Large Scale QA in the PostgreSQL Ecosystem
Maintaining 8000 Packages

Christoph Berg <christoph.berg@credativ.de>
PostgreSQL Consultant
PostgreSQL Contributor
Debian Developer

31st January 2015
Packages
apt.postgresql.org

Hosts packages for . . .

- **6 PostgreSQL releases**
  8.4 9.0 9.1 9.2 9.3 9.4 (9.5)
- **7 Debian and Ubuntu releases**
  squeeze wheezy jessie sid lucid precise trusty
- **2 architectures**
  amd64 i386
- **$6 \times 7 \times 2 = 84$ targets**

- About 70 source packages
- Some are built “only” per distribution/architecture (14 targets)
- Most depend on the PostgreSQL version as well (84 targets)
Simple packages

pkg 1.0 tarball

debian/ dir from Git

pkg-source

generate_pgdg_source

pkg_1.0-1.pgdg14.10+1.dsc
pkg_1.0-1.pgdg70+1.dsc
pkg_1.0-1.pgdg60+1.dsc

pkg-binaries

build_and_provide_package

pkg_1.0-1.pgdg60+1_amd64.deb
pkg_1.0-1.pgdg60+1_i386.deb
pkg_1.0-1.pgdg60+1_amd64.deb
pkg_1.0-1.pgdg60+1_i386.deb

apt.postgresql.org
reprepro
dput

pkg_1.0-1.pgdg60+1_amd64.deb
pkg_1.0-1.pgdg60+1_i386.deb
pkg_1.0-1.pgdg60+1_amd64.deb
pkg_1.0-1.pgdg60+1_i386.deb

pkg_1.0-1.pgdg60+1_amd64.deb
pkg_1.0-1.pgdg60+1_i386.deb
PostgreSQL extension packages
No one is going to test 8000 packages
We need testsuites
PostgreSQL has regression tests
Many extensions have regression tests
`postgresql-common` has a testsuite on top of the PostgreSQL packages
Various application level testsuites of varying quality
Quality Assurance

- No one is going to test 8000 testsuites
- We need automation
Tests
Jenkins

- Continuous Integration Server
- Takes your code and compiles it all the time
- More generally a framework for running scripts
- Jobs with “configurations”
- “source” jobs take a tarball and a VCS with a debian/ directory to build source packages per distribution
- “binaries” jobs take source packages and build (many) binary packages
Project orafce-source

Build Debian source package of orafce from git.
Do not edit this job through the web interface, it is generated via jenkins-job-builder!

Configurations

Downstream Projects

https://pgdgbUILD.dus.dg-i.net/jenkins/job/orafce-source/lastFailedBuild/
Project orafce-binaries

Build Debian binary package of orafce.
Do not edit this job through the web interface, it is generated via jenkins-job-builder!

<table>
<thead>
<tr>
<th>Configuration Matrix</th>
<th>amd64</th>
<th>386</th>
</tr>
</thead>
<tbody>
<tr>
<td>sid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jessie</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wheezy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>squeeze</td>
<td></td>
<td></td>
</tr>
<tr>
<td>utopic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trusty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>precise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lucid</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Latest Test Result (no failures)

Upstream Projects

orafce-source
oraefce Console Output

Jenkins  › orafce-binaries  › #19  › amd64, sid

```
17:08:26  ## installcheck 9.4 ##
17:08:29  Creating new cluster 9.4/regress ...
17:08:29  config /etc/postgresql/9.4/regress
17:08:29  data /var/lib/postgresql/9.4/regress
17:08:29  locale en_US.utf8
17:08:30  port 5432
17:08:32  /pg_regress --inputdir=/tmp/adt-run.8HvDm7/tree --psqldir='/usr/lib/postgresql/9.4/bin' --load-language=plpgsql --schedule=parallel_schedule --encoding=utf8
17:08:32  --dbname=contrib_regression orafce dbms_output dbms_utility files varchar2 nvarchar2 aggregates nlssort dbms_random
17:08:32  (using postmaster on localhost, port 5432)
17:08:32  =============== dropping database 'contrib_regression' ===============
17:08:32  NOTICE: database 'contrib_regression' does not exist, skipping
17:08:32  DROP DATABASE
17:08:32  =============== creating database 'contrib_regression' ===============
17:08:33  CREATE DATABASE
17:08:33  ALTER DATABASE
17:08:33  =============== installing plpgsql ===============
17:08:33  CREATE LANGUAGE
17:08:33  =============== running regression test queries ===============
17:08:33  test init ... ok
17:08:33  parallel group (2 tests):  dbms_pipe_session_A dbms_pipe_session_B
17:08:43  dbms_pipe_session_A ... ok
17:08:43  dbms_pipe_session_B ... ok
17:08:43  test oracle ... ok
17:08:43  test dbms_output ... ok
17:08:43  test dbms_utility ... ok
17:08:43  test files ... ok
17:08:43  test varchar2 ... ok
17:08:43  test nvarchar2 ... ok
17:08:43  test aggregates ... ok
17:08:43  test nlssort ... ok
17:08:44  test dbms_random ... ok
17:08:44  All 12 tests passed.
```
pg_builddext

