

Wer sich jemals mit dem inzwischen in die Jahre gekommenen Jumpstart Server von Solaris beschäftigt hat, kennt die damit verbundenen gewaltigen Nachteile und Probleme sehr gut.

Solaris 11 Deployment – willkommen in der Neuzeit

Ralf Germann, Trivadis AG

Selten funktioniert eine Installation von Anfang an wie gewollt. Unzureichende Mechanismen für Post-Installation-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. 300 Solaris-Server sollen mit einem neuen Software-Agenten bestückt werden, doch wo ist das Hilfsmittel dazu? Wie können wir unsere Software auf den neuesten Stand bringen?

Mit den neuen Deployment-Mechanismen von Solaris 11 hat Oracle einen Quantensprung in Richtung Neuzeit gemacht und das Betriebssystem mit Möglichkeiten ausgestattet, die von einem modernen Betriebssystem erwartet werden. Dieser Artikel zeigt die Unterschiede zwischen Solaris 10 und 11 hinsichtlich des Deployment und erklärt die neuen Mechanismen. Bei Jumpstart wird jeweils Bezug auf den puren Mechanismus ohne allfällige Hilfsmittel wie JET oder dergleichen genommen.

Die wesentlichen Unterschiede

Auf den ersten Blick lässt sich feststellen, 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 dafür ist das Jumpstart Enterprise Tool (JET), das ursprünglich als internes Projekt einiger Sun-Entwickler gestartet wurde, die mit den Funktionen ihres hauseigenen Deployment nicht zufrieden waren.

Funktionen, die zusammengehören sollten, befinden sich in getrennten

Komponenten. Dies hat zwangsweise zur Folge, dass sich die Administratoren in wesentlich mehr „Produkte“ und deren Eigenheiten sowie Kommando-Strukturen einarbeiten müssen. Zudem sind diese von ihrer Logik her meistens auch nicht konsistent aufgebaut.

Schon wesentlich angenehmer sieht es unter Solaris 11 aus. Die Deployment-Struktur wurde von Grund auf überarbeitet und konsolidiert. Mit durchgehendem ZFS und den Boot-Environments ist eine gute Basis entstanden, die auch bei Notfällen optimal unterstützt. Dies war unter Solaris 10 mit Live-Upgrade beziehungsweise den „lu“-Kommandos bereits möglich, allerdings nicht so komfortabel gelöst wie mit dem einheitlichen „bootadm(1M)“-Befehl unter Solaris 11.

Das Packaging System (IPS) erhielt endlich Komponenten, die viele Administratoren unter Solaris 10 stark vermisst haben (siehe Abbildung 1). Mit Repositories ist nun auch die zentrale Verwaltung von Software möglich. Die Installation und Deinstallation so-

wie das Aktualisieren von Produkten ist wesentlich vereinfacht. Wer sich mit gängigen Unix- und Linux-Paketmanagern auskennt, stellt mit Freude fest, dass die Bedienung sehr intuitiv und ohne große Einarbeitung funktioniert.

Auch die Basis-Installation mit dem Automated Installer ist um einiges angenehmer, übersichtlicher und einfacher geworden und arbeitet sehr gut mit den IPS-Repositories zusammen. Es wurden zudem Bestrebungen unternommen, Kommandos einheitlicher zu konzipieren. Aufbauend auf der Kommando-Struktur, die 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. Einige Funktionen, die bisher nicht zusammengeführt wurden, haben sehr ähnliche Namen.

Der Automated Installer

AI steht nicht für den englischen Begriff „Artificial Intelligence“, was so viel wie künstliche Intelligenz bedeutet. Soweit ist Oracle mit ihrem automatisierten Installationswerkzeug Automated Installer 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 (siehe Abbildung 2).

Die AI-Software muss auf einem zentral zugänglichen Server installiert werden. Darin enthalten sind das IPS-Paket „installadm“ sowie dessen direkte Abhängigkeiten. Ein DHCP-Server und die Dienste TFTP und HTTP/S sind ebenfalls unerlässlich. Mittels sogenannter „Manifeste“ kann man seine Clients und die eigentliche Installation vorbereiten.

