Unix Application Software

Migration Scenarios towards the New Distribution Scheme

Stephan Wiesand

DESY Zeuthen – DV -

August 26, 2003

Reminder: What this was about

- distributing software for supported Unix flavours
  - Solaris >= 2.6, DESY Linux >= 4
- suitable for
  - systems with small disks (remote software)
  - mission critical systems w/ large disks (local install)
  - typical systems (partially local installation)
  - mobile/home systems w/o network
  - external institutes
- flexible, delegatable configuration
- well defined, easily reproducible state w/o backup

Solution chosen

- RPM packages
  - provided in an AFS repository
  - installed into an AFS "reference installation"
    - can be used by clients through a single symlink
      - like /products today
    - or a single symlink per package and version
      - + a small local "default package" w/ symlinks
  - packages can also be installed locally
- tool for dealing with all this in a simple way
  - define what should be installed in which way
  - run tool to turn "is" state into defined state

Where we are

- the tool is being finalized
  - bugs are found as we start using it
  - but functionality complete
    - automatic dependency resolution
    - hierarchic, delegatable configuration
      - reasonable default from DV/IT
      - can be modified (overridden) by group admins
        - redelegation possible
      - can be overridden by DV/IT for critical packages
    - special mode for maintaining the reference installation
    - convenient queries
  - missing: logging

Where we are

- many software packages foreseen are available
  - for DL4 and DL5, Solaris lacks build host
  - gcc 2.95.3 (DL4/5), gcc 3.3 (DL5) - 2.91.66 anyone ?
  - several berkeley db & openssl versions
  - kerberos 4 & 5 (heimdal/MIT), sasl -> pine, pam
  - perl 5.8.0, + some 100 modules
  - apache incl. mod_perl, embperl, php, expat
  - maple 8, acroread, libxml2, libxslt, autoconf
- some important ones are still missing
  - cernlib, root, mozilla, openoffice, [x]emacs (?)

Differences for Users

- binaries in /opt/products/bin, man pages in .../man, ...
  - makes a difference for users who set their PATHS
  - no problem for people who just modify default
- libraries in /opt/products/lib, includes in .../include
  - BUT: /opt/products/lib NOT searched by dynamic linker by default (not in ld.so.conf or in LD_LIBRARY_PATH)
  - instead: use compiler & linker flags at compile time
    - takes some getting used to
    - but is the only way not to reproduce the /usr/local mess
    - and protects you against silent changes by us

Compatibility

- some compatibility links will be provided
  - example: /products/perl/bin/perl
    - today: /products/perl/i386_linux24 -> v5.8.0
    - new scheme: i386_linux24 -> /opt/products/perl/5.8.0
    - there will also be a /usr/local/bin/perl link
      - until the end of time
- what exists in /products today will stay
  - maybe for many years
- but eventually
  - /usr/local/bin will be depopulated (YOUR request)
  - /products will be frozen

Migration Scenarios

- on DL5, software will be provided in the NEW scheme
  - ... most of it
  - ... and with many compatibility links
  - /usr/local will be local, and relatively empty
    - in particular, /usr/local/lib and /usr/local/include
- what about the old platforms ?
  - DL4
  - Solaris 2.6, 2.7, 8, 9
    - do you (still) care for application software there ?
  - DL3, HP-UX, AIX, IRIX are frozen anyway

Scenario 1

- no changes at all on existing platforms
- new scheme for DL5 and Solaris 10 only
- pros:
  - users hate changes, and love continuity
- cons:
  - hard cut from old to new platforms
  - migration to DL5 becomes harder
  - additional work to provide software in 2 schemes
    - this field is severly underpowered anyway
    - that's one reason why we need to change something

Scenario 2

- freeze /products content for existing platforms
- deploy /opt/products on all supported platforms
  - produce future versions in new scheme only,
    - wherever reasonable
  - provide compatibility links at least for old platforms
- pros:
  - much more efficient
    - building packages for 2nd , 3rd,... platform is cheap
  - old platforms benefit, smoother migration to DL5
- cons: it may hurt occasionally, despite all efforts

Scenario 3

- there is no scenario 3
- nobody even remotely considers wiping out /products and /usr/local tomorrow

Summary

- Unix software distribution **will** change
  - we have to satisfy additional needs
    - mobile & mission critical systems
  - we have to do this with the same (more likely: less) person power as (than) in the past
- migration will not be completely painless
- but we can choose our poison
- what would YOU prefer?