

w.c.s. - Bug #18519

cron qui s'emballa

07 septembre 2017 17:19 - Thomas Noël

Statut:	Fermé	Début:	07 septembre 2017							
Priorité:	Normal	Echéance:								
Assigné à:		% réalisé:	0%							
Catégorie:		Temps estimé:	0:00 heure							
Version cible:		Planning:								
Patch proposed:	Oui									
Description										
Vu sur auquo.entrouvert.org :										
root	10672	0.0	0.0	61300	1552	?	S	17:00	0:00	_ /usr/sbin/CRON -f
wcs	10674	0.0	0.0	4284	636	?	Ss	17:00	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	10677	0.0	0.0	4284	684	?	S	17:00	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	10685	36.8	1.4	347716	185548	?	R	17:00	5:03	_ python /usr/lib/w
cs/manage.py cron										
root	11390	0.0	0.0	61300	1552	?	S	17:01	0:00	_ /usr/sbin/CRON -f
wcs	11391	0.0	0.0	4284	640	?	Ss	17:01	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	11392	0.0	0.0	4284	688	?	S	17:01	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	11394	0.3	0.6	167236	75756	?	S	17:01	0:02	_ python /usr/lib/w
cs/manage.py cron										
root	11685	0.0	0.0	61300	1552	?	S	17:02	0:00	_ /usr/sbin/CRON -f
wcs	11687	0.0	0.0	4284	644	?	Ss	17:02	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	11689	0.0	0.0	4284	684	?	S	17:02	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	11692	0.5	0.6	167236	75748	?	S	17:02	0:03	_ python /usr/lib/w
cs/manage.py cron										
root	11984	0.0	0.0	61300	1552	?	S	17:03	0:00	_ /usr/sbin/CRON -f
wcs	11985	0.0	0.0	4284	644	?	Ss	17:03	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	11986	0.0	0.0	4284	684	?	S	17:03	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	11988	0.4	0.6	167236	75752	?	S	17:03	0:02	_ python /usr/lib/w
cs/manage.py cron										
root	12353	0.0	0.0	61300	1552	?	S	17:04	0:00	_ /usr/sbin/CRON -f
wcs	12355	0.0	0.0	4284	640	?	Ss	17:04	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	12357	0.0	0.0	4284	684	?	S	17:04	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	12360	0.5	0.6	167236	75748	?	S	17:04	0:03	_ python /usr/lib/w
cs/manage.py cron										
root	12627	0.0	0.0	61300	1552	?	S	17:05	0:00	_ /usr/sbin/CRON -f
wcs	12629	0.0	0.0	4284	644	?	Ss	17:05	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	12630	0.0	0.0	4284	688	?	S	17:05	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	12632	0.6	0.6	167236	75752	?	S	17:05	0:03	_ python /usr/lib/w
cs/manage.py cron										
root	13170	0.0	0.0	61300	1552	?	S	17:06	0:00	_ /usr/sbin/CRON -f
wcs	13172	0.0	0.0	4284	640	?	Ss	17:06	0:00	_ /bin/sh -c /usr/bin/wcs-m
anage cron										
wcs	13174	0.0	0.0	4284	684	?	S	17:06	0:00	_ /bin/sh /usr/bin/wcs-
manage cron										
wcs	13178	0.6	0.6	167236	75748	?	S	17:06	0:03	_ python /usr/lib/w
cs/manage.py cron										

