

Solaris 11 Deployment – Willkommen in der Neuzeit

Ralf Germann
Trivadis AG
Glattbrugg

Schlüsselworte

Solaris 11, Solaris 10, Deployment, Automated Installer, Distribution Constructor, Boot Environment, Text Installer, Imaging Packaging System, Ablösung Jumpstart, Repositories, First-Boot-SMF-Services, Zonen

Einleitung

Wer sich jemals mit dem inzwischen in die Jahre gekommenen Jumpstart Server von Solaris beschäftigt hat, kennt die gewaltigen Nachteile und Probleme, welche damit verbunden sind, sehr gut. Selten funktioniert eine Installation von Anfang an wie gewollt. Unzureichende Mechanismen für After-Installations-Tasks führen dazu, dass die meisten Administratoren eine eigene Lösung dafür entwickeln oder auf Produkte von Drittherstellern zurückgreifen müssen. An eine Einheit ist und war nicht zu denken. Ich möchte meine 300 Solaris Server mit einem neuen Softwareagenten bestücken, doch wo ist das Hilfsmittel dazu? Wie kann ich meine Software auf den neusten Stand bringen? Was tun, wenn der Disasterfall eintritt? Wie kann ich mein System schnellstmöglich wieder herstellen?

Mit den neuen Deployment Mechanismen von Solaris 11 hat Oracle einen Quantensprung in Richtung Neuzeit gemacht und ihr Betriebssystem mit Möglichkeiten ausgestattet, die von einem modernen Betriebssystem erwartet werden. Dieses Dokument soll die Unterschiede zwischen Solaris 11 und Solaris 10 in Bezug auf das Deployment aufzeigen und die neuen Mechanismen erklären.

Die wesentlichen Unterschiede auf einen Blick

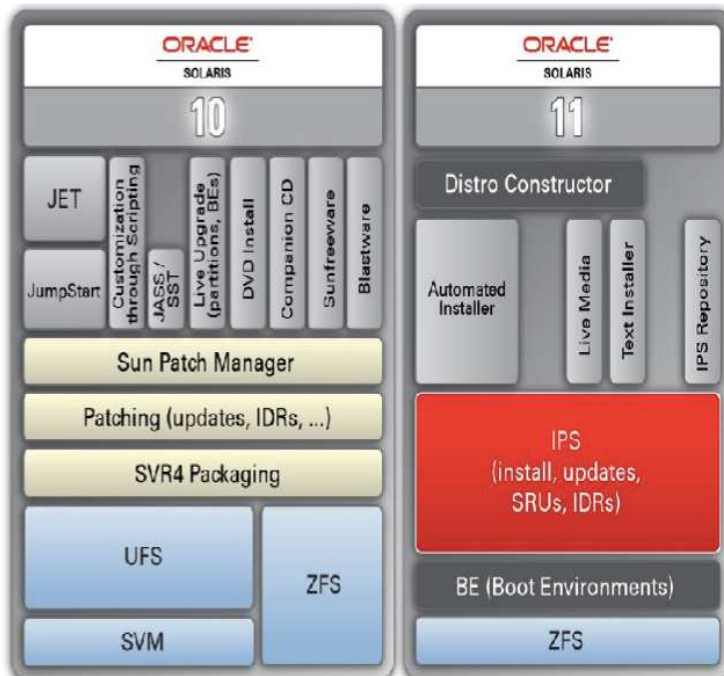


Abb. 1: Vergleich der Deployment Lösungen von Solaris 10 und Solaris 11

Auf den ersten Blick stellen wir fest, dass das Deployment von Solaris 10 ein Sammelsurium diverser Komponenten und Tools ist. Einiges davon wurde nachträglich entwickelt und hinzugefügt, um etwas Erleichterung im Deployment Alltag zu schaffen. Bestes Beispiel hierfür ist vermutlich das Jumpstart Enterprise Tool (JET), welches ursprünglich als internes Projekt von einigen SUN Entwicklern gestartet wurde.

Funktionen, die zusammengehören sollten, «liegen» in getrennten Komponenten. Dies hat zwangsweise zur Folge, dass sich die Administratoren in wesentlich mehr «Produkte» und deren Eigenheiten und Kommandostrukturen einarbeiten müssen, die von ihrer Logik her zumeist auch nicht konsistent aufgebaut sind.

