

Authentic 2 - Bug #60035

build cassé, exécution concurrentielle de safe_get_or_create en erreur

23 décembre 2021 09:23 - Paul Marillonnet

Statut:	Fermé	Début:	23 décembre 2021
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
On avait une borne supérieure de 10% d'erreurs dans ce test, qui n'est plus respectée du tout :			
<pre>transactional_db = None, concurrency = 100 def test_safe_get_or_create(transactional_db, concurrency): EMAIL = 'john.doe@example.net' barrier = threading.Barrier(concurrency) users = [] exceptions = [] threads = [] def thread_run(): try: barrier.wait() user, created = safe_get_or_create(User, email=EMAIL, defaults={'email': EMAIL}) except Exception as e: exceptions.append(e) else: if created: users.append(user) finally: connection.close() for _ in range(concurrency): threads.append(threading.Thread(target=thread_run)) threads[-1].start() for thread in threads: thread.join() assert len(users) == 1 assert User.objects.count() == 1 assert all(isinstance(exception, MultipleObjectsReturned) for exception in exceptions) > assert len(exceptions) < (0.1 * concurrency) # 10% of failure is ok with a lot of concurr ency E AssertionError: assert 39 < (0.1 * 100) E + where 39 = len([MultipleObjectsReturned('get() returned more than one User -- it retur ned 86!'), MultipleObjectsReturned('get() retur...than one User -- it returned 5!'), MultipleObjec tsReturned('get() returned more than one User -- it returned 5!'), ...])</pre>			
Demandes liées:			
Lié à Authentic 2 - Development #60645: tests: réduire la variable concurrency		Rejeté	13 janvier 2022

Révisions associées

Révision bfb0226f - 13 janvier 2022 19:45 - Paul Marillonnet

tests: loosen failure threshold value on test_safe_get_or_create (#60035)

Historique

#1 - 23 décembre 2021 09:23 - Paul Marillonnet

- Sujet changé de build cassé, exécution concurrentielle de User.objects.get_or_create en erreur à build cassé, exécution concurrentielle de

safe_get_or_create en erreur

#2 - 23 décembre 2021 12:21 - Paul Marillonnet

Bon je ne reproduis pas en local, je vais regarder plus en détails.

#3 - 23 décembre 2021 12:23 - Benjamin Dauvergne

C'est peut-être une modification à la configuration du postgres local, il faudrait voir avec Guillaume et Pierre si quelque chose a été modifié de ce côté là, ensuite on pourra modifier le ratio. Le test est particulièrement agressif.

#4 - 13 janvier 2022 16:56 - Thomas Noël

Ca bloque les build aujourd'hui, le test pourrait être adapté ? (moins agressif) ?

#5 - 13 janvier 2022 17:28 - Paul Marillonnet

Sur le salon technique il est question de monter ce seuil à 50%. S'attendre à ce que moins de 10% des threads échouent ne me paraît pas trop agressif.

Ce qui se passe dans django est le [QuerySet.create_objects_from_params](#) qui fait remonter une IntegrityError lorsque se produit le cas où la création de l'objet a échoué, et où dans un second temps sa tentative de récupération via une QuerySet nouvellement instanciée échoue elle aussi.

Je serais plutôt pour d'abord voir s'il y a eu des changements dans la configuration des accès concurrentiels à PG, je vais créer un ticket admin sys.

#7 - 13 janvier 2022 17:37 - Guillaume Baffoin

le test est fait sur quel base ? en local sur jenkins ?