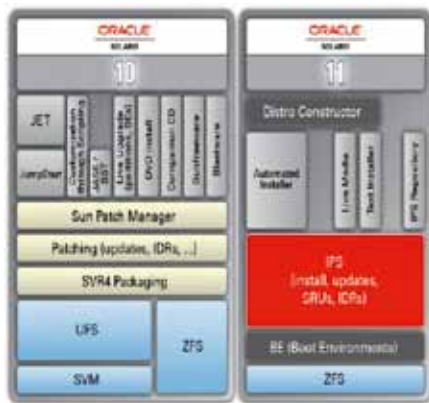


Abbildung 1: Vergleich der Deployment-Lösungen von Solaris 10 und Solaris 11 (Quelle: Oracle)

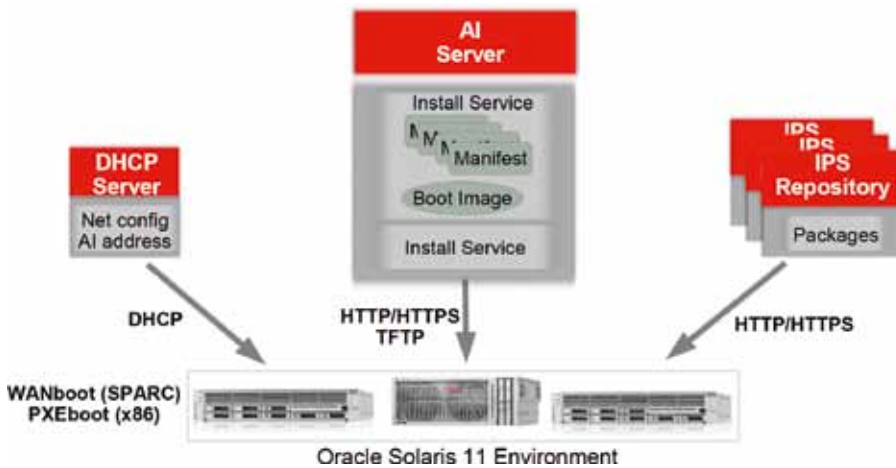


Abbildung 2: Funktionsübersicht des Automated Installer (Quelle: Oracle)

Manifeste sind vergleichbar mit Kickstart-Files aus den Linux-Distributionen. Partitionierungen, 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. Mit den mitgelieferten Bordmitteln von Jumpstart unter Solaris 10 war eine automatische Zonen-Installation nach der Client-Installation beispielsweise nicht möglich.

Ohne ein oder mehrere Repositories geht natürlich gar nichts. In diesen Repositories befinden sich alle benötigten System-Pakete sowie zusätzliche Produkte (von Drittherstellern oder eigene Pakete), die man gern mitinstallieren möchte. Die benötigten Komponenten können selbstverständlich auf einen oder mehrere Server mit unterschiedlichen Funktionen verteilt sein. Voraussetzung ist die reibungslose Kommunikation dieser Server untereinander.

Als zentrales Tool für die Verwaltung von AI-Tasks dient uns das Kommando „installadm(1M)“, das von der Struktur her auf der bereits von Solaris 10 bekannten ZFS-Verwaltung basiert. Beinahe alle benötigten Schritte lassen sich damit ausführen, was die Handhabung sehr vereinfacht. Dazu ein Beispiel für die Erstellung eines Clients (siehe Listing 1), ein Beispiel für die Erstellung und Zuweisung eines Manifests (siehe Listing 2) sowie die Anzeige der Zusammenfassung (siehe Listing 3).

Es besteht die Möglichkeit, Manifeste an einen oder mehrere Server zu knüpfen (siehe Beispiel). Die Steuerung übernehmen die sogenannten „Criteria“. Es können Werte wie „Architektur“, „MAC-Adressen“, „Memory“, „Plattformen“, „Hostnamen“ oder „IP-Adressen“ definiert werden. Es sind sowohl einzelne Werte als auch komplette Ranges 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

```
# installadm create-client -e
00:0C:29:31:AD:9F -n sol1111x86
```

Listing 1

```
# installadm export -n sol1111x86 -m orig_default -o /export/
manifest/TVD-SOL-11-CLIENT.xml
# installadm create-manifest -n sol1111x86 -f /export/manifest/
TVD-SOL-11-CLIENT.xml -m TVD-SOL-11-CLIENT -c mac="00:0C:29:31:AD:9F"
```

Listing 2

```
# installadm list -n sol1111x86 -m -c -p

Service Name Client Address Arch Image Path
-----
sol1111x86 00:0C:29:31:AD:9F i386 /install/sol1111x86

Service/Manifest Name Status Criteria
-----
sol1111x86
TVD-SOL-11-CLIENT mac = 00:0C:29:31:AD:9F
orig_default Default None

Service/Profile Name Criteria
-----
sol1111x86
TVD-SOL-11-CLIENT mac = 00:0C:29:31:AD:9F
```

Listing 3

ein. Dies gilt insbesondere für sehr homogene Server-Umgebungen.

