



PostgreSQLコミュニティに飛び込もう ～ PostgreSQLのパッチをレビュー ～

2015年11月14日
株式会社NTTデータ 澤田雅彦

NTT DATA

PostgreSQLのコミュニティ開発って、

- 興味はあるけどなんか難しそう。。英語とか。
- 「コミュニティ開発」=「すごい機能を開発」

と思っていませんか？

新しい機能を作るだけでなく、パッチのレビュー、typoの修正も一つのコミュニティ開発です。

本日は、PostgreSQLコミュニティにおける「**パッチのレビュー方法**」についてできるだけ具体的に解説します。

資料での説明後、実際にパッチをレビューする時間を設けていますので、一緒にやってみましょう。

The eventual committer will do their own review before the patch goes into the code base. The task of a reviewer is to take off load from committers by catching simple problems.

The reviewer's job is not necessarily to guarantee some level of quality, but just to report any problems they are able to find. That task is done if you think the patch is ready for in-depth review from a committer.

レビューアの仕事は、前もって問題を取り除くことで、コミッタの仕事を減らすことです。

レビューアには完璧を求められることはなく、見つけられるだけの問題を見つけることが仕事になります。もしあなたが、そのパッチがコミッタの詳細レビューにたる品質になったと判断したら、そこでレビューは完了します。

https://wiki.postgresql.org/wiki/Reviewing_a_Patch/ja より

パッチをレビューすることで、

- PostgreSQLの最前線に身を置き、最新情報を知れる。
- レビューしたパッチやその周辺機能に(ソースコードレベルで)詳しくなれる。
- 途中の議論も知れるので、なぜその機能が入ったのか？なぜそのような仕様になったのか？残課題はあるか？どのような条件でバグが発生するか？など、その機能(やバグ修正)の知識がより深まる。
- 他人のパッチやコミットされたコードを見ることで、プログラミングスキルが向上する。(PostgreSQL開発における勘所を学べる)
- カンファレンス等でパッチ作成者とコミュニケーションを取りやすくなる。

