

Solaris 11 Repositories Best Practices

Detlef Drewanz
Oracle Deutschland B.V. & Co. KG
Berlin

Schlüsselworte

Oracle Solaris 11, IPS, Repository

Einleitung

Oracle hat mit Oracle Solaris 11 das Konzept der IPS Software Repositories eingeführt, die für Installationen und Updates des Betriebssystems, aber auch für andere Software genutzt werden. Der Umgang mit Repositories und deren Einbindung in Unternehmensnetzwerke oder in Abläufe im Rechenzentrum ist nach wie vor für Kunden und Partner Neuland. Dieser Text diskutiert den Aufbau und die unterschiedlichen Formen von Repositories, sowie deren Einsatzvarianten.

Oracle Solaris 11 Pakete, Repositories und Updates im Überblick

Oracle Solaris 11 Software wird mit Image Packaging System (IPS) Paketen verteilt. Diese IPS-Pakete werden in IPS Package Repositories durch IPS Publisher bereitgestellt.

Mit dem Oracle Solaris 11 11/11 Release wird ein komplettes Repository aller Oracle Solaris 11 Pakete bereitgestellt. Updates der Paketen sind bis zum Erscheinen des nächsten Release über Support Repository Updates (SRU) erhältlich, die monatlich erscheinen.

Grundlegende Prinzipien des IPS-Paketmanagements sind Versionsnummern der Pakete und Abhängigkeiten zwischen Paketen, die vom IPS automatisch ausgewertet und durchgesetzt werden.

Die Versionsnummern werden durch die Paketverwaltungs-Tools ausgewertet. Ein Oracle Solaris 11 Paket nutzt die folgende Versionierung für Pakete:

```
name@release_version,release-trunk_id.update.SRU.reserved.buildid.nightlyid
```

Für das Paket *entire* vom SRU 7.5 stellt sich die Versionsnummer z.B. wie folgt dar:

```
detlefd@cantaloup:~$ pkg list entire
NAME (PUBLISHER)          VERSION          IFO
entire                   0.5.11-0.175.0.7.0.5.0  i--
```

Für unsere Zwecke betrachten wir lediglich den letzten Teil des Versionsstrings: 0.175.0.7.0.5.0

Die *trunk_id* ist bei Oracle Solaris 11 immer 0.175.

Das Paket ist aus Update 0 (=FCS), das SRU ist 7 und die *buildid* des SRU ist 5.

Die Versionsnummern der Pakete werden u.a. während des Oracle Solaris 11 Update Prozesses verwendet. Ein Update eines Systems wird durch den Befehl `pkg update` initiiert. Dazu wird das Incorporation-Paket *entire* verwendet. Dieses Paket enthält keine Software, sondern optionale Abhängigkeiten zu allen Paketen, die zu einem Update Release oder SRU gehören. Optionale Abhängigkeit bedeutet in diesem Fall: wenn ein Paket installiert ist, muss es eine bestimmte Version haben. Ist es nicht installiert, wird es nicht angefordert. Welche Pakete installiert werden, ist z.B.

durch das Gruppenpaket `solaris-small-server` abgebildet. Während des Updates werden die Versionsnummern der installierten Pakete mit den Versionsnummern in dem im Repository neu vorliegenden `entire` Paket abgeglichen, das mit einem SRU oder einem neuen Release Update zusammen ausgeliefert wird. Entsprechende neue Versionen von Paketen werden nachinstalliert. Die zu installierenden Pakete werden dazu aus dem konfigurierten Repository geladen. Der eigentliche Download wird jedoch nur auf die tatsächlich veränderten Dateien minimiert, so dass nicht komplette Pakete, sondern in der Regel nur Teile davon heruntergeladen und installiert werden.

Oracle Solaris 11 IPS Repositories

Oracle Solaris 11 IPS Repositories enthalten die Betriebssystempakete. Da die IPS Pakete multiplattformfähig sind, enthält ein Repository sowohl den x86-, als auch den SPARC-Anteil der Pakete. Der Client, der das Paket von dem Repository installiert, lädt nur den für ihn relevanten Teil herunter.