Mit einem optionalen „Sysconfig“-Profil lassen sich die nach einer Installation notwendigen Einstellungen wie „Benutzer“, „Passwörter“, „Zeitzone“, „Hostnamen“, „Netzwerk-Konfigurationen“ etc. im Voraus definieren (siehe Abbildung 3). Dies hat den Vorteil, dass man sich nach der Installation den Gang zur Konsole spart. Der Befehl „sysconfig create-profile“ ruft ein interaktives Menü auf, das der Eingabemaske entspricht und die Einstellungen in ein vordefiniertes File speichert. Listing 4 zeigt ein Beispiel für die Erstellung eines Sysconfig-Profiles.

Für Konfigurationsaufgaben, die sich nicht während der Installation erledigen lassen, eignet sich der sogenannte „First Boot SMF Service“. Dieser ist optional und entspricht ungefähr der „Post Install Script“-Funktion unter Jumpstart. Nicht unerwähnt sei das Tool „js2ai(1M)“. Wer seine alten „sysidcfg“-Dateien in AI-Manifeste umwandeln möchte, hat damit das geeignete Werkzeug. Bei komplizierteren Gebilden wird nicht immer zu 100 Prozent 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

```
# sysconfig create-profile -o /export/profiles/TVD-SOL-11-CLIENT.xml
```

Listing 4: Beispiel für die Erstellung eines Sysconfig-Profiles

Datei. In Tabelle 1 sind die einzelnen Aufgaben zwischen Jumpstart und Automated Installer direkt miteinander verglichen. Nebenbei bemerkt: 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 – Paketmanagement unter Solaris 11

Wer sich zum Beispiel mit dem Paketmanager „yum“ aus den Red-Hat-Derivaten oder anderen ähnlichen Verwaltungs-Werkzeugen auskennt, wird sich mit dem Imaging Packaging System (IPS) von Solaris 11 schnell zurecht finden. Mit einem oder mehreren Software-Repositories, die entweder direkt übers Internet bezogen oder auf Servern in der eigenen Umgebung abgelegt werden können, lassen sich sehr einfach Software-Komponenten installieren, deinstallieren oder aktualisieren. Da es sich um ein zentrales Paket-Management handelt, ist die Belieferung mehrerer Server kein Problem mehr. Die Verwendung von einem oder mehreren Spiegel-Servern ist ebenfalls möglich. Auf grafischen Systemen kann direkt mit dem Paketmanager (siehe Abbildung 4) 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 mit installiert.

Standardmäßig wird das Public-Repository von Oracle vorkonfiguriert (siehe <http://pkg.oracle.com/solaris/release/en/index.shtml>), Vertragskunden erhalten für ihre Updates ein separates Repository (siehe <https://pkg.oracle.com/solaris/support>).

Der Distribution Constructor

Mit dem Distribution Constructor (siehe

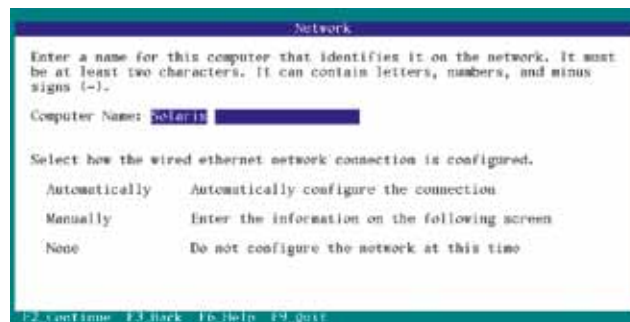


Abbildung 3: Einstiegsbild des System Configuration Tool

he Abbildung 5) lassen sich angepasste Installations-Images oder virtuelle Maschinen erstellen. Die Oracle-Solaris-Entwickler benutzen dieses Tool übrigens auch, um ihre Produkte zu erstellen.

