

Image Packaging System

Dipl.-Chem. Rainer Orth
Technische Fakultät
Universität Bielefeld
ro@TechFak.Uni-Bielefeld.DE

Übersicht

- Probleme mit System V-Packaging
- Indiana als Spielwiese
- Image Packaging System
- Snap Upgrade
- Automated Installation
- Slim Install, Distribution Constructor

Probleme mit System V-Packaging

- kein/eingeschränkter Support für Netzwerk-Repositories
- keine automatische Installation von Dependencies
- Dependencies unvollständig
- keine Gruppierungen (nur im Installer)
- Upgrade- und andere Metadaten nur im Installer
- flacher Namensraum (Packagenamen lange auf 9 Zeichen beschränkt, jetzt 32)
- häufig interaktiv per Default

Probleme mit System V-Packaging (ii)

- einige Features dazugehackt: signierte Packages, Zonen-Support
- schwierig für Entwickler: Unterstützung verschiedener Umgebungen wie Zonen, diskless Clients
- keine Upgrades außer durch Patches
- Patches (Sammlung von sparse packages) aktualisieren oft mehrere Packages, aber Probleme bei Nachinstallation
- schlechte Versionsübersicht
- beliebige Kombinationen von Patches erzeugen unüberschaubare viele ungetestete Konfigurationen ('Dim Sum patching')
- keine Nutzung von ZFS

Indiana als Spielwiese

- neue OpenSolaris-Distribution: nur Open Source-Komponenten
- Experimentierfeld für neue Installationstechniken und mehr
- auf Entwickler ausgerichtet
- aber: teilweise umstritten
 - Namenswahl an der Community vorbei
 - kontroverse Änderungen: „Linuxisierung“
 - einzige OpenSolaris-Distribution mit Support, trotz Alpha-Status
 - eingeschränkte Funktionalität durch OpenSource-Fixierung: kein Motif, CDE, ...

Indiana-Realisation

- mehrere Previews seit Oktober 2007, erstes Release 2008.05, gerade 2008.11, Updates in Halbjahresabständen geplant
- bisher nur x86, SPARC-Support geplant für 2009.04 (initial `sun4v`)
- LiveCD zum Ausprobieren, Installation
- (bisher?) kein Support für Upgrades von Solaris 10/Solaris Express

Image Packaging System

- nach Evaluation diverser existierender Systeme: Neuentwicklung unter Berücksichtigung von OpenSolaris-Features und -Anforderungen
- pkg (5) : initial HTTP/HTTPS-basiertes Packaging-System
- z.Zt. Fileformat noch nicht definiert
- weitgehend in Python
- aber: warum nicht Weiterentwicklung von SVr4-Packaging unter Beibehaltung des Formats?

Image Packaging System: Die Kommandos

- Clients: `pkg(1)` für Installation/Deinstallation/Upgrade, `packagemanager(1)` als GUI-Client, `updatemanager(1)` für einfache Updates per GUI
- Server: `pkg.depotd(1M)`
- Publikations-/Retrieval-Clients: `pkgsend(1)`, `pkgrecv(1)`

Image Packaging System: Termini

- Package: versioniertes Software-Paket
- FMRI: `pkg://[authority]/[pkg_name]@[version][,build]-[branch]:[timestamp]`
- Image: Full, Partial, User
- Authority: Packager von Binär-Software
- Repository: Netzwerk-Quelle für Software, potentiell von mehreren Authorities, Default: `http://pkg.opensolaris.org/`

Image Packaging System: Features

- potentiell plattform-übergreifend (cf. Update Center 2, z.B. für GlassFish v3, OpenDS, ...)
- Komplexität ins Packaging-System verschoben: Root-/User-Split, Zones-Support
- bei Updates werden nur geänderte Files installiert
- nutzt ZFS-Snapshots bei Upgrades
- Mirroring z.Zt. nur von Daten (per `rsync`), Metadaten nur vom Master

Image Packaging System: Features to come

- Dependencies werden bei der Publikation gecheckt; Package wird zurückgewiesen, wenn unvollständig
- kryptographische Signaturen
- z.Zt. werden Packages aus Solaris Express importiert, aber in Vorbereitung:
 - hierarchische Umbenennung zur besseren Kategorisierung
 - Aufhebung des Root-/User-Splits