Jedes IPS Paket enthält Metainformationen (Paket-Manifest) und den eigentlichen Content Payload, d.h. die mit dem Paket zu installierenden Dateien. Während die Manifeste im Repository unkomprimiert vorliegen, wird der Content Payload komprimiert abgelegt.

Repositories verfügen neben den Daten der Pakete über weitere unkomprimierte Metadaten wie den Catalog (die Liste der verfügbaren Pakete), den Manifesten der Pakete des Repositories und den Such-Index. Der Zugriff auf Repositories kann direkt über das Filesystem, per NFS oder per HTTP erfolgen. Der HTTP-Zugriff auf Repositories wird durch zu konfigurierenden `pkg.depotd (1M)` ermöglicht.

Es gibt zwei Typen von Repositories:

- Origin Repositories, die alle Metadaten und den Content der Pakete enthalten
- Mirror Repositories, die nur den Content von Paketen enthalten

Einsatzmöglichkeiten von Repositories

Vor dem Roll-out von Repositories sollten die unterschiedlichen Einsatzmöglichkeiten diskutiert werden. Hierbei sind Fragen zur Performance, Skalierbarkeit, Security und der Vereinheitlichung von Installationen über einen längeren Zeitraum zu betrachten. Die folgenden Fragen können bei der Qualifikation helfen.

- Sollen die zentralen Full- und Service-HTTP-Repositories von Oracle oder lokale Kopien verwendet werden ?
- Wie soll der Zugriff auf die lokalen Repositories erfolgen - durch das lokale Filesystem, per NFS, per direktem HTTP oder per HTTP-Proxy ?
- Sollten die SRU in die Full Repositories integriert werden ?
- Wird für jedes Update Release ein separates Repository bereitgestellt oder wird ein Repository für alle Release Updates verwendet ?
- Werden mehrere lokalen Kopien der Repositories benötigt ? Nach welchen Kriterien sind diese Kopien anzuordnen ?
- Wie können lokale Kopien untereinander synchronisiert werden ?

Zentrale Oracle Repositories oder eine lokale Kopie

Oracle stellt das Oracle Solaris 11 Repository auf <http://pkg.oracle.com/solaris/release/> bereit.

Die Support Repository Updates können von <https://pkg.oracle.com/solaris/support/> bezogen. Die dazu notwendigen Zertifikate können auf <https://pkg-register.oracle.com/> erzeugt werden.

Aus Security- und Performance-Gründen wird vielfach die Einrichtung von lokalen Kopien des Full- und des Support-Repositories bevorzugt bzw. die Nutzung der Internet Repositories steht überhaupt nicht zur Diskussion. Eine Verbindung von Systemen im Rechenzentrum zu Systemen im Internet ist oft nicht erlaubt oder möglich. Selbst Proxy-Verbindungen sind außerhalb des Erlaubten. Wird ein Repository Mirror verwendet, ist hier trotzdem noch der Zugriff auf die Metadaten im Netzwerk erforderlich.

Lokale Repositories bieten jedoch in den meisten Fällen einen schnelleren Zugriff. Der RZ-Betreiber stellt die lokale Repository-Kopie auf eigenen Servern bereit und kann so durch CPU-, Speicher- und Disk-Auswahl, sowie Netzwerk-Anbindung direkten Einfluß auf die Performance des Repository-Servers nehmen.

Zur Erzeugung eines lokalen Repositories werden die Full-Repository Images von <http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html> heruntergeladen. Die Support Repository Updates können für Support-Kunden als zip-Dateien von <http://support.oracle.com> bezogen werden (Siehe Oracle Solaris 11 Support repository Updates Index: <https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1372094.1>). Es sind entsprechende Prozesse im Rechenzentrum aufzusetzen, die für den automatisierten Bezug der neuesten Images sorgen.

Der Zugriff auf lokale Repositories

Der Zugriff auf die Repositories kann durch das Filesystem, per NFS oder per HTTP erfolgen.