PostgreSQL開発サイクル

PostgreSQLはどのように開発されている？

Commit Festとは？

PostgreSQLのパッチをレビューをしてみよう

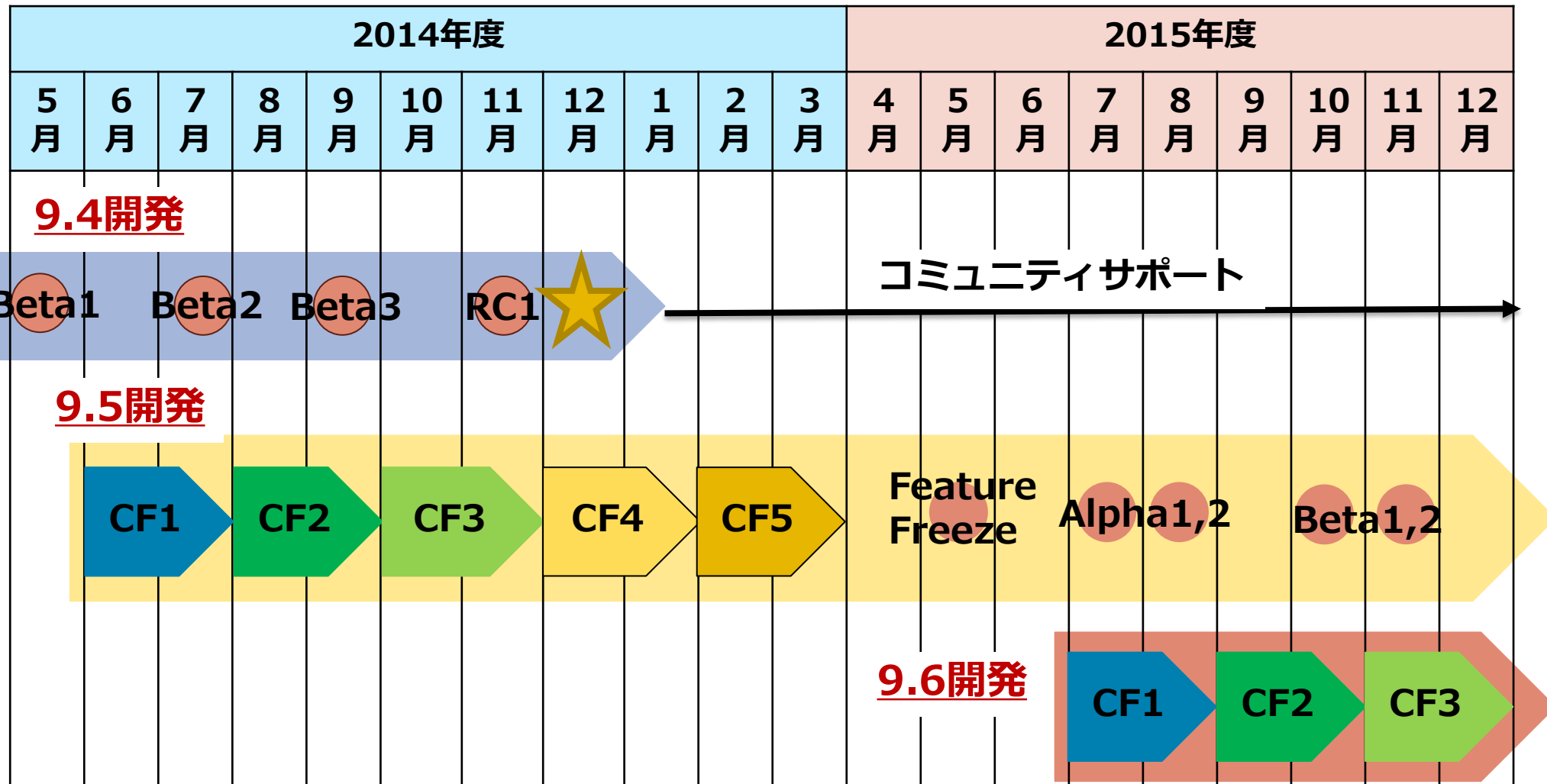
パッチレビューに必要なもの

パッチレビューのやり方

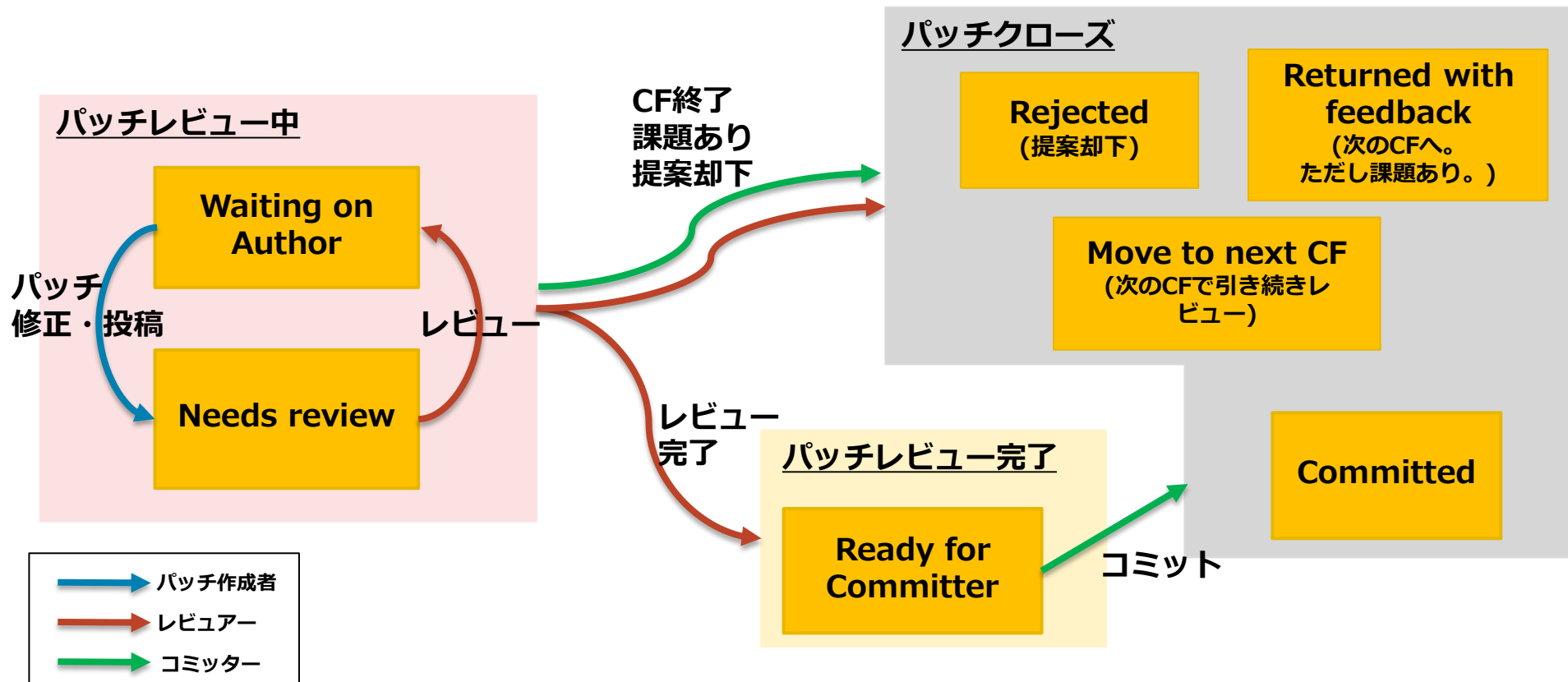
おすすめパッチ紹介



PostgreSQLの開発サイクル



- パッチレビューを集中的に行うサイクル
- 「パッチ作成：1か月、パッチレビュー：1か月」で1サイクル
- 1パッチ、1レビューのルール
- パッチはCommit Fest Appで管理



- アカウントの作成が必要ですので、登録します。
 - <https://www.postgresql.org/account/signup/>
- Commit Fest Appで現在のCFに移動します。
 - <https://commitfest.postgresql.org/>

Commitfest 2015-09

Search/filter

Shortcuts ▾

Status summary: Needs review: 26. Waiting on Author: 11. Ready for Committer: 6. Committed: 30. Moved to next CF: 4. Rejected: 3. Returned with Feedback: 11. Total: 91.

Active patches

Patch	Status	Reviewers	Committer	Latest activity	Latest mail
Bug Fixes					
Improve handling of OOM errors in libpq making process handling for COPY and bind (take 2)	Read	Amit Kapila (amitkapila), Heikki Linnakangas (heikki)		2015-10-15 14:30	2015-10-16 02:00
EPQ recheck failure if ForeignScan/CustomScan replaced built-in Join logic				2015-08-26 04:53	2015-10-21 10:55
Clients					
extends pgbench expressions with functions	Needs review	Fabien Coelho (calvin)	BeomYong Lee (leebeomyong)	2015-08-26 07:42	2015-09-18 18:35
pgbench scripts & stats	Needs review	Fabien Coelho (calvin)		2015-09-10 12:12	2015-10-03 06:11
