

Hobo - Development #23045

parallélisation du migrate_schema

09 avril 2018 21:10 - Thomas Noël

| | | | |
|---|----------|----------------------|---------------|
| Statut: | En cours | Début: | 09 avril 2018 |
| Priorité: | Bas | Echéance: | |
| Assigné à: | | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | Non |
| Patch proposed: | Oui | | |
| Description | | | |
| Permettre que les migrations se passent en parallèle, avec un nombre de process paramétrable. | | | |

Historique

#1 - 09 avril 2018 21:16 - Thomas Noël

Idées en vrac :

- soit des options de migrate_schema, soit une nouvelle commande dédiée pour ne pas tout mélanger
- permettre de passer un nombre de processus max en ligne de commande, ou dans les settings, sinon par défaut nombre de coeurs
- utilisation de <https://docs.python.org/2/library/multiprocessing.html> (imap_unordered)
- si possible permettre le démarrage «dès que possible» des workers uwsgi :
 - préparer la migration en posant un "on-maintenance.lock" sur tous tenants
 - avoir un retour 503 dans hobo.multitenant.middleware.TenantMiddleware sur un tenant tant que ce lock existe, cela "ferme" le tenant
 - quand la migration d'un tenant est terminée, libérer le lock (ie supprimer le fichier), le middleware fait alors son boulot normal, le tenant est ouvert
 - ça demande que cette commande manage de migration rende la main aussitôt que les locks sont posés, et travaille en tâche de fond, laissant le lancement de uwsgi se faire, qui attendra la libération des tenants un par un
 - étudier au passage comment NE PAS lancer les cron sur les tenants lockés (voire, autre sujet, quand le serveur uwsgi n'est pas lancé)

#2 - 09 avril 2018 21:33 - Benjamin Dauvergne

Thomas Noël a écrit :

- étudier au passage comment NE PAS lancer les cron sur les tenants lockés (voire, autre sujet, quand le serveur uwsgi n'est pas lancé)

Je me dis qu'on pourrait utiliser des locks explicites postgres plutôt que de passer par le FS (et ça résoud le problème quand on a plusieurs front en plus): <https://www.postgresql.org/docs/9.4/static/explicit-locking.html>

#3 - 09 avril 2018 21:49 - Benjamin Dauvergne

Thomas Noël a écrit :

- ça demande que cette commande manage de migration rende la main aussitôt que les locks sont posés, et travaille en tâche de fond, laissant le lancement de uwsgi se faire, qui attendra la libération des tenants un par un
- étudier au passage comment NE PAS lancer les cron sur les tenants lockés (voire, autre sujet, quand le serveur uwsgi n'est pas lancé)

En fait on aurait rien besoin de locker si on s'astreignait à une certain discipline car il y a deux types de migrations :

- ajout de tables ou de colonnes
- suppressions de tables ou de colonnes

Si on ordonnait systématiquement les migrations d'ajout avant les suppression on pourrait avoir le processus suivant:

- appliquer les ajouts
- relancer les serveurs et les crons sur le nouveau code
- appliquer les suppressions

Dans cet ordre là il n'est pas nécessaire de locker tout peut se faire en parallèle, il peut y avoir un peu de contention entre les migrations et le code de l'application (cron ou uwsgi) mais nos migrations étant généralement assez courtes je ne crois pas qu'on ait des contentions importantes.

À voir comment on pourrait automatiser ce fonctionnement.

#4 - 10 avril 2018 01:14 - Thomas Noël

Benjamin Dauvergne a écrit :

En fait on aurait rien besoin de locker si on s'astreignait à une certaine discipline car il y a deux types de migrations :

- ajout de tables ou de colonnes
- suppressions de tables ou de colonnes

Je pense qu'il y a aussi les renommages (sans doute à gérer en premier) et sans doute d'autres choses encore. Mais je n'arrive pas à voir comment on peut séparer les choses dans Django sauf à mettre les doigts dans le code de migration. Ou bien je rate qqchose ?

#5 - 07 janvier 2019 14:24 - Frédéric Péters

C'est peut-être nouveau de django-tenant-schemas mais je vois maintenant tenant_schemas/migration_executors/parallel.py et dans ce fichier :

```
def run_tenant_migrations(self, tenants):
    if tenants:
        processus = getattr(settings, 'TENANT_PARALLEL_MIGRATION_MAX_PROCESSES', 2)
        chunks = getattr(settings, 'TENANT_PARALLEL_MIGRATION_CHUNKS', 2)
```

#6 - 07 janvier 2019 15:05 - Thomas Noël

Pas si nouveau (mars 2007 <https://github.com/bernardopires/django-tenant-schemas/commit/323cfcfd38a60d61a9ba9b96eb01fa4a9a2e30d82>) et c'est dispo sur nos prod.

#7 - 07 janvier 2019 15:10 - Frédéric Péters

2017. Et du coup chez nous ça a fait partie de #18178.

#8 - 15 janvier 2019 14:55 - Thomas Noël

- Fichier 0001-migrate_schemas-add-executor-option-from-django-tena.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Un patch qui ajoute l'option executor, extraite de django-tenant-schemas. Au passage ça fait nettoyer des trucs pour du Django < 1.8.

La parallélisation se fait alors via « migrate_schemas --executor=parallel » sachant que par défaut j'ai posé une limite à 4 workers * 2 chunks (cf variables TENANT_PARALLEL_MIGRATION_* dans debian_config_common.py, par défaut c'est 2*2)

#9 - 02 mars 2019 11:20 - Benjamin Dauvergne

- Assigné à mis à Thomas Noël

#10 - 02 mars 2019 12:16 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

À rebaser, repousser pour test mais ça m'a l'air ok; vérifier au passage que les tests multitenant prennent bien en compte le paramétrage pour le parallélisme.

#11 - 10 avril 2020 14:28 - Thomas Noël

- Statut changé de Solution validée à En cours
- Priorité changé de Normal à Bas

Ce patch serait à revoir (rebaser, au moins) et notamment voir s'il est toujours efficace au regard des optimisations dans django >= 1.11.

#12 - 14 août 2022 01:12 - Thomas Noël

- Tags mis à accessible
- Assigné à Thomas Noël supprimé

Thomas Noël a écrit :

Ce patch serait à revoir (rebaser, au moins) et notamment voir s'il est toujours efficace au regard des optimisations dans django >= 1.11.

Aujourd'hui le nombre de tenant a explosé et un retour sur ce ticket serait bienvenu. Le patch doit être plus facile encore aujourd'hui avec les nettoyages qui ont eu lieu depuis.

#13 - 15 août 2022 19:53 - Thomas Noël

- Tags accessible supprimé

J'ai regardé quand même et après [#29522](#) je pense qu'on ne peut plus profiter de l'executor aussi facilement... je retire le tag "accessible".

Fichiers

| | | | |
|---|---------|-----------------|-------------|
| 0001-migrate_schemas-add-executor-option-from-django-tena.patch | 9,49 ko | 15 janvier 2019 | Thomas Noël |
|---|---------|-----------------|-------------|