zonen an, die auf dem System konfiguriert sind, während das System und alle Zonen aktiv sind. Diese Kopie wird entweder auf separate Dateisysteme kopiert oder, falls ZFS genutzt wird, als ZFS Snapshot angelegt. Der Update-Prozess wird auf diesen Kopien durchgeführt, ebenfalls, während das Gesamtsystem produktiv ist. Ein Aktivieren des neuen Softwarestandes erfolgt durch einen Reboot des Systems.
- **Manuelles Duplizieren einer Zone und nachfolgender Update**: Ein Duplikat der laufenden Zone wird erzeugt (was ein Duplizieren der Daten mit einschließen kann). Die duplizierte Zone wird dann an ein anderes System mit neuerem Softwarestand angehängt und ein Update-on-attach durchgeführt. Die Originalzone wird dann gestoppt und das Duplikat auf dem anderen System gestartet.
- **Update einer neuen Zone aus einer Referenzzone**: Eine vollständige Referenzzone wird dupliziert und dann mit Update-on-attach an einem neuen System angehängt. Alternativ kann auch die Referenzzone selbst auf dem neuen System zur Verfügung gestellt werden.
- **Skript-gesteuertes Neuanlegen einer Zone auf einem neuen System**

UPDATE PROZESS	AUFFWAND, DIE ZONE ZU DUPLIZIEREN	ERHALT DER ÄNDERUNGEN AN DER ZONE	INDIVIDUELLER ZONEN-UPDATE MÖGLICH	ZWEITES SYSTEM NOTWENDIG	AKTIVIEREN DES UPDATES
Live Upgrade	Gering - lucreate	LU kopiert den aktuellen Stand	Nein	Nein	Reboot
Duplizieren und Update der Zone	Gering – zfs snapshot	Das Duplizieren, dupliziert auch alle Anpassungen	Ja	Ja	Stop der alten Zone, Start der neuen Zone
Duplizieren einer Referenzzone	Mittel – ständiges Anpassen der Referenzzone und dann Duplizieren der Referenzzone	Change Management der Referenzzone	Ja	Ja	Stop der alten Zone, Start der neuen Zone
Neuanlegen einer Zone mit einem angepassten Skript	Hoch – ständiges Anpassen des Skripts und Neuanlegen der Zone mit Hilfe des Skripts	Change Management des Skripts	Ja	Ja	Stop der alten Zone, Start der neuen Zone

Tabelle 1: Vergleich verschiedener Update-Methoden

Es ist wichtig, den Inhalt der zweiten Spalte etwas näher zu erläutern. Aus theoretischer Sicht des Change Management sind sicherlich Verfahren, die eine ständige Anpassung von Referenzinstallationen oder Skripts verfolgen, vorzuziehen. Damit ist gewährleistet, dass jederzeit, überall und von jedem Nutzer (wenn möglich) eine standardisierte Umgebung entweder aus dieser Referenzumgebung oder mit Hilfe eines Werkzeugs erzeugt werden kann. Dies erfordert natürlich entsprechenden Aufwand bei der ständigen Aktualisierung, der sich wiederum nur lohnt, wenn nicht nur eine Kopie einer Referenz erzeugt wird, sondern viele.

Aus vielen Diskussionen mit Kunden wissen wir, dass zwar aufwändiges Change Management betrieben wird, aber doch lieber das aktuell produktive System dupliziert wird. So werden mit geringstem Aufwand und hoher Sicherheit auch wirklich alle Änderungen des Systems mitgenommen, wenn ein System mit neuem Softwarestand zu erzeugen ist.

Empfehlungen für die Struktur einer Zonenumgebung

Die Empfehlungen für die Struktur unterscheiden sich nicht von denen für eine Anwendungsumgebung in einer globalen Zone. Anwendungsdaten sollten grundsätzlich vom Zonenpfad getrennt sein, damit sie auch von einer Anwendungsumgebung auf einem anderen Server oder ein einer anderen Zone genutzt werden können.

Die Frage nach dem Installationsort für die Anwendungsprogramme selbst ist immer wieder umstritten. Sollen die Programme einmal und separat von den Daten installiert werden, z.B. im Zonenpfad ? Oder ist es besser, sie auch auf „shared Storage“ abzulegen, damit ihre Pflege nur einmal notwendig ist. Im Allgemeinen hat sich die Erkenntnis durchgesetzt, dass eine Installation „lokal“, z.B. im Zonenpfad, sinnvoller ist. Mit dieser Methode sind Updates der Programme zusammen mit dem System Update möglich, während die Daten von einer anderen Zone und dem aktuellen Programm in Benutzung sind.

