

w.c.s. - Bug #7818

problème sur l'index workflow_roles

10 juillet 2015 14:24 - Frédéric Péters

Statut:	Fermé	Début:	10 juillet 2015
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
<p>Il arrive (??? sur des workflows toujours trop complexes pour en tirer quelque chose) que l'index workflow_roles ne soit plus correct, ce qui fait que limiter un listing à un statut particulier ne fonctionne pas (zéro entrées, là où le listing global affichait des demandes dans le statut donné).</p> <p>(occurrence du jour, https://demarches.vincennes.fr/backoffice/management/completer-un-dossier-creer-avec-un-agent-d-accueil-envoi-de-pieces-justificatives/)</p>			

Révisions associées

Révision fa801967 - 15 juillet 2015 18:29 - Frédéric Péters

storage: add lock around index update (#7818)

Historique

#1 - 11 juillet 2015 11:53 - Frédéric Péters

- Fichier 0001-storage-add-lock-around-index-update-7818.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

Tout à fait incapable de reproduire les conditions qui pourraient amener ça, j'ai pris un peu de recul et considéré le seul truc qui restait, qu'il y ait des écritures concurrentes sur les index, ce qui peut quand même pas trop difficilement arriver avec les hashed_indexes, il me semble.

Du coup j'ai ajouté un test avec quelques forks et facilement il y a tout qui explose, d'une part la génération d'id, exécutée lors de la création de l'objet et non lors de l'écriture, ça laisse une opportunité énorme à d'autres objets pour acquérir le même id (et boum que je t'écrase). D'autre part, les index.

Pour le premier problème j'ai retiré la génération précoce de l'id (il y a dans le storage.py du code pour créer le fichier avec (os.O_CREAT | os.O_EXCL) pour éviter ça, autant l'utiliser).

Pour le second problème j'ai regardé un peu ce qu'il existait comme bibliothèque de lock et j'ai importé le code de la plus agréable.

#2 - 13 juillet 2015 12:19 - Benjamin Dauvergne

- Fichier 0001-use-gdbm-to-store-hashed-indexes.patch ajouté

Il me semble qu'avec le atomic_write ça ne devrait jamais arriver qu'on lise dans le fichier d'un StorableObject pendant que quelqu'un y écrit, par contre c'est possible avec les fichiers pickle produits pour _hashed_indexes vu que ça n'utilise pas atomic_write pour sa mise à jour. À un moment on pourrait même remplacer cela par un fichier GDBM ou SQLite qui gèrent eux mêmes les locks entre lecture et écriture.

J'ai bidouillé un support gdbm qui passe ton test (mais a d'autres problèmes)... SQLite serait peut-être plus sûr.

#3 - 13 juillet 2015 12:35 - Frédéric Péters

Ça détourne la conversation; il y a un patch avec le lock, est-ce que ça corrige le soucis présent ? Sans doute.

Pour refaire les trucs, sans passer par postgresql, tu crées un autre ticket ?

#4 - 13 juillet 2015 14:17 - Benjamin Dauvergne

Le lock corrige le bug (même sans fsync il est impossible qu'un update_indexes() ne lise pas la mise à jour précédente, sauf crash disque), mais aussi à mon avis utiliser atomic_write aurait suffi en plus de la suppression d'une allocation précoce de l'id qui est le principal problème je pense.

#5 - 15 juillet 2015 18:28 - Thomas Noël

Ack pour 0001-storage-add-lock-around-index-update-7818.patch

#6 - 15 juillet 2015 18:29 - Frédéric Péters

- *Statut changé de En cours à Résolu (à déployer)*

```
commit fa801967ef22450e3802a24d35e37ce954190516
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sat Jul 11 11:44:03 2015 +0200
```

```
storage: add lock around index update (#7818)
```

#7 - 26 août 2015 13:53 - Thomas Noël

- *Statut changé de Résolu (à déployer) à Fermé*

Fichiers

0001-storage-add-lock-around-index-update-7818.patch	11,2 ko	11 juillet 2015	Frédéric Péters
0001-use-gdbm-to-store-hashed-indexes.patch	7,93 ko	13 juillet 2015	Benjamin Dauvergne