root	13349	0.0	0.0	61300	1552	?	S	17:07	0:00	_ /usr/sbin/CRON -f
wcs	13350	0.0	0.0	4284	644	?	Ss	17:07	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	13351	0.0	0.0	4284	688	?	S	17:07	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	13353	0.8	0.6	167236	75752	?	S	17:07	0:03	_ python /usr/lib/wcs/manage.py cron
root	13662	0.0	0.0	61300	1552	?	S	17:08	0:00	_ /usr/sbin/CRON -f
wcs	13664	0.0	0.0	4284	640	?	Ss	17:08	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	13665	0.0	0.0	4284	684	?	S	17:08	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	13669	0.8	0.6	167236	75756	?	S	17:08	0:03	_ python /usr/lib/wcs/manage.py cron
root	13868	0.0	0.0	61300	1552	?	S	17:09	0:00	_ /usr/sbin/CRON -f
wcs	13869	0.0	0.0	4284	640	?	Ss	17:09	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	13870	0.0	0.0	4284	684	?	S	17:09	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	13872	1.1	0.6	167236	75752	?	S	17:09	0:03	_ python /usr/lib/wcs/manage.py cron
root	14226	0.0	0.0	61300	1552	?	S	17:10	0:00	_ /usr/sbin/CRON -f
wcs	14228	0.0	0.0	4284	640	?	Ss	17:10	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	14229	0.0	0.0	4284	684	?	S	17:10	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	14233	1.3	0.6	167236	75756	?	S	17:10	0:02	_ python /usr/lib/wcs/manage.py cron
root	14828	0.0	0.0	61300	1552	?	S	17:11	0:00	_ /usr/sbin/CRON -f
wcs	14829	0.0	0.0	4284	640	?	Ss	17:11	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	14830	0.0	0.0	4284	684	?	S	17:11	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	14832	1.6	0.6	167236	75752	?	S	17:11	0:02	_ python /usr/lib/wcs/manage.py cron
root	15142	0.0	0.0	61300	1552	?	S	17:12	0:00	_ /usr/sbin/CRON -f
wcs	15143	0.0	0.0	4284	644	?	Ss	17:12	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	15145	0.0	0.0	4284	684	?	S	17:12	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	15149	2.9	0.6	167236	75752	?	S	17:12	0:03	_ python /usr/lib/wcs/manage.py cron
root	15340	0.0	0.0	61300	1552	?	S	17:13	0:00	_ /usr/sbin/CRON -f
wcs	15341	0.0	0.0	4284	640	?	Ss	17:13	0:00	_ /bin/sh -c /usr/bin/wcs-manage cron
wcs	15342	0.0	0.0	4284	688	?	S	17:13	0:00	_ /bin/sh /usr/bin/wcs-manage cron
wcs	15344	6.1	0.6	167236	75752	?	S	17:13	0:02	_ python /usr/lib/wcs/manage.py cron

Comme le cron est lancé toute les minutes et ne vérifie par qu'un autre cron est en cours, on a jusqu'à 15 cron lancés en parallèle (quand on tombe sur des cron qui gère des centaines d'expiration à vérifier).

Mais si on lock, un seul workflow lent pourra bloquer toute la mécanique pendant un gros bout de temps.

Révisions associées

Révision 5b582638 - 26 mars 2018 15:35 - Thomas Noël

don't start a new cron if locked (#18519)

Historique

#1 - 07 septembre 2017 17:21 - Thomas Noël

Idées en vrac :

- lancer un process "cron" par instance
- voire un process par formdef

#2 - 07 septembre 2017 17:55 - Benjamin Dauvergne

Utiliser systemd:

wcs-cron.service

```
[Unit]
Description=w.c.s. cron
```

```
[Service]
ExecStart=/usr/bin/wcs-manage cron
```

wcs-cron.timer

```
[Unit]
Description=w.c.s. cron
```

```
[Timer]
OnCalendar=*-*-* *:*:*
AccuracySec=1 minute
```

#3 - 07 septembre 2017 17:55 - Benjamin Dauvergne

systemd ne lance la commande qu'une fois et attend qu'elle termine par défaut.

#4 - 08 septembre 2017 00:10 - Thomas Noël

- Fichier 0001-don-t-start-a-new-cron-if-locked-18519.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

Je propose déjà de ne pas lancer un nouveau cron si le lock est présent (actuellement on attend que le lock se libère)