Mithilfe von spezifizierten Parametern in den Distribution-Constructor-Manifesten (XML-Format) erstellt das Tool ISO-Images oder Images für virtuelle Maschinen. Basierend auf x86-ISO-Images kann auch ein bootbares Image für USB-Geräte erzeugt 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

Obwohl die Technologie nur entfernt etwas mit dem Deployment zu tun hat, wird das Thema trotzdem kurz gestreift. Unter Solaris 10 waren Boot Environments ebenfalls möglich. Ohne einen „ZFS-rpool2“ musste man allerdings einige Einschränkungen in Kauf nehmen, die unter Solaris 11 mit dem durchgängigen ZFS nicht mehr vorhanden sind (siehe Abbildung 6).

Mit dem zentralisierten „bootadm(1M)“-Befehl lassen sich Boot Environments (BE) schnell und unkompliziert erstellen. Weil die Technologie mit ZFS-Funktionen arbeitet, wird bei neuen BEs meist nur wenig zusätzlicher Speicherplatz für ein Delta belegt.

Wer Updates installieren möchte, aber keine große Downtime dafür zur Verfügung hat, der wird Boot Environments ebenfalls sehr mögen. Es besteht die Möglichkeit, Updates/Installationen in einem neuen BE auszuführen, das erst zum Zeitpunkt der Umschaltung aktiv wird. Somit ist das produktive System von der Installation nicht betroffen, solange man nicht das neue BE aktiv nimmt. Nebenbei bemerkt: Boot Environments sind auch in Zonen möglich.

Neuerungen unter Solaris 11.1

Es wird nicht auf alle Neuerungen eingegangen, die das Update 11.1 enthält, sondern nur auf die wesentlichen Änderungen im Deployment. Eine der spürbarsten Verbesserungen ist ganz

Aufgabe	Jumpstart	Automated Installer
Installations-Server bereitstellen	Es wird das „setup_install_server(1M)“-Kommando benötigt	Mittels „installadm create-service“ wird ein neuer Installationsdienst bereitgestellt
Clients einer Installation hinzufügen	Durch das Kommando „add_install_client(1M)“	Geht ganz einfach mit „installadm create-client“
Installationsbedingungen definieren	Mit den sogenannten „Profile“-Dateien realisierbar	Über ein AI-Manifest-File
Spezifizieren von Client-Regeln	Wird mit den „Rules“-Dateien realisiert, die einer „Profile-Datei“ zugewiesen werden müssen	Mit dem Kommando „installadm set-criteria“ lassen sich Clients mit AI-Manifesten verknüpfen
Konfigurationen nach der Installation ausführen	Es werden das „sysidcfg“-File sowie „finish“-Skripte benötigt	Mittels der Manifeste und eines First-Boot-SMF-Service umsetzbar

Tabelle 1

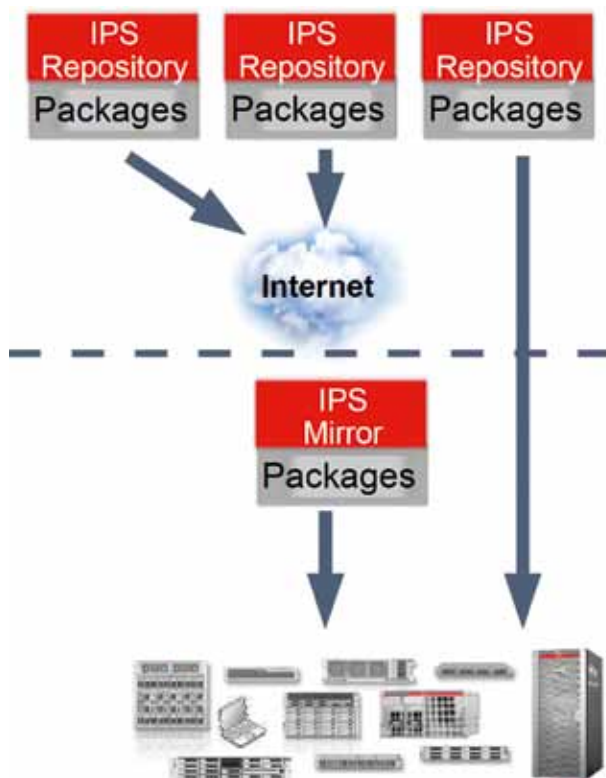


Abbildung 4: Überblick Paket-Management (Quelle: Oracle)

klar der Paket-Manager (IPS). Oracle hat diesem einen gewaltigen Performance-Schub verpasst. Unter 11.0 dauerten zum Beispiel Installationen und Updates unnachvollziehbar lange, was viele Administratoren als störend empfunden haben. Mit 11.1 ist die Performance deutlich akzeptabler.

