

Hobo - Development #15470

Ajout de verrous globaux et par application

17 mars 2017 10:30 - Frédéric Péters

Statut:	Fermé	Début:	17 mars 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Dans un environnement haute dispo il ne faut pas que les jobs s'exécutent sur les mêmes données depuis les différentes VM.

Idée d'ajouter un --skip-if-filename à la commande tenant_command puisqu'elle est utilisée pour le lancement de tous nos jobs.

Révisions associées

Révision d432df9f - 19 avril 2018 14:32 - Frédéric Péters

general: add possibility to skip all cron jobs (#15470)

This is useful for load balancing as jobs should only be run on one host.

Original was amended with a condition on "--all-tenants" and help messages.

Signed-off-by: Christophe Siraut <csiraut@entrouvert.com>

Historique

#1 - 05 avril 2017 16:29 - Frédéric Péters

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Je ne suis plus arrivé à me souvenir ce qui avait motivé l'idée du --skip-if-filename; à l'implémenter je me suis dit que ça ne serait pas pratique, ça exige d'aller modifier tous les appels à tenant_command pour ajouter cette option.

Du coup, pourquoi ne pas simplement avoir la possibilité de mettre DISABLE_CRON_JOBS = True dans le settings.py ? (patch qui fait ça)

#2 - 06 avril 2017 02:56 - Thomas Noël

Frédéric Péters a écrit :

Du coup, pourquoi ne pas simplement avoir la possibilité de mettre DISABLE_CRON_JOBS = True dans le settings.py ? (patch qui fait ça)

Yep. Mais peut-être uniquement en cas de --all-tenants ? (c'est "tenant_command --all-tenants" qui est utilisé par les crons)

Et aussi :

- ajouter un petit help sur force_job
- en cas de verbose, un petit message si le DISABLE_CRON_JOBS est coupé
- pour mémoire et guise de simili-documentation, poser un DISABLE_CRON_JOBS = False dans debian/debian_config_common.py avec un mini commentaire qui explique que ça bloque les commandes lancées avec "tenant_command --all-tenants"

#3 - 06 avril 2017 10:44 - Benjamin Dauvergne

J'aime pas trop même si ça fait le job, tout ce qu'on lance avec tenant_command n'est pas un job cron, on va avoir de plus en plus de machines en TMA autre que la notre, ça fera un truc en plus à expliquer (faudra faire tenant_command --force-job shell pour avoir un shell sur ces machines par exemple, ou créer un superuser quand on en a pas, ou que sais je d'autre).

Ça ne vous dirait pas plus de simplement changer les debian/package.cron pour if ajouter des

```
test -f /etc/package/no-cron-job || la_command
```

je sais que ça fait beaucoup plus de modifications sur plusieurs paquets mais c'est moins magique.

#4 - 06 avril 2017 10:44 - Benjamin Dauvergne

Pas vu la remarque sur --all-tenants, effectivement ce serait plus acceptable dans ce cas.

#5 - 06 avril 2017 10:51 - Thomas Noël

Benjamin Dauvergne a écrit :

Ça ne vous dirait pas plus de simplement changer les debian/package.cron pour if ajouter des "test -f /etc/package/no-cron-job"

J'avoue que c'est compréhensible à un unixien de base (tel que moi). Je ne vois pas de désavantage. Fred, pourquoi on était parti sur un truc dans le code de l'appli ? On avait une raison ? (si oui je l'ai pas notée)

#6 - 06 avril 2017 11:03 - Frédéric Péters

On avait une raison ? (si oui je l'ai pas notée)

Non. Moi ça me va aussi de ne **rien faire**, juste dans les notes d'installation redondante dire qu'il faut désactiver tels jobs.

#7 - 07 avril 2017 13:28 - Thomas Noël

En fait, à y réfléchir, l'idée du "DISABLE_CRON_JOBS = True dans le settings.py" elle est pas super parce que ça rend un peu difficile une éventuelle bascule d'une machine à l'autre.

Avoir des "/etc/<package>/cron-disabled" c'est quand même pratique (bascule avec rm/touch et hop).

#9 - 10 avril 2018 10:01 - Christophe Siraut

Thomas Noël a écrit :

En fait, à y réfléchir, l'idée du "DISABLE_CRON_JOBS = True dans le settings.py" elle est pas super parce que ça rend un peu difficile une éventuelle bascule d'une machine à l'autre.

Avoir des "/etc/<package>/cron-disabled" c'est quand même pratique (bascule avec rm/touch et hop).

Moi j'aime bien le patch de Fred, il permet d'ajouter la variable dans /etc/\$module/settings.d/cron-disabled.py comme ça tout le monde est content.