pgbench with multi-line scripts	Waiting on Author	Kyotaro Horiguchi (horiguti)	Fabien Coelho (calvin)	2015-10-14 08:51	2015-10-14 08:49
group command option for psql	Needs review	Pavel Stehule (okbobcz)		2015-08-27 04:02	2015-09-21 14:46
Miscellaneous					
In-core regression test suite for recovery	Ready for Committer	Michael Paquier (michael-kun)	Simon Riggs (simon), Amir Rohan (amirrohan)	2015-10-10 13:46	2015-10-11 10:31

レビューする
パッチを選択
します

Home / Commitfest 2015-09 / Freeze avoidance of very large table / Log in

Freeze avoidance of very large table

Edit Comment Status

Title	Freeze avoidance of very large table
Topic	Server Features
Created	2015-04-20 07:47:41
Last modified	2015-10-05 10:36:04 (2 weeks, 6 days ago)
Latest email	2015-10-24 08:54:16 (1 day, 16 hours ago)
Status	2015-09: Needs review
Authors	Masahiko Sawada (masahikosawada)
Reviewers	Simon Riggs (simon), Amit Kapila (amitkapila), Fujii Masao (masaofujii)
Committer	
Links	
Emails	Freeze avoidance of very large table. ✕ First at 2015-04-03 05:59:41 by Sawada Masahiko <sawada.mshk at gmail.com> Latest at 2015-10-24 08:54:16 by Masahiko Sawada <sawada.mshk at gmail.com> Latest attachment (000_add_frozen_bit_into_visibilitymap_v19.patch) at 2015-10-22 17:10:11 from Masahiko Sawada <sawada.mshk at gmail.com> + Add annotation

