

w.c.s. - Development #71441

Passer les nonces dans la base de données

18 novembre 2022 08:38 - Pierre Ducroquet

Statut:	Fermé	Début:	18 novembre 2022
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		

Description

Aujourd'hui, les nonces sont représentés par des fichiers dans les dossiers nonces des tenants. Cette solution fonctionne, mais a un défaut majeur : elle va générer un nombre important d'entrées dans le dossier. Certes, les fichiers sont supprimés ultérieurement, mais cela a des conséquences en fonction du système de fichiers, et ext4, que l'on utilise présentement, gère "mal" ce cas.

```
pducroquet@node1.prod:/srv/drbd0/lib/wcs$ du -shc */nonces
1,5G    total
```

C'est beaucoup pour 50000 fichiers de 0 octets, et ça semble à l'origine de gros pics d'iowait la nuit lors de l'exécution des crons de nettoyage.

Démo du problème :

```
$ mkdir demo-huge-dir
$ cd demo-huge-dir
$ for i in `seq 0 50000`; do touch $i ; done
$ rm *
$ ls -hal .. | grep "demo-huge-dir"
drwxr-xr-x  2 pierre pierre 384K Nov 18 08:21 demo-huge-dir
$ for i in `seq 0 50000`; do touch $i ; done
$ rm *
$ ls -hal .. | grep "demo-huge-dir"
drwxr-xr-x  2 pierre pierre 1.1M Nov 18 08:21 demo-huge-dir
```

En fait, ext4 n'est pas capable de réindexer à chaud un dossier. Il existe une solution pour forcer la réindexation, mais il s'agit de démonter le système de fichiers et forcer un fsck avec l'option -D.

Du coup, contrairement à ce qu'on pourrait imaginer, il n'est pas inutile de considérer une migration des nonces dans la base de données.

PG sera largement capable d'optimiser ça, à la limite en augmentant l'agressivité de l'autovacuum sur cette nouvelle table.

La table pourrait être de la forme

```
CREATE TABLE nonces (
  id uuid primary key,
  expiration timestamp with time zone not null
);
```

Vu que l'expiration est faite dans un cron, et qu'on a rarement beaucoup d'éléments, inutile d'envisager un index sur cette colonne. La conversion des formats existants en uuid est transparente (et a l'avantage de considérablement accélérer l'indexation et la recherche)

```
test=> insert into nonces (id) values ('d408b86c9282548e15f210b810dc1a7f') returning *;
```

id	expiration
d408b86c-9282-548e-15f2-10b810dc1a7f	2022-11-18 08:35:03.052132+01

```
(1 row)
```

```
INSERT 0 1
```

```
test=> select * from nonces where id = 'd408b86c9282548e15f210b810dc1a7e';
      id | expiration
-----+-----
d408b86c-9282-548e-15f2-10b810dc1a7e | 2022-11-18 08:33:48.556903+01
(1 row)
```

Si le format UUID pose problème en python, on peut passer outre avec :

```
In [18]: cur.execute("select uuid_send(id) from nonces limit 1")
```

```
In [19]: cur.fetchone()[0].hex()
```

```
Out [19]: 'd408b86c9282548e15f210b810dc1a7e'
```

Demandes liées:

Lié à w.c.s. - Development #71455: Passer les form_tokens dans la base de don...

Fermé

18 novembre 2022

Révisions associées

Révision f7a70b4c - 13 janvier 2023 12:50 - Frédéric Péters

misc: save received nonces in token table (#71441)

Historique

#1 - 18 novembre 2022 08:50 - Frédéric Péters

ça semble à l'origine de gros pics d'iowait la nuit lors de l'exécution des crons de nettoyage

```
cls.register_cronjob(CronJob(cls.clean_nonces, minutes=range(0, 60, 5), name='clean_nonces'))
```

Le job de nettoyage des nonces est programmé pour tourner toutes les cinq minutes, je pense qu'il y a erreur dans l'analyse.

#2 - 18 novembre 2022 08:57 - Pierre Ducroquet

Je me suis fait avoir par le nom de l'autre cron qui tournait et qui pour moi était déjà en base. Clean_sessions nettoie en fait les form_tokens qui ont exactement le même fonctionnement et le même problème que les nonces.

Donc il y aura le même ticket à faire, mais pour form_tokens. Je suis d'avis de commencer par nonces parce que ça me semble plus facile à faire et donnera un premier aperçu de l'évolution.

#3 - 18 novembre 2022 09:09 - Benjamin Dauvergne

Pierre Ducroquet a écrit :

Je me suis fait avoir par le nom de l'autre cron qui tournait et qui pour moi était déjà en base. Clean_sessions nettoie en fait les form_tokens qui ont exactement le même fonctionnement et le même problème que les nonces.

Donc il y aura le même ticket à faire, mais pour form_tokens. Je suis d'avis de commencer par nonces parce que ça me semble plus facile à faire et donnera un premier aperçu de l'évolution.