Die Bereitstellung im lokalen Filesystem ermöglicht lediglich dem Host selber, auf das Repository zuzugreifen. Für Test- und Laborumgebungen oder Einzelumgebungen ist es jedoch die einfachste Variante.

Der Zugriff per NFS setzt eine NFS-Infrastruktur voraus. Diese ist vielfach vorhanden, da sie bereits in vielen Jumpstart-Installationen genutzt wurde. Für einen weit verteilten Zugriff - ggf. sogar per Proxy Diensten - ist NFS jedoch nicht geeignet, da in weit verteilten NFS-Umgebungen die Freischaltung von NFS-Ports in Firewalls oftmals zu komplex oder gar nicht möglich ist.

Die Verteilung von IPS Paketen per HTTP-Zugriff ist die flexibelste Variante. So können die Verbindungen auch über Proxy-, Cache- oder Load Balancer-Server verteilt werden. Eine Zugangsbegrenzung per Firewall erfordert nur die Freischaltung eines einzelnen Ports.

In der überwiegenden Zahl der Anwendungsfälle kommen HTTP-Repositories zum Einsatz.

Weiterhin ist zu prüfen, inwiefern die Full-Repositories aller Solaris 11 Updates in ein Repository und damit in ein Filesystem integriert werden oder ob für jedes Update ein eigenes Repository erzeugt wird. Kleinere Repositories für jedes Release können Performance Vorteile bieten, erfordern allerdings einen höheren administrativen Aufwand, da mit jedem Erscheinen eines Updates, die Publisher-Definition am Client zu erweitern bzw. zu verändern ist.

Repositories im Filesystem

Für die Einrichtung eines lokalen Repositories ergeben sich die folgenden Anforderungen und Empfehlungen:

- Der Host muß Oracle Solaris 11 als Betriebssystem benutzen.
- Das Repository sollte in ein eigenes ZFS-Dataset installiert werden. Wenn möglich kann auch ein eigener ZPOOL verwendet werden. Auf Grund der Art und Weise des Aufbaus und des Zugriffs auf Repositories sollten für große zentrale Repositories mit vielen Client-Zugriffen separate, ausreichend schnelle Festplatten genutzt werden. Separate ZFS-Datasets bieten die Möglichkeit für separierte ZFS-Snapshots und Recovery.
- Das ZFS-Dataset sollte die folgenden Eigenschaften haben:
- Das ZFS Dataset sollte `atime=off` als Attribute benutzen: So erfolgt keine Modifikation des Access-Time Attributes im Filesystem bei Filezugriff. Es wird Disk-Performance gespart.
- Das ZFS Dataset sollte `compression=on` als Attribute benutzen: Obwohl der Payload Content der Pakete im Repository bereits komprimiert vorliegt, lohnt sich die ZFS-Kompression noch für die Manifeste, die Katalogdaten und den Search-Index. Es wird Disk-Performance und Plattenplatz gespart, jedoch muß CPU-Zeit zur Kompression aufgewendet werden.
- Es sind mindestens 15 GB freier Plattenplatz erforderlich.
- Das System muß über ausreichend freien Plattenplatz (> 3 GB) in `/var/tmp` für die Repository Operationen verfügen.

Das folgende Beispiel zeigt exemplarisch die Erzeugung eines lokalen Repositories, das anschließende Neuerzeugen der index-Informationen des Repositories, die Einrichtung des `pkg.depotd(1M)`-Servers und das Umsetzen der Publisher Festlegungen am Client. Anschließend wird der Status des Repositories geprüft.

```
# zfs create -o mountpoint=/repo -o compression=on -o atime=off rpool/repo
# zfs create rpool/repo/s11
# mount -F hsfs full-repo-iso-image.iso /mnt
# cd /mnt/repo
# tar cf - . | (cd /repo/s11;tar xfp -)
# umount /mnt

# pkgrepo -s /repo/s11 refresh

# svccfg -s pkg/server setprop pkg/inst_root=/repo/s11
# svccfg -s pkg/server setprop pkg/readonly=true
# svcadm refresh pkg/server
# svcadm enable pkg/server

# pkg set-publisher -G 'http://pkg.oracle.com/solaris/release/' \
-g http://mein-repo-server solaris

# pkgrepo info -s http://mein-repo-server
PUBLISHER PACKAGES STATUS          UPDATED
solaris   4292      online          2011-10-26T17:17:30.230911Z
```

