w.c.s. - Bug #22355

le remapping des statuts sur changement d'un workflow est parfois trop long

07 mars 2018 17:35 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	07 mars 2018
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Voir #22259, il y a 7293 demandes, on fait un update suivi d'un commit pour chaque demande, sur du disque rotatif on est à minimum 10 ms de latence et à Montpellier qui n'a pas des foudres de guerre comme serveurs certainement plus, soit minimum 70 secondes pour le remapping.

Fred suggère de faire le remapping en dehors de la requête en mettant le formulaire dans un statut maintenance (statut qui n'existe pas actuellement).

Il serait aussi possible d'accélérer la mise à jour en faisant tout cela dans une unique transaction (mais on finira par trouver une instance avec trop de demandes plus tard), à noter quand même qu'une erreur en cours de migration pourrait laisser les formulaires dans un état intermédiaire (je me demande d'ailleurs où ça en est actuellement à Montpellier sur ce formulaire...) ce qui me laisse penser qu'une transaction serait vraiment nécessaire (en plus de la suggestion de Fred).

Demandes liées:

Lié à w.c.s. - Development #38579: Mapping de workflow en mode postgresql	**Fermé**	**17 décembre 2019**

Historique

#2 - 07 mars 2018 17:36 - Benjamin Dauvergne

- Description mis à jour

#3 - 11 mars 2018 09:05 - Frédéric Péters

Je viens (enfin!) de regarder le code et il y a deux temps au remapping :

- la modification des colonnes status_id des tables du formdata et de ses évolutions, c'est workflow_status_remapping_submit et très simplement, il faudrait juste du code particulier quand on est en mode SQL, pour faire ça à base de UPDATE sur l'ensemble des lignes ; (workflow_status_remapping_submit).
- la mise à jour des autorisations d'accès (rebuild_security), là c'est a priori moins évident de réduire le nombre de requêtes.

#4 - 12 mars 2018 09:27 - Thomas Noël

Et donc, dans un premier temps, faire :

- la modification des colonnes status_id des tables du formdata et de ses évolutions

lors du submit lui-même (synchrone, avec transaction et plantage possible) ; et

- la mise à jour des autorisations d'accès (rebuild_security), là c'est a priori moins évident de réduire le nombre de requêtes.

lors d'un afterjob (n'est pas si urgent, on peut survivre à une situation à les demandes n'ont pas de "security" à jour à mon sens, mais je peux me tromper)

#5 - 12 mars 2018 11:58 - Benjamin Dauvergne

Oui le rebuild_security est idempotent, on peut foirer recommencer on finira par converger.

Par contre sur la première partie j'ai vraiment l'impression que là ou ça a foiré. les statuts sont en vrac et difficilement récupérable (en vrai je suis très inquiet pour Montpellier, mélange d'anciens et de nouveaux statuts), et oui utiliser UPDATE est la bonne solution, mais c'est peut-être plus de code que de simplement faire en sorte de pouvoir faire un BEGIN/COMMIT en évitant que store() fasse un conn.close() (paramètre close_connection=False?).

#6 - 12 mars 2018 12:36 - Frédéric Péters

C'est plus de code mais c'est une question de perf aussi, ne pas charger tous les formdata.

#7 - 27 mai 2020 17:24 - Frédéric Péters

- Lié à Development #38579: Mapping de workflow en mode postgresql ajouté

#8 - 26 novembre 2020 11:41 - Benjamin Dauvergne

- Statut changé de Nouveau à Fermé

Je ferme au profit de #38579 pas la peine de garder deux tickets pour dire la même chose.