- Helps with building packages for multiple PostgreSQL versions at once
- Loops over versions listed in debian/pgversions
- Supported versions defined by /usr/share/postgresql-common/supported-versions
- Ideally no change in debian/* needed to target a new PostgreSQL version
debian/rules

include /usr/share/postgresql-common/pgxs_debian_control.mk

override_dh_auto_build:
  +pg_buildext build build-%v

override_dh_auto_test:
  # nothing to do here, see debian/tests/* instead

override_dh_auto_install:
  +pg_buildext install build-%v postgresql-%v-foobart

override_dh_installldocs:
  dh_installldocs --all README.*

override_dh_auto_clean:
  +pg_buildext clean build-%v

%:
  dh $@
Build-time testing vs. testing installed packages

- Regression tests are nice, but don’t catch packaging errors
- Run tests on installed packages as well (or instead)
- Extensions only support “make installcheck” anyway
- ** autopkgtest**
- postgresql-common: 1274 tests for postgresql-9.4
- 1954 for postgresql-9.3+9.4 incl. upgrade tests
autopkgtest

- debian/tests/control
  Tests: run-testsuite
  Depends: @, build-essential, hunspell-en-us, locales, netcat-openbsd, net-tools, logrotate
  Restrictions: needs-root

- debian/tests/run-testsuite

```
#!/bin/sh
cd /usr/share/postgresql-common
./testsuite
```

====== Running all tests with tight umask 077 ======

--- Running test 001_packages.t ... ---

1..19

# PostgreSQL versions installed: 9.4
ok 1 - postgresql-9.4 installed
ok 2 - postgresql-plpython-9.4 installed
ok 3 - postgresql-plpython3-9.4 installed
ok 4 - postgresql-plperl-9.4 installed
ok 5 - postgresql-server-dev-9.4 installed
ok 6 - postgresql-contrib-9.4 installed
ok 7 - libecpg-dev installed
ok 8 - logrotate installed
ok 9 - procps installed
ok 10 - netcat-openbsd installed
ok 11 - hunspell-en-us installed
ok 12 - system has a default UTF-8 locale
autopkgtest and pg_buildext

- Simple debian/tests/installcheck
  
  ```bash
  #!/bin/sh
  pg_buildext installcheck
  ```

- Complex debian/tests/installcheck
  
  ```bash
  #!/bin/sh
  set -e

  for v in $(pg_buildext supported-versions); do
    case $v in
      8*|9.0) # don't bother to test the extension here
        continue 
      esac

      PATH=$PATH:/usr/lib/postgresql/$v/bin
      pg_buildext installcheck-$v
    done
  ```
# Packages overview for Debian PostgreSQL Maintainers

**Maintainer:** Debian PostgreSQL Maintainers <pkg-postgresql-public@lists.alioth.debian.org> — Bugs: open · RC · all · submitted · WNPP · — Reports: Dashboard · Build · Patch · Debian · Piuparts · Portfolio

## main (34)

<table>
<thead>
<tr>
<th>Source Name</th>
<th>Bugs</th>
<th>Version</th>
<th>VCS</th>
<th>Binary Package</th>
<th>Buildd</th>
<th>Deb Check</th>
<th>Limitian</th>
<th>E+W</th>
<th>Plu Parts</th>
<th>CI</th>
<th>Popcon</th>
<th>Watch</th>
</tr>
</thead>
<tbody>
<tr>
<td>orafe</td>
<td>-</td>
<td>3.0.1.1</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>check postgres</td>
<td>-</td>
<td>2.19.0-1</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>postresql-pgmp</td>
<td>-</td>
<td>-</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>pg-orig</td>
<td>-</td>
<td>-</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>postgresql-common</td>
<td>19</td>
<td>113&lt;squel&gt;</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>pgaadmin</td>
<td>6</td>
<td>1.14.o-1</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>postgresql-9.4</td>
<td>16</td>
<td>113&lt;squel&gt;</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>postgresql-multicorn</td>
<td>-</td>
<td>-</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>libpqtypes</td>
<td>-</td>
<td>1.5.1-2</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>psqljdbc</td>
<td>2</td>
<td>109.o-1.0</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>pgpool24</td>
<td>1</td>
<td>3.1.3-5</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>pgmemcache</td>
<td>-</td>
<td>2.0-1</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>pgmcclient</td>
<td>-</td>
<td>1.0.1-1</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
<tr>
<td>postgresql-filedump</td>
<td>-</td>
<td>9.1.0-1</td>
<td>Git</td>
<td>OK 1</td>
<td>14+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20+</td>
<td>16</td>
<td>3.0.14</td>
<td></td>
</tr>
</tbody>
</table>

