

w.c.s. - Development #65485

Table snapshot très volumineuse (-> partitionner la table)

20 mai 2022 08:12 - Pierre Ducroquet

Statut:	Fermé	Début:	20 mai 2022
Priorité:	Normal	Echéance:	
Assigné à:	Pierre Ducroquet	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Sur test.saas.entrouvert.org, j'observe que la base wcs_demarches_orleans_test_entrouvert_org est de loin la plus volumineuse (39GB). Or, tout l'espace est pris par la table snapshot et sa colonne "patch".			
Est-ce-que ce comportement est connu et voulu, ou s'agit-il d'un bug ? Si c'est voulu, alors je ferai le patch nécessaire pour que la table soit partitionnée par timestamp afin qu'elle cesse d'être un seul bloc massif qui casse les différents outils capable de traiter des tables en parallèle (dump, backup, restore, vacuumdb...)			
Demandes liées:			
Lié à w.c.s. - Development #65497: moins attendre pour préférer un snapshot c...		Fermé	20 mai 2022

Révisions associées

Révision 44bc83f0 - 16 juin 2022 11:44 - Pierre Ducroquet

sql: add indexes on snapshots table (#65485)

Historique

#1 - 20 mai 2022 08:29 - Frédéric Péters

Est-ce-que ce comportement est connu et voulu, ou s'agit-il d'un bug ?

Il faut regarder dans la table si quelque chose aurait l'air anormal.

Pour donner un contexte :

- dans les workflows il peut y avoir des modèles de documents, qui peuvent être assez lourds,
- historiquement il y avait une règle basique type "si ça fait plus d'1Mo, on n'enregistre pas de snapshot",
- on a fait évoluer pour permettre de plutôt stocker la différence entre deux versions successives, dans l'idée qu'ainsi une modification sur un statut A peut se faire sans avoir à restocker le gros modèle de document du statut B,
- et seulement quand le diff se trouvait plus gros que le workflow en lui-même, enregistrer la totalité, ([#57299](#))

Quelque chose d'anormal ce serait le même workflow enregistré, à répétition, en totalité alors qu'a priori un diff aurait du être enregistré.

#2 - 20 mai 2022 09:13 - Pierre Ducroquet

```
wcs_demarches_orleans_test_entrouvert_org=# select pg_size_pretty(sum(coalesce(length(comment), 0))) as comments, pg_size_pretty(sum(coalesce(length(serialization), 0))) as serializations, pg_size_pretty(sum(coalesce(length(patch), 0))) as patches from snapshots;
comments | serializations | patches
-----+-----+-----
```

```
1946 kB | 1539 MB | 37 GB
(1 ligne)
```

```
wcs_demarches_orleans_test_entrouvert_org=# with foo as (select object_type, object_id, sum(coalesce(length(patch), 0)) from snapshots group by 1, 2 order by 3 desc limit 30) select foo.object_type, foo.object_id, pg_size_pretty(sum) from foo;
object_type | object_id | pg_size_pretty
-----+-----+-----
```

```
workflow | 178 | 35 GB
workflow | 168 | 735 MB
workflow | 162 | 418 MB
```

Donc oui, y'a un workflow qui a un problème...

```
wcs_demarches_orleans_test_entrouvert_org=# select extract(week from timestamp), count(*) from snapshots where
object_type = 'workflow' and object_id = '178' group by 1 order by 1;
```

date_part	count
5	792
6	782
7	821
8	84
13	183
14	252
15	241
16	251

(8 lignes)

Plusieurs centaines de changements par semaine.

#3 - 20 mai 2022 09:18 - Frédéric Péters

Plusieurs centaines de changements par semaine.

Mais en soit ça ne devrait pas poser de problème, sauf si c'est à chaque fois "je remplace ce modèle de 20 Mo par un autre nouveau modèle de 20 Mo".

#4 - 20 mai 2022 09:34 - Pierre Ducroquet

En moyenne 8MB par patch, donc 2GB par semaine, sauf la semaine 6 où on avait 31MB par patch, 25GB sur la semaine.

