

Zoo - Development #18481

nanterre: sécuriser les flux contenant des créations de fédération

06 septembre 2017 02:33 - Benjamin Dauvergne

Statut:	Fermé	Début:	06 septembre 2017
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
<ul style="list-style-type: none">• ne pas les rejouer automatiquement (i.e. le statut sur une erreur sera toujours UNRECOVERABLE_ERROR)• locker les individus concernés lors de la requête et lors du rejeu• sur un rejeu (i.e. job repassé dans l'état TODO) qui échoue, on efface la clé de fédération• on sauvegarde les anciennes clés en cas d'écrasement dans une liste 'anciennes_clés_de_fédération' composée de tuples: <pre>[nom de l'application, date de l'écrasement, valeur de l'ancienne clé]</pre>			

Révisions associées

Révision 5732cc61 - 06 septembre 2017 15:26 - Benjamin Dauvergne

nanterre: sécurise les créations initiales et en rejeu des fédérations (fixes #18481)

- ne pas les rejouer
- locker les individus pour lesquels on va potentiellement créer une fédération
- en cas de rejeu forcé et d'échec on nettoie la fédération
- on sauvegarde les anciennes clés en cas d'écrasement dans une liste 'anciennes_clés_de_fédération' composée de tuples:

```
[nom de l'application, date de l'écrasement, valeur de l'ancienne clé]
```

Historique

#1 - 06 septembre 2017 02:59 - Benjamin Dauvergne

- Fichier 0001-nanterre-s-curise-les-cr-ations-initiales-et-en-reje.patch ajouté
- Patch proposed changé de Non à Oui

#3 - 06 septembre 2017 08:04 - Thomas Noël

Le select_for_update() va attendre que «l'autre» requête se soit terminée, est-ce ce que l'on veut ? Avec un nowait=True on pourrait obtenir l'info qu'un lock est en place par ailleurs et renvoyer un message d'erreur (via un try: select_for_update(nowait=True) except: erreur) ... Je ne sais pas ce qui est le mieux, à vrai dire.

#4 - 06 septembre 2017 09:34 - Benjamin Dauvergne

J'arrivais mieux à penser la chose avec un simple lock bloquant mais oui c'est plus efficace de ne pas bloquer.

#5 - 06 septembre 2017 09:46 - Benjamin Dauvergne

- Fichier 0001-nanterre-s-curise-les-cr-ations-initiales-et-en-reje.patch ajouté

Voilà, j'ai du mal à envisager un test simplement.

#6 - 06 septembre 2017 10:02 - Thomas Noël

MMmhh t'as pas géré pour les select_for_update, là (y'en a deux sans noawait, et un sans try/except)

#7 - 06 septembre 2017 10:32 - Benjamin Dauvergne

Bon finalement je ne suis plus si sûr, on a deux moment où il faut un lock:

- à l'exécution du flux (pour éviter une écriture concurrente dans 'cles_de_federation')
- à la création du flux (pour être sûr qu'on prend la bonne décision concernant le choix entre créer une fédération ou utiliser l'existante)

Je me dis qu'à la création on veut toujours bloquer parce que jusqu'à présent on a aucune erreur attendue à la génération d'un flux et là on en créerait une, non seulement ça a échoué mais on a généré aucun flux; j'aime pas inventer des situations inconnues.

À l'exécution effectivement on peut simplement abandonner.

Dans mon dernier patch j'ai viré les lock au niveau de def synchronize() et CalculerQF.create() (d'ailleurs je pourrai factoriser encore un peu plus maintenant que j'ai mon flag FragmentBuilder.lock_individus et mettre ça dans FragmentBuilder.create()).

#8 - 06 septembre 2017 10:33 - Benjamin Dauvergne

- Fichier 0001-nanterre-s-curise-les-cr-ations-initiales-et-en-reje.patch ajouté

Mauvais patch.

#9 - 06 septembre 2017 10:39 - Benjamin Dauvergne

Et oui bloquer à la création qui est aussi le moment d'exécution de 99% des flux, c'est bloquer tout le temps (sachant qu'on a un timeout de 20s sur les appels RSU, bien trop long finalement sachant qu'on peut synchroniser jusqu'à 3 applications), on a des blocages en synchro jusqu'à 30s. De toute façon si technocarte timeout on aura tous les workers bloqués assez vite (5 personnes lance une synchro techno, personne utilise zoo pendant 20s). Idéalement on devrait beaucoup baisser les timeout et tant pis si ça passe pas (je le leur dirai), on peut aussi monter le nombre de workers et/ou utiliser massivement le mode thread/greenlet d'uwsgi parce que finalement zoo c'est que des I/O (peut-être plus greenlet parce que psycpg2 et threads ça m'inquiète toujours un peu, même si logiquement on a pas de connections partagées entre threads).

#10 - 06 septembre 2017 15:06 - Thomas Noël

Je suis pour poser ce patch ainsi sans toucher aux timeouts pour l'instant, sujet encore trop sensible face à une réalité qu'on ne connaît pas (on n'a jamais vu ce que tout cela donne en prod, "quand les applications sont chargées", etc). Comme toi, pas envie de trouver des solutions à des problèmes inconnues.

#11 - 06 septembre 2017 15:30 - Benjamin Dauvergne

- Statut changé de Nouveau à Résolu (à déployer)

- % réalisé changé de 0 à 100

Appliqué par commit [5732cc61c405f707ec222694d5fa3b8702af7713](https://commit-hash.com/5732cc61c405f707ec222694d5fa3b8702af7713).

#12 - 14 septembre 2017 17:01 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-nanterre-s-curise-les-cr-ations-initiales-et-en-reje.patch	9,45 ko	06 septembre 2017	Benjamin Dauvergne
0001-nanterre-s-curise-les-cr-ations-initiales-et-en-reje.patch	10,8 ko	06 septembre 2017	Benjamin Dauvergne
0001-nanterre-s-curise-les-cr-ations-initiales-et-en-reje.patch	9,73 ko	06 septembre 2017	Benjamin Dauvergne