Schon wesentlich angenehmer sieht es unter Solaris 11 aus. Die Deploymentstruktur wurde von Grund auf überarbeitet und konsolidiert. Mit durchgehendem ZFS und den Boot Environments erhalten wir eine gute Basis, die uns bei Notfällen optimal unterstützt. Im Packaging System (IPS) wurden endlich Komponenten hinzugefügt, die ich persönlich unter Solaris 10 stark vermisst habe. Mit Repositories ist nun auch die zentrale Verwaltung von Software möglich. Die Installation und Deinstallation sowie das Aktualisieren von Produkten wurde wesentlich vereinfacht. Wer sich zum Beispiel mit dem Paketmanager yum aus den RedHat Distributionen auskennt, stellt mit Freude fest, dass die Bedienung sehr intuitiv und ohne große Einarbeitung funktioniert.

Auch die Basisinstallation mit dem Automated Installer ist um einiges angenehmer, übersichtlicher und einfacher geworden und arbeitet meiner Meinung nach sensationell mit den IPS Repositories zusammen.

Es wurden zudem Bestrebungen unternommen, Kommandos einheitlicher zu konzipieren. Aufbauend auf der Kommandostruktur, welche wir bereits von der ZFS-Verwaltung her kennen, ist dies Oracle auf den ersten Blick auch gelungen. Leider sind die Befehle zum Teil etwas verwirrend und es besteht Verwechslungsgefahr. Es gibt immer noch einige Funktionen, die bisher nicht zusammengeführt wurden und sehr ähnliche Namen haben.

AI – Der Automated Installer

Nein, AI steht nicht für den englischen Begriff „Artificial Intelligence“, was so viel wie künstliche Intelligenz bedeutet. Soweit ist Oracle mit ihrem automatisierten Installationswerkzeug leider noch nicht. Auch wenn man neidlos zugeben muss, dass damit ein guter Job gemacht wurde und eine gewisse Intelligenz des Produkts nicht abzustreiten ist.

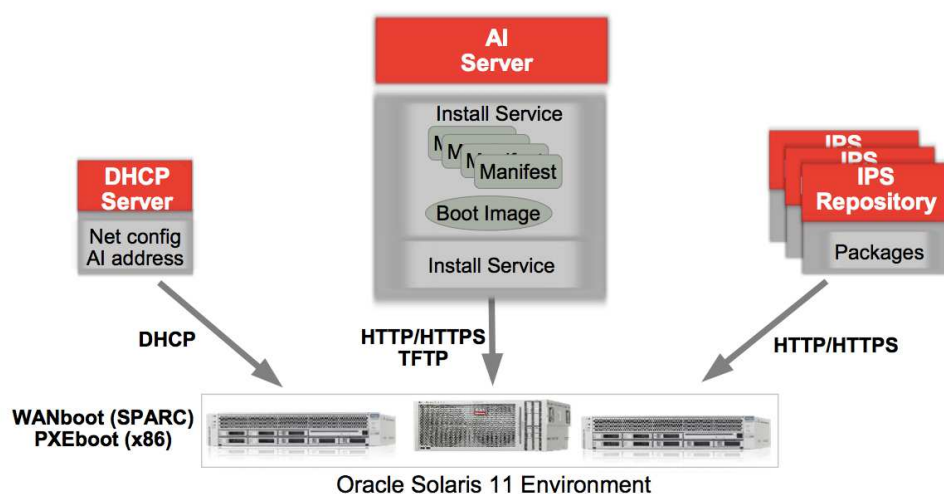


Abb. 2: Funktionsübersicht des Automated Installer (AI)

Was benötigen wir, damit wir Server mittels Automated Installer aufsetzen können? Die AI-Software muss auf einem zentral zugänglichen Server installiert werden. Darin enthalten ist das IPS-Paket `installadm` sowie dessen direkte Abhängigkeiten. Ein DHCP-Server und die Dienste TFTP und HTTP/S sind ebenfalls unerlässlich.

Mittels sogenannten Manifesten können wir unsere Clients und die eigentliche Installation vorbereiten. Manifeste sind vergleichbar mit Kickstart Files aus den RedHat Distributionen. Partitionierung, zusätzliche Repositories und Pakete sind nur einige der Konfigurationsmöglichkeiten innerhalb eines Manifests. Sollen auf dem physischen Server auch gleich eine oder mehrere Zone(n) mitinstalliert werden, ist das Manifest ebenfalls der richtige Ort um dies zu definieren.

