

w.c.s. - Development #22383

migrate : séparer les deux étapes (migrations / réindexations)

08 mars 2018 15:29 - Thomas Noël

Statut:	Fermé	Début:	08 mars 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Pour ne pas qu'une mise à jour bloque la mise à jour trop longtemps.			
Il faut retirer du sql::migrate.py :			
<pre>for user_id in SqlUser.keys(): SqlUser.get(user_id).store()</pre>			
et			
<pre>from wcs.formdef import FormDef for formdef in FormDef.select(): for formdata in formdef.data_class().select(): formdata.store()</pre>			
et poser cela dans des commandes "rebuild-indexes --users --formdata ..." ?			
Demandes liées:			
Lié à w.c.s. - Development #23305: sql: permettre à select() de fonctionner c...			Fermé 20 avril 2018

Révisions associées

Révision 3961764a - 20 avril 2018 14:38 - Thomas Noël

sql: delay re-index operations on first cron (#22383)

Historique

#2 - 21 mars 2018 11:05 - Thomas Noël

- Fichier 0001-misc-do-not-rebuild-fts-by-default-on-migrate-comman.patch ajouté
- Patch proposed changé de Non à Oui

Proposition :

- par défaut migrate ne fait jamais les rebuild FTS
- mais on peut les lancer "quand on veut" avec --rebuild-fts-user et --rebuild-fts-formdata (y compris, en fait, en dehors d'une migration)

Une proposition alternative pourrait être d'avoir une nouvelle commande « rebuild-fts » ?

#3 - 21 mars 2018 11:31 - Frédéric Péters

Ces étapes post-migrations de schéma (et il y a aussi la création de nouveaux index, que tu laisses exécutée au démarrage), je pense vraiment qu'elles doivent être exécutées de manière automatique.

Et sur la forme précise, pour ne pas partir dans des forks, je me dis que ça pourrait être déclaré comme job cron (interne, register_cronjob), qui regarde une ligne supplémentaire de la table wcs_meta et lance la "post-migration" si nécessaire.

#4 - 21 mars 2018 12:06 - Thomas Noël

Frédéric Péters a écrit :

Ces étapes post-migrations de schéma (et il y a aussi la création de nouveaux index, que tu laisses exécutée au démarrage), je pense vraiment qu'elles doivent être exécutées de manière automatique.

Je ne comprends pas. C'est justement ce qui nous gêne actuellement, non ?

#5 - 21 mars 2018 12:12 - Frédéric Péters

Ce qui (me) gêne c'est que w.c.s. doit attendre qu'elles aient été exécutées pour être up.

#6 - 21 mars 2018 12:13 - Thomas Noël

Frédéric Péters a écrit :

Ce qui (me) gêne c'est que w.c.s. doit attendre qu'elles aient été exécutées pour être up.

Ok, la prochaine fois je lirais tes commentaires jusqu'au bout ! :-)

#7 - 26 mars 2018 01:59 - Thomas Noël

- Patch proposed changé de Oui à Non

Patch à refaire, une simple première étape serait de forker avant de lancer les "re-indexations" potentiellement longues et qui peuvent être lancées après le démarrage de wcs, ie après les mises à jour des schémas, qui doivent rester "bloquantes" avant le démarrage.

#8 - 04 avril 2018 01:31 - Thomas Noël

Voici mon plan :

- sur migrate :
 - retirer de sql.migrate les reindexation user et formdata
 - les remplacer par des valeurs dans wcs_meta : need_reindex_user et need_reindex_formdata, qui passent à value=1 quand une reindexation doit avoir lieu
- au lancement du processus démon
 - parcourir les sites qui ont need_reindex_* à 1, les ré-indexer et passer la valeur à 0 quand c'est fait

Mon soucis pour l'instant c'est cette seconde étape... je ne sais pas où la coder :/ Pour l'instant ce que j'arrive à imaginer c'est une nouvelle commande "reindex" qui se lancerait en daemon, et serait donc lancée après migrate par systemd, mais ne serait pas bloquant. Toute idée est bienvenue.