Um die vollen Möglichkeiten, die Oracle Solaris bietet auszunutzen, gehen wir davon aus, dass der Zonenpfad einer Zone auf ZFS liegt. Nur so können die einfachen und sicheren Methoden für Snapshots u.ä. genutzt werden.

Individuelles Update von nicht geclusterten Zonen

Das vorgeschlagene, getestete und dokumentierte Verfahren nutzt die Eigenschaften von ZFS, um ein Duplikat der aktuell laufenden Zone zu erzeugen. Dieses Duplikat wird dann in zwei Schritten einem zweiten System, das einen neuere Softwarestand aufweist, übertragen und dort mit dem sog. Update-on-Attach Verfahren auf den aktuellen Softwarestand gebracht. All dies passiert, ohne dass die originale Zone beeinträchtigt oder sogar angehalten wird. Erst im letzten Schritt wird die alte Zone gestoppt und die neue Zone auf dem neuen System gestartet. Beim Umschalten auf die neue Zone werden auch die Anwendungsdaten der alten Zone der neuen Zone zur Verfügung gestellt und beim Start genutzt.

Allein in dieser letzten kurzen Umschalt-Phase steht die Anwendung, die in der Zone betrieben wurde, nicht zur Verfügung – also eine wahrliche Minimierung der Unterbrechungszeit für die Nutzer der Anwendung während der Update-Phase.

Details für nicht geclusterte Zonen

Wir benutzen als Beispiel zwei Systeme mit Oracle Solaris, die TOM und JERRY genannt werden, wobei JERRY den aktuelleren Softwarestand enthält. Die benutzte Zone nennen wir ora_zone. Diese hat ihr Rootfilesystem in dem ZFS dataset ora_zpool/ora_zone.

Das vorgeschlagene Verfahren lässt sich in wenigen Schritte zusammenfassen, die im folgenden prinzipiell an kurzen grundlegenden Kommandos erläutert werden.

- Duplizierung des Rootfilesystems der produktiven Zone auf dem produktiven System (TOM)
- Erzeugung und Update der neuen Zone aus dem Duplikat auf einem neuen System (JERRY)
- Testen der neu erzeugten Zone
- Umschalten der Anwendung von der alten auf die neue Zone (ora_zone auf JERRY)

Duplizierung des Rootfilesystems der produktiven Zone im laufenden Betrieb:
Das Ziel des Verfahrens ist, mit möglichst wenig Unterbrechungszeit für die Anwendung den Softwarestand der Laufzeitumgebung zu aktualisieren. Dafür wird eine Kopie der Laufzeitumgebung benötigt, die unabhängig aktualisiert werden kann.

Unsere Laufzeitumgebung ist ein Oracle Solaris Container. D.h. wir erzeugen eine Kopie des Rootfilesystems der Zone. Die Daten der Anwendung sind hierbei nicht enthalten, da sie sich auf einem separaten Dateisystem, also nicht im Rootfilesystem der Zone befinden.

Mit einem ZFS Snapshot (`zfs snapshot`) erzeugen wir einen konsistenten Zustand des Rootfilesystems einer Zone. Aus diesem Snapshot wird anschließend ein identisches Rootfilesystem der Zone in einem neuen zpool erzeugt (`zfs send | zfs receive`). Der unabhängige zpool kann nun sehr einfach von einem System zu einem anderen System (von TOM zu JERRY) bewegt werden, wenn der für den zpool verwendete Storage, d.h. die LUN, in einem SAN liegt. Dazu wird der zpool auf TOM exportiert (`zpool export`) und auf JERRY importiert (`zpool import`). Hierbei ist zu beachten, dass der zpool mit dem Duplikat eine andere Bezeichnung trägt als der zpool mit dem Original. Nun liegt das Duplikat des Rootfilesystems von ora_zone auf JERRY vor.

Erzeugung und Update der neuen Zone aus dem Duplikat auf einem neuen System:
Aus dem Rootfilesystem der Zone kann nun eine Zone erzeugt werden. Dazu wird eine Zonen-Konfiguration benötigt, die entweder bereits vorhanden oder identisch zu der produktiven Zone anzulegen ist. Um spätere Verwechselungen auszuschließen, sollte die neu zu erzeugende Zone einen anderen Namen erhalten als die produktive Zone.

Die Erzeugung und das Update der Zone auf den neueren Softwarestand des Systems erfolgt mit der update-on-attach Funktion von Solaris (`zoneadm attach -U`). Hierbei wird der Softwarestand der globalen Zone mit dem Softwarestand aus dem Rootfilesystem der Zone verglichen, die Differenzen ermittelt und diese in die Zone eingespielt. Im Ergebnis liegt eine installierte Zone vor, die den Softwarestand der globalen Zone hat.

