

Passerelle - Development #82379

toulouse-maelis: ré-introduire du cache sur le catalogue général

16 octobre 2023 08:54 - Nicolas Roche

Statut:	Fermé	Début:	16 octobre 2023
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description			
Il a été désactivé :			
<ul style="list-style-type: none">pour y lire le nombre de places restantes https://dev.entrouvert.org/issues/78141pour laisser le portail piloter l'intervalle des dates d'interrogation https://dev.entrouvert.org/issues/78141 mais ce n'est pas satisfaisant, les requêtes sont trop lentes https://dev.entrouvert.org/issues/81351			
L'idée serait de réintroduire le cache du catalogue général, puis qu'au début de la démarche d'inscription, la capacité restante soit contrôlée sur le catalogue personnalisé.			
Demandes liées:			
Lié à Passerelle - Development #82961: Avoir une entrée "every5min" dans les ...		Fermé	30 octobre 2023
Lié à Intégrations graphiques Publik - Development #82964: toulouse-2022: [pa...		Fermé	30 octobre 2023
Lié à Passerelle - Development #82966: Ne plus proposer de date sur le WS qui...		Fermé	30 octobre 2023

Révisions associées

Révision a7ff9bbc - 10 novembre 2023 09:29 - Nicolas Roche

toulouse-maelis: put global catalog in cache (#82379)

Historique

#1 - 18 octobre 2023 09:13 - Nicolas Roche

- Description mis à jour

Stéphane me dit que mettre en cache le catalogue toutes les heures ne serait pas satisfaisant ; qu'il faudrait plutôt viser une mise à jour toutes les 5 ou 10 minutes.

Je me demande si ce ne serait pas trop "sale" de me greffer sur le cron "availability" en créant un job qui met le catalogue en cache, dans "check_status".

#4 - 20 octobre 2023 11:32 - Thomas Noël

Alors, je note que ici cache = stocker une copie chaque référentiel en base. Autrement dit, on n'utilise pas le cache de Django, mais on maintient une copie des référentiels, c'est pas un "cache" mais plutôt une "synchro régulière". C'est ce qui est fait dans Referential et c'est bien.

Selon moi, la technique de gestion d'une synchro avec une mise à jour non bloquante est plutôt celle ci :

- lorsqu'un référentiel est utilisé, renvoyer immédiatement ce qui est présent en local
- puis, si la date de dernière synchro a plus de "x" minutes, lancer un job de mise à jour (self.add_job)

Il faut donc deux choses en plus de la copie local du référentiel : sa date de dernière mise à jour, et un système de "lock" pour ne lancer qu'un seul job de mise à jour.

Pour le lock, pour ne pas lancer un job qui serait déjà en cours, je pense qu'il faut regarder ce qui est dans self.jobs_set (en ne considérant pas les jobs avec un statut failed/completed) ; ici je verrais bien un ticket spécifique pour ajouter une méthode add_job_once à côté de add_job. Ou pas :)

Reste qu'avec cette solution, si le référentiel n'est pas utilisé pendant 3 heures, alors la synchro ne se fait pas pendant 3 heures.

Pour contrer ça, on peut imaginer un cron hourly qui va lancer un job de synchro. Mais comme ça n'est que "hourly", s'il ne faut jamais que le cache ait plus d'une heure, alors on peut imaginer lancer un job asynchrone dans self.jobs, qui serait du genre :

```
def jobs(self):
    self.add_job_once('sync_referentials')
    super().jobs()
```

(self.jobs est un cron qui est lancé chaque 5 minutes, comme availability)

#5 - 20 octobre 2023 15:34 - Nicolas Roche

self.jobs est un cron qui est lancé chaque 5 minutes, comme availability

\o/ ça me plaît.

Le code de mise à jour des référentiels n'utilise pas de verrou.