#10 - 10 avril 2018 10:01 - Christophe Siraut

ack.

#11 - 10 avril 2018 11:05 - Thomas Noël

un-ack ; cf mes indications sur #1, le patch tel quel est trop brutal.

#12 - 10 avril 2018 11:35 - Christophe Siraut

- Assigné à mis à Christophe Siraut

je prends

#13 - 11 avril 2018 16:16 - Christophe Siraut

- Fichier 0002-management-commands-tenant_command.py-add-help-messa.patch ajouté

- Fichier 0003-management-commands-tenant_command.py-notify-task-di.patch ajouté

- Fichier 0004-management-commands-tenant_command.py-only-consider-.patch ajouté

- Fichier 0005-settings.py-document-DISABLE_CRON_JOBS.patch ajouté

Quelques ajouts pour tenir compte des remarques de Thomas; il reste un souci: en l'état DISABLE_CRON_JOBS doit être posé sur chaque brique (le patch 5 est incorrect, je ne comprends pas encore pourquoi). J'aimerais pouvoir poser cette option uniquement dans le settings du serveur hobo.

#14 - 11 avril 2018 16:44 - Thomas Noël

Christophe Siraut a écrit :

J'aimerais pouvoir poser cette option uniquement dans le settings du serveur hobo.

Le serveur hobo est installé sur une machine différente, donc c'est pas possible. Ou bien tu parlais d'autre chose ?

#15 - 16 avril 2018 16:05 - Christophe Siraut

- Fichier 0005-settings.py-document-DISABLE_CRON_JOBS.patch supprimé

#16 - 16 avril 2018 16:21 - Christophe Siraut

- Fichier 0005-debian_config_common-document-option-DISABLE_CRON_JO.patch ajouté

Voici, en attente de commentaire/validation.

#17 - 17 avril 2018 15:45 - Christophe Siraut

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch ajouté

- Fichier 0002-debian-debian_config_common.py-add-inline-documentat.patch ajouté

- Assigné à changé de Christophe Siraut à Thomas Noël

Voici une version squashée.

#18 - 17 avril 2018 16:37 - Christophe Siraut

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch ajouté

#19 - 17 avril 2018 16:39 - Christophe Siraut

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch supprimé

#20 - 17 avril 2018 16:41 - Christophe Siraut

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch ajouté

#21 - 17 avril 2018 16:44 - Thomas Noël

Vu ensemble déjà, ça me va bien ; pousse ça dès que tu as testé une exécution d'un truc en local.

#22 - 18 avril 2018 09:48 - Christophe Siraut

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch ajouté

Je veux bien une relecture supplémentaire: j'ai ajouté le traitement de l'option verbosity en copiant le code de django BaseCommand. La ligne 40 contextualise la situation. J'ai testé le fonctionnement.

#23 - 18 avril 2018 15:09 - Thomas Noël

C'est pas ok :

```
args_namespace, args = args_parser.parse_known_args(argv)
```

va supprimer le verbosity de args, qui en sera pas envoyé à la commande lancée sur les tenants cibles.

Il faudrait lire le verbosity avec un parser à part, qui ne touche pas à args, genre :

```
verbosity_parser = argparse.ArgumentParser()
verbosity_parser.add_argument('-v', '--verbosity', action='store', dest='verbosity', default=1, type=int)
args_verbosity, _ = verbosity_parser.parse_known_args(args)
# et on a bien un args_verbosity.verbosity (à 1 par défaut)
```

#24 - 19 avril 2018 09:56 - Christophe Siraut

- Fichier 0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch ajouté

#25 - 19 avril 2018 10:50 - Thomas Noël

Ack

#26 - 27 avril 2018 12:23 - Christophe Siraut

- Statut changé de En cours à Résolu (à déployer)

#27 - 27 avril 2018 19:27 - Benjamin Dauvergne

- Fichier *0001-use-pg_advisory_lock-to-prevent-running-cronjobs-dur.patch* ajouté

Une proposition basée sur un lock global dans postgres, ça bloque aussi le fait de faire tourner les crons durant les migrations, et inversement.

#28 - 27 avril 2018 19:28 - Benjamin Dauvergne

- Fichier *0001-use-pg_advisory_lock-to-prevent-running-cronjobs-dur.patch* ajouté

Avec le fichier principal c'est mieux.

#29 - 30 avril 2018 09:55 - Christophe Siraut

Une proposition basée sur un lock global dans postgres, ça bloque aussi le fait de faire tourner les crons durant les migrations, et inversement.