#9 - 09 avril 2018 21:04 - Thomas Noël

- Fichier 0001-sql-delay-re-index-operations-on-first-cron-22383.patch ajouté

- Statut changé de Nouveau à En cours

Première approche simple pour démarrer "assez vite" sans les FTS : on passe celles-ci dans le premier cron.

Le patch joint présente une implémentation (test_sql doivent être élargis et adaptés, ne passent plus en l'état).

J'utilise deux clés "reindex_user" et "reindex_formdata" dans wcs_meta, qui seront posées à "needed" par sql.migrate si besoin. Lors de chaque cron (et notamment le premier) on lance en premier lieu les re-indexations qui sont à "needed". A la fin du process on met "done" dans la clé.

Si ça crashe, la clé est toujours à "needed" tant que la ré-indexation n'est pas terminée : un futur cron relancera la reindexation. Par sécurité (lock de cron en rade, cron sur deux machines distinctes, etc.) on pourrait passer la clé à "in-progress" pendant qu'a lieu la ré-indexation, mais ça compliquerait les rattrapages en cas de pépin.

#10 - 20 avril 2018 11:32 - Thomas Noël

- Fichier 0001-sql-delay-re-index-operations-on-first-cron-22383.patch ajouté

- Patch proposed changé de Non à Oui

Voilà, avec les tests où on vérifie que la ré-indexation a bien eu lieu après le premier cron.

#11 - 20 avril 2018 11:58 - Frédéric Péters

Je déplacerais les changements sur les fixtures dans un premier commit.

Petite hésitation sur l'état qui peut juste être needed/done. (mais ça doit juste être une crainte d'exécutions parallèles qui n'est pas fondée)

Pour les utilisateurs on passe par une récupération une par une des lignes, plutôt que le .select(); je peux comprendre ça sur l'idée que le select() n'est pas un itérateur mais je serais pour le permettre (et pour ne pas prendre de risque sur le code actuel, que ça soit uniquement en présence d'un iterator=True en paramètre, défaut à False).

#12 - 20 avril 2018 12:49 - Thomas Noël

- Lié à Development #23305: sql: permettre à select() de fonctionner comme un itérateur ajouté

#13 - 20 avril 2018 14:04 - Thomas Noël

- Fichier 0002-sql-delay-re-index-operations-on-first-cron-22383.patch ajouté

- Fichier 0001-tests-use-sql_mode-publisher-creation-in-test_sql.patch ajouté

Voilà, avec un select itérable. J'ai aussi renforcé un poil les tests en vérifiant que la migration ne joue pas la re-indexation (ie qu'elle est bien faite uniquement après le premier cron).

#14 - 20 avril 2018 14:33 - Frédéric Péters

Ok, go.

#15 - 20 avril 2018 14:39 - Thomas Noël

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

```
commit 3961764ac9300727a6d22376774b6df460dd859b (HEAD -> master, origin/master, origin/HEAD)
```

```
Author: Thomas NOEL <tnoel@entrouvert.com>
```

```
Date: Fri Apr 20 12:17:55 2018 +0200
```

```
sql: delay re-index operations on first cron (#22383)
```

```
commit 0b56cd640ef7ecc8f20d5bfe734fa8fb0b98f4ba
```

```
Author: Thomas NOEL <tnoel@entrouvert.com>
```

```
Date: Fri Apr 20 12:17:39 2018 +0200
```

```
tests: use sql_mode publisher creation in test_sql
```

#16 - 23 décembre 2018 14:42 - Frédéric Péters

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

Fichiers

0001-misc-do-not-rebuild-fts-by-default-on-migrate-comman.patch	5,83 ko	21 mars 2018	Thomas Noël
0001-sql-delay-re-index-operations-on-first-cron-22383.patch	4,32 ko	09 avril 2018	Thomas Noël
0001-sql-delay-re-index-operations-on-first-cron-22383.patch	7,41 ko	20 avril 2018	Thomas Noël
0002-sql-delay-re-index-operations-on-first-cron-22383.patch	6,15 ko	20 avril 2018	Thomas Noël
0001-tests-use-sql_mode-publisher-creation-in-test_sql.patch	1,95 ko	20 avril 2018	Thomas Noël