Connecting the Debian and PostgreSQL worlds

Christoph Berg
christoph.berg@credativ.de
myon@debian.org

PGconf.EU 2011, Amsterdam
October 20th, 2011
Debian Developer
- Quality Assurance, New Members
- maintaining PostgreSQL programs and extensions

senior consultant at credativ GmbH in Mönchengladbach, Germany (not too far from here)
- PostgreSQL support and operation
- general Debian topics
Agenda

- PostgreSQL and Debian Releases
- Packaging PostgreSQL Extensions
- Repository
- Open Ends
- Wrapup
PostgreSQL and Debian Releases
## PostgreSQL and Debian Releases

<table>
<thead>
<tr>
<th>PostgreSQL</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.4 November 2003</td>
<td></td>
</tr>
<tr>
<td>8.0 January 2005</td>
<td>3.1 Sarge June 2005</td>
</tr>
<tr>
<td>8.1 November 2005</td>
<td></td>
</tr>
<tr>
<td>8.2 December 2006</td>
<td>4.0 Etch April 2007</td>
</tr>
<tr>
<td>8.3 February 2008</td>
<td>5.0 Lenny February 2009</td>
</tr>
<tr>
<td>8.4 July 2009</td>
<td></td>
</tr>
<tr>
<td>9.0 September 2010</td>
<td>6.0 Squeeze February 2011</td>
</tr>
<tr>
<td>9.1 September 2011</td>
<td>7.0 Wheezy Freeze June 2012</td>
</tr>
</tbody>
</table>

- **PostgreSQL:** about every year
- **Debian:** about every 2 years
### Version Matrix

<table>
<thead>
<tr>
<th></th>
<th>7.4</th>
<th>8.0</th>
<th>8.1</th>
<th>8.2</th>
<th>8.3</th>
<th>8.4</th>
<th>9.0</th>
<th>9.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Sarge</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0 Etch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0 Lenny</td>
<td></td>
<td></td>
<td></td>
<td>.</td>
<td>X</td>
<td></td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>6.0 Squeeze</td>
<td></td>
<td></td>
<td></td>
<td>.</td>
<td>.</td>
<td>X</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>7.0 Wheezy</td>
<td></td>
<td></td>
<td></td>
<td>(X)</td>
<td>(X)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unstable</td>
<td></td>
<td></td>
<td></td>
<td>(X)</td>
<td>(X)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- since Etch: several PostgreSQL versions installable in parallel
- usually only one shipped
- server extension packages are available for a subset of these
- **goal: fill in missing X/·,** and **build extensions for them**
Packaging PostgreSQL Extensions
Debian Packages in a Nutshell

- source package = orig.tar.gz + diff + pkg_version.dsc file
- binary package = pkg_version_arch.deb file
- debian/control: package name, (build) dependencies, description, maintainer
- debian/rules: Makefile with targets to build, install, and build binary package
- dpkg-buildpackage invokes targets to build one or more binary packages from source tree
- example: source postgresql-plproxy_2.1-1.dsc builds postgresql-8.4-plproxy_2.1-1_amd64.deb (armel, i386, ia64, kfreebsd-amd64, kfreebsd-i386, mips, mipsel, powerpc, s390, sparc)
Status Quo

- Extension packages build-depend on postgresql-server-dev-X.Y
- Version change needs manual editing of several files in debian/
- Usually no support to build for two PostgreSQL versions

debian/control:

Source: postgresql-plproxy
Build-Depends: debhelper (>= 7), postgresql-server-dev-8.4, flex, bison
Standards-Version: 3.9.1

Package: postgresql-8.4-plproxy
Architecture: any
Depends: postgresql-8.4, ${misc:Depends}, ${shlibs:Depends}
Description: PL/Proxy is a proxy language for PostgreSQL 8.4
#!/usr/bin/make -f

PG_CFG84 = /usr/lib/postgresql/8.4/bin/pg_config
CFLAGS84 = $(shell $(PG_CFG84) --cflags)
SRCDIR = $(CURDIR)
TGT84 = $(CURDIR)/debian/prefix-8.4

install: build
    mkdir -p $(TGT84)
    cd $(TGT84) && $(MAKE) -f $(SRCDIR)/Makefile \
        CFLAGS="$(CFLAGS)" PG_CONFIG="$(PG_CFG84)" \
        VPATH="$(SRCDIR)"
    dh_install -ppostgresql-8.4-prefix
#!/usr/bin/make -f

PACKAGE := $(shell dh_listpackages)
PGVERSION = $(patsubst postgresql-%-plproxy,%,$(PACKAGE))
PG_CONFIG = /usr/lib/postgresql/$(PGVERSION)/bin/pg_config

build:
  $(MAKE) PG_CONFIG=$(PG_CONFIG) PG_CPPFLAGS='$(PG_INC)'

install: build
  $(MAKE) install PG_CONFIG=$(PG_CONFIG) PG_CPPFLAGS='$(PG_INC)'
  DESTDIR=$(CURDIR)/debian/postgresql-$(PGVERSION)-plproxy
The Future: pg_buildext