#8 - 13 janvier 2022 18:17 - Guillaume Baffoin

je regarde les logs dans jenkins, est ce que ces erreurs sont "normales" (elles sont attendues et testés) ?

sinon on dirait qu'il n'arrive pas à purger la base à la fin.

```
*** /tmp/authentic-2447/pg_virtualenv.9jU6vC/log/postgresql-13-regress.log (last 100 lines) ***
2022-01-13 16:48:46.218 CET [335719] LOG:  démarrage de PostgreSQL 13.5 (Debian 13.5-0+deb11u1) on x86_64-pc-l
inux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit
2022-01-13 16:48:46.218 CET [335719] LOG:  en écoute sur IPv6, adresse « ::1 », port 44441
2022-01-13 16:48:46.218 CET [335719] LOG:  en écoute sur IPv4, adresse « 127.0.0.1 », port 44441
2022-01-13 16:48:46.218 CET [335719] LOG:  écoute sur la socket Unix « /tmp/.s.PGSQL.44441 »
2022-01-13 16:48:46.220 CET [335720] LOG:  le système de bases de données a été arrêté à 2022-01-13 16:48:46 C
ET
2022-01-13 16:48:46.224 CET [335719] LOG:  le système de bases de données est prêt pour accepter les connexion
s
2022-01-13 16:56:50.363 CET [337978] jenkins@test_authentic2 ERREUR:  la valeur d'une clé dupliquée rompt la c
ontrainte unique « authentic2_auth_oidcaccount_user_id_key »
2022-01-13 16:56:50.363 CET [337978] jenkins@test_authentic2 DÉTAIL:  La clé « (user_id)=(3311) » existe déjà.
2022-01-13 16:56:50.363 CET [337978] jenkins@test_authentic2 INSTRUCTION :  INSERT INTO "authentic2_auth_oidc
account" ("created", "modified", "provider_id", "user_id", "sub") VALUES ('2022-01-13T15:56:50.362967+00:0
0'::timestampz, '2022-01-13T15:56:50.362987+00:00'::timestampz, 56, 3311, '4567') RETURNING "authentic2_auth
_oidcaccount"."id"
2022-01-13 17:01:29.731 CET [351208] jenkins@test_authentic2 ERREUR:  la valeur d'une clé dupliquée rompt la c
ontrainte unique « authentic2_auth_fc_account_sub_order_a4c5049e_uniq »
2022-01-13 17:01:29.731 CET [351208] jenkins@test_authentic2 DÉTAIL:  La clé « (sub, "order")=(1234, 0) » exis
te déjà.
2022-01-13 17:01:29.731 CET [351208] jenkins@test_authentic2 INSTRUCTION :  INSERT INTO "authentic2_auth_fc_fc
account" ("created", "modified", "user_id", "sub", "order", "token", "user_info") VALUES ('2022-01-13T16:01:29
.731558+00:00'::timestampz, '2022-01-13T16:01:29.731577+00:00'::timestampz, 6169, '1234', 0, '{"access_token
": "4da777b1-c473-4348-baf1-78fe34cfef82", "id_token": "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJ4eHgiLCJleHAiOiJlNDIwO
Dk3NDksImZlcyI6Imh0dHBzOi8vZmNwLmludGVudGVudEuZGVZLWZyYW5jZWNvbml1Y3QzbnIvIiwibm9uY2UiOiJfRHRFRHRwTjtk2aGV3a01saR
laDZnT1NzYXc3amdzQjgwOVJzTnNlVlR0dDZiIiwic3ViIjoiaMTIzNCJ9.v1lna3qLxv_vzvRhaq2ec2hAg4zOKdjGcSyzrCIXiM"}', '{"sub":
"1234"}') RETURNING "authentic2_auth_fc_fcaccount"."id"
2022-01-13 17:01:36.162 CET [351208] jenkins@test_authentic2 ERREUR:  la valeur d'une clé dupliquée rompt la c
ontrainte unique « authentic2_auth_fc_fcaccount_user_id_order_9be8b543_uniq »
2022-01-13 17:01:36.162 CET [351208] jenkins@test_authentic2 DÉTAIL:  La clé « (user_id, "order")=(6181, 0) »
existe déjà.
2022-01-13 17:01:36.162 CET [351208] jenkins@test_authentic2 INSTRUCTION :  INSERT INTO "authentic2_auth_fc_fc
account" ("created", "modified", "user_id", "sub", "order", "token", "user_info") VALUES ('2022-01-13T16:01:36
.161853+00:00'::timestampz, '2022-01-13T16:01:36.161876+00:00'::timestampz, 6181, '4567', 0, '{"access_token
": "ecfe020f-cbe3-4779-b59c-959c6fcd854b", "id_token": "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJ4eHgiLCJleHAiOiJlNDIwO
Dk3NDksImZlcyI6Imh0dHBzOi8vZmNwLmludGVudGVudEuZGVZLWZyYW5jZWNvbml1Y3QzbnIvIiwibm9uY2UiOiJfRHRFRHRwTjtk2aGV3a01saR
laDZnT1NzYXc3amdzQjgwOVJzTnNlVlR0dDZiIiwic3ViIjoiaMTIzNCJ9.v1lna3qLxv_vzvRhaq2ec2hAg4zOKdjGcSyzrCIXiM"}', '{"family_name": "Fr\u00e9d\u00e9rique", "given_name": "\u0178u\u00fle", "email": "john.doe@examp
```

```
le.com", "sub": "4567"}') RETURNING "authentic2_auth_fc_fcaccount"."id"
2022-01-13 17:03:24.326 CET [364129] jenkins@test_authentic2 ERREUR: n'a pas pu supprimer extension pg_trgm car d'autres objets en dépendent
2022-01-13 17:03:24.326 CET [364129] jenkins@test_authentic2 DÉTAIL: index custom_user_name_gist_idx dépend de classe d'opérateur gist_trgm_ops pour la méthode d'accès gist
2022-01-13 17:03:24.326 CET [364129] jenkins@test_authentic2 ASTUCE : Utilisez DROP ... CASCADE pour supprimer aussi les objets dépendants.
2022-01-13 17:03:24.326 CET [364129] jenkins@test_authentic2 INSTRUCTION : DROP EXTENSION IF EXISTS pg_trgm Dropping cluster 13/regress ...
```