Für diesen Schritt gilt es, die Zonenkonfiguration zu beachten. Da diese aus der Originalzone abgeleitet wurde, ist hier auch die Benutzung der Storagekomponenten festgelegt. Diese sind jedoch auf dem produktiven System noch in Benutzung und dürfen während der Ausführung von update-on-attach nicht benutzt werden. Hier muss entweder für diesen Zeitraum die Konfiguration kurzzeitig modifiziert oder dafür gesorgt werden, dass die Nutzung der vorhanden Konfiguration nicht zu Konflikten mit der produktiven Zone führt.

Testen der neu erzeugten Zone:
Nach dem erfolgten Update kann nun die neue Zone das erste Mal gestartet werden. Um die Zone mitsamt ihren Anwendungen testen zu können, können anstelle der Produktivdaten auch Beispieldaten während des Tests verwendet werden. Auch hier gilt es die Zonenkonfiguration zu beachten. So sind die konfigurierten IP-Adressen identisch zu denen in der produktiven Zone. Zur Vermeidung von IP-Adresskonflikten mit der produktiven Zone kann die Netzwerkkonfiguration während der Testphase kurzzeitig umkonfiguriert werden. Alternativ können die Tests auch in einer abgeschlossenen Netzwerkumgebung erfolgen. Bei diesem abschließenden Test wird sichergestellt, dass die Aktualisierung der Zone erfolgreich war.

Umschalten der Anwendung von der alten auf die neue Zone:
Nun ist die Zeit für den Rollentausch der Zonen gekommen. Die alte Zone wird angehalten und die neue Zone wird als produktive Zone gestartet. Zunächst müssen dafür der neuen Zone aber die aktuellen Anwendungsdaten zur Verfügung gestellt werden. Diese befinden sich noch auf TOM in einem separaten zpool. Durch das Anhalten der Zone wurde die enthaltene Anwendung beendet und hat ihre Datenbereiche in diesem zpool geschlossen. Dieser zpool wird nun an TOM exportiert (`zpool export`) und an JERRY importiert (`zpool import`) und steht so der neuen Zone beim Start zur Verfügung.

Damit ist der Prozess zum Update der Zone abgeschlossen. Die Ausfallzeit für die Anwendung war lediglich die Zeit zwischen dem Anhalten der alten Zone auf TOM bis nach dem Starten der neuen Zone auf JERRY.

Nicht nur für den Fall, dass ein Update fehl schlägt, sollte eine Fallback Variante vorgesehen werden. Nach dem letzten Schritt haben die alte und die neue Zone die Rollen getauscht. Somit steht die alte Zone nach dem Rollenwechsel noch zur Verfügung und kann als Fallback Variante genutzt werden. In dem Fall werden die Anwendungsdaten wieder zurück zu TOM übertragen und von der alten Zone beim Start benutzt.

Was ändert sich bei Zonen, die unter Oracle Solaris Cluster Kontrolle sind ?

Das beschriebene Verfahren ist einfach und sicher. Da viele Kunden solche Failover Zonen allerdings unter (Oracle Solaris) Cluster Kontrolle, d.h. unter der Kontrolle des HA Container Agenten verwenden, stellt sich nun aber die Frage, ob und um wieviel das Verfahren komplexer wird. Interessanterweise sind zwar einige zusätzliche Schritte durchzuführen, andererseits fallen aber auch einige Schritte weg und der Umstieg von der alten auf die neue Zone – und auch der rückwärtige Weg (Fallback) im Falle eines Problems, können mit wenigen Schritten elegant bewältigt werden.

Grundsätzlich gibt es vier wesentliche Unterschiede zum Verfahren mit nicht-geclusterten Zonen:

- Die neue Zone auf dem zweiten Rechner muss nicht erst konfiguriert werden, da sie in einem Cluster, in dem die Zone unter der Kontrolle des HA Container Agenten steht, schon existiert.
- Sobald einer der Clusterknoten auf einem anderen Softwarestand ist, muss verhindert werden, dass geclusterte Zonen zwischen Clusterknoten geschwenkt werden.
- Das Stoppen der alten und Starten der neuen Zone geschieht mit Hilfe von Cluster-Kommandos.
- Clusterknoten dürfen nur für den Zweck und den Zeitraum eines „Rolling“ oder „Dual Partition“ Update mit unterschiedlichen Softwareversionen in einem Cluster betrieben werden. Die Zeitspanne, in der Zonen individuell auf einen neuen Stand gebracht werden, ist also kurz.

Details für geclusterte Zonen

Die Unterschiede zu nicht-geclusterten Zonen beginnen in dem Moment, in dem auf dem zweiten System eine neue Zone konfiguriert werden muss, um das Duplikat zu beherbergen. Diese Zonenkonfiguration existiert in einer Clusterkonfiguration schon, da dies eine Voraussetzung dafür ist, dass eine Zone unter Kontrolle des HA Container Agenten betrieben wird.

