Continuous Integration for Commitfests
Testing all the patches all the time

Thomas Munro, PGCon 2018, Ottawa
$ whoami

• PostgreSQL hacker at EnterpriseDB (~3 years)

• Some things I’ve worked on: Parallel Hash Join, various parallel query infrastructure, transition tables for triggers (sous-chef), remote_apply, replay_lag, SKIP LOCKED, various portability stuff
<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Author(s)</th>
<th>Status</th>
<th>Windows Passing</th>
<th>Linux Passing</th>
</tr>
</thead>
<tbody>
<tr>
<td>18/1500</td>
<td>csv output format for psql</td>
<td>Daniel Vérité</td>
<td>patch</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>18/1619</td>
<td>de-deduplicate code in DML execution hooks in postgres_fdw</td>
<td>Ashutosh Bapat</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1591</td>
<td>fix constraint exclusion failure for certain partition key types</td>
<td>Amit Langote</td>
<td>patch</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>18/1620</td>
<td>generalized expression syntax for partition bounds</td>
<td>Tom Lane, Kyotaro Horiguchi, Amit Langote</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1640</td>
<td>generate bootstrap entries for array types</td>
<td>John Naylor</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1536</td>
<td>get rid of StdRdOptions, use individual binary reoptions representation for each...</td>
<td>Nikolay Shaplov</td>
<td>patch</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>18/1631</td>
<td>hostorder and failover_timeout for libpq</td>
<td>Ildar Musin</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1587</td>
<td>kNN for SP-GIST</td>
<td>Nikita Glukhov</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1518</td>
<td>lc_messages parameter doesn't work on Windows</td>
<td>Aleksandr Parfenov</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1635</td>
<td>libpq compression</td>
<td>Konstantin Knizhnik</td>
<td>patch</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1490</td>
<td>macOS Secure Transport SSL Support</td>
<td>Daniel Gustafsson</td>
<td>patch</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

- List of current proposed patches
- Does the patch apply, do the tests pass on Windows, do the tests pass on Linux?
- Recent changes highlighted
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Author(s)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>18/1491</td>
<td>Handling better supported channel binding types for SSL implementations</td>
<td>Michael Paquier</td>
<td></td>
</tr>
<tr>
<td>18/1204</td>
<td>Implement NULL-related checks in object address functions to prevent cache lookups...</td>
<td>Michael Paquier</td>
<td></td>
</tr>
<tr>
<td>18/1642</td>
<td>Fix some error handling for read() and errno</td>
<td>Michael Paquier</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/528</td>
<td>Fix the optimization to skip WAL-logging on table created in same transaction</td>
<td>Heikki Linnakangas, Michael Paquier, Kyotaro Horiguchi</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1633</td>
<td>Make description of heap records more talkative for flags</td>
<td>Michael Paquier</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>18/1609</td>
<td>Simplify final sync in pg_rewind's target folder and add --no-sync</td>
<td>Michael Paquier</td>
<td></td>
</tr>
<tr>
<td>18/1630</td>
<td>Temporary WAL segments files not cleaned up after an instance crash</td>
<td>Michael Paquier</td>
<td></td>
</tr>
</tbody>
</table>

Per author view
<table>
<thead>
<tr>
<th>Pull Request</th>
<th>Title</th>
<th>Author</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>18/669</td>
<td>pgbench - allow to store query results into variables</td>
<td>Fabien Coelho</td>
<td>✗</td>
</tr>
<tr>
<td>18/1388</td>
<td>pgbench - option to build using ppoll() for larger connection counts</td>
<td>doug rady</td>
<td>✔️</td>
</tr>
<tr>
<td>18/1389</td>
<td>verify ALTER TABLE SET NOT NULL by valid constraints</td>
<td>Sergei Kornilov</td>
<td>✔️</td>
</tr>
</tbody>
</table>