Tu peux ouvrir un ticket spécifique pour les form_tokens ? Parce que pour les form_tokens la situation est particulière, ils sont sensés être liés à une session¹ et qu'il y a déjà un moyen de déporter le stockage en table principale des sessions via sql.TransientData utilisé actuellement pour les magictokens.

Est-ce que ce ne serait pas plus simple d'utiliser memcache (rendu cohérent pour l'occasion) pour les nonces ? L'expiration n'est qu'à 30 secondes et on peut se permettre de ne pas être totalement ACID et éventuellement consistant sur un système qui limite le rejeu (ce que je dirai qu'on est déjà via NFS).

¹ ce qui n'est plus vrai dans l'implémentation via le filesystem actuelle, vu que le chemin ne contient que l'identifiant du token et pas celui de la session dans laquelle on l'a créé, mais passons, ça se corrigera tout seul si on change le mode de stockage.

#4 - 18 novembre 2022 09:44 - Pierre Ducroquet

Benjamin Dauvergne a écrit :

Pierre Ducroquet a écrit :

Je me suis fait avoir par le nom de l'autre cron qui tournait et qui pour moi était déjà en base. Clean_sessions nettoie en fait les form_tokens qui ont exactement le même fonctionnement et le même problème que les nonces.

Donc il y aura le même ticket à faire, mais pour form_tokens. Je suis d'avis de commencer par nonces parce que ça me semble plus facile à faire et donnera un premier aperçu de l'évolution.

Tu peux ouvrir un ticket spécifique pour les form_tokens ? Parce que pour les form_tokens la situation est particulière, ils sont sensés être liés à une session¹ et qu'il y a déjà un moyen de déporter le stockage en table principale des sessions via sql.TransientData utilisé actuellement pour les magictokens.

Yep, c'est prévu, je regarde un peu le code pour réfléchir au truc avant de faire le ticket.

Est-ce que ce ne serait pas plus simple d'utiliser memcache (rendu cohérent pour l'occasion) pour les nonces ? L'expiration n'est qu'à 30 secondes et on peut se permettre de ne pas être totalement ACID et éventuellement consistant sur un système qui limite le rejeu (ce que je dirai qu'on est déjà via NFS).

Rendu cohérent pour l'occasion, en voilà une phrase qu'elle est belle :)

Oui ça serait sûrement plus simple, mais je n'ai pas connaissance de l'utilisation de memcache au sein de wcs, et je n'ai jamais regardé comment rendre le memcache "cohérent" entre les nœuds.

¹ ce qui n'est plus vrai dans l'implémentation via le filesystem actuelle, vu que le chemin ne contient que l'identifiant du token et pas celui de la session dans laquelle on l'a créé, mais passons, ça se corrigera tout seul si on change le mode de stockage.

C'est effectivement ce qu'il me semble aussi de ce que j'ai commencé à lire (mais ça posera la question de la migration)

#5 - 18 novembre 2022 10:06 - Pierre Ducroquet

- Lié à Development #71455: Passer les form_tokens dans la base de données ajouté

#6 - 18 novembre 2022 13:23 - Benjamin Dauvergne

Pierre Ducroquet a écrit :

Rendu cohérent pour l'occasion, en voilà une phrase qu'elle est belle :)

Oui ça serait sûrement plus simple, mais je n'ai pas connaissance de l'utilisation de memcache au sein de wcs, et je n'ai jamais regardé comment rendre le memcache "cohérent" entre les nœuds.

En général ça veut dire n'en avoir qu'un et le partager.

#7 - 11 janvier 2023 20:16 - Frédéric Péters

- Statut changé de Nouveau à En cours

- Assigné à changé de Pierre Ducroquet à Frédéric Péters

Je prends dans la suite de [#71455](#) et pareil ici on peut réutiliser une table existante (token, qui gère déjà l'expiration).

#8 - 11 janvier 2023 21:20 - Robot Gitea

- Statut changé de En cours à Solution proposée

Frédéric Péters (fpeters) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://gitea.entrouvert.org/entrouvert/wcs/pulls/28>
- Titre : misc: save received nonces in token table ([#71441](#))
- Modifications : <https://gitea.entrouvert.org/entrouvert/wcs/pulls/28/files>

#9 - 12 janvier 2023 09:37 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Lauréline Guérin (lguerin) a approuvé une pull request sur Gitea concernant cette demande :

- URL : <https://gitea.entrouvert.org/entrouvert/wcs/pulls/28>
- Commentaire :

N/A

#10 - 13 janvier 2023 12:50 - Robot Gitea

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

Frédéric Péters (fpeters) a mergé une pull request sur Gitea concernant cette demande :

- URL : <https://gitea.entrouvert.org/entrouvert/wcs/pulls/28>

- Titre : misc: save received nonces in token table ([#71441](#))
- Modifications : <https://gitea.entrouvert.org/entrouvert/wcs/pulls/28/files>

#11 - 13 janvier 2023 16:14 - Transition automatique

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

#12 - 19 mars 2023 04:42 - Transition automatique

Automatic expiration