Das folgende Beispiel mit Kommandos der Oracle Solaris Cluster Kommandozeilenschnittstelle zeigt, wie in wenigen Schritten die alte Zone gestoppt, eine Cluster-Ressource umkonfiguriert und dann die neue Zone gestartet wird. Im folgenden Beispiel wird eine Oracle Solaris Cluster Ressource Gruppe, ora_zone-rg, betrachtet. Diese besteht aus drei Ressourcen - ora_zone-rs, hazone und ora_zone-hasp - und kann auf den beiden Knoten TOM mit altem und JERRY mit neuen Softwarestand laufen.

```
clrg set -p nodelist=TOM,JERRY ora_zone-rg
```

Hier wird die Liste der durch die Ressourcegruppe verwalteten Knoten auf den reduziert, der das alte Software-Release nutzt. Diese Aktion sollte natürlich schon zu dem Zeitpunkt durchgeführt werden, zu dem der Software-Update auf dem zweiten Knoten beginnt.

```
clrs disable ora_zone-rs
```

Dieses Kommando stoppt die Ressource, die den Container repräsentiert und fährt auch die Zone herunter.

```
clrs set -p Zpools=ora_zpool-cloned,ora_dpool ora_zone-hasp
```

Die Cluster-Ressource, die dafür sorgt, dass beim Start der Zone die notwendigen Zpools verfügbar sind wird so geändert, dass sie den neuen Zpool der duplizierten Zone verwendet.

```
clrg switch -n JERRY ora_zone-rg
```

Mit diesem Kommando wird die gesamte Ressourcegruppe auf den „neuen“ Rechner geschwenkt. D.h. dass auf dem System JERRY die Ressourcegruppe gestartet wird.

```
clrg set -p nodelist=JERRY ora_zone-rg
```

Die Liste der nutzbaren Knoten wird nun auf den neuen Knoten gesetzt, da ja die Zone mit dem neuen Softwarestand nur auf dem neuen Knoten laufen darf.

```
clrs enable ora_zone-rs
```

Und schließlich wird die Container-Ressource wieder gestartet, wodurch implizit die Zone gestartet wird.

Die Änderungen an dieser Kommandosequenz die notwendig sind, um die Zone so zurück zu schwenken, dass die alte Zone auf dem anderen Knoten wieder gestartet wird, sind offensichtlich. Die Serviceunterbrechung ist allein die Zeit zwischen dem „disable“ bis zur Beendigung des „enable“ der Zonenressource.

Zusammenfassung

Das beschriebene Verfahren zeigt, wie durch die konsequente Ausnutzung verschiedener Oracle Solaris Technologien die geplanten Ausfallzeiten beim Update von Zonen und Software minimiert werden können. Dieses Verfahren ist um so wichtiger, je mehr Zonen auf einem physischen System betrieben werden und je höher die Anforderungen an die Verfügbarkeit sind. So kann der Softwarestand jeder Zone selbst in geclusterten Umgebungen separat, sicher, schnell und mit wenig Aufwand aktualisiert werden.

Literaturverzeichnis

- How to Upgrade and Patch with Oracle Solaris Live Upgrade, Oracle Whitepaper, May 2010 by Jeff McMeekin
  http://www.oracle.com/technetwork/server-storage/solaris/solaris-live-upgrade-wp-167900.pdf

- System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones
  http://download.oracle.com/docs/cd/E18752_01/html/817-1592

- Oracle Solaris Cluster Data Service for Solaris Containers Guide
  http://download.oracle.com/docs/cd/E18728_01/html/821-2677/

- Maintaining Solaris with Live Upgrade and Update On Attach; Sun Microsystems Blueprint, September 2009 by Hartmut Streppel, Dirk Augustin and Martin Müller
  http://www.oracle.com/technetwork/server-storage/archive/a11-028-sol-live-upgrade-455915.pdf

- Using Live Upgrade in complex environments, Enda O'Connor
  http://blogs.oracle.com/patch/entry/using_live_upgrade_in_complex

- ZFS Best Practices Guide
  http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide

- Oracle Solaris ZFS Administration Guide
  http://download.oracle.com/docs/cd/E18752_01/html/819-5461

Kontaktadresse:

Hartmut Streppel
ORACLE Deutschland B.V. & Co. KG
Riesstr. 25
D-80992 München

Telefon:	+49 (0) 89 1430-2588
Fax:	+49 (0) 89 1430-1150
E-Mail	Detlef.Drewanz@oracle.com; Hartmut.Streppel@oracle.com
Internet:	http://www.oracle.com