```
5839 Build FAILED.
5840 "C:\projects\postgresql\pgsql.sln" (default target) (1) ->
5841 "C:\projects\postgresql\pgbench.vcxproj" (default target) (72) ->
5842 (ClCompile target) ->
5843 src/bin/pgbench/pgbench.c(6002): warning C4033: 'ignore_socket' must return a value
5844 [C:\projects\postgresql\pgbench.vcxproj]
5845 src/bin/pgbench/pgbench.c(6120): warning C4013: 'ppoll' undefined; assuming extern returning int
5846 [C:\projects\postgresql\pgbench.vcxproj]
5847 [C:\projects\postgresql\pgbench.vcxproj]
5848 "C:\projects\postgresql\pgsql.sln" (default target) (1) ->
5849 "C:\projects\postgresql\pgbench.vcxproj" (default target) (72) ->
5850 (ClCompile target) ->
5851 src/bin/pgbench/pgbench.c(6004): error C2027: use of undefined type 'pollfd'
5852 src/bin/pgbench/pgbench.c(6000): error C2036: 'socket_set:*' unknown size
5853 [C:\projects\postgresql\pgbench.vcxproj]
5854 [C:\projects\postgresql\pgbench.vcxproj]
5855 src/bin/pgbench/pgbench.c(6000): error C2036: 'socket_set:*' unknown size
5856 [C:\projects\postgresql\pgbench.vcxproj]
5857 src/bin/pgbench/pgbench.c(6000): error C2037: 'fd' specifies undefined object 'union pollfd'
5858 src/bin/pgbench/pgbench.c(6000): error C2037: left of 'revents' specifies undefined object 'union pollfd'
5859 src/bin/pgbench/pgbench.c(6000): error C2065: 'POLLRDHUP': undeclared identifier
5860 src/bin/pgbench/pgbench.c(6000): error C2065: 'POLLRDHUP': undeclared identifier
6605 t/090_reindexdb.pl ........ ok
6606 t/091_reindexdb_all.pl .... ok
6607 t/100_vacuumdb.pl ......... ok
6608 t/101_vacuumdb_all.pl ...... ok
6609 t/102_vacuumdb_stages.pl .. ok
6610 t/290_connstr.pl .......... ok
6611 All tests successful.```

Linux and Windows logos with Tux.
Command exited with code 1

perl dumprepr.pl

== $path ==

C:/projects/postgresql/src/test/regress/expected/psql.out

Thu May 17 05:10:50 2018

C:/projects/postgresql/src/test/regress/results/psql.out

Thu May 17 05:21:38 2018

**********

### 261,267 ####

border 1

columns 0

expanded off

fieldsep ''

fieldsep_zero on

footer on

format aligned

--- 261,267 ----

border 1

columns 0

expanded off

fieldsep not used

fieldsep_zero off

footer on

format aligned
Motivation
pgsql-hackers@postgresql.org

• ~140 people contributing code
• ~500 people contributing to discussions
• Up to ~250 proposed patches in consideration at a time
4 times a year patches are reviewed and committed in a month-long ‘commitfest’

Patch submission and review is done entirely through the pgsql-hackers, pgsql-bugs, pgsql-committers mailing lists

Patches are tracked through the commitfest.postgresql.org web app; registering a thread in the CF app is approximately like making a ‘pull request’ in many other projects
Patch inflation

Moved  Committed  Returned  Rejected
Welcome, new contributors

Distinct patch authors
How long do patches live?

Age (no. commitfests) of patches that reached final state in CF 2018-03
Reviewer & committer bandwidth is precious
Automatically discoverable problems

- Bitrot: please rebase!
- Other compilers are pickier than yours
- Tests fail (maybe with obscure build options or full TAP tests)
- Portability bugs (endianness, word size, OS, libraries)
- Uninitialised data, race conditions, …
- Documentation is broken
Build farm

• The build farm will find some of these problems automatically

• ... but that happens after commit, and consumes committer time and energy

• People will shout at you — ask me how I know

• Let’s apply some of that sort of automation to proposals, during the review phase
Implementation
This time last year

- Daily cronjob to check for bitrot in time for morning coffee

- Various experiments with executing tests, but … how safe is that?

