

w.c.s. - Development #65700

Performances: éviter de récupérer toutes les données des tables evolution systématiquement

25 mai 2022 11:54 - Pierre Ducroquet

Statut:	Fermé	Début:	25 mai 2022
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		

Description

Les tables `_evolution` des bases wcs ont une colonne de type `bytea`, `parts`, qui représente la majeure partie du poids de la table (plus de 90%), et qui en prime est désérialisée pour chaque instance récupérée.

Or, plusieurs chemins dans le code de wcs n'ont a priori pas besoin de cette colonne, dont la récupération coûte très cher (potentiellement un facteur 100 côté temps de requête PG, sans compter le temps passer dans pickle).

Il serait donc préférable de ne pas charger cette colonne systématiquement, mais plutôt à la demande.

À chaud, j'envisage deux pistes :

- soit avoir une propriété `evolution_lazy` dans `SqlDataMixin` qui soit utilisée par les chemins de code dont on sait qu'ils ne se soucient pas de `parts`,
- soit ne jamais charger `parts` et en faire une propriété dans `evolution` qui aille requêter en base à la demande.

Historique

#1 - 25 mai 2022 12:09 - Frédéric Péters

qui en prime est désérialisée pour chaque instance récupérée

À noter quand même que ces données (la table `..._evolution`) ne sont déjà chargées qu'en cas d'accès à `formdata.evolution`; en pratique on est ici sur des accès individuels à une demande. (ou le cas particulier de `load_all_evolution`).

- soit avoir une propriété `evolution_lazy` dans `SqlDataMixin` qui soit utilisée par les chemins de code dont on sait qu'ils ne se soucient pas de `parts`,

Ils peuvent être difficile à identifier, il suffit qu'il y ait un gabarit et toutes les données doivent devenir accessibles.

- soit ne jamais charger `parts` et en faire une propriété dans `evolution` qui aille requêter en base à la demande.

Ce qui peut multiplier les appels et au final faire perdre davantage, j'ai un peu peur, sans mesures. (peut-être que ça amènerait un entredeux, sur un accès à `parts` on chargerait toutes les "parts" du `formdata` concerné, sur l'idée que si on "creuse" comme ça dans un bout de statut il y a une grande chance qu'on le fasse aussi pour le statut suivant, etc.).

#2 - 09 juillet 2023 20:35 - Frédéric Péters

- Statut changé de Nouveau à Fermé

Il y a eu [#69109](#) après un profilage, pour ne plus unpickler les `parts`.