w.c.s. - Development #60469

Requête lente récurrente sur WCS

10 janvier 2022 17:26 - Pierre Ducroquet

Statut:	Fermé	**Début:**	10 janvier 2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Pierre Ducroquet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

En regardant les logs nouvellement mis en place sur lille-prod, je me rends compte qu'une requête est souvent réalisée sous une forme qui ne peut être indexée (et qui de toute façon utilise des données non indexées).

Requête sur lille-prod:

```
SELECT id FROM formdata_113_demander_mon_pass_lillemoi WHERE '906c88eff7e047c5af891cf2e6b19c64' =
ANY (concerned_roles_array);
```

=> 200 à 300ms.

Or, on peut faire mieux:

```
create index concurrently test_pierre on formdata_113_demander_mon_pass_lillemoi using gin(
concerned_roles_array);
SELECT id FROM formdata_113_demander_mon_pass_lillemoi WHERE concerned_roles_array @> ARRAY[
'906c88eff7e047c5af891cf2e6b19c64'];
```

Le select prend alors une milliseconde environ. L'index pèse 136kB pour une table de 364MB, ce qui est tout à fait acceptable je pense.

Je ne maîtrise pas du tout wcs, mais j'ai pu identifier comment corriger la requête, il me manque juste la partie migrations:

```
diff --git a/wcs/sql.py b/wcs/sql.py
index bb8cbe52..78f2213d 100644
--- a/wcs/sql.py
+++ b/wcs/sql.py
@@ -2329,7 +2329,7 @@ class SqlDataMixin(SqlMixin):
             value = str(value)

         if '%s_array' % index in [x[0] for x in cls._table_static_fields]:
-            sql_statement = '''SELECT id FROM %s WHERE %%(value)s = ANY (%s_array)''' % (
+            sql_statement = '''SELECT id FROM %s WHERE %s_array @> ARRAY[%%(value)s]''' % (
                 cls._table_name,
                 index,
             )
```

Révisions associées

Révision 5edf9301 - 10 janvier 2022 18:01 - Pierre Ducroquet

Add and use indexes on concerned_roles_array and actions_roles_array (#60469)

After reading some slow queries log in lille-prod, I noticed these two columns
had no usable index (and their queries were not able to use them anyway).
This patch fixes it.

Révision 0e8868dd - 14 janvier 2022 15:43 - Pierre Ducroquet

sql: add and use indexes on concerned_roles_array and actions_roles_array (#60469)

After reading some slow queries log in lille-prod, I noticed these two columns
had no usable index (and their queries were not able to use them anyway).
This patch fixes it.

Historique

#1 - 10 janvier 2022 17:38 - Frédéric Péters

Je ne maîtrise pas du tout wcs, mais j'ai pu identifier comment corriger la requête, il me manque juste la partie migrations:

Il y a une méthode do_formdef_indexes pour la création des index.

Dans wcs/sql.py, def migrate() reprend l'ensemble des migrations, il y a un niveau global SQL_LEVEL, à monter, et ça détermine ensuite ce qu'il y aura à exécuter, ici ça donnerait a priori

```
@@ -752,6 +752,10 @@ def do_formdef_indexes(formdef, created, conn, cur, concurrently=False):
                 % {'create_index': create_index, 'table_name': table_name, 'attr': attr}
             )

+    cur.execute(
+        # quelque chose
+    )
+

 @guard_postgres
 def do_user_table():
@@ -3499,7 +3503,7 @@ def get_period_total(
 # latest migration, number + description (description is not used
 # programmaticaly but will make sure git conflicts if two migrations are
 # separately added with the same number)
-SQL_LEVEL = (55, 'update full text normalisation (switch to unidecode)')
+SQL_LEVEL = (56, 'add gin indexes to concerned_roles_array')

 def migrate_global_views(conn, cur):
@@ -3638,7 +3642,7 @@ def migrate():
         # 51: add index on formdata blockdef fields
         # 55: update full text normalisation (switch to unidecode)
         set_reindex('formdata', 'needed', conn=conn, cur=cur)
-    if sql_level < 46:
+    if sql_level < 56:
         from wcs.carddef import CardDef
         from wcs.formdef import FormDef

@@ -3646,6 +3650,7 @@ def migrate():
         # 35: add indexes on formdata(receipt_time) and formdata(anonymised)
         # 36: add index on formdata(user_id)
         # 45 & 46: add index on formdata(status)
+        # 56: add gin indexes to concerned_roles_array
         for formdef in FormDef.select() + CardDef.select():
             do_formdef_indexes(formdef, created=False, conn=conn, cur=cur)
     if sql_level < 32:
```