From: Cron Daemon <munro@asterix>
Subject: Cron <munro@asterix> /home/munro/patches/patchmon.sh

7 out of 8 hunks failed while patching src/backend/libpq/auth.c
Failed to apply /home/munro/patches/ldap-diagnostic-message-v3.patch
1 out of 2 hunks failed while patching configure
1 out of 2 hunks failed while patching configure.in
Failed to apply /home/munro/patches/kqueue-v7.patch
Let’s execute random code from the internet…

What could possibly go wrong?
patch -p1 < foo.patch

- CVE-2018-1000156
- CVE-2016-10713
- CVE-2015-1418
- CVE-2015-1416
- CVE-2015-1395
- CVE-2015-1196
- CVE-2014-9637
- CVE-2010-4651

- patch: runs arbitrary shell commands
- patch: writes to files outside the target source tree
- patch: denial of service
Step 1: Quarantine and apply

1. pristine source tree, patch tools → cloned ZFS filesystem
2. patches
3. Apply patches in jail
4. Push branch to GitHub as commitfest/18/1234
5. Destroy jail, filesystem

github.com/postgresql-cfbot/postgresql
### Overview

- **Default branch**
  - **master** Updated 5 months ago by tglfsfdc

### Active branches

<table>
<thead>
<tr>
<th>Branch</th>
<th>Updated Time</th>
<th>Updated By</th>
<th>Protection Status</th>
<th>Merge Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>commitfest/16/1394</td>
<td>Updated 5 minutes ago by Commitfest B...</td>
<td>853</td>
<td><img src="" alt=" " /></td>
<td><img src="" alt=" " /></td>
</tr>
<tr>
<td>commitfest/16/1389</td>
<td>Updated 10 minutes ago by Commitfest ...</td>
<td>853</td>
<td><img src="" alt=" " /></td>
<td><img src="" alt=" " /></td>
</tr>
<tr>
<td>commitfest/16/1388</td>
<td>Updated 15 minutes ago by Commitfest ...</td>
<td>853</td>
<td><img src="" alt=" " /></td>
<td><img src="" alt=" " /></td>
</tr>
<tr>
<td>commitfest/16/1386</td>
<td>Updated 25 minutes ago by Commitfest ...</td>
<td>853</td>
<td><img src="" alt=" " /></td>
<td><img src="" alt=" " /></td>
</tr>
</tbody>
</table>

[View more active branches]
[CF 16/1386] Atomic pgrename on Windows

This commit was automatically generated by cfbot at commitfest.cputube.org. It is based on patches submitted to the PostgreSQL mailing lists and registered in the PostgreSQL Commitfest application.

This branch will be overwritten each time a new patch version is posted to the email thread or the master branch changes.

Commitfest entry: https://commitfest.postgresql.org/16/1386
Patch(es): https://www.postgresql.org/message-id/CAPpHfds7duGZt%2BPf2GL9qSSVv00ZnjNwqiCPjN7mirDw882tA%40mail.gmail.com
Author(s): Alexander Korotkov

Commitfest Bot committed 26 minutes ago
1 parent 1cc4f53
commit 5c2270755ec14258ba7209a53dc177ddc91552a5

Showing 3 changed files with 77 additions and 3 deletions.

- .travis.yml
- src/backend/postmaster/pgstat.c
- src/port/dirmod.c
Step 2: Build and test

- Many wonderful, generous, free-for-open-source build-bot providers
- Running untrusted code in throw-away virtual machine images is their core business
- travis-ci.org for Ubuntu, macOS
  appveyor.com for Windows
  … there are many more
- Friendly result pages and APIs
How to

• Tell travis-ci.org, appveyor.com, … to watch your github.com, bitbucket.com, … public source repository and build any branch with a control file in it

• Add the control file to your branch (.travis.yml, appveyor.yml etc as appropriate):

  `script: ./configure ... && make -j4 && make check`

• This is a nice way to test your branches before you submit patches, and can send you emails, provide ‘badges’ for your web page, tell your IRC channel, release homing pigeons etc

• This talk is about plugging an old school mailing list workflow into this technology!
cfbot information flow

- git.postgresql.org
- commitfest.postgresql.org
- archives.postgresql.org

GitHub

Travis CI
AppVeyor CI
<table>
<thead>
<tr>
<th>Branch</th>
<th>#</th>
<th>Status</th>
<th>Commit</th>
<th>Label</th>
<th>Build Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>commitfest/16/1479</td>
<td>🧪 14 builds</td>
<td># 28284 started</td>
<td>🧪 -</td>
<td>🔄 efb690</td>
<td>⚠️ unknown</td>
</tr>
<tr>
<td>commitfest/17/1478</td>
<td>🧪 14 builds</td>
<td># 28283 started</td>
<td>🧪 -</td>
<td>🔄 5531ca5</td>
<td>✅ Commitfest Bot</td>
</tr>
<tr>
<td>commitfest/17/1476</td>
<td>🧪 17 builds</td>
<td># 28282 started</td>
<td>🧪 -</td>
<td>🔄 85e4147</td>
<td>✅ Commitfest Bot</td>
</tr>
<tr>
<td>commitfest/17/1474</td>
<td>✔️ 19 builds</td>
<td># 28281 passed</td>
<td>🧪 4 minutes ago</td>
<td>🔄 acbf300</td>
<td>✅ Commitfest Bot</td>
</tr>
<tr>
<td>commitfest/17/1470</td>
<td>✔️ 19 builds</td>
<td># 28280 passed</td>
<td>🧪 8 minutes ago</td>
<td>🔄 94781ee</td>
<td>✅ Commitfest Bot</td>
</tr>
<tr>
<td>commitfest/17/1469</td>
<td>✔️ ✔️</td>
<td># 28279 passed</td>
<td>🧪 -</td>
<td>🔄 f3e1aae</td>
<td>✅ Commitfest Bot</td>
</tr>
</tbody>
</table>
Step 3: Collect results

- CI providers have APIs where you can collect the results
- Collecting them in a small database allows consolidated reporting in one place
- You can also browse results directly at CI websites
Active battles
Windows

- Currently able to run `make check` on appveyor.com CI, but the tablespace test fails so I just exclude it

- Not yet attempting to run `check-world`

- If you know how to fix this, please see me after, I will pay you in beer
Rare transient false negatives

- coverage .gdca files getting trampled on by multiple backends (later GCC will fix that)
- Failure to fetch “winflexbison” from sf.net
- Failure to fetch XSL files from oasis-open.org, sf.net
- Timeout of crash-restart TAP test —undiagnosed!
Plans for the future
## Commitfest 2018-09


### Active patches

<table>
<thead>
<tr>
<th>Patch</th>
<th>Status</th>
<th>Author</th>
<th>Reviewers</th>
<th>Committer</th>
<th>Latest activity</th>
<th>Latest mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix the optimization to skip WAL-logging on table created in same transaction</td>
<td>Needs review</td>
<td>Heikki Linnakangas (heikki), Michael Paquier (michael-kun), Kyoto Horiguchi (horiguti)</td>
<td>Michael Paquier (michael-kun), Kyotaro Horiguchi (horiguti), satyanarayana Nariapuram (snarlap)</td>
<td>2018-04-10 18:17</td>
<td>2018-03-30 01:06</td>
<td></td>
</tr>
<tr>
<td>Fix a bug that can prevent standby from restarting</td>
<td>Needs review</td>
<td>Kyotaro Horiguchi (horiguti)</td>
<td>Michael Paquier (michael-kun)</td>
<td>2018-04-10 18:17</td>
<td>2018-05-14 06:59</td>
<td></td>
</tr>
<tr>
<td>Replication status in logical replication</td>
<td>Ready for Committer</td>
<td>Masahiko Sawada (masahikosawada)</td>
<td>Simon Riggs (simon), Petr Jelinek (pjmodos), vaishnavi prabakaran (vaishnavip)</td>
<td>simon</td>
<td>2018-04-10 18:18</td>
<td></td>
</tr>
</tbody>
</table>
• Run Coverity and other static analysis tools?

• Run Valgrind, Clang asan etc to look for bugs?

• Add a big endian 32 bit non-Linux system for maximum portability bug detection with one stone?

• Display built documentation for review?

• Make Travis/AppVeyor fetch and apply patches themselves?

• Put .travis.yml, .appveyor.yml files in the tree?

• Andreas Seltenreich’s SQL Smith?

• Code coverage report? (that is, reinstate)

• Automated performance testing…?
Questions, ideas?

- Thanks to Andres Freund, Dagfinn Ilmari Mannsåker, Andrew Dunstan, Peter van Hardenberg, Oli Bridgman for ideas and scripting improvements
- Thanks to Travis CI and AppVeyor CI for supporting open source
- Thanks to pgsql-hackers for all the patches