Die neuen Sub-Kommandos „update-service“, „update-profile“ und „set-service“ des „installadm(1M)“-Befehls vereinfachen die Anpassung von

Installations-Services und bieten Administratoren mehr Flexibilität. Neu ist auch die direkte Installation mit der „interactive text“-Methode und dem „live media installer“ auf iSCSI-LUNs. Die Fortschrittsausgabe während des Deployment ist viel ausführlicher und zuverlässiger geworden.

Wer nicht gern mit XML-Files arbeitet, dem wird die nächste Neuerung sicherlich gefallen. Mit dem Kommando „svcbundle(1M)“ lassen sich

neue SMF-Manifeste und -Profile ohne XML-Kenntnisse erstellen. Auch beim Update von Zonen hat sich in Sachen „Performance“ gewaltig etwas geändert. Dies ist vor allem der Möglichkeit zu verdanken, mehrere Zonen auf einem Server gleichzeitig aktualisieren zu können. Gemäß Oracle ist das Updaten von 20 Zonen somit vier Mal schneller als bisher. Zudem wurde die Zeit für eine Zonen-Installation um 27 Prozent verbessert.

Wer sich über weitere Neuerungen und Verbesserungen unter Solaris 11.1 informieren möchte, dem kann das offizielle Dokument von Oracle unter <http://www.oracle.com/technetwork/server-storage/solaris11/documentation/solaris11-1-whatnew-1732377.pdf> weiterhelfen.

Fazit

Oracle hat mit dem Deployment unter Solaris 11 sehr viel richtig gemacht. Es scheint, als wäre der Neuaufbau ein guter Schritt, der sich zwangweise durch die vielen Abgänge ergab. Vieles wirkt nun modern, zeitgemäß und durchdacht. Wenn Oracle sich nun die Zeit nimmt und weitere Optimierungen am Paket-Manager vornimmt oder die Kommando-Struktur noch mehr zu vereinfachen beziehungsweise zu konsolidieren versucht, kann man schlussendlich mit gutem Gewissen sagen, dass Solaris – vor allem was das Deployment angeht – endlich in der Neuzeit angekommen ist.

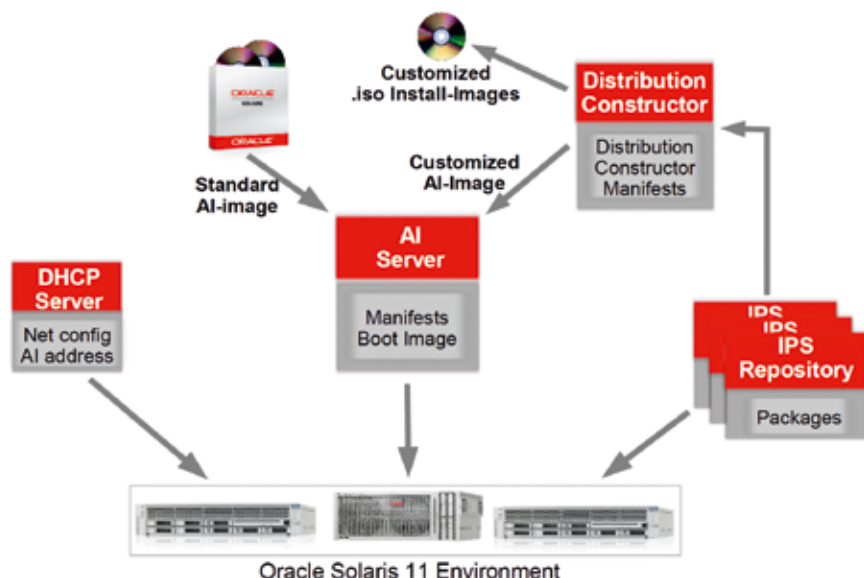


Abbildung 5: Überblick über den Distribution Constructor (Quelle: Oracle)



Abbildung 6: Boot-Environments-Beispiel (Quelle: Oracle)

Ralf Germann
ralf.germann@trivadis.com

