System Administration with pkgsrc

<seb@ssr.univ-paris7.fr>
a.k.a <seb@NetBSD.org>

PkgsrcCon 2006

Who ? What ?

- Networks (data & voice) department of Paris 7 University
- About 20 supporting servers
- All running NetBSD, /i386 or /sparc64
- Of course all using pkgsrc!

Which kind of packages ?

- About 300 packages used (leaf & non-leaf), all architectures and NetBSD versions accounted for
- Mostly “server type” packages (read no KDE or Gnome)

Building method

- Limited bulk builds (mk/bulk/build -s)
- Builds done by each sysadmins in “personal” pkg_comp(8) sandboxes
- Builds usually done on “washing machines”, see http://asg.web.cmu.edu/depot/SoftMgmt.html
- 1 washing machine per architecture
- pkg_comp's+(lib)kver(3): 1 washing machine serves all supported NetBSD releases on its architecture

mk.conf

- 1 mk.conf used for all builds
- General build settings
- Packages configuration and options
- Inventory of packages per host via Make variables

```
SITE_SPECIFIC_PKGS+=    devel/subversion-base
...
SPECIFIC_PKGS.snmp-tools+=      local/cisco-mibs
SPECIFIC_PKGS.snmp-tools+=      net/net-snmp
....
YHOSTS+=        grissom
YHOSTS+=        yamaguchi
...
.for h in ${YHOSTS}
HOST_SPECIFIC_PKGS.${h}+=       local/arpwatch
HOST_SPECIFIC_PKGS.${h}+=       net/bind9
...
HOST_SPECIFIC_PKGS.${h}+=       ${SPECIFIC_PKGS.snmp-tools}
.endfor
```

List of packages to build

pkgsrc/local/etc/mk.conf

```
SITE_SPECIFIC_PKGS:=    ${SITE_SPECIFIC_PKGS:O:u}

.for h in ${HOSTS.${OPSYS}-${MACHINE_ARCH}}
GROUP_SPECIFIC_PKGS+=   ${HOST_SPECIFIC_PKGS.${h}}
.endfor
GROUP_SPECIFIC_PKGS:=   ${GROUP_SPECIFIC_PKGS:O:u}

USER_SPECIFIC_PKGS+=    ${USER_SPECIFIC_PKGS.${OPSYS}}
USER_SPECIFIC_PKGS+=    ${USER_SPECIFIC_PKGS.${OPSYS}-${MACHINE_ARCH}}
USER_SPECIFIC_PKGS:=    ${USER_SPECIFIC_PKGS:O:u}
```

pkgsrc/local/etc

- Makefile, pkg_comp(8) configuration templates, mk.conf, build.conf
- pkgsrc/local/etc/Makefile's convenience targets
  - Cvs update pkgsrc
  - Create pkg_comp sandboxes
  - Launch limited bulk builds in sandboxes
  - Upload packages
  - Generate manifest of packages per hosts from *_SPECIFIC_PKGS

```
# from     $Id: mk.conf 6107 2006-04-20 13:27:39Z
seb $
bozeman archivers/unzip
bozeman archivers/zip
...
```

Binary packages sets

- Binary packages set = binary packages built by one bulk build run
- Sets are pruned (PRUNEPACKAGES=yes in build.conf) and uploaded on a FTP server (BULK_ID set at upload time)
- Sets also contains:
  - lintpkgsrc “internal database” build by the bulk build framework (LINTPKGSRC_CACHE=yes in build.conf)
  - pkgsrcmap file built by lintpkgsrc -g from pkgsrc/local/etc/Makefile

Sets on our FTP

All packages sets:

```
:;ls
NetBSD-2.0_STABLE-i386                 NetBSD-3.0_STABLE-i386_eh-20060330
NetBSD-2.0_STABLE-i386_eh-20060330     NetBSD-3.0_STABLE-i386_seb-20060420
NetBSD-2.0_STABLE-i386_seb-20060420    NetBSD-3.0_STABLE-sparc64
NetBSD-2.0_STABLE-sparc64              NetBSD-3.0_STABLE-sparc64_eh-20060330
NetBSD-2.0_STABLE-sparc64_eh-20060330  NetBSD-3.0_STABLE-sparc64_seb-20060420
NetBSD-2.0_STABLE-sparc64_seb-20060420 distfiles
NetBSD-2.1_STABLE-sparc64              hosts-pkgs
NetBSD-3.0_STABLE-i386
```

Blessed packages sets:

```
:;ls -l NetBSD-3.0_STABLE-i386 NetBSD-3.0_STABLE-sparc64
lrwxr-xr-x  1 seb  staff  35 Apr 20 16:12 NetBSD-3.0_STABLE-i386 -> NetBSD-3.0_STABLE-i386_seb-20060420
lrwxr-xr-x  1 seb  staff  38 Apr 20 21:59 NetBSD-3.0_STABLE-sparc64 -> NetBSD-3.0_STABLE-sparc64_seb-20060420
```

Now what ?

