Backlog Courrier - Anomalie #16925

[PDO] les requêtes préparées ne sont pas libérées

23/04/2021 11:08 - Quentin RIBAC

Statut:	Résolu	**Début:**	23/04/2021
Priorité:	2-Sérieux	**Echéance:**	
Assigné à:			
Catégorie:			
Version cible:	20.10		
Version source:	2301	**Tags Courrier:**	

Description

Maarch Courrier 20.10.10

Pour mon problème de script qui prenait toute la RAM, j'ai avancé. Voir l'image jointe pour se rendre compte de la gravité du problème, toute la mémoire vive et le swap sont occupés, soit plus de 20Go.

Ce sont des process postgres qui prennent la mémoire, pas php. En coupant mon script PHP la mémoire n'est pas libérée, ni en redémarrant PostgreSQL, mais en faisant un restart d'apache ça libère la mémoire. Donc pour moi le problème vient de Courrier. J'ai été voir dans les sources qui gèrent le PDO et j'ai fait cette petite modification :

```
diff --git a/src/core/models/DatabasePDO.php b/src/core/models/DatabasePDO.php
index 7328049bc5..08deb5b045 100755
--- a/src/core/models/DatabasePDO.php
+++ b/src/core/models/DatabasePDO.php
@@ -20,6 +20,10 @@ class DatabasePDO
     private static $type            = null;
     private static $preparedQueries = [];

+   public function __destruct() {
+       self::$preparedQueries = [];
+   }
+
     public function __construct(array $args = [])
     {
         if (!empty(self::$pdo)) {
```

et ça semble fonctionner.

J'ai testé en mettant self::reset(); dans le destructeur mais la recherche ne fonctionnait alors plus vu qu'il y a apparemment des tables temporaires dans la connexion.

Cette solution intermédiaire de détruire les requêtes préparées semble améliorer grandement l'utilisation de la mémoire, peut-être au détriment de la vitesse d'exécution, mais pas d'une façon que j'ai pu ressentir en tant qu'utilisateur.

Pouvez-vous valider que cette modification n'aura pas d'impact fonctionnel bloquant pour les utilisateurs, hormis un léger ralentissement ?
La modification ne serait à appliquer que le temps d'exécution d'un script de reprise de données à usage unique.

Historique

#1 - 23/04/2021 11:13 - Emmanuel DILLARD

- Statut changé de A qualifier à R&D - A étudier

- Assigné à changé de EDI PO à Florian AZIZIAN

- Version cible changé de Inscription Backlog Courrier à 20.10 (Fin de vie)

#2 - 23/04/2021 14:29 - Quentin RIBAC

Nouvelle suggestion de git diff :

```
diff --git a/rest/index.php b/rest/index.php
index e8314ba876..09ffeb1dd8 100755
--- a/rest/index.php
+++ b/rest/index.php
@@ -661,3 +661,4 @@ $app->put('/registeredMails/import', \RegisteredMail\controllers\RegisteredMailC
 $app->put('/registeredMails/{resId}', \RegisteredMail\controllers\RegisteredMailController::class . ':update'
);

 $app->run();
+\SrcCore\models\DatabasePDO::clearPreparedQueries();
diff --git a/src/core/models/DatabasePDO.php b/src/core/models/DatabasePDO.php
index 7328049bc5..b83a8ac19d 100755
--- a/src/core/models/DatabasePDO.php
+++ b/src/core/models/DatabasePDO.php
@@ -20,6 +20,10 @@ class DatabasePDO
     private static $type            = null;
     private static $preparedQueries = [];

+   public function clearPreparedQueries() {
+       self::$preparedQueries = [];
+   }
+
     public function __construct(array $args = [])
     {
         if (!empty(self::$pdo)) {
```

Ici on supprime les requêtes préparées une fois par appel REST et non à chaque destruction de DatabasePDO.

#3 - 23/04/2021 15:12 - Florian AZIZIAN

- Statut changé de R&D - A étudier à Complément d'Informations

- Assigné à changé de Florian AZIZIAN à Quentin RIBAC

Cette deuxième suggestion me semble mieux.

Cependant, je trouve ça dangereux de l'intégrer dans le standard, car les requêtes SQL sont le cœur absolu de l'application.
Cela peut effectivement réduire les performances.

Je te propose de l'intégrer dans le code de ton client le temps de lancer le script de reprise (ne pas oublier supprimer le code quand c'est terminé). Ou sinon, il est peut être possible d'optimiser ton script pour lancer le batch par lot de 1000 (c'est un chiffre au hasard car je ne connait pas ce que fait ton script). Comme ça la variable static $preparedQueries se videra tous les 1000 et ne grossira pas (et donc pas d'augmentation de la RAM).

#4 - 06/05/2021 19:01 - Emmanuel DILLARD

- Priorité changé de 1-Majeur à 2-Sérieux

#5 - 10/05/2021 15:35 - Emmanuel DILLARD

- Statut changé de Complément d'Informations à 17

#6 - 14/05/2021 15:46 - Quentin RIBAC

- Assigné à changé de Quentin RIBAC à EDI PO

- Priorité changé de 2-Sérieux à 0-Bloquant

Lors de la mise en place de cette modification sur Maarch Courrier 20.03 sur le serveur, puis au lancement du script, le serveur est tombé : Out of Memory.

Je peux fournir le code de mon script pour que nous testions ensemble sur une instance locale.

Lors des essais sans modification du code de l'application, le serveur était out of memory à cause de postgres ; ici c'est le système de fichier répliqué glusterfs qui a été tué automatiquement pour libérer de la mémoire, coupant l'accès aux fichiers de configuration (custom) et stockage (docservers) pour les clients.

Il faut fournir une optimisation, sans quoi le script, que nous avons interrompu avant sa fin, ne pourra pas être exécuté entièrement ni les fois suivantes prévues.

C'est surtout un signe que l'API REST de Maarch Courrier ne supporte pas les appels massifs.

#7 - 17/05/2021 10:06 - Emmanuel DILLARD

- Statut changé de 17 à R&D - A étudier

- Assigné à changé de EDI PO à Florian AZIZIAN

- Priorité changé de 0-Bloquant à 1-Majeur

#8 - 18/05/2021 16:06 - Emmanuel DILLARD

- Projet changé de Backlog à CURRENT SPRINT

- Statut changé de R&D - A étudier à R&D - En cours

#9 - 18/05/2021 16:43 - Emmanuel DILLARD

- Echéance mis à 01/06/2021

#11 - 18/05/2021 16:52 - Florian AZIZIAN

- Assigné à Florian AZIZIAN supprimé

#12 - 01/06/2021 15:24 - Emmanuel DILLARD

- Projet changé de CURRENT SPRINT à Backlog

- Statut changé de R&D - En cours à R&D - A planifier

#13 - 04/06/2021 12:54 - Florian AZIZIAN

- Echéance 01/06/2021 supprimé

#14 - 08/06/2021 15:13 - Emmanuel DILLARD

- Priorité changé de 1-Majeur à 2-Sérieux

#15 - 09/06/2021 10:34 - Emmanuel DILLARD

- Projet changé de Backlog à Backlog Courrier

- Version cible changé de 20.10 (Fin de vie) à 20.10

#18 - 24/05/2023 11:19 - Emmanuel DILLARD

- Statut changé de R&D - A planifier à R&D - A étudier

#20 - 24/07/2023 11:58 - Emmanuel DILLARD

- Statut changé de R&D - A étudier à Résolu

- Version source mis à 2301

Pas de changement notables.
Risques potentiels élevés selon GHE

Fichiers

ram99pct.png	166 ko	23/04/2021	Quentin RIBAC