Become reviewer

Attach thread

「Become reviewer」
を押してレビュアーと
して登録します

Home / Commitfest 2015-09 / Freeze avoidance of very large table

Freeze avoidance of very large table

Edit Comment Status

Title Freeze avoidance of very large table

Topic Server Features

Created 2015-04-20 07:47:44

Last modified

Latest

Submitted

Accepted

Reviewed

Committer

Links

Emails

Freeze avoidance of very large table. ✕

First at 2015-04-03 05:59:41 by Sawada Masahiko <sawada.mshk at gmail.com>

Latest at 2015-10-24 08:54:16 by Masahiko Sawada <sawada.mshk at gmail.com>

Latest attachment (000_add_frozen_bit_into_visibilitymap_v19.patch) at 2015-10-22 17:10:11 from Masahiko Sawada <sawada.mshk at gmail.com> +

Add annotation

Status

Open statuses

Needs review

Waiting on Author

Ready for Committer

Closed statuses

Rejected

Returned with feedback

Move to next CF

Committed

Become reviewer

Attach thread

パッチのステータス
を変えるときはここ



PostgreSQLのパッチレビュー

コミュニティへの貢献方法は

- 新規機能の開発
- バグ修正
- バグ報告
- パッチのレビュー

など様々。

もちろん、

- ソースコード内のコメントのtypo修正
- ドキュメントのtypo修正

もコミュニティへの貢献です。

pgsql: Fix typo in comment.

From: Fujii Masao <fujii(at)postgresql(dot)org>
To: pgsql-committers(at)postgresql(dot)org
Subject: pgsql: Fix typo in comment.
Date: 2014-01-20 17:25:51
Message-ID: E1W5Ibf-0006Li-FQ@gemulon.postgresql.org (view [raw](#) or [whole thread](#))
Thread: 2014-01-20 17:25:51 from Fujii Masao <fujii(at)postgresql(dot)org>
Lists: [pgsql-committers](#)

Fix typo in comment.

Sawada Masahiko

Branch

master

Details

<http://git.postgresql.org/pg/commitdiff/5363c7f2bc5e77caf2f308bed93e3e881a9113cf>

Modified Files

src/backend/commands/cluster.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

pgsql: Defend against bad trigger definitions in contrib/lo's lo_manage

From: Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
To: pgsql-committers(at)postgresql(dot)org
Subject: pgsql: Defend against bad trigger definitions in contrib/lo's lo_manage
Date: 2013-11-24 03:47:00
Message-ID: E1VkQey-0004CN-5w@gemulon.postgresql.org (view [raw](#) or [whole thread](#))
Thread: 2013-11-24 03:47:00 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
Lists: [pgsql-committers](#)

Defend against bad trigger definitions in contrib/lo's lo_manage() trigger.

This function formerly crashed if called as a statement-level trigger,
or if a column-name argument wasn't given.

In passing, add the trigger name to all error messages from the function.
(None of them are expected cases, so this shouldn't pose any compatibility
risk.)

Marc Cousin, reviewed by Sawada Masahiko

Branch

master

Details

<http://git.postgresql.org/pg/commitdiff/64d15e424091d8fd2c882356ee4bf07cfc5ff69d>

Modified Files

contrib/lo/lo.c | 14 ++++++
1 file changed, 12 insertions(+), 2 deletions(-)

- 開発環境
 - Linux, Windowsどちらでも
 - PostgreSQLがコンパイル、起動できるように
 - PostgreSQLのソースコードが見れるように
 - gitがあると便利です(例) Mac + VMFusion(CentOS) + git + emacs
- ネット環境
 - ソースコードの入手、MLへの投稿で必要(例) git使うときとメール送るときしかネットを使わないので、テザリング環境でも十分
- メールアドレス
 - コミュニティで議論するために必要(例) gmailアカウントを作る
- PostgreSQLコミュニティのアカウント
 - Commit Fest Appで必要
- C言語の知識
 - C言語で書かれたソースをレビューするので
- 少しの英語力
 - 中学生レベルの英語でOK(だと思います)