- system provides /usr/share/postgresql-common/supported-versions
- package provides debian/pgversions
- **pg_buildext** works on intersection of these lists
- Build-Depends: postgresql-server-dev-all

debian/rules:

build:

  +pg_buildext build $(CURDIR) build-%v

install: build

  +pg_buildext install $(CURDIR) build-%v \
  postgresql-%v-plsh
debian/control generated from debian/control.in:

Source: postgresql-plproxy
Build-Depends: debhelper (>= 7), flex, bison, asciidoc, postgresql-server-dev-all (>= 119~)
Standards-Version: 3.9.2

Package: postgresql-PGVERSION-plproxy
Architecture: any
Depends: postgresql-PGVERSION, ${misc:Depends}, ${shlibs:Depends}
Description: database partitioning for PostgreSQL PGVERSION
Repository
Repository

- http://pgapt.debian.net/
- repository with PostgreSQL server and extension packages
  - Debian Lenny/Squeeze/Wheezy/unstable
  - PostgreSQL 8.2/8.3/8.4/9.0/9.1
- packages are imported from unstable
- build daemons build for all releases
- http://pgapt.debian.net/bin/packages.cgi
- currently: hstore-new, ip4r, pgfincore, plr, pljava, PLLua, plproxy, plsh, prefix, preprepare, skytools(3)

/etc/apt/sources.list:

```
deb http://pgapt.debian.net/ lenny-pgapt main  # 5.0
deb http://pgapt.debian.net/ squeeze-pgapt main  # 6.0
```
Building Packages

- custom /usr/share/postgresql-common/supported-versions file to cover all PostgreSQL versions
- debian/pgversions extended where needed
- add new changelog entry && dpkg-buildpackage

- packages built have version suffix ~pgapt+1 (unstable)
  ~pgaptNN+1 (others)
- ~ sorts before “empty string”
- 1.0~pgapt50+1 < 1.0~pgapt60+1 < 1.0~pgapt+1 < 1.0
- postgresql-9.1-plproxy_2.2-3~pgapt60+1_i386.deb
Packages Database

- repository is imported in a PostgreSQL database (qa.debian.org schema)
  - package (version, architecture, maintainer, ...)
  - suite (distribution, architecture)
  - packagelist (suite_id, package_id)
  - package_source (package_id, source_id)
  - other meta data

- extended to cover new package–PostgreSQL version combinations
New Package–Version Combinations

- table pgversion – list of targeted versions
  - 8.2 8.3 8.4 9.0 9.1
- view all_versioned_packages – synthetic list of all possible packages (postgresql-*-extension)

```sql
SELECT DISTINCT
  -- prefix, e.g. 'postgresql-
  regexp_replace (package, E'[0-9]+\.[0-9]+', '') ||
  pgversion ||
  -- suffix, e.g. '-plproxy'
FROM package NATURAL JOIN pgversion
WHERE package ~ E'[0-9]\.[0-9]';
```
What to Build

- **view missing_packages** – packages not yet built (per Debian release/architecture)
- **view outdated_packages** – packages needing rebuild

CREATE VIEW outdated_packages AS
SELECT ap.*, s.suite, s.architecture, p.version AS oldversion
FROM all_versioned_packages ap
JOIN package p ON (ap.package = p.package)
JOIN packagelist pl ON (p.package_id = pl.package_id)
JOIN suite s ON (pl.suite_id = s.suite_id)
WHERE p.version::debversion < ap.version::debversion; -- FIXME

---

| pgversion | 9.0 |
| package   | postgresql-9.0-pljava-gcj |
| source    | postgresql-pljava |
| version   | 1.4.3-1~pgapt70+1 |
| suite     | wheezy-pgapt |
| architecture | i386 |
| oldversion | 1.4.2-4 |
Open Ends
Repository

- actually use the database for automated building
- pgq?
- remember package-version combinations that are not supported
- “best-effort” support for old releases
many packages not yet converted
many extensions do not support out-of-tree builds – “VPATH”
mkdir build-8.4
cd build-8.4
make -f ../Makefile VPATH=$PWD/..
maybe a PGXS deficiency
pg_buildext might do in-place build&&install instead
- testing extensions manually on all PostgreSQL versions is infeasible
- `pg_regress` supported by many extensions
- but `pg_regress` wants files in their final location (`:libdir`)
- at build time only present in `debian/$pkg/usr/...`
- idea: fakerooid/fakechroot-like wrapper to create an overlay filesystem (`LD_PRELOAD`)
Wrapup
What you can do as an extension author

- document PostgreSQL versions supported
- support 9.1’s extension control files
- make your Makefile VPATH-friendly
- provide regression tests
What you can do as a package maintainer

- use pg_buildext (at least “supported-versions”)
- put all supported versions into debian/pgversions, not just the current one
- try to avoid debian/$pkg.install and friends
What you can do as a DBA and User

- test & report
- does this meet your needs?
Wrapup

- building extensions for many targets
- ongoing work
- visit http://pgapt.debian.net/
- thanks to Dimitri Fontaine!