Ohne ein oder mehrere Repositories geht natürlich gar nichts. In diesen Repositories befinden sich alle benötigten System-Pakete sowie zusätzliche Produkte, welche wir allenfalls gerne mitinstallieren möchten.

Die benötigten Komponenten können selbstverständlich auf einen oder mehrere Server mit unterschiedlichen Funktionen verteilt werden, sofern diese reibungslos miteinander kommunizieren können.

Als zentrales Tool für die Verwaltung von AI-Tasks dient uns das Kommando `installadm (1M)`, welches von der Struktur her auf der bereits von Solaris 10 bekannten ZFS-Verwaltung basiert. Beinahe alle benötigten Schritte lassen sich damit ausführen.

Es besteht die Möglichkeit Manifeste an einen oder mehrere Server zu knüpfen. Dies steuern die sogenannten «Criteria». Es können Werte wie zum Beispiel Architektur, MAC-Adressen, Memory, Plattformen, Hostnamen oder IP-Adressen definiert werden. Einzelne Werte als auch ganze Ranges sind möglich. Kennt man also seine zukünftige Umgebung und nimmt sich die notwendige Zeit für eine Planung und gute Struktur, spart man für die Umsetzung einiges an Zeit ein. Dies gilt insbesondere für sehr homogene Serverumgebungen.

Mit einem optionalen Sysconfig-Profil lassen sich die nach einer Installation notwendigen Einstellungen wie Benutzer, Passwörter, Zeitzonen, Hostnamen, Netzwerkkonfigurationen, etc. im Voraus definieren. Dies hat den Vorteil, dass man sich nach der Installation den Gang zur Konsole spart. Mit dem Befehl `sysconfig create-profile` wird ein interaktives Menu aufgerufen, welches der Eingabemaske einer manuellen Textinstallation entspricht und die Einstellungen in ein vordefiniertes File speichert.

Für Konfigurationsaufgaben, die sich nicht während der Installation erledigen lassen, eignet sich der sogenannte First-Boot-SMF-Service. Dieser ist natürlich ebenfalls optional und entspricht in etwa der Post-Install-Script-Funktion unter Jumpstart.

Nicht unerwähnt möchte ich das Tool `js2ai (1M)` lassen. Wer seine alten `sysidcfg` Dateien in AI-Manifeste umwandeln möchte, dem bietet Oracle damit das geeignete Werkzeug. Bei komplizierteren Gebilden wird einem nicht immer zu 100% die Nacharbeit erspart, jedoch ist diese nach der Umwandlung in das XML-Format um einiges einfacher, fehlerfreier und komfortabler, als bei der manuellen Übertragung in eine Datei.

In der nachfolgenden Tabelle werden die einzelnen Aufgaben zwischen Jumpstart und Automated Installer verglichen.

Aufgabe	Jumpstart	Automated Installer
Installations-Server bereitstellen.	Es wird das <code>setup_install_server(1M)</code> Kommando benötigt.	Mittels <code>installadm create-service</code> wird ein neuer Installationsdienst bereitgestellt.
Clients einer Installation hinzufügen.	Benutzen Sie das Kommando <code>add_install_client(1M)</code> .	Geht ganz einfach mit <code>installadm create-client</code> .
Installationsbedingungen definieren.	Mit den sogenannten <code>profiles</code> Dateien realisierbar.	Benutzen Sie ein AI Manifest File.
Spezifizieren von Client Regeln.	Wird mit den <code>rules</code> Dateien realisiert, welche einem <code>profile</code> File zugewiesen werden müssen.	Mit dem <code>installadm set-criteria</code> Kommando können Clients mit AI-Manifesten verknüpft werden.
Konfigurationen nach der Installation ausführen.	Es werden das <code>sysidcfg</code> File sowie <code>finish</code> Scripts benötigt.	Mittels den Manifesten und einem <code>First-Boot-SMF-Service</code> umsetzbar.

By the way: Installationen mit dem Automated Installer lassen sich sowohl für SPARC als auch für x86 Systeme durchführen, egal auf welcher Architektur der AI-Server und dessen Komponenten basieren...