#5 - 20 mai 2022 09:39 - Frédéric Péters

Si je prends les deux plus récentes modifications, je télécharge, et je diff,

```
--- snapshot-workflow-21301-20220424-1811.wcs 2022-05-20 09:33:15.837723115 +0200
+++ snapshot-workflow-21302-20220424-1812.wcs 2022-05-20 09:33:09.497675588 +0200
@@ -209977,7 +209977,7 @@
     <prefill>
       <locked>False</locked>
       <type>string</type>
-     <value>{{mairie_creation_8_var_mairie|default:""}}</value>
+     <value>{% if form_name == "Demande de livret de famille suite &#224; erreur" %}{form_var_r
emise_lf_orleans}}{% else %}{mairie_creation_8_var_mairie|default:""}}{% endif %}</value>
     </prefill>
     <items>
       <item>Mairie centrale</item>
```

mais dans la table il y a un diff qui est bien plus volumineux.

(...)

Ce qui est logique le diff est fait par rapport à la dernière version complète.

Tu peux produire pour le workflow 178 la liste des snapshots (order by id desc), avec en timestamp, comment + colonnes la taille des colonnes serialization et patch. ? Pour tenter d'imaginer un meilleur seuil.

#6 - 20 mai 2022 09:42 - Frédéric Péters

8MB

Deux approches pour faire évoluer :

```
if len(obj.serialization) > len(patch):
    # serialization is bigger than patch, store patch
```

j'étais juste avant sur l'idée de faire un $\text{len}(\text{obj.serialization}) / 10$ et chercher ce facteur mais peut-être que c'est aussi à combiner avec un truc statique genre $\text{len}(\text{patch}) > 1_000_000$.

#7 - 20 mai 2022 09:47 - Pierre Ducroquet

- Fichier log-wcs-wf-178.txt ajouté

Cf ci-joint pour les valeurs historiques du coup.

Et s'il est certain que l'on garde les données, je vais faire un patch pour qu'on partitionne la table. Si les données historiques sont froides, autant les garder dans des tables qui ne seront jamais modifiées.

#8 - 20 mai 2022 09:53 - Frédéric Péters

- Lié à [Development #65497](#): moins attendre pour préférer un snapshot complet à un patch ajouté

#9 - 23 mai 2022 09:35 - Frédéric Péters

- *Sujet changé de Table snapshot très volumineuse. à Table snapshot très volumineuse (-> partitionner la table)*

- Assigné à mis à [Pierre Ducroquet](#)

#10 - 23 mai 2022 16:35 - Pierre Ducroquet

- *Fichier 0001-sql-add-indexes-on-snapshots-table-65485.patch ajouté*

- *Statut changé de Nouveau à Solution proposée*

- *Patch proposed changé de Non à Oui*

En préliminaire, j'ai ce patch pour améliorer les performances...

#11 - 24 mai 2022 17:48 - Pierre Ducroquet

Étant donné qu'on a encore des personnes avec des instances en PostgreSQL 9.6 (plus supporté pourtant) et que le partitionnement déclaratif a été introduit dans PostgreSQL 10... je suppose qu'il vaut mieux attendre ?

#12 - 24 mai 2022 17:53 - Frédéric Péters

Oui. (mais ça doit se réduire)

#16 - 16 juin 2022 09:08 - Frédéric Péters

- *Statut changé de Solution proposée à Solution validée*

En préliminaire, j'ai ce patch pour améliorer les performances...

Go.

#17 - 16 juin 2022 11:46 - Pierre Ducroquet

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

Fait

```
commit 44bc83f0bfff04de116d36a78519955b087effacc (HEAD -> main, origin/wip/65485/snapshots-table-partitionning, origin/main, origin/HEAD, wip/65485/snapshots-table-partitionning)
```

```
Author: Pierre Ducroquet <pducroquet@entrouvert.com>
```

```
Date: Mon May 23 16:31:03 2022 +0200
```

```
sql: add indexes on snapshots table (#65485)
```

Je vais faire un nouveau ticket pour le partitionnement, pour le jour où nous n'aurons plus de 9.6 ou équivalent dans les installations connues.

#18 - 17 juin 2022 08:14 - Transition automatique

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

#19 - 21 août 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

log-wcs-wf-178.txt	187 ko	20 mai 2022	Pierre Ducroquet
0001-sql-add-indexes-on-snapshots-table-65485.patch	2,06 ko	23 mai 2022	Pierre Ducroquet