PostgreSQLコミュニティでのコミュニケーションはメーリングリストが主体です。

登録しておくべきメーリングリストは3つ

- pgsql-hackers
 - pgsql-hackers@postgresql.org
 - 開発、バグ修正関連の議論はここで行われる。パッチレビューコメントを送るのもここ。
- pgsql-committers
 - pgsql-committers@postgresql.org
 - コミットされた内容が流れる。何がコミットされたかを知る為に必要。
- pgsql-bugs
 - pgsql-bugs@postgresql.org
 - バグレポートの内容や、その後の議論がここで行われる。

<http://www.postgresql.org/community/lists/subscribe/> から今すぐ登録！

レビューするパッチを決める



これまでの議論を追う



パッチをレビューする



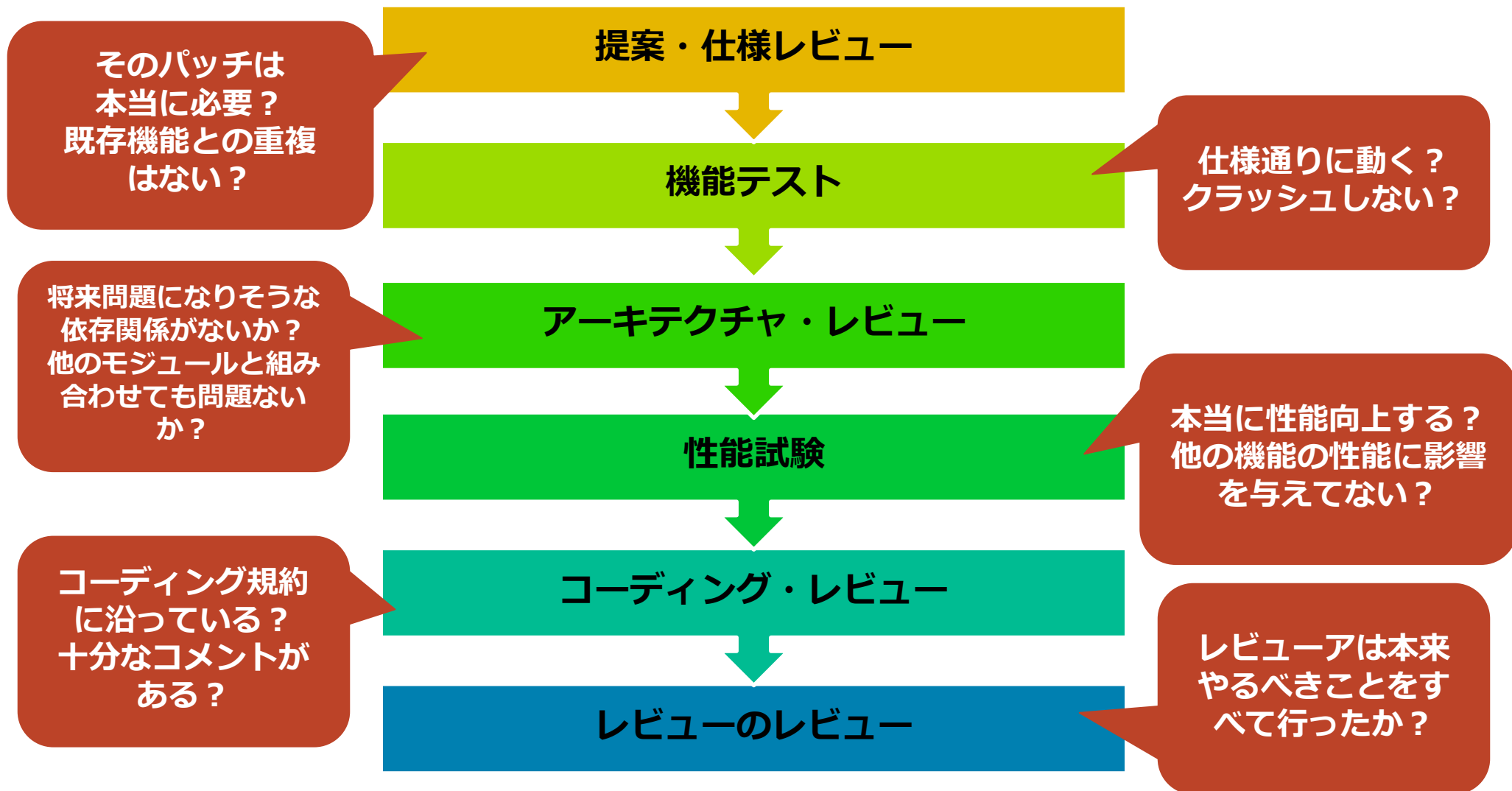
レビューコメントを送る

決め方は人それぞれ

- 機能、分野別
 - プランナ関連、インデックス関連、レプリケーション関連…
 - contribモジュール別、クライアントアプリケーション別…
 - 実業務で関連しそうな所
 - 自分が興味ある所
- 難易度別
 - PostgreSQLのバックエンド機能(プランナ、レプリケーションなど)より、PostgreSQLのフロントエンド機能(psql, pg_dump, pgbenchなど)の方が、読むコード量が少ない場合が多く、取り掛かりやすい
- 規模別
 - 大きい機能をレビューしてみたい！(UPSERT、BRINなど)
 - 簡単なバグ修正

既に議論が進んでいる場合は、これまでの議論をキャッチアップします。

- どんなことを解決する機能なのか？ユースケースは？
- どのようなインタフェースで作られているか？
- 議論の雲行きを見る。
- それまでの議論での合意事項、未合意事項。



Re: Explicit psqlrc

From: Mark Wong <markwkm(at)gmail(dot)com>
To: David Christensen <david(at)endpoint(dot)com>
Cc: gabrielle <gorthx(at)gmail(dot)com>, pgsql-hackers(at)postgresql(dot)org
Subject: Re: Explicit psqlrc
Date: 2010-06-17 03:54:52
Message-ID: 20100616205452.4a3ead73@nqhung (view [raw](#) or [whole thread](#))
Thread: 2010-06-17 03:54:52 from Mark Wong <markwkm(at)gmail(dot)com>
Lists: [pgsql-hackers](#)

Hi David,

At a pdxpug gathering, we took a look at your patch to psql for supporting multiple -f's and put together some feedback:

REVIEW: Patch: support multiple -f options

https://commitfest.postgresql.org/action/patch_view?id=286

==Submission review==

Is the patch in context diff format?

yes

Does it apply cleanly to the current CVS HEAD?

yes