IPS – yum unter Solaris

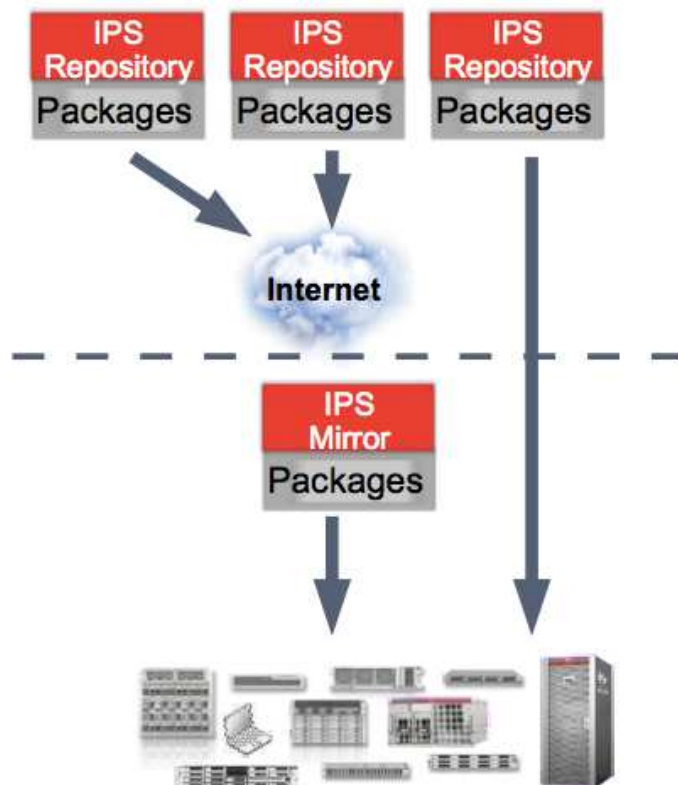


Abb. 3: IPS Überblick

Wer sich mit dem Paketmanager yum oder ähnlichen Tools auskennt, wird sich im Imaging Packaging System von Solaris 11 schnell zu Recht finden.

Mit einem oder mehreren Softwarerepositories, welche entweder direkt vom Internet bezogen oder auf lokalen Servern abgelegt werden können, lassen sich sehr einfach Softwarekomponenten installieren, deinstallieren oder aktualisieren. Da es sich um ein zentrales Paketmanagement handelt, ist die Belieferung von mehreren Servern kein Problem mehr. Die Verwendung von einem oder mehreren Spiegelservern ist ebenfalls möglich.

Auf grafischen Systemen kann direkt mit dem Paketmanager oder dem webbasierten Tool gearbeitet werden. Server ohne grafische Komponenten benutzen das `pkg (1M)` Kommando für ihre Tasks.

Pakete lassen sich versionieren und die benötigten Abhängigkeiten werden automatisch mitinstalliert.

Standardmäßig wird das Public-Repository von Oracle vorkonfiguriert:

<http://pkg.oracle.com/solaris/release/en/index.shtml>

Vertragskunden erhalten für Ihre Updates ein separates Repo:

<https://pkg.oracle.com/solaris/support>

Der Distribution Constructor

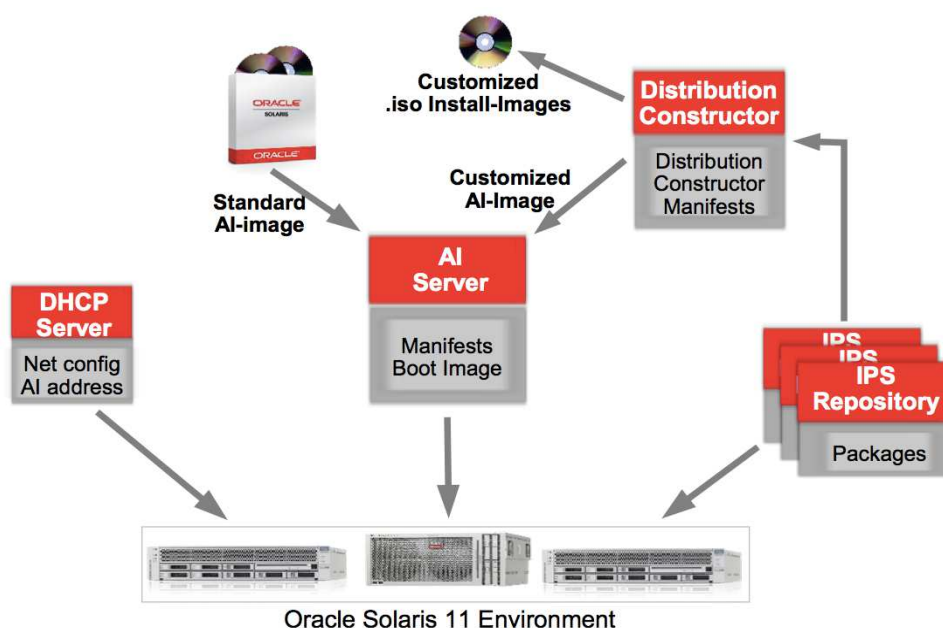


Abb.4: Überblick über den Distribution Constructor

Mit dem Distribution Constructor haben Sie die Möglichkeit angepasste Installations-Images oder virtuelle Maschinen zu erstellen. Die Oracle Solaris Entwickler benutzen dieses Tool übrigens auch, um ihre Produkte zu erstellen.

Mit Hilfe von spezifizierten Parametern in den Distribution Constructor Manifests (XML-Format) erstellt das Tool ISO-Images oder Images für virtuelle Maschinen. Basierend auf x86-ISO-Images kann auch ein bootbares Images für USB-Geräte erstellt werden.

Beim Erstellen von Images besteht die Möglichkeit den Prozess an verschiedenen Stellen zu stoppen und später wieder zu starten. Für Überprüfungen und Debugging ist dies unerlässlich. Diese Funktion nennt sich checkpointing.

Das Boot Environment – Ihr Lebensretter

Stellen Sie sich vor, Sie haben Ihr System aktualisiert und danach läuft nichts mehr wie es sollte. Im schlimmsten Fall startet Ihr System nicht mehr komplett. Was nun?

Mussten Sie früher komplizierte und zeitaufwendige Wiederherstellungsszenarien anwenden, lässt sich dies unter Solaris 11 sehr einfach und zeitsparend mit Boot Environments lösen. Sie erstellen vor den Wartungsarbeiten ein aktuelles BE. Sollten die Wartungsarbeiten einen Rollback erfordern, schalten Sie unkompliziert in das Environment vor den Tasks und die Welt ist wieder in Ordnung.



Abb.5: Boot Environments Beispiel

Weil die Technologie mit ZFS-Funktionen arbeitet, wird bei neuen BEs meist nur wenig Speicherplatz für ein Delta benötigt.

Wer Updates installieren möchte, aber keine grosse Downtime dafür verwenden darf, der wird Boot Environments ebenfalls sehr mögen. Es besteht die Möglichkeit die Updates / Installationen in einem neuen BE auszuführen, welches erst zum Zeitpunkt der Umschaltung aktiv wird. Somit ist Ihr produktives System von der Installation nicht betroffen, solange Sie nicht das neue BE aktiv nehmen.

By the way: Boot Environments sind auch in Zonen möglich...

Fazit – Keep on working...

Oracle hat mit dem Deployment unter Solaris 11 sehr viel richtig gemacht. Es scheint als wäre der Neuaufbau ein guter Schritt, der sich zwangsweise durch die vielen Abgänge ergab. Vieles wirkt nun modern, zeitgemäß und durchdacht.

Wenn Oracle sich nun die Zeit nimmt, um dem etwas langsamen Paketmanager einen Geschwindigkeitsschub zu verpassen oder die Kommandostruktur noch mehr zu vereinfachen beziehungsweise zu konsolidieren kann man schlussendlich mit gutem Gewissen sagen, dass Solaris, was das Deployment angeht, endlich in der Neuzeit angekommen ist.

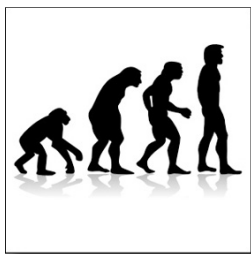


Abb. 6: Von der Steinzeit in die Neuzeit

Kontaktadresse:

Ralf Germann
Consultant
Trivadis AG
Europa-Straße 5
CH-8152 Glattbrugg (Zürich)

Telefon: +41 (0) 76-337 1983
Fax: +41 (0) 44-808 7021
E-Mail: ralf.germann@trivadis.com
Internet: www.trivadis.com