---

Christoph Berg <christoph.berg@credativ.de> Large Scale QA in the PostgreSQL Ecosystem 31st January 2015
Bugs found
Bugs found: peer authentication

Found by postgresql-common:

commit b777be0d48a042f500cac72140ffb50392973aa2
Author: Tom Lane <tgl@sss.pgh.pa.us>
Date: Fri Mar 28 10:30:37 2014 -0400

Un-break peer authentication.

Commit 613c6d26bd42dd8c2dd0664315be9551475b8864 sloppily replaced a lookup of the UID obtained from getpeereid() with a lookup of the server's own user name, thus totally destroying peer authentication. Revert. Per report from Christoph Berg.

In passing, make sure get_user_name() zeroes *errstr on success on Windows as well as non-Windows. I don't think any callers actually depend on this ATM, but we should be consistent across platforms.
Bugs found: stack size detection

Found by the PostgreSQL regression tests:

```plaintext
***************
*** 444,447 ****
| select infinite_recurse()
| language sql;
| \set VERBOSITY terse
| select infinite_recurse();
| ! ERROR: stack depth limit exceeded
--- 444,447 ----
| select infinite_recurse()
| language sql;
| \set VERBOSITY terse
| select infinite_recurse();
| ! connection to server was lost
```

General Linux issue on 32bit archs, mitigated by disabling build hardening.
Bugs found: non-reproducible EXPLAIN output

Found by the multicorn regression tests on 9.4:
During the 9.4 development, EXPLAIN grew an extra "Planning time" output line that made it impossible to use this in regression tests:

```
explain select * from testmulticorn m1 inner join testmulticorn m2 on m1.test1 = m2.test1;
```

```
QUERY PLAN

-------------------------------------------------------------------------------------
| Nested Loop (cost=20.00..806.05 rows=2 width=128) |
| Join Filter: ((m1.test1)::text = (m2.test1)::text) |
| --> Foreign Scan on testmulticorn m1 (cost=10.00..400.00 rows=20 width=20) |
| --> Materialize (cost=10.00..400.10 rows=20 width=20) |
| --> Foreign Scan on testmulticorn m2 (cost=10.00..400.00 rows=20) |
(5 rows)
```

Resolved by showing the Planning time for EXPLAIN ANALYZE only

Christoph Berg <christoph.berg@credativ.de> Large Scale QA in the PostgreSQL Ecosystem 31st January 2015 25 / 35
Bugs found: timezone mess

First spotted on Solaris, but it was a Debian problem as well:
When using `--with-system-tzdata`:

```
1937c1937
< Sat Oct 25 22:00:00 2014 UTC
---
> Sat Oct 25 21:00:00 2014 UTC
```

```
1943c1943
< Sat Oct 25 22:00:01 2014 UTC
---
> Sat Oct 25 21:00:01 2014 UTC
```

(Blame the Russians)

Resolved by using testcases for similar TZ changes from 2012 instead
Bugs found: shm_mq problems

```bash
../../src/test/regress/pg_regress --inputdir=/${PKGBUILDDIR}/build/../contrib/test_shm_mq --temp-install=./tmp_check --top-builddir=../.. --port=3133 --extra-install=contrib/test_shm_mq --dbname=contrib_regression test_shm_mq
```

============== creating temporary installation ==============
============== initializing database system ==============
============== starting postmaster ==============
running on port 3133 with PID 29665
============== creating database "contrib_regression" ==============
CREATE DATABASE
ALTER DATABASE
============== running regression test queries ==============
E: Caught signal ‘Terminated’: terminating immediately
make[4]: *** [check] Terminated
make[3]: *** [check-test_shm_mq-recurse] Terminated
test test_shm_mq ... /${PKGBUILDDIR}/build/..../src/makefiles/pgxs.mk:305: recipe for target 'check-world-contrib-recurse' failed
GNUmakefile:69: recipe for target 'check-world-contrib-recurse' failed
Build killed with signal TERM after 300 minutes of inactivity