Support Repository Updates

Zur Behebung von Software-Fehlern in Oracle Solaris 11 IPS-Paketen werden Updates der Pakete in Support Repository Updates (SRU) für Service-Kunden bereitgestellt. Anders als in früheren Solaris Versionen werden keine Patches von Software Paketen erzeugt, sondern neue Pakete mit einer höheren Versionsnummer. Anhand der höheren Versionsnummer der Pakete erkennt `pkg update` die Verfügbarkeit von Updates und installiert diese. Wird nichts spezielles angegeben, wird jeweils die neueste Version eines Paketes installiert.

Updates von Softwarepaketen erscheinen nicht einzeln, sondern in monatlich Support Repository Updates (SRU). Diese enthalten jeweils die neueste Version aller Pakete, die seit dem Release eines Solaris 11 Updates entstanden sind, auf das sich das SRU bezieht.

Um also ein aktuelles Oracle Solaris 11 mit allen Updates in einem Repository bereitzustellen, werden das Full Repository und das letzte SRU benötigt. Ist jedoch beabsichtigt auch frühere SRU-Stände zu benutzen, so sind auch zwischenzeitig veröffentlichte diese SRU zu installieren, da Teile der älteren SRU bereits durch Teile neuerer SRU aktualisiert sein könnten. Die folgende Tabelle zeigt exemplarisch diese Zusammenhänge auf.

Paket	Version mit Solaris 11	Bestand SRU1	Bestand SRU2	Bestand SRU 3	Bestand SRU 4
pkgA	1.0	1.1 (neu)	1.1 (aus SRU1)	1.2 (neu)	1.2 (aus SRU3)
pkgB	2.7	-	2.8 (neu)	2.8 (aus SRU2)	2.8 (aus SRU2)
pkgC	3.4	-	-	-	4.0 (neu)
entire	1.0	1.1 (neu)	1.2 (neu)	1.3 (neu)	1.4 (neu)

SRU können auf verschiedene Art lokal verwendet werden:

- Jedes SRU wird als separates Repository bereit gestellt. Am Client ist für jedes zusätzliche Repository ein weitere Publisher nachzutragen.
- Alle SRU bezüglich eines Solaris 11 Update Releases werden zusammen in ein SRU-Repository integriert.
- Alle SRU aller Solaris 11 Update Releases werden in ein globales SRU-Repository integriert.
- Die SRU werden in das Full-Repository integriert.

Am einfachsten ist es, die erscheinenden SRU in das Full-Repository zu integrieren und nur einen Publisher an den Clients zu definieren. Mit steigender Anzahl der integrierten SRU steigt jedoch auch die Laufzeit des Integrations-Kommandos am Repository Server. Der Zeitaufwand zur Integration der Repositories wird hier jedoch weniger eine Rolle spielen, da dieser Prozeß automatisierbar ist und

nicht den Client betrifft. Andererseits ist eine Aufteilung der SRU auf mehrere Repositories aus administrativer Sicht am Client kaum praktikabel. Es ist jedoch zu prüfen, inwiefern ein Update an einem Client - ausgeführt von einem kombinierten Repository - genauso schnell ist und genauso viel Ressourcen benötigt, wie ein Update aus mehreren Repositories.