#9 - 13 janvier 2022 19:05 - Paul Marillonnet

Guillaume Baffoin a écrit :

je regarde les logs dans jenkins, est ce que ces erreurs sont "normales" (elles sont attendues et testées) ?

sinon on dirait qu'il n'arrive pas à purger la base à la fin.

[...]

Hmm, les trucs genre l'extension qu'on n'arrive pas à supprimer, je ne pense pas que ce soit des erreurs "normales"...

Mais ce qui m'intéresserait ici est d'avoir les logs SQL pour le test en question.

En local, j'ai tapé des DEBUG et des DEBUG_DB partout, inspecté tous les fichiers de log possibles, je ne trouve rien. Je suspecte que les logs disparaissent avec la base de test test_authentic2 créée lorsque les tests a2 s'exécutent en local, mais je ne suis plus dans un état de fraîcheur suffisant pour creuser plus loin :)

Je reprends demain.

#10 - 13 janvier 2022 19:19 - Paul Marillonnet

Et directement sur jenkins.eo.org, les fichiers /var/log/postgresql/postgresql*.log contiennent les logs d'erreurs SQL de toutes les briques sauf ... authentic :)

Bref, en local je vais creuser pour exploiter les wal directement, parce qu'à part ça je suis à court d'idées.

#11 - 13 janvier 2022 19:35 - Paul Marillonnet

J'ai poussé une branche pour avoir plus d'infos côté Jenkins, et maintenant ça plante sur la ligne d'au dessus !

```
    assert len(users) == 1
    assert User.objects.count() == 1
> assert all(isinstance(exception, MultipleObjectsReturned) for exception in exceptions)
E     assert False
E     + where False = all(<generator object test_safe_get_or_create.<locals>.<genexpr> at 0x7fcc2ea06580>)
```

(sans que je sois parvenu à avoir les logs applicatifs à temps pour savoir de quelle exception exactement il s'agissait).

