PostgreSQL and Debian

Michael Banck <michael.banck@credativ.de> and Christoph Berg <christoph.berg@credativ.de>

DebConf17, Montreal, 12 Aug 2017
Uploading activity by year for the pkg-postgresql team.
PostgreSQL - Overview

- Extensible, object-relational database system
- Created as a research project at Berkeley, community-based development since the mid-90s
- Vendor-neutral, commercial support available from multiple companies
- “Postgres Global Development Group”, core team (5 members), release team (3 members), around 20 committers
- Yearly release-cycle, 5 years maintenance per release
- Quarterly patch-releases, whitelisted by Debian SRM
- No copyright assignments, no open-core, no dual licensing
- BSD/MIT-style licence
- Many (also proprietary) forks
PostgreSQL - Main Features

▶ Rock Solid
▶ Good and consistent coverage of the SQL-standard, useful modern SQL extensions
▶ Cost-based query-planner
▶ Large number of extensions and associated projects
▶ Transactional changes to database structure (DDL)
▶ Drivers for many programming languages
▶ Many different procedural languages
▶ Foreign-data-wrapper (FDW) for federated access to many other data sources/databases
▶ Packaged in Debian
New Features Between jessie and stretch - Version 9.5

- INSERT .. ON CONFLICT UPDATE/IGNORE (UPsert)
- Row-Level Security (RLS)
- Block-Range Indexe (BRIN)
- GROUPING SETS, CUBE und ROLLUP for analytical queries
- Rewinding an old master as new standby
- ALTER TABLE ... SET LOGGED / UNLOGGED
- TABLESAMPLE in defined time
- IMPORT FOREIGN SCHEMA for foreign-data-wrappers
- Improved scalability
New Features Between jessie and stretch - Version 9.6

- Parallel query
- Improved VACUUM (automatic freeze map)
- More than one synchronous standby
- Consistent reads from synchronous standbys
- Remote Joins/Sorts/UPDATES/DELETEs in Postgres FDW
- Improved activity overview for locks and vacuum
- Improved scalability

Michael Banck <michael.banck@credativ.de> and Christoph Berg <christoph.berg@credativ.de>
New Features - Version 10

- Logical replication
- Extended parallel query
- Native, declarative partitionierung
- Client-based connection failover
- Quorum-commit for synchronous standbys
- Simplified replication configuration
- Salted Challenge Response Authentication Mechanism (SCRAM)
- Improved and prodution-ready hash-index
- Faster evaluation of expressions
- Multi-column statistics for the query planer
New Extensions / Projects Since jessie

- citus - Sharding and distributed joins extension
- pg-partman - Partition manager
- pg-repack - Reorganize tables with minimal locks
- pgbackrest - Backup & restore
- pldebugger - PL/pgSQL debugger API
- postgresql-mysql-fdw - Foreign data wrapper for MySQL
- sqlsmith - Random SQL query generator and fuzzer

Since stretch:

- pglogical - Logical replication extension
- pgaudit - Audit extension
- bgw-replstatus - Report whether node is master or standby
PostgreSQL in Debian

Simple plan:

- There is Debian
- There is PostgreSQL
- Package it
- Done!
Reality is more complex

- There is Debian sid, stretch, jessie, wheezy
- There is PostgreSQL 9.6, 9.5, 9.4, 9.3, 9.2, 9.1, 9.0, ...
- PostgreSQL major releases have incompatible on-disk format
- Upgrading needs both versions installed in parallel (or lots of disk space and a complicated plan)
Solution: make packages for 9.5, 9.6 etc. co-installable

postgresql-common takes care of creating database clusters in the correct locations

Still, Debian stretch has 9.6 only (jessie has 9.4, wheezy had 9.1)

- Users want to try new PostgreSQL versions
- Users want to upgrade Debian without upgrading PostgreSQL
Apt repository hosting packages of all upstream supported PostgreSQL versions built for all Debian (and Ubuntu) releases

Somewhat a superset of what backports would do

6 PostgreSQL releases:
9.2 9.3 9.4 9.5 9.6 (10)

7 Debian and Ubuntu releases:
wheezy jessie stretch sid trusty xenial zesty

3 architectures: amd64 i386 ppc64el

6x7x3 = 126 targets

Around 160 source packages

Some are built “only” per distribution/architecture (28 targets)

Extension modules depend on the PostgreSQL version as well (126 targets)
PostgreSQL LTS - squeeze-lts

- Ships PostgreSQL-8.4
- Upstream support ended when squeeze-lts began
- LTS stable branch maintained by credativ
- Backported around 250 commits
- 6 releases: 8.4.22lts1 - 8.4.22lts6

- https://github.com/credativ/postgresql-lts
PostgreSQL LTS - wheezy-lts

- Ships PostgreSQL-9.1
- Upstream support ended last September
- Regular security support for wheezy EOL in February 2016
- Only security fixes (so far) backported by credativ this time
- 9.1.24lts1 and 9.1.24lts2 released
- Committed to continue to work on postgresql-9.1 for wheezy-lts
PostgreSQL Package Layout