Das folgende Beispiel zeigt exemplarisch die Integration von SRU4 und die Neuerzeugung des Search-Index in das oben erzeugte Repository. Nach der Integration des SRU ist der Depot-Server neu zu starten.

```
# mount -F hsfs sru4-iso-image.iso /mnt
# pkgrecv -s /mnt/repo -d /repo/s11 '*'
# pkgrepo rebuild -s /repo/s11
# svcadm restart pkg/server
```

Das Kopieren von Repositories

In großen verteilten Netzwerken kann es erforderlich sein, mehrere lokale Kopien der Repositories vorzuhalten. In diesem Falle ist zu klären, wie die Repositories effizient auf einem Stand gehalten werden können. Dafür bieten sich zwei Varianten an:

- Erzeugung von ZFS Snapshot der Repositories, Übertragung der Snapshots mit `zfs send/receive` und Ersetzung der alten Repositories: Dieser Prozeß kann durch die Nutzung von inkrementellen ZFS-Snapshots sehr effizient umgesetzt werden, überträgt jedoch trotzdem eine Menge an Daten über das Netzwerk. Außerdem ist der Prozeß selbst zu entwickeln, kann dadurch aber auch an die spezielle RZ-Umgebung angepasst werden.
- Nutzung von `pkgrecv` zur online Synchronisation der Repositories: Dies ist der von Solaris 11 vorgesehene Prozeß zur Übertragung von Repositories. Mit dem Aufruf eines Kommandos wird der gesamte Prozeß angestoßen. `pkgrecv` gleicht die zu übertragenden Pakete und Metainformationen mit den bereits vorhanden ab und überträgt nur die wirklich neuen Informationen. Der Prozeß ist sehr netzwerkeffizient, benötigt jedoch auch eine längere Zeitspanne und Ressourcen (Hauptspeicher) auf dem Server auf den das Repository übertragen wird.

Ein Domainname für mehrere Repositories

Wenn mehrere Repositories verwendet werden und auf diese per HTTP-Protokoll zugegriffen werden soll, müssen mehrere `pkg.depotd`-Server konfiguriert und gestartet werden. Im Ergebnis existieren mehrere Repositories, die über mehrere Ports erreicht werden. In Weitverkehrsnetzen erfordert das eine komplexere Konfiguration, ggf. die Einrichtung mehrerer Proxy-Server oder die Freischaltung weiterer Ports an den Firewalls.

Zur Vereinfachung der Architektur wird ein Proxy-Server aufgesetzt, der hinter einem Domainnamen und einem HTTP-Port über mehrere Prefixes auf mehrere Repositories zeigt. Im Ergebnis ist z.B. das folgende Setup umsetzbar:

<http://mein-proxy/s11-repo>
<http://mein-proxy/s11-sru>
<http://mein-proxy/private>

zeigt auf mein-proxy:2000 mit Repository /repo/s11-repo
zeigt auf mein-proxy:2001 mit Repository /repo/sru
zeigt auf mein-proxy:2002 mit Repository /repo/private

Zusammenfassung

Für das Roll-out von Oracle Solaris 11 sind Solaris Repositories notwendig. In den meisten Fällen werden sich Betreiber dafür, lokale Repositories aufzusetzen. Vor dem Roll-out von Repositories sollten jedoch die unterschiedlichen Möglichkeiten diskutiert werden. Hierbei sind Fragen zur Performance, Skalierbarkeit, Security und der Vereinheitlichung von Installationen über einen längeren Zeitraum zu betrachten. Weiterhin ist zu klären, ob ein einzelnes großes oder mehrere kleine Repositories genutzt werden und ob die SRU in das Full-Repository integriert werden.

Literaturverzeichnis

- Evaluating Oracle Solaris 11

<http://www.oracle.com/technetwork/server-storage/solaris11/overview/evaluate-1530234.html>

- Copying and Creating Oracle Solaris 11 Package Repositories

http://docs.oracle.com/cd/E23824_01/html/E21803/index.html

- How to Create Multiple Internal Repositories for Oracle Solaris 11

<http://www.oracle.com/technetwork/articles/servers-storage-dev/int-s11-repositories-1632678.html>

- The Oracle Solaris 11 Package Branch Version Scheme [ID 1378134.1]

<https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1378134.1>

Kontaktadresse:

Detlef Drewanz
Oracle Deutschland B.V. & Co. KG
Komturstr. 18a
D-12099 Berlin

Telefon: +49 (0) 331-200 7341

E-Mail Detlef.Drewanz@oracle.com

Internet: <http://www.oracle.com>