Bonne idée, je n'ai pas testé. Je vois que tu hardcode un temps d'attente de 1 heure pour les migrations, cela me semble beaucoup, et vu que les migrations sont interactives j'ajouterais un message à l'utilisateur pendant l'attente. Par rapport au lock global sur les cron, pour les installations de Publik sur un postgresql centralisé, nous ne ferons pas l'économie d'organiser les tâches cron des différentes briques (au risque que ce soit toujours la même qui prenne le verrou)

#30 - 30 avril 2018 11:49 - Benjamin Dauvergne

Qu'est-ce qui exigerait d'avoir les CRONs lancés toujours sur le même serveur ? Le fait que les machines ne soient pas toutes à jour en même temps ?

Mon avis sur les CRONs c'est qu'ils devraient être écrit pour pouvoir s'exécuter en parallèle autant que possible, avec des `select_for_update()` et à partir de Django 1.11 des `select_for_update(skip_locked=True)`, mais qu'en attendant on peut utiliser PG comme gestionnaire de lock distribué.

#31 - 30 avril 2018 12:51 - Benjamin Dauvergne

- Fichier *0002-use-pg_advisory_lock-to-prevent-running-cronjobs-dur.patch* ajouté

- Fichier *0001-tests_multitenant-PEP8ness-and-cleanup-15470.patch* ajouté

J'ai un peu amélioré les choses en définissant un lock unique pour les migrations et des locks variables pour les crons à utiliser via `--lock <lock_id>` sur la commande `tenant_command` qui par contre prend un lock partagé sur le lock des migrations.

J'ai aussi transformé la fonction de lock en `context_manager` pour aussi délocker le lock (normalement c'est automatique en fin de session avec PG mais pour les tests c'est plus pratique de savoir que le lock est parti tout de suite et ça mange pas de pain).

Donc:

- on peut avoir au mieux un processus de migration en cours, il bloque les CRONs, et si un CRON tourne il attend qu'il finisse pour commencer sa migration,
- on peut avoir autant de processus CRON qui tournent pourvu qu'ils soient de type différent et n'utilisent pas le même lock, ils peuvent tourner sur n'importe quelle machine

On pourrait aussi utiliser le lock de CRON pour mettre l'application en maintenance pendant une migration avec un middleware adapté.

#32 - 30 avril 2018 12:52 - Benjamin Dauvergne

(Reporté commentaire et patch que j'avais malencontreusement posé sur le [#23471](#)).

#33 - 30 avril 2018 21:23 - Christophe Siraut

- *Sujet changé de Ajout d'un paramètre pour ne pas exécuter les job cron à Ajout de verrous globaux et par application*

- *Statut changé de Résolu (à déployer) à En cours*

Qu'est-ce qui exigerait d'avoir les CRONS lancés toujours sur le même serveur ? Le fait que les machines ne soient pas toutes à jour en même temps ?

Rien. (je faisais seulement remarquer avant ta modification de verrou par application qu'une application pourrait avoir le monopole de l'exécution de ses tâches)

Dans le code actuel de PGLock(); est-ce qu'une tentative d'exécution de cron est empêchée par le verrou de global de migration?

#34 - 30 avril 2018 21:27 - Christophe Siraut

Dans le code actuel de PGLock(); est-ce qu'une tentative d'exécution de cron est empêchée par le verrou de global de migration?

je ne comprends pas encore pourquoi on doit dupliquer try_lock et take_lock, je réessaie demain.

#35 - 30 avril 2018 22:02 - Benjamin Dauvergne

Christophe Siraut a écrit :

Dans le code actuel de PGLock(); est-ce qu'une tentative d'exécution de cron est empêchée par le verrou de global de migration?

Oui toujours, j'ai pensé que c'était plus logique de faire comme cela, par contre je ne renvoie pas de code retour particulier, je ne sais pas si CommandError le permet, ce serait peut-être utile pour ne pas émettre de mail d'erreur inutile, mais bon c'est un détail.

#36 - 30 avril 2018 22:04 - Benjamin Dauvergne

Christophe Siraut a écrit :

Dans le code actuel de PGLock(); est-ce qu'une tentative d'exécution de cron est empêchée par le verrou de global de migration?

je ne comprends pas encore pourquoi on doit dupliquer try_lock et take_lock, je réessaie demain.

C'est juste que try_lock() renvoie une réponse tout de suite alors que take_lock prend un lock au minimum d'1s, j'aurai pu jouer la dessus et produire moins de code, je voulais juste jouer avec l'API de postgresql, aucun souci pour simplifier.

#37 - 14 mai 2018 09:31 - Christophe Siraut

Dans quel cas d'utilisation de tenant_command.py voudrait-on spécifier manuellement un lock_id? Au lieu d'ajouter l'option '--lock' pourrait-on utiliser le nom de l'application (authentic/combo/etc) comme identifiant de verrou?

je ne comprends pas encore pourquoi on doit dupliquer try_lock et take_lock, je réessaie demain.