Des choses restent encore à éclaircir, affaire à suivre.

#12 - 13 janvier 2022 19:49 - Paul Marillonnet

- Fichier 0001-tests-loosen-failure-threshold-value-on-test_safe_ge.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Thomas me dit sur le salon tech qu'il aimerait bien une solution à court-terme pour avoir un paquet à coup sûr au cas où des correctifs sont à appliquer rapidement pendant/après la mât de ce soir.

Patch joint ici, poussé sur wip/60035-wip-alleviate-safe-get-or-create-threshold.

#13 - 13 janvier 2022 20:07 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#14 - 13 janvier 2022 20:15 - Paul Marillonnet

- Statut changé de Solution validée à Résolu (à déployer)

```
commit bfb0226f2f7702178b393968bad510d3a6dff214
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date: Thu Jan 13 19:45:22 2022 +0100
```

```
tests: loosen failure threshold value on test_safe_get_or_create (#60035)
```

#17 - 13 janvier 2022 22:16 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

#18 - 13 janvier 2022 22:59 - Benjamin Dauvergne

Une autre possibilité c'est aussi de diminuer le nombre de threads, je vais ouvrir un ticket dans ce sens.

#19 - 13 janvier 2022 23:08 - Benjamin Dauvergne

- Lié à Development #60645: tests: réduire la variable concurrency ajouté

#20 - 14 janvier 2022 09:55 - Pierre Ducroquet

Paul Marillonnet a écrit :

Et directement sur jenkins.eo.org, les fichiers /var/log/postgresql/postgresql*.log contiennent les logs d'erreurs SQL de toutes les briques sauf ... authentic :)

Bref, en local je vais creuser pour exploiter les wal directement, parce qu'à part ça je suis à court d'idées.

C'est pas une bonne idée ça.

En local tu peux mettre un log_min_duration_statement à 0, voire le spécifier sur une seule base, voire trander ton django pour le faire sur ton test uniquement. Mais le WAL ne t'aidera pas.

#21 - 14 janvier 2022 10:12 - Pierre Ducroquet

Pour revenir sur le fond du problème, je doute très fortement qu'un changement dans PostgreSQL puisse être à l'origine d'un soucis ici. Il n'y a pas vraiment de "configuration des accès concurrents" en jeu, encore moins une configuration qui serait changée ici.

S'agissant par contre d'un code multi-thread, en python, une fluctuation dans l'ordre de lâcher des GIL et d'ordonnement des fils d'exécution peut changer dramatiquement le profil d'exécution, donc l'ordre des requêtes SQL, et ipso facto provoquer ce changement sur cette fonction.

D'où l'intérêt d'avoir un log des requêtes SQL exécutées, et pas autre chose. Tout autre élément nous éloignera de la réalité de la communication entre l'application et PostgreSQL.

#22 - 14 janvier 2022 11:09 - Paul Marillonnet

Pierre Ducroquet a écrit :

Paul Marillonnet a écrit :

Et directement sur jenkins.eo.org, les fichiers /var/log/postgresql/postgresql*.log contiennent les logs d'erreurs SQL de toutes les briques sauf ... authentic :)

Bref, en local je vais creuser pour exploiter les wal directement, parce qu'à part ça je suis à court d'idées.

C'est pas une bonne idée ça.

En local tu peux mettre un log_min_duration_statement à 0, voire le spécifier sur une seule base, voire trander ton django pour le faire sur ton test uniquement. Mais le WAL ne t'aidera pas.

D'ac, j'avais suggéré cette idée un peu au comble du désespoir, je vais tenter ce que tu proposes avec le log_min_duration_statement, merci pour le tuyau. Bon, cela dit, a priori on est parti sur une solution plus radicale, [#60658](#).

#23 - 20 mars 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-tests-loosen-failure-threshold-value-on-test_safe_ge.patch 986 octets 13 janvier 2022

Paul Marillonnet