- Allow multiple major versions to be co-installed
- Instances ("Clusters") are identified by version and cluster name
- Every new instance gets an incremented port, starting from 5432
- Server package is `postgresql-<version>`
- Client package is `postgresql-client-<version>`
- Database directories are in
  /var/lib/postgresql/<version>/<cluster>
- Logile is
  /var/log/postgresql/postgresql-<version>-<cluster>.log
- Configuration files are in
  /etc/postgresql/<version>/<cluster>
  - `postgresql.conf`, `pg_hba.conf`, `pg_ident.conf`
  - `pg_ctl.conf`, `start.conf` (Debian-specific)
postgresql-common Commands

- **pg_wrapper**: Wrapper for PostgreSQL client binaries allowing to select instance
  - `psql --cluster 9.4/main`
  - `PGCLUSTER=9.4/main psql`
  - `~/.postgresqlrc`
  - `/etc/postgresql-common/user_clusters`

- **pg_*cluster**: Debian-specific cluster administration commands
  - `pg_createcluster <version> <name>`
  - `pg_dropcluster <version> <name>`
  - `pg_renamecluster <version> <old name> <new name>`
  - `pg_lsclusters`
  - `pg_ctlcluster <version> <name> <command>`
  - `pg_upgradecluster`

- **pg_conftool**: Read and edit parameters in configuration files
Options for Cluster Creation

- `pg_createcluster` behaviour can be configured in
  `/etc/postgresql-common/createcluster.conf`
  - `create_main_cluster`: Whether to create the default main cluster on package installation
  - `start_conf (auto/manual/disabled)`: Default value for `start.conf`
  - `data_directory`: Default database directory
  - `xlogdir`: Optional external directory for transaction logs
  - `initdb_options`: Option to pass during database initialization, e.g. `--data-checksums`
  - All further config parameters are copied into `postgresql.conf`
**pg_buildext**

- 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`
- Examples in `pg_buildext(1)`
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-foobar
override_dh_installdocs:
    dh_installdocs --all README.*
override_dh_auto_clean:
    +pg_buildext clean build-%v
%
    dh $@

Michael Banck <michael.banck@credativ.de> and <christoph.berg@credativ.de>
Regression tests are nice, but don’t catch packaging errors
Run tests on installed packages as well
\texttt{autopkgtest}
\texttt{postgresql-common}: 1274 tests for \texttt{postgresql-9.4}
1954 for \texttt{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-pltcl-9.4 installed
ok 6 - postgresql-server-dev-9.4 installed
ok 7 - postgresql-contrib-9.4 installed
ok 8 - libecpg-dev installed
ok 9 - logrotate installed
[...]
autopkgtest and pg_builddext

- Run "make installcheck" (via PGXS)
- Simple debian/tests/installcheck

#!/bin/sh
pg_builddext installcheck
autopkgtest and pg_builddext

- Complex debian/tests/installcheck

```bash
#!/bin/sh
set -e
for v in $(pg_builddext 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_builddext installcheck-$v
  )
done
```
PostgreSQL Appliance

- Joint offering with Thomas-Krenn GmbH

PostgreSQL Appliance Dashboard

- Home
- pgAdmin4
- Grafana
- Prometheus
- pgBadger
- pgBackRest
- Cockpit
- Shell
- Documentation
- Support

PostgreSQL Cluster

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Port</th>
<th>Data directory</th>
<th>Service</th>
<th>Log</th>
<th>Report</th>
<th>Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>online 9.6/foobar</td>
<td>5434</td>
<td>/var/lib/postgresql/9.6/foobar</td>
<td>Service</td>
<td>Log</td>
<td>Report</td>
<td>Backup Now</td>
</tr>
<tr>
<td>online 9.6/main</td>
<td>5432</td>
<td>/var/lib/postgresql/9.6/main</td>
<td>Service</td>
<td>Log</td>
<td>Report</td>
<td>Backup Now</td>
</tr>
<tr>
<td>online 9.6/test</td>
<td>5433</td>
<td>/var/lib/postgresql/9.6/test</td>
<td>Service</td>
<td>Log</td>
<td>Report</td>
<td>Backup Now</td>
</tr>
</tbody>
</table>
PostgreSQL Appliance

- pgAdmin4 - Web-Based Administration of PostgreSQL
- Grafana - Monitoring Dashboards
- Prometheus - Monitoring
- pgBadger - Logfile Analysis
- pgBackRest - Backups
- Cockpit - System and Services Administration
- Shell In A Box - Web-Based Terminal Emulator
PostgreSQL Appliance - Grafana Dashboard

Michael Banck <michael.banck@credativ.de> and <christoph.berg@credativ.de>
PostgreSQL Appliance - Grafana Dashboard

CPU Usage (by mode) [5m]

CPU Usage (by core) [5m]

Memory Distribution

Memory Dirty

Swap Usage

Used: Min: 438 MiB Max: 472 MiB Avg: 454 MiB
Buffers: Min: 83 MiB Max: 83 MiB Avg: 83 MiB
Cached: Min: 2.757 GiB Max: 2.772 GiB Avg: 2.761 GiB
Free: Min: 563 MiB Max: 609 MiB Avg: 591 MiB
PostgreSQL Appliance

- Can be deployed via Vagrant
- `elephant-shed-*` packages provide configuration file integration and systemd services

```
deb http://packages.credativ.com/public/postgresql/stretch-test main
https://packages.credativ.com/public/postgresql/aptly.key
```

- Source repository to be released soon
Contact

- Michael Banck <michael.banck@credativ.de>
- https://www.credativ.de
- https://www.credativ.de/postgresql
- https://www.credativ.de/jobs
- https://github.com/credativ/postgresql-lts