#5 - 08 septembre 2017 00:37 - Thomas Noël

Ensuite j'imaginerai bien un multiprocessing.pool qui itère sur les tenants, afin d'avoir un process par tenant, genre (poc/wip absolument pas testé) :

```
import tempfile
import time
import os
import multiprocessing.pool

from django.core.management.base import BaseCommand
from qommon.publisher import get_publisher_class

from qommon import get_logger
from qommon.vendor import locket
from qommon.cron import cron_worker

def launch_cron(context):
    publisher = context['publisher']
    app_dir = context['app_dir']
    hostname = context['hostname']
    now = context['now']
    publisher.app_dir = os.path.join(app_dir, hostname)
    cron_worker(publisher, now)

class Command(BaseCommand):
    help = 'Execute cronjobs'

    def handle(self, verbosity, **options):
        lockfile = os.path.join(tempfile.gettempdir(), 'wcs-cron-in-progress.lock')
        try:
            with locket.lock_file(lockfile, timeout=0):
                now = time.localtime()
                publisher_class = get_publisher_class()
                publisher_class.register_cron = True
                publisher = publisher_class.create_publisher()
                app_dir = publisher.app_dir

                contexts = []
                for hostname in publisher.get_tenants():
                    contexts.append({
```

```
'publisher': publisher,  
'app_dir': app_dir,  
'hostname': hostname,  
'now': now,  
})
```

```
pool = multiprocessing.pool.ThreadPool()  
try:  
    list(pool.imap_unordered(launch_cron, contexts))  
finally:  
    pool.close()  
    pool.terminate()
```

```
except locket.LockError:  
    get_logger().warn('can not start cron job - locked by %s' % lockfile)
```

#6 - 21 mars 2018 16:56 - Thomas Noël

- Fichier 0001-cron-fork-each-job-in-each-tenant-with-a-lock-18519.patch ajouté

Dans l'idée de granulariser plus, tout en lockant mieux, un patch qui lance les cron en parallèle, avec chacun son lock (c'est un draft, ça met plein de lock dans /var/lib/wcs/site/, pas super joli).

Ca permettrait de mieux gérer les opération lourdes telles que les reindexations fts, sans (trop) bloquer le reste.

Par défaut ça lance autant de cron que de coeurs sur la machine, et donc on serait quand même bloqués si on a plus de sites que de coeurs, on pourrait ne pas avoir cette limite, voire même catégoriser deux familles de cron, ceux qui peuvent être lancés "tous en parallèle" (genre les sauts) et ceux qu'on limite au nombre de coeurs (reindexation fts).

#8 - 23 mars 2018 18:52 - Thomas Noël

- Fichier 0001-don-t-start-a-new-cron-if-locked-18519.patch ajouté

Retour dans le passé, sur l'objet de départ du ticket, à savoir des crons qui s'attendent les uns les autres... si la durée d'un cron est supérieure à la minute et qu'il est lancé chaque minute (JUMP_TIMEOUT_INTERVAL=60), ça peut vite devenir la cata.

Je reviens donc à ma première proposition, à savoir que le cron abandonne en présence d'un lock. Patch rebasé sur la commande Django, et avec des tests.

(on pense à passer à systemd aussi, mais pour l'instant on n'a pas de solution pour concilier les deux, or il faut continuer à gérer les machines sans systemd)

#9 - 26 mars 2018 14:32 - Emmanuel Cazenave

- il me semble qu'on a pas besoin pas besoin d'appeler super(Command, self).add_arguments(parser)
- pas très fan du fait que l'option de verbosité ait un impact sur l'exit code du processus. Je trouverai plus propre si une autre option (--fail-if-lock ?) déclenchait l'exception en cas de lock.
- en parlant de verbosité pour faciliter l'analyse, puisque le cron peut durer et que ça commence à poser problème, peut être afficher sur stdout à quel tenant en est le cron, avec un compteur (genre "demarches-fondettes.xxxx.yyyy 6/40) ?

#10 - 26 mars 2018 15:36 - Thomas Noël

- Fichier 0001-don-t-start-a-new-cron-if-locked-18519.patch ajouté

Yep pour tout. J'ai donc juste posé des print selon la verbosité, y compris à la place du CommandError.

#11 - 26 mars 2018 17:00 - Emmanuel Cazenave

si je pinaille je dirais que tu n'as pas mis le compteur qui dirait combien de tenant on a déjà fait et combien il reste à faire, mais je te laisse juger, ack.

#12 - 26 mars 2018 17:37 - Thomas Noël

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

```
commit 5b582638d532c65a419c561a87ab7263ab8fd5a9  
Author: Thomas NOEL <tnoel@entrouvert.com>  
Date: Fri Mar 23