C'est juste que try_lock() renvoie une réponse tout de suite alors que take_lock prend un lock au minimum d'1s, j'aurai pu jouer la dessus et produire moins de code, je voulais juste jouer avec l'API de postgresql, aucun souci pour simplifier.

oui ça me plairait bien de simplifier, (ou de rajouter du texte sur ce à quoi correspond take_lock et try_lock); par ailleurs le block suivant ne me semble pas logique (soit on attend, soit on attend pas):

```
+         if self.wait == 0:
+             [...]
+         else:
+             if self.wait:
```

#38 - 14 mai 2018 09:35 - Christophe Siraut

- Assigné à changé de Thomas Noël à Benjamin Dauvergne

(et le patch doit être rebasé sur le code actuel, pas compliqué:)

#39 - 14 mai 2018 10:18 - Benjamin Dauvergne

Christophe Siraut a écrit :

Dans quel cas d'utilisation de tenant_command.py voudrait-on spécifier manuellement un lock_id? Au lieu d'ajouter l'option '--lock' pourrait-on utiliser le nom de l'application (authentic/combo/etc) comme identifiant de verrou?

De façon évidente non, postgresql ne supporte que des entiers comme identifiant de lock (32 ou 64 bits, je crois). Maintenant on pourrait générer un entier à partir d'une chaîne via une fonction de hash (genre (str.hash() % 0xFFFF) << 12, on se laisse 12bits bas de marge).

De base ça n'a pas vraiment d'importance on peut sans trop de soucis utiliser le lock_id "1" pour tout le monde au départ, et faire de --lock un simple booléen et plus un paramètre entier.

je ne comprends pas encore pourquoi on doit dupliquer try_lock et take_lock, je réessaie demain.

C'est juste que try_lock() renvoie une réponse tout de suite alors que take_lock prend un lock au minimum d'1s, j'aurai pu jouer la dessus et produire moins de code, je voulais juste jouer avec l'API de postgresql, aucun souci pour simplifier.

oui ça me plairait bien de simplifier, (ou de rajouter du texte sur ce à quoi correspond take_lock et try_lock); par ailleurs le block suivant ne me semble pas logique (soit on attend, soit on attend pas):

[...]

Ok, je vais voir pour faire ça et supprimer vraisemblablement lock_id.

#40 - 04 mars 2019 11:19 - Benjamin Dauvergne

- Statut changé de *En cours* à *Information nécessaire*

- Assigné à *changé de Benjamin Dauvergne* à *Christophe Siraut*

Je t'assigne voir si on s'en fout et tu peux fermer, ou si ça t'intéresse à terme.

#41 - 19 avril 2019 13:59 - Christophe Siraut

- Assigné à *Christophe Siraut* supprimé

#42 - 19 avril 2019 14:00 - Christophe Siraut

- Statut changé de *Information nécessaire* à *Fermé*

Fichiers

0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch	1,8 ko	05 avril 2017	Frédéric Péters
0002-management-commands-tenant_command.py-add-help-messa.patch	1,36 ko	11 avril 2018	Christophe Siraut
0003-management-commands-tenant_command.py-notify-task-di.patch	1,52 ko	11 avril 2018	Christophe Siraut
0004-management-commands-tenant_command.py-only-consider-.patch	25 ko	11 avril 2018	Christophe Siraut
0005-debian_config_common-document-option-DISABLE_CRON_JO.patch	1,03 ko	16 avril 2018	Christophe Siraut
0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch	2,66 ko	17 avril 2018	Christophe Siraut
0002-debian-debian_config_common.py-add-inline-documentat.pat	960 octets	17 avril 2018	Christophe Siraut
0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch	3,33 ko	17 avril 2018	Christophe Siraut
0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch	3,3 ko	18 avril 2018	Christophe Siraut
0001-general-add-possibility-to-skip-all-cron-jobs-15470.patch	3,39 ko	19 avril 2018	Christophe Siraut
0001-use-pg_advisory_lock-to-prevent-running-cronjobs-dur.patch	4,12 ko	27 avril 2018	Benjamin Dauvergne
0001-use-pg_advisory_lock-to-prevent-running-cronjobs-dur.patch	4,83 ko	27 avril 2018	Benjamin Dauvergne
0002-use-pg_advisory_lock-to-prevent-running-cronjobs-dur.patch	8 ko	30 avril 2018	Benjamin Dauvergne
0001-tests_multitenant-PEP8ness-and-cleanup-15470.patch	3,02 ko	30 avril 2018	Benjamin Dauvergne