

Performant Package Management for Solaris

Overview & Guidelines

What's a Package?

– slide 1

- ▷ a coherent (atomic) set of software
e.g. GNU Bison, version 1.33
- ▷ the form in which this set is distributed
e.g. `bison-1.33-src.noarch.rpm`

Package Relations

– slide 2

- ▷ complex boolean relations:
 - dependencies
 - mutual exclusion
- ▷ provides
- ▷ recommends
- ▷ virtual packages
- ▷ ...

Package Distribution

– slide 3

three options for package maintainers:

- ▷ as a source tarball
(plus platform- and OS-specific patches)
- ▷ as a “source” or “binary” package object
- ▷ opaque (only via a package manager)

Actions on Packages (1)

– slide 4

the abstract (high-level) package operations are :

- ▷ *install* a package
- ▷ *remove* a package

the end-users / admins need it that simple!

Actions on Packages (2)

– slide 5

the actual (low-level) ops :

1. find and get it (download ; copy ; unpack)
2. recursively install dependencies
3. build it (option) (patch ; compile ; move file tree)
4. configure it
5. remove it

Package Management Systems (1)

– slide 6

they perform the low-level ops, *transparently*

sample one-liners:

▷ `pkgadd package`

(Solaris / SVr4)

▷ `apt-get install package`

(Debian GNU/Linux)

▷ `fink install package`

(GNU/Darwin)

Package Management Systems (2)

– slide 7

need to perform several choices to design a PMS:

- ▷ packaging format deb, rpm, pkg, ... ?
- ▷ local file hierarchy sandbox or mixed ?
- ▷ dependency graph centralized or not ?
- ▷ patch maintenance ...
- ▷ package distribution ...
- ▷ bug-tracking ...

Discussion: Package Formats (1)

– slide 8

several quality criteria:

- ▷ rich metadata (signatures, checksums, dependencies, ...)
- ▷ embedded package programs (auto configuration)
- ▷ scalability (no hard-coded limits, format extension)
- ▷ usability by standard tools (GNU fileutils, GNU tar, etc.)
- ▷ ... (see solarpack.sf.net)

Discussion: Package Formats (2)

– slide 9

	deb	rpm	tgz	slp	pkg
security, authentication	+	+	+	-	-
usability by standard linux tools	+		+	-	
rich metadata	+	-	+	-	-
embedded programs	+	+	+	-	-
scalability	+	+	+	-	

conclusion:

deb (Debian) and tgz (NetBSD) formats meet the criteria
but the choice depends on the PMS

Discussion: Local Hierarchy (1)

– slide 10

two options:

- ▷ mixed with system (in `/usr` and/or `/usr/local`)
- ▷ in a “sandbox” (e.g. in `/opt`)

Discussion: Local Hierarchy (2)

– slide 11

	sandbox	mixed
pros	<ul style="list-style-type: none">▷ secure▷ optional (add-on)▷ shareable ?	<ul style="list-style-type: none">▷ zero user setup▷ no redundancy
cons	<ul style="list-style-type: none">▷ need to setup▷ filesystem complexity	<ul style="list-style-type: none">▷ risk of conflicts▷ unclear fs structure

Discussion: Centralization

– slide 12

	centralized	not centralized
pros	<ul style="list-style-type: none">▷ non-redundant▷ one-line maintenance	<ul style="list-style-type: none">▷ easy for upstream developers▷ balances load
cons	<ul style="list-style-type: none">▷ requires a very active maintenance	<ul style="list-style-type: none">▷ risk of conflicts▷ redundant efforts▷ slow maintenance

Case Study: Redhat

– slide 13

- ▷ package format: RedHat rpm
- ▷ package manager: rpm
- ▷ advantages:
 - widespread; well-known commands; known by upstream maintainers; huge community
- ▷ drawbacks:
 - no standard, no build-from-source support, no centralization

Case Study: Debian

– slide 14

- ▷ package format: Debian deb
- ▷ package manager: `dpkg` / `apt-get` / `dselect`
- ▷ advantages:
 - widespread; dependency tree & binaries fully centralized; huge community; well documented
- ▷ drawbacks:
 - lacks a decent build-from-source support; complex

Case Study: Solaris

– slide 15

- ▷ package format: SVr4 pkg
- ▷ package manager: pkg*
- ▷ advantages:
 - uses existing Solaris PMS
- ▷ drawbacks:
 - dispersed community; zero build-from-source support; little centralization; OS-specific

Case Study: *BSD

– slide 16

- ▷ package format: enhanced `tgz` tarballs
- ▷ package manager: `pkg_*`
- ▷ advantages:
 - centralized; multi-platform; sandboxed; bulk-build support; large community
- ▷ drawbacks:
 - incomplete binary distribution system

Case Study: GNU/Darwin

– slide 17

- ▷ package format: Debian deb
- ▷ package manager: fink / dselect
- ▷ advantages:
 - fully centralized, ease of use, “add-on” system
- ▷ drawbacks:
 - mono-platform (though portable), medium-sized community

Comparative chart

– slide 18

	SVr4	RedHat	Debian	*BSD	Fink
user simplicity	+	+	+	-	+
easy maintenance	-	-	+	+	+
easy packaging	+	-	-	-	+
build-from-source			-	+	+
sources centralized			-	+	+
install binaries	+	+	+	-	+
binaries centralized	-		+	-	+
sandbox	-		-	+	+

Current Solaris Packaging Efforts

– slide 19

- ▷ pkg-based: SunFreeware, iBiblio
binary SVr4 package repositories; dispersed though well maintained
- ▷ tgz-based: Zoularis
uses NetBSD's source package tree, well maintained, not Solaris-specific
- ▷ deb-based: SoftAgency
ported dpkg and apt; binaries only; not maintained