(aussi, peut-être ça vaut le coup de mettre le même index sur actions_roles_array)

#2 - 10 janvier 2022 18:02 - Pierre Ducroquet

- Fichier 0001-Add-and-use-indexes-on-concerned_roles_array-and-act.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Assigné à mis à Pierre Ducroquet

- Patch proposed changé de Non à Oui

#3 - 10 janvier 2022 18:13 - Thomas Noël

Sur le contenu du patch, ça me semble bel et bon.

Pour le message de commit, indique que ça concerne le module sql. Et on met pas de majuscule, c'est comme ça, donc plutôt :

```
sql: add and use indexes on concerned_roles_array and actions_roles_array (#60469)
```

Ensuite, créer une branche wip/60469-add-indexes (ce qui est important c'est que son nom commence par wip/NNNNN où NNNNN est le numéro du ticket) et y pousser ce patch.

Jenkins va alors faire tourner les tests sur cette branche (ça sera visible dans https://jenkins.entrouvert.org/job/wcs-wip/). En cadeau il y aura en bas de ce ticket un petit javascript qui en montrera le résultat (vert/rouge).

#4 - 10 janvier 2022 19:39 - Thomas Noël

- Statut changé de Solution proposée à En cours

Crash parce que do_formdef_indexes est appelé sur les nouvelles tables alors que les colonnes n'existent pas encore. Désolé de ne pas avoir vu ça.

Sur le salon, proposition de Fred (à intégrer au patch) :

```
@@ -540,6 +540,7 @@ def do_formdef_tables(formdef, conn=None, cur=None, rebuild_views=False, rebuild
                        AND table_name = %s''',
         (table_name,),
     )
+    new_table = False
     if cur.fetchone()[0] == 0:
         cur.execute(
             '''CREATE TABLE %s (id serial PRIMARY KEY,
@@ -564,7 +565,7 @@ def do_formdef_tables(formdef, conn=None, cur=None, rebuild_views=False, rebuild
                                    formdata_id integer REFERENCES %s (id) ON DELETE CASCADE)'''
             % (table_name, table_name)
         )
-        do_formdef_indexes(formdef, created=True, conn=conn, cur=cur)
+        new_table = True

     cur.execute(
         '''SELECT column_name FROM information_schema.columns
@@ -705,6 +706,9 @@ def do_formdef_tables(formdef, conn=None, cur=None, rebuild_views=False, rebuild
         # them even if not asked to.
         redo_views(conn, cur, formdef, rebuild_global_views=rebuild_global_views)

+    if new_table:
+        do_formdef_indexes(formdef, created=True, conn=conn, cur=cur)
+
     if own_conn:
         conn.commit()
         cur.close()
```

#5 - 10 janvier 2022 19:52 - Pierre Ducroquet

- Fichier 0001-sql-add-and-use-indexes-on-concerned_roles_array-and.patch ajouté

- Statut changé de En cours à Solution proposée

#6 - 10 janvier 2022 20:13 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Cette fois ci je valide en étant totalement sûr de moi (tousse tousse).

Attention cependant, ne pas pousser dans main, on est en "semaine de release".

Je suis du genre conservateur et je préfère éviter, et attendre plutôt vendredi ou lundi (ie après la release de ce 2ème jeudi du mois). Sauf si Frédéric est plus confiant que moi et a envie de ce patch right now (mais il est largement aussi conservateur que moi).

#7 - 10 janvier 2022 20:15 - Frédéric Péters

Yep une fois que c'est bon dans jenkins on attend vendredi.

#8 - 10 janvier 2022 21:00 - Pierre Ducroquet

- Fichier 0001-sql-add-and-use-indexes-on-concerned_roles_array-and.patch ajouté

- Statut changé de Solution validée à Solution proposée

J'ai changé le nommage de l'index, c'est trop long actuellement. Puis j'en ai profité pour utiliser "IF NOT EXISTS", c'est présent depuis PostgreSQL 9.5 tout de même...
Comme le nom de la table commence par un discriminant, j'ai mis le nom de la table en dernier élément pour que comme ça ce soit ce nom qui soit tronqué.