- On servers daily audit-packages(8) of course
- To update packages:
  - Get package set “manifest”
  - Compare with installed version
  - Remove obsolete ones
  - Install required packages
  - ...
- Lots of typing (ftp, lintpkgsrc, audit-packages, pkg_delete, pkg_add, ...)

To save typing: mpkg.sh 1/3

```
usage: mpkg.sh [options] operations [operations...]
add=PKG
        Install package PKG from PKG_PATH
db
        If PKGSRCDB_URL is set, get lintpkgsrc internal database from
        PKGSRCDB_URL and pkgsrc map from PKGSRCMAP_URL.
        If PKGSRCDB_URL is not set or option -f is given build lintpkgsrc
        internal databse and pkgsrc map. This requires lintpkgsrc(1) and
        PKGSRCDIR.
        If option -e is given build or fetch may be skipped altogether.
check
        Check installed packages versions, with lintpkgsrc(1) -i
        against latest versions available as listed in lintpkgsrc
        internal database.  Check security vulnerabilities of
        installed packages with audit-packages(8).
```

To save typing: mpkg.sh 2/3

getmanifest

Get manifest of packages per hosts from PKGMANIFEST_URL and save it in VARDIR/hosts-pkgs.

checkreq

Check if required packages as listed in manifest of packages per hosts are installed.

installreq

Install missing required packages as listed in manifest of packages per hosts.

perm

Fix package's files permissions via mtree(8) using mtree specification files in MTREEDIR/*.mtree.
When a specification filename is foo.mtree it is only considered if a package named 'foo' is installed.
The specification file 'pkg.mtree' is always considered.

To save typing: mpkg.sh 3/3

pkgdepgraph
- Generate various files and scripts with pkgdepgraph(1)
- VARDIR/pkgs-delete-order
  - list of out-of-date and vulnerable packages that need to be deleted, in least required first order
- VARDIR/pkgs.sh
  - generic "do it" script for each out-of-date and vulnerable packages VARDIR/pkgs.dot
  - dependencies graphs of installed packages
- VARDIR/pkgs-out-of-date.dot
  - dependencies graphs of out-of-date installed packages
- VARDIR/pkgs-add.sh
  - package add script

autopilot
- runs canned sequence of operations:
- db, getmanifest, check, checkreq, pkgdepgraph.

checkpoint
- Not yet implemented.

pkgpath.sh

```
# helper script to set PKG_PATH
# BULK_ID?
: ${BULK_ID:=}
# Sanitize BULK_ID
case "${BULK_ID}" in""|_*) : ;;*) BULK_ID="_${BULK_ID}" ;;esac
# default PKG_PATH
DEFAULT_PKG_PATH="ftp://ftp.univ-paris7.fr/pub/SSR/packages/NetBSD-$(uname -r)-$(uname -m)
${BULK_ID}/All"
# PKG_PATH in env?
: ${PKG_PATH:=${DEFAULT_PKG_PATH}}
# export it
export PKG_PATH
```

Typical usage of mpkg.sh

```
$ /etc/sf/mpkg autopilot | BRAIN
...
# pkg_delete $(cat /var/mpkg/pkgs-delete-order)
...
# /etc/sf/mpkg installreq
...
# /etc/sf/mpkg perm
...
# pkg_admin check
...
```

Mpkg check output

```
:;/etc/sf/mpkg check
=> check
pkg_info: disabling PKG_PATH when operating on all packages.
Reading pkgsrc database: 5915 / 5915 (6026 / 6026) pkgs...done
Version mismatch: 'Radiator' 3.13nb3 vs 3.14p649
Version mismatch: 'amanda-client' 2.4.4p4nb1 vs 2.4.4p4nb3
...
Package php-4.4.1nb2 has a inject-smtp-headers vulnerability
(vulnid:unknown), see
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-3883
...
Missing: foobar-0.1
...
:;
```

/var/mpkg/*

```
:;ls /var/mpkg/
hosts-pkgs                pkgs-delete-order         pkgs.sh
pkgdepgraph.in            pkgs-out-of-date.dot      pkgsrc.db
pkgs-add.sh               pkgs.dot                  pkgsrcmap
:;head -2 /var/mpkg/pkgs.sh
( pkg_info -qe perl || ( cd /usr/pkgsrc/lang/perl5 && make "$@"
 ) ) &&
( pkg_info -qe p5-MIME-Base64 || ( cd /usr/pkgsrc/converters/p5-
MIME-Base64 && make "$@" ) ) &&
```

Future work

- Implement mpkg's checkpoint operation
- Integrate/work on recent pkg_summary(5) to get ride of lintpkgsrc -i in mpkg check operation ?
- May be return again on pkg_chk(8) ?
- Support per host packages configuration using BULK_ID

That's it

- mpkg.sh as of today available at http://www.ssr.univ-paris7.fr/~seb/mpkg.sh
- Thanks for your attention!
- Any questions ?