(de mémoire je suis parti du connecteur BAN <https://dev.entrouvert.org/issues/70982#note-15>)

J'ai peut-être manqué quelque-chose (il faut me le dire si c'est flagrant) :

je procède par mise à jour des items via update_or_create

et à la fin je supprime ceux dont la date n'a pas été modifiés : .filter(..., updated__lt=last_update).delete()

De cette façon j'ai l'impression que je pourrais continuer à me passer d'un verrou, et traiter le catalogue comme les autres référentiels, sans lancer de job asynchrone :

```
def daily(self):
    super().daily()
    self.update_referentials()
```

#6 - 20 octobre 2023 16:48 - Thomas Noël

Le problème si tu n'as pas de verrou est le risque de lancer "n" mise à jour en même temps.

Pour un truc qui ne prend que quelques minutes, le risque est nul si tu fais juste du "cron hourly". (Sur la BAN c'est carrément du daily).

Sur le "cron jobs" qui se lance chaque 5 minutes, il faut être un peu plus sûr de toi... mais si on parle d'un truc qui prend une minute, ça doit passer.

Je ne pense pas qu'on augmente un jour la fréquence du cron jobs. Si c'est le cas, je compte sur toi pour rappeler que "ça fait pas uniquement les jobs" :) Blague à part je me dis qu'une solution plus propre, au lieu de réutiliser un cron pas prévu pour, ça serait d'ajouter un "every5min" dans nos FREQUENCY de cron, que tu serais le premier à utiliser -- ça doit se faire en quelques lignes dans commands/cron.py + passerelle/base/models.py + debian/uwsgi.ini, ça serait un premier commit facile je pense.

(si on parle d'un lancement de synchro encore plus fréquent, genre "dès que le référentiel a plus d'une minute" alors il faut vraiment locker, mais mon avis est que dans un tel besoin c'est d'abord au webservice en face d'accélérer ;))

#8 - 30 octobre 2023 16:42 - Nicolas Roche

- Lié à Development #82961: Avoir une entrée "every5min" dans les cron ajouté

#9 - 30 octobre 2023 18:17 - Nicolas Roche

- Lié à Development #82964: toulouse-2022: [parsifal] Ne plus passer de date de référence sur la cellule catalogue ajouté

#10 - 30 octobre 2023 18:17 - Nicolas Roche

- Lié à Development #82966: Ne plus proposer de date sur le WS qui renvoie le catalogue général. ajouté

#11 - 01 novembre 2023 21:34 - Robot Gitea

- Statut changé de Nouveau à Solution proposée

- Assigné à mis à Nicolas Roche

Nicolas Roche (nroche) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/passerelle/pulls/377>
- Titre : toulouse-maelis: put global catalog in cache (#82379)
- Modifications : <https://git.entrouvert.org/entrouvert/passerelle/pulls/377/files>

#12 - 06 novembre 2023 10:52 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne (bdauvergne) a approuvé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/passerelle/pulls/377>

#13 - 10 novembre 2023 09:24 - Robot Gitea

- Statut changé de *Solution validée* à *En cours*

Nicolas Roche (nroche) a fermé une pull request sur Gitea concernant cette demande.

#14 - 10 novembre 2023 09:34 - Robot Gitea

- Statut changé de *En cours* à *Solution proposée*

Nicolas Roche (nroche) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/passerelle/pulls/397>
- Titre : toulouse-maelis: put global catalog in cache (#82379)
- Modifications : <https://git.entrouvert.org/entrouvert/passerelle/pulls/397/files>

#15 - 10 novembre 2023 09:49 - Robot Gitea

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

Nicolas Roche (nroche) a mergé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/passerelle/pulls/397>
- Titre : toulouse-maelis: put global catalog in cache (#82379)
- Modifications : <https://git.entrouvert.org/entrouvert/passerelle/pulls/397/files>

#16 - 13 novembre 2023 11:15 - Transition automatique

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

#17 - 14 janvier 2024 04:42 - Transition automatique

Automatic expiration