Do all tests pass?

yes

Does it include reasonable tests, necessary doc patches, etc?

- tests: inconclusive

- docs: no:

psql --help does not mention that you can use multiple -f switches; do we want it to? also, no doc update was included in the patch.

==Usability review==

Read what the patch is supposed to do, and consider:

Does the patch actually implement that?

yes

Do we want that?

sure!

Do we already have it?

no

Does it follow SQL spec, or the community-agreed behavior?

NA

Does it include pg_dump support (if applicable)?

NA

Are there dangers?

- subject to the usual Dumb Mistakes (see "have all the bases been covered") Have all the bases been covered?

Scenarios we came up with:

- how does it handle wildcards, eg psql -f *.sql?

Does not - this is a shell issue. psql is designed to take the database as the last argument; giving the -f option a wildcard expands the list, but does not assign multiple -f switches...so if you name one of your files the same as one of your databases, you could accidentally run updates you don't want to do. This is a known feature of psql, and has already bitten these reviewers in the butt on other occasions, so nothing to worry about here.

- how does it handle the lack of a file?

as expected:

```
gabrielle(at)princess~/tmp/bin/::--> ./psql -f
./psql: option requires an argument -- 'f'
```

Try "psql --help" for more information.

- how does it handle a non-existent file?

as expected:

```
gabrielle(at)princess~/tmp/bin/::--> ./psql -f beer.sql
beer.sql: No such file or directory
```

- how does it handle files that don't contain valid sql?

as expected, logs an error & carries on with the next file.

