PSARC/2008/190 pkg(5): Image packaging system

David Comay, Danek Duvall, Stephen Hahn, Brad Hall, K Johansen, Dan Price, Brock Pytlik, Bart Smaalders, BJ Wahl, and Shawn Walker

Introduction and overview

- See the overview slides for an introduction
- Following is an outline of the topics to be treated as part of a packaging system architectural review
  - Need to provide a successor to Application Packagers Guide
  - Do not want to follow “huge design doc” model of PSARC/2002/547

Relationship with libbe, ZFS

- Image packaging uses SnapUpgrade to take consistent cross-filesystem snapshots
  - PSARC/2007/039: Caiman Umbrella
- No rollback for non-ZFS presently designed

Multiplatform aspects

- Working with Update Center team to develop a cross-platform packaging system
  - This case is primarily OpenSolaris-focussed, but architectural feedback will affect all multiplatform development
  - Sometimes multiplatform restrictions will constrain choices
  - LSARC/2008/411: Update Center 2 Toolkit

Client use

- Example usage, as given in the overview slides
- Two client consumers in OpenSolaris
  - pkg(1) CLI
  - packagemanager(1) GUI
- Expecting to define MIME type/control document format for BUI-based install

Developer and operator use

- pkgsend(1) and pkgrecv(1) give developer access to
  - Publication of new package versions
  - Retrieval of raw package contents (without installation)
- Debate: new operational command, depotadm(1M), versus enhancements to pkgsend(1)
  - Delete package version, mark historical, toxic; depot garbage collection, etc.

Package

- A **package** is composed of a series of **versions**
- Each version is in turn composed by a set of **actions**
  - A file is an action
  - Version-version transitions are install/update/removal of the sets of actions of the two versions
- A package version is installed into an **image**

Package states

- One **package version** can be installed into an **image**
- Each package version has a known, exclusive state within an image:
  - Known
  - Installed
  - Faulty
- Transitional states (Known-Installing, Installed-Uninstalling) needed

Image types

- **Full:** a typical OS install
- **Partial:** linked to a full image
  - Expected to handle sparse zone, diskless equivalents
  - Under discussion by Zones team
- **User:** dependent on full image
  - Per-user installs
  - May or may not be updated with full image

Authorities

- An **authority** is a publisher of packages
  - Each **repository** contains package content from one authority
    - Debate: one and only one, or at least one?
- Packaging client can install packages from multiple authorities
  - Package dependencies can cross authorities

Security model

- All files in the repository are stored based on cryptographic hash
  - Currently SHA-1; will support multiple hash algorithms
- Each package version has a manifest that refers to these hashes
  - Manifests will be signed
- Catalog points to versions
  - Catalog to be signed
- SSL a transport option

Use of SSL

- Packaging client can associate one key/certificate with each authority
  - Depot server does SSL client-side authentication to determine general access
  - Per-package entitlement API a possibility
- Question regarding per-mirror key/certificate association
  - Seems complex

pkgsend / pkgrecv

- Update of a package is delivery of a new package version
  - A transaction with the repository
- pkgrecv allows retrieval in transaction form

Transaction states

- Publication also involves a set of states
  - Open, closed-hold, closed-accept
- Server evaluates transaction for completeness
  - Hold versus accept is based on completeness
  - Dependency analysis, multiarchitecture package transaction

Publication authentication

- Need authentication/authorization model for package publication
  - No design at present
- One option is to map HTTP authentication to package update operations
  - Forces transport, which may not be appropriate for all deployments
  - Can reuse Unix credentials

pkg.depotd(1M)

- Two purposes
- Retrieval
  - Support /filelist/0 operation
  - Support partial /catalog/0 update op
  - Mirror and caching modes
- Publication
  - Support all publication APIs
  - Deactivate with –read-only mode

Protocol / network format

- Primary interaction is network-based
  - Operations on single/multi-package are a subset of network ops
- HTTP
  - API is split into two: retrieval (for client installs) and publication (for package uploads)
  - Server will also provide BUI access
    - Search, updates-as-feed, package information

Retrieval API

- /catalog/0
  - Retrieve updates to catalog
- /manifest/0
  - Retrieve specific package manifest
- /file/0, /filelist/0
  - Retrieve single file, multiple files
- /search/0
  - Search repository for words/keywords

Retrieval API, 2

- /info/0
  - Retrieve formatted package information
- /feed/0
  - Retrieve Atom feed for depot updates

Publication API

- /open/0
  - Begin transaction
- /add/0
  - Add a package action to current txn
- /close/0
  - Close transaction
- More operations to come
  - Rename, historical, end-of-life
  - Removal of packages, package versions

On-disk format

- An on-disk format for packages is needed
  - Not designed at this time
- Debate: simple archive, transaction format, repository format
  - Tools alignment versus standard tools
- Goal is simple package manipulation

Repository format

- Repository is
  - Catalog, manifest, files (content); search indices
- Stable format means that pkgsend(1) can optionally publish direct to repository format

Image format

- Metadata
  - /var/pkg for full, partial images
  - .org.opensolaris,pkg for user images
- Contents
  - catalogs for subscribed authorities
  - manifests for installed/queried package versions
  - search indices for installed package versions

Actions

- A **package** is composed of **actions**
- Implemented:
  - set, depend; file, dir, user, group, driver, license, legacy
- Needed:
  - service, ...
- An action is always needed for an OS resource that must be configured for boot
- Actions can restart services

Configuration transition

- Avoid configuration file merging
- Choices:
  - Assistive application (add_drv(1M), update_drv(1M), etc.)
  - Post-boot assembly (GNOME desktop-cache/ services)
    - Closest to past practice
  - Use of extended smf(5) profiles
  - Combinations of these
    - E.g. delivery of separate files to a directory, merged by a tool/service

Migration and compatibility

- pkgsend(1M) can convert SysV packages
  - But no support for scripting conversions
- SysV and pkg(5) can coexist
  - But query operations are mostly disjoint
  - legacy action makes pkg(5) installs appear to satisfy SysV package requirements
- Debate: How far to go?

opensolaris™

PSARC/2008/190 pkg(5): Image packaging system

pkg-discuss@opensolaris.org

http://opensolaris.org/os/project/pkg