#9 - 11 janvier 2022 11:37 - Thomas Noël

Ok pour la longueur, bien vu

Pour pinailler, sur le "IF NOT EXISTS", comme on ne l'utilise pas encore ailleurs, par mimétisme j'aurais bien vu le code rester avec un bêbête « if ... not in existing_indexes: » dans un premier temps.

... et donc un autre ticket-patch "utiliser IF NOT EXISTS dans dans la création des index" (do_formdef_indexes, do_loggederrors_table).

#10 - 11 janvier 2022 11:45 - Pierre Ducroquet

Thomas Noël a écrit :

> Ok pour la longueur, bien vu
>
> Pour pinailler, sur le "IF NOT EXISTS", comme on ne l'utilise pas encore ailleurs, par mimétisme j'aurais bien vu le code rester avec un bêbête « if ... not in existing_indexes: » dans un premier temps.
>
> ... et donc un autre ticket-patch "utiliser IF NOT EXISTS dans dans la création des index" (do_formdef_indexes, do_loggederrors_table).

J'ai choisi de le faire comme ça pour ne pas réimplémenter en Python la limitation en longueur des noms de relations dans PG. Je prévoyais de faire un patch ultérieur pour simplifier la création des index et utiliser IF NOT EXISTS à la place. Après si vous préférez qu'on fasse par mimétisme d'abord puis qu'on modernise tout d'un coup... As you wish (lorraine).

#11 - 11 janvier 2022 14:21 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Pierre Ducroquet a écrit :

> J'ai choisi de le faire comme ça pour ne pas réimplémenter en Python la limitation en longueur des noms de relations dans PG. Je prévoyais de faire un patch ultérieur pour simplifier la création des index et utiliser IF NOT EXISTS à la place. Après si vous préférez qu'on fasse par mimétisme d'abord puis qu'on modernise tout d'un coup... As you wish (lorraine).

C'est malin d'écrire ça alors que j'ai faim.

Ok, allons-y ainsi, ta méthode est la bonne. Je vais un autre ticket pour moderniser les autres. Je pense qu'on est bien en 9.6 partout donc ça va passer.

(encore une fois, juste pour rappel, attendre vendredi pour pousser cela)

#12 - 11 janvier 2022 14:23 - Thomas Noël

Thomas Noël a écrit :

> Je vais un autre ticket pour moderniser les autres.

Et ça, c'est fait → #60507

#13 - 14 janvier 2022 13:45 - Pierre Ducroquet

Thomas Noël a écrit :

> (encore une fois, juste pour rappel, attendre vendredi pour pousser cela)

Du coup, on est vendredi, je peux pousser dans main ?

#14 - 14 janvier 2022 13:46 - Frédéric Péters

yes.

#15 - 14 janvier 2022 15:44 - Pierre Ducroquet

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

Et c'est pushèd

#16 - 16 janvier 2022 16:16 - Frédéric Péters

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

#17 - 17 janvier 2022 09:53 - Pierre Ducroquet

- Statut changé de Solution déployée à Fermé

On a diminué considérablement le nombre de requêtes de plus de 200ms sur l'instance lille-prod (où j'avais observé le problème initialement).
Il y a d'autres cibles faciles qui apparaissent désormais, pour de futurs tickets.

#18 - 17 janvier 2022 12:15 - Emmanuel Cazenave

Pierre Ducroquet a écrit :

> Et c'est pushèd

Une fois que tu as poussé, notre usage est d'inclure dans le ticket une référence au commit, façon #60675#note-6.

#19 - 17 janvier 2022 12:22 - Pierre Ducroquet

```
commit 0e8868dd5dac75a9823a2977dc9bf3a9c76aba71
Author: Pierre Ducroquet <pducroquet@entrouvert.com>
Date:   Mon Jan 10 18:01:08 2022 +0100

    sql: add and use indexes on concerned_roles_array and actions_roles_array (#60469)
```

Fichiers

0001-Add-and-use-indexes-on-concerned_roles_array-and-act.patch	2,9 ko	10 janvier 2022	Pierre Ducroquet
0001-sql-add-and-use-indexes-on-concerned_roles_array-and.patch	4 ko	10 janvier 2022	Pierre Ducroquet
0001-sql-add-and-use-indexes-on-concerned_roles_array-and.patch	3,98 ko	10 janvier 2022	Pierre Ducroquet