Issue on some mips(el) build hosts, still unresolved
Bugs found: psqlodbc problems

Found by the psqlodbc regression tests on s390x:

```c
struct IPDFields_
{
-#if (ODBCVER >= 0x0300)
  -  SQLUINTEGER       *param_processed_ptr;
-#else
  -  SQLULEN           *param_processed_ptr; /* SQLParamOptions */
-#endif /* ODBCVER */
+  +  SQLULEN           *param_processed_ptr;
  +  SQLUSMALLINT      *param_status_ptr;
```
Bugs found: extensions not ported to 9.x yet

Dozens of cases where extensions were not ported to 9.3 9.4 9.5 yet:

*** /tmp/adt-run.PNuMnw/tree/test/expected/mpq.out 2015-01-16 01:47:29
--- /tmp/adt-run.PNuMnw/tree/results/mpq.out 2015-01-17 21:45:27
***************
*** 486,491 ****
--- 486,492 ----
insert into test_mpq_idx select generate_series(1, 10000);
create index test_mpq_btree_idx on test_mpq_idx using btree (q);
create index test_mpq_hash_idx on test_mpq_idx using hash (q);
+ WARNING: hash indexes are not WAL-logged and their use is discouraged
-- Hash is compatible with mpz
select mpq_hash(0) = mpz_hash(0);
t

Regression failures usually easy to work around, code changes more involved
Bugs NOT found
Bugs NOT found: SSL problems in libpq

One of our servers is currently running on postgres 9.2 using the 9.2.9-1.pgdg70+1 packages from pgdg. After an apt update this morning which brought in the libpq5 package version 9.4.0-1.pgdg70+1, connections to the tabase started failing with SSL errors logged on the server:

```
[unknown] [unknown] LOG: could not accept SSL connection:
```

Rolling back the server and client to libpq5 version 9.3.5-2.pgdg70+1 fixed it.

This is running on an otherwise up-to-date Debian Wheezy. The SSL certificate is locally issued using an internal CA which has been added to the local trust store. SSL-related config options are left set to the defaults.

9.4’s libpq only supports TLS, not SSL, for which 512bit keys are too small.
Bugs NOT found: libpq and NSS lookups

In minimal chroots, 9.4's libpq is unusable:

Jan 10 00:11:40 lehmann postfix/trivial-rewrite[29960]:
  warning: connect to pgsql server localhost:5432:
    out of memory?
Jan 10 00:11:40 lehmann postfix/trivial-rewrite[29960]:
  warning: pgsql:/etc/postfix/pgsqltest lookup error for "*

/etc/passwd lookup failures will be non-fatal again in 9.4.1
TODO
Real CI (see #postgresql-apt)

06:45 -Graviton- Project postgresql-9.5-source-git-head build #115: SUCCESS

06:45 -Graviton- * tgl: Handle unexpected query results, especially NULLs,

06:45 -Graviton- * bruce: doc: clarify libpq's 'verify-full' host name check

06:45 -Graviton- * Stephen Frost: Fix BuildIndexValueDescription for expressions

07:06 -Graviton- Project dput build #978: SUCCESS in 15 sec: https://pgdgbuild.dus.dg-i.net/jenkins/job/dput/978/

07:06 -Graviton- Parameters:

07:06 -Graviton- binaries=postgresql-9.5-binaries-sid-amd64

07:06 -Graviton- queue=Incoming

18:45 -Graviton- Project postgresql-9.5-source-git-head build #116: SUCCESS

18:45 -Graviton- * kgrittn: Allow pg_dump to use jobs and serializable transactions

18:45 -Graviton- * heikki.linnakangas: Fix query-duration memory leak with

18:45 -Graviton- * tgl: Fix assorted oversights in range selectivity estimation

18:46 -Graviton- Project postgresql-9.5-binaries-sid-amd64 » amd64,sid build #114: FAILURE in 1 min 9 sec: https://pgdgbuild.dus.dg-i.net/jenkins/job/postgresql-9.5-binaries-sid-amd64/architecture=amd64,distribution=sid/114/ (configure: error: header file <libxml/parser.h> is required for XML support)

Do this for some/most/all other packages as well

Christoph Berg <christoph.berg@credativ.de> Large Scale QA in the PostgreSQL Ecosystem 31st January 2015 34 / 35
Things not automated yet

- Some packages don’t have tests
- Some packages have, well, complicated testsuites
- Triggering builds of updated packages
- Nice webpages for packages in the repository

Credits:

- Marco Nenciarini, Markus Wanner, Dimitri Fontaine
- PostgreSQL Sysadmins
- DG-i for hosting the build server, credativ for hosting me