Image Packaging System: Non-Features

- kein Skripting:
 - zu riskant
 - schlecht zu testen in diversen Umgebungen
 - schwierig rückgängig zu machen
 - inkonsistente Codeduplikation für wiederkehrende Aufgaben
- stattdessen: fertige Aktionen für übliche Aufgaben, z.B. `file`, `dir`, `link`, `hardlink`, `driver`, `depend`, `license`, `legacy`, `set`, `group`, `user`
- keine Integration mit Build-System: zu divers

Image Packaging System: Benutzung

- `pkg(1)`-Subkommandos:
 - `install, uninstall`
 - `verify, fix`
 - `info, contents, list, search`
 - `refresh`
 - `image-create, image-update`
 - `set-property, unset-property, property`
 - `set-authority, unset-authority, authority`
 - `history, purge-history`

Snap Upgrade

- Nachfolger für Live Upgrade
- letzteres ursprünglich `ksh`-Skripten, inzwischen auch Teile in C, aber AT&T-Copyright, kein Redistributionsrecht für Sun
- Upgrade zwischen verschiedenen Software-Ständen/-Versionen „am lebenden Objekt“
- Boot Environment: bootfähige Installation, die separat geupgraded werden kann

Snap Upgrade: Nutzung

- Management mit `beadm (1m)`, Nachfolger der diversen `lu*`-Kommandos
- Subkommandos wie gewohnt: `create`, `destroy`, `list`, `mount`, `unmount`, `rename`, `activate`
- initial nur mit ZFS
- Unterstützung von Zone-Upgrades
- Upgrade von Solaris 10/Solaris Express wohl für später geplant
- separate Library `libbe` zur Nutzung z.B. durch `pkg (1)`

Automated Installation

- Nachfolger für JumpStart
- einfacher zu benutzen: `installadm(1M)` -Kommando zur Einrichtung und Konfiguration des Servers
- diverse XML-Profile:
 - AI-Manifest beschreibt Zieldisk, Partitionierung, Authorities und zu installierende Software
 - SC-Manifest beschreibt Systemkonfiguration: Root-Password, Zeitzone, Default-User
 - Criteria-Manifest verbindet die beiden; Client-IP- und MAC-Adressen, Speichergröße, Architektur als Kriterien

Automated Installation: Ablauf

- Client-Profilen enthalten nur Service-Namen, Clients lokalisieren Server per mDNS (optional: DNS), Download zusätzlicher Files per HTTP (NFS für später geplant)
- x86: PXE-Boot, dann HTTP-Zugriff auf Installer-CD-Image (Subset der LiveCD)
- SPARC: initial WANboot geplant, d.h. nur für neuere Systeme mit entsprechendem OBP-Support
- geplant: Konversion von JumpStart-Profilen

Der Rest vom Caiman: Slim Install

- Slim Install
 - massiv vereinfachter graphischer Installer
 - nur die Fragen, die zur Installation nötig sind
 - Konfiguration soweit irgend möglich hinterher auf dem laufenden System
 - bisher keine text-basierte interaktive Installation
- z.Zt. kein Ersatz für Flash-Archive, Diskless Clients

Noch mehr Caiman: Distribution Constructor

- Erzeugung von LiveCD-Images, Boot-Images für USB-Sticks etc.
- war bisher selbst innerhalb von Sun eine schwarze Kunst
- jetzt massiv vereinfacht mit `distro_const` (1M)
- nutzt XML-Profil zur Beschreibung der Installation

Weitere Informationen

- **OpenSolaris-Installation and Packaging-Community:** <http://www.opensolaris.org/os/community/install/>
- **OpenSolaris-SVR4 Packaging-Projekt:** http://www.opensolaris.org/os/project/svr4_packaging/
- **OpenSolaris-Image Packaing System-Projekt:**
<http://www.opensolaris.org/os/project/pkg/>
- **OpenSolaris-Caiman: Solaris Install Revisited-Projekt:**
<http://www.opensolaris.org/os/project/caiman/>
- **OpenSolaris-Indiana-Projekt:**
<http://www.opensolaris.org/os/project/indiana/>