

w.c.s. - Development #66315

Fonction sql::SqlDataMixin.rebuild_security trop aggressive

16 juin 2022 12:06 - Pierre Ducroquet

Statut:	Fermé	Début:	16 juin 2022
Priorité:	Normal	Echéance:	
Assigné à:	Pierre Ducroquet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
<p>La fonction sql::SqlDataMixin.rebuild_security est assez "violente" dans son fonctionnement. Dans une transaction, elle va réaliser un UPDATE inconditionnel sur chaque formdata d'un formdef. Avec l'introduction de wcs_all_forms sous forme de table maintenue par triggers, ce comportement est exacerbé et peut aboutir à des deadlock, en particulier sur les instances où le serveur PostgreSQL n'a pas des disques aussi bons que les nôtres. Je vois donc deux améliorations à faire sur cette fonction :</p> <ol style="list-style-type: none">1. Ne pas mettre à jour inutilement, en ajoutant des critères sur l'UPDATE.2. Étant donné le code, je pense que l'on peut faire des COMMIT intermédiaires sans craintes, par exemple un commit tous les 100 objets modifiés. <p>Le second correctif a l'avantage de réduire considérablement la durée des locks qui seront pris par la fonction. Actuellement, toutes les lignes se retrouvent verrouillées pour toute la durée de l'opération.</p>			
Demandes liées:			
Lié à w.c.s. - Bug #65416: TransactionRollbackError: ERREUR: Bloquage mortel...		Fermé	19 mai 2022

Révisions associées

Révision 0c08fc01 - 30 juin 2022 13:49 - Pierre Ducroquet

sql: rework the rebuild_security function (#66315)

The function rewrote the whole table without checking if updates were really required. Instead, we now check before updating, and we commit every 100 rows to prevent any long lock

Historique

#1 - 16 juin 2022 12:13 - Pierre Ducroquet

- Lié à Bug #65416: TransactionRollbackError: ERREUR: Bloquage mortel détecté ajouté

#2 - 20 juin 2022 12:57 - Pierre Ducroquet

- Fichier 0001-sql-rework-the-rebuild_security-function-66315.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Ci-joint, un patch qui fait le taf a priori:

- 1) les update ne sont plus systématiques
- 2) en cours d'update, des commits sont faits périodiquement

#5 - 23 juin 2022 18:52 - Frédéric Péters

L'approche minimaliste ici me convient tout à fait.

Sur la forme on fera plutôt

```
for i, formdata in enumerate(formdatas):
```

Aussi must_update est bizarrement nommée dans la mesure où True ou False il y a un UPDATE dans les deux cas; ça serait il me semble mieux en étant renommé en quelque chose comme update_all.

#6 - 23 juin 2022 19:04 - Benjamin Dauvergne

Dans le cas `must_update=False`, vu qu'on a chargé le `formdata` est-ce qu'on ne pourrait pas complètement éviter la requête `UPDATE` en comparant directement la cible avec ce qui a été chargé ?

#7 - 29 juin 2022 14:17 - Pierre Ducroquet

Ci-joint un patch mis à jour.

Par contre lors du rebase sur `origin/main` j'ai une erreur qui est apparue qui ne me semble pas liée à ce patch.

J'ai renommé le `must_update` en `update_all`, comme suggéré. Pour la suggestion de faire la vérification côté Python, je n'ai pas réussi à obtenir un résultat satisfaisant, et ça complique le code. Donc je pense qu'on peut se contenter du patch actuel.

#8 - 29 juin 2022 14:32 - Pierre Ducroquet

- Fichier `0001-sql-rework-the-rebuild_security-function-66315.patch` ajouté

#9 - 29 juin 2022 14:50 - Frédéric Péters

Par contre lors du rebase sur `origin/main` j'ai une erreur qui est apparue qui ne me semble pas liée à ce patch.

N'hésite pas dans ces situations à lancer un nouveau build (ça peut juste être jenkins un peu trop chargé et des tests qui dépendent du timing).

#10 - 29 juin 2022 16:35 - Pierre Ducroquet

Frédéric Péters a écrit :

Par contre lors du rebase sur `origin/main` j'ai une erreur qui est apparue qui ne me semble pas liée à ce patch.

N'hésite pas dans ces situations à lancer un nouveau build (ça peut juste être jenkins un peu trop chargé et des tests qui dépendent du timing).

En effet, c'était juste ça, j'ignorais qu'on avait des tests dépendant du timing

#11 - 30 juin 2022 12:22 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *Solution validée*

Dans `wcs/formdata.py` le paramètre `force_update` s'appelle toujours comme ça. Sinon c'est bon.

#12 - 30 juin 2022 13:50 - Pierre Ducroquet

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

Merci, mergé (avec la correction du nom de paramètre dans `formdata.py`)

```
commit 0c08fc01ecf8413946ddb2e899dcb16f970132bd (HEAD -> main, origin/main, origin/HEAD, wip/66315-lighten-rebuild_security)
```

```
Author: Pierre Ducroquet <pducroquet@entrouvert.com>
```

```
Date: Thu Jun 16 12:32:36 2022 +0200
```

```
sql: rework the rebuild_security function (#66315)
```

```
The function rewrote the whole table without checking if updates were really required. Instead, we now check before updating, and we commit every 100 rows to prevent any long lock
```

#13 - 30 juin 2022 22:14 - Transition automatique

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

#14 - 04 septembre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

<code>0001-sql-rework-the-rebuild_security-function-66315.patch</code>	3,91 ko	20 juin 2022	Pierre Ducroquet
<code>0001-sql-rework-the-rebuild_security-function-66315.patch</code>	3,92 ko	29 juin 2022	Pierre Ducroquet