- パッチをHEADにあててみる
- `git diff --check`
- コンパイル
- リグレーションテスト(`make check-world`)
- 仕様通り動くか？
- 性能試験
- 思いつくいろんな使い方をする
 - 境界値テストとかでもOK
 - すごい大量に実行してみたり、すごい長い文字を入れてみたり
 - レプリケーション環境でもちゃんと動くか？ `pg_upgrade`しても問題ないか？
など

ここまでで何か問題があれば、それを開発者に伝えてあげましょう！

Re: TODO : Allow parallel cores to be used by vacuumdb [WIP]

From: Sawada Masahiko <sawada(dot)mshk(at)gmail(dot)com>
To: Dilip kumar <dilip(dot)kumar(at)huawei(dot)com>
Cc: PostgreSQL-development <pgsql-hackers(at)postgresql(dot)org>, Jan Lentfer <Jan(dot)Lentfer(at)web(dot)de>, Euler Taveira <euler(at)timbira(dot)com(dot)br>
Subject: Re: TODO : Allow parallel cores to be used by vacuumdb [WIP]
Date: 2014-06-25 18:07:24
Message-ID: CAD21AoBZxDYkQn7jS_aPbs4teNLJJ-8a-66=SzH4YSuoht2E3Q@mail.gmail.com (view [raw](#) or [whole thread](#))
Thread: 2014-06-25 18:07:24 from Sawada Masahiko <sawada(dot)mshk(at)gmail(dot)com>
Lists: [pgsql-hackers](#)

Hi,

I got following FAILED when I patched v3 to HEAD.

```
$ patch -d. -p1 < ../patch/vacuumdb_parallel_v3.patch
patching file doc/src/sgml/ref/vacuumdb.sgml
Hunk #1 succeeded at 224 (offset 20 lines).
patching file src/bin/scripts/Makefile
Hunk #2 succeeded at 65 with fuzz 2 (offset -1 lines).
patching file src/bin/scripts/vac_parallel.c
patching file src/bin/scripts/vac_parallel.h
patching file src/bin/scripts/vacuumdb.c
Hunk #3 succeeded at 61 with fuzz 2.
Hunk #4 succeeded at 87 (offset 2 lines).
Hunk #5 succeeded at 143 (offset 2 lines).
Hunk #6 succeeded at 158 (offset 5 lines).
Hunk #7 succeeded at 214 with fuzz 2 (offset 5 lines).
Hunk #8 FAILED at 223.
Hunk #9 succeeded at 374 with fuzz 1 (offset 35 lines).
Hunk #10 FAILED at 360.
Hunk #11 FAILED at 387.
3 out of 11 hunks FAILED -- saving rejects to file
src/bin/scripts/vacuumdb.c.rej
```


Re: [PATCH] pg_basebackup: progress report max once per second

From: Sawada Masahiko <sawada(dot)mshk(at)gmail(dot)com>
To: Mika Eloranta <mel(at)ohmu(dot)fi>
Cc: PostgreSQL-development <pgsql-hackers(at)postgresql(dot)org>
Subject: Re: [PATCH] pg_basebackup: progress report max once per second
Date: 2013-11-18 05:53:20
Message-ID: CAD21AoAnaukeO8zyM5CqYc_tNR8x_9w6p1uTjMsBLEca6odw=A@mail.gmail.com (view [raw](#) or [whole thread](#))
Thread: 2013-11-18 05:53:20 from Sawada Masahiko <sawada(dot)mshk(at)gmail(dot)com>
Lists: [pgsql-hackers](#)

```
On Thu, Nov 14, 2013 at 6:27 PM, Mika Eloranta <mel(at)ohmu(dot)fi> wrote:
> On 13 Nov 2013, at 20:51, Mika Eloranta <mel(at)ohmu(dot)fi> wrote:
>
>> Prevent excessive progress reporting that can grow to gigabytes
>> of output with large databases.
>
> Same patch as an attachment.
>
> --
> Mika Eloranta
> Ohmu Ltd. http://www.ohmu.fi/
>
```

I got error with following scenario.

```
$ initdb -D data -E UTF8 --no-locale
/* setting the replication parameters */
$ pg_basebackup -D 2data
Floating point exception
LOG:  could not send data to client: Broken pipe
ERROR:  base backup could not send data, aborting backup
FATAL:  connection to client lost
```

- コーディング規約
 - インデントはスペースは4つ。コメントの書き方もチェック。
 - コーディング規約：<http://www.postgresql.org/docs/current/static/source.html>
- コメント
 - 適切なコメントが書いてあるか？コメントの更新忘れはないか？
- ドキュメントパッチ
 - ドキュメントの内容に間違いはないか？typoはない？
- リグレーションテストパッチ
 - テストすべきテストケースがリグレーションテストにあるか？
- 既存コードの流用
 - 既存のコードをうまく利用して作られているか？無駄なコードはないか？
- 環境依存(Windows対応)
 - Windowsでも動くように。
- レビュアーそれぞれの視点
 - 実装方法がいまいちだと思う。この既存機能と衝突しそう。
 - ここのロジックはもっとシンプルに実装できそう。

