w.c.s. - Development #24424

Ne plus stocker last_jump_datetime dans les évolutions

11 juin 2018 12:00 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	11 juin 2018
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Suite du #23348.

Ça bourrine trop la base de donnée et empêche de journaliser les modifications correctement (on ne peut plus archive le WAL, on est obligé de le supprimer), il faut trouver une autre manière de conserver la date du dernier saut pour calculer l'expiration.

~~Dans le #23348 je soutiens que c'est du cache dans la mesure où en l'absence de la valeur (en cas de perte de celle-ci) on peut se baser sur formdata._last_update_time / evolutions[-1].time / formdata.receipt_time sans que ce soit un problème si juste après on repose un last_update_time dans un stockage volatile (en gros on va boucler un peu plus vite en cas de perte du stockage volatile, mais ça n'aura pas de conséquence grave). Peut-être que je me trompe donc si quelqu'un voit un cas ou l'absence de la donnée exacte serait préjudiciable, dites moi.~~

~~Le seul besoin que je verrai de matérialiser ça en base c'est si on avait une requête optimisée pour obtenir la liste des formdata avec last_update_time plus ancien qu'un certain point (ça augmenterait les perfs du CRON un poil), mais on n'a pas ça et on s'en passe bien pour l'instant, et même si on le voulait on pourrait toujours lister les valeurs depuis le cache (en utilisant une queue de priorité dans redis par exemple) en plus de faire la requête SQL.~~

Nouvean plan :
Le stockage du last_jump_datetime pour chaque évolution ne semble pas avoir d'utilité car seul certaines évolutions se verront assigner un last_jump_datetime différent de leur time

- créer une nouvelle table formdata_..evolution_last_jump_datetime avec deux colonnes evolution_id et @last_jump_datetime
- faire la jointure lors des select, si NULL renvoyer evolution.time à la place
- mettre à jour via un accesseur sur FormData set_last_jump_datetime() qui en SQL ira updater la bonne table et seulement celle là.

La même idée pourrait être utilisée pour last_update_time sur FormData.

Historique

#2 - 11 juin 2018 13:09 - Frédéric Péters

J'ai l'impression que c'est mal intitulé ? Que ton souhait est plutôt que l'information soit dans une autre table ?

#3 - 25 janvier 2019 11:58 - Benjamin Dauvergne

- Sujet changé de Ne pas mettre à jour last_jump_time sur un saut sur place à Ne plus stocker last_jump_datetime dans les évolutions

Frédéric Péters a écrit :

> J'ai l'impression que c'est mal intitulé ? Que ton souhait est plutôt que l'information soit dans une autre table ?

Oui ça devrait déjà améliorer les choses de ségréguer les champs fortement mis à jour dans leur propre table parce que ça limitera la quantité de données balancées dans le WAL.

#4 - 25 janvier 2019 12:08 - Benjamin Dauvergne

- Description mis à jour

#5 - 09 juillet 2023 21:20 - Frédéric Péters

- Statut changé de Nouveau à Fermé

- Planning mis à Non

#65744 a modifié le stockage des évolutions pour que seules les dernières soient enregistrées.