- 日本語を書いて英語に訳す？最初から英語で考える？
- 具体的な例を出して説明する
 - For example ...
- 他の人が書いている英語のマネをする
- 文を無理に長くしなくてもよい
 - I applied the patch to HEAD. But I got FAILED. ...
- 箇条書きにする
- 分からないことは素直に聞いてみる
 - PostgreSQLコミュニティの人は優しい人が多い

例えば、

Thank you for updating the patch.

The followings are my review comments and questions.

- * pg_stat_vacuum_progress should have the oid of relation begin vacuumed.
- * progress_message variable in PgBackendStatus is not used at all.
IIRC, progress_message variable is set the description of processing.
- * The progress of VACUUM FULL seems wrong.
When I run VACUUM FULL for a table, I got following progress.
=# SELECT * FROM pg_stat_vacuum_progress;
:
:
- * The vacuum by autovacuum is not displayed.
Could you check this on your environment?

Regards,

--

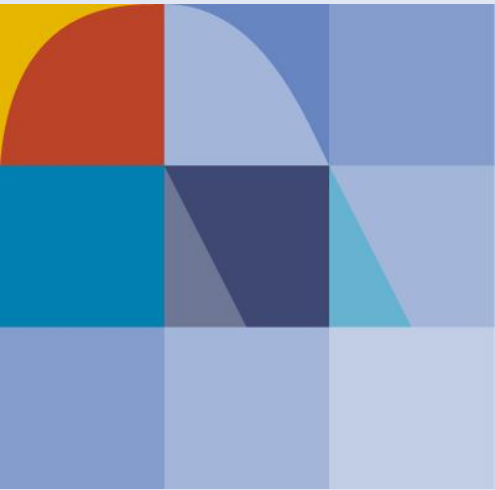
Masahiko Sawada

- 返信は必ずpgsql-hackersをTOに入れる。(作成者個人宛てに送らない)
 - 議論の途中であれば、議論に参加している他の開発者もTOまたはCCに入れる。
 - 基本「reply all」で問題なし。
- メールを送ったら、パッチのステータスを「Waiting on Author」に変更

ここから本当のコミュニティ活動が始まります。
議論→レビューを繰り返し、自分が満足いくまでレビューをしましょう。
また、必要に応じて議論にも参加しましょう！

「"Ready for Committer" is reviewer saying, "Based on my review, I would have committed this if I had been the patch's committer."」

- CF3はすでに始まっています！
- 約100個のパッチが登録
- その中で「Needs review」は約半数(本日時点)
- ちなみに、FreezeMap(PageInfoMap)パッチをレビューしてくれるかた募集中です！
 - これまでの議論の経緯、パッチの説明、全部します。



NTT DATA

Global IT Innovator

- 開発版のソースコードを入手
 - `$ git clone git://git.postgresql.org/git/postgresql.git`
※既に開発版を入手している場合は、「`$ git pull origin master`」等で最新にします。
- パッチをHEADにあててみる
 - `$ cd postgresql`
 - `$ patch -p1 < <パッチへのパス>`
または、
 - `$ git apply <パッチへのパス>`
※`patch`コマンドではあてられるHUNKは全て実際に変更を加えるのに対し、`git apply`ではHUNKが1つでもあたらない場合に、パッチ全体の適用をやめます(つまり、`git apply`ではパッチの適用がアトミック)
- `git diff --check`
 - `$ git diff --check`
※ソースコード末尾に空白等がある場合に検知できます。
- コンパイル
 - `$ make -j 4 > /dev/null`
※こうするとコンパイルWARNINGが出た時だけわかります。ただし、
In file included from gram.y:13990:
scan.c: In function 'yy_try_NUL_trans':
scan.c:10192: 警告: unused variable 'yyg'
は、いつも出るWARNINGなので無視します。
- リグレッションテスト(`make check-world`)
 - `$ make check-world | tee /tmp/check-world.log`
 - `$ less /tmp/check-world.log`
※`make check-world`の結果を目で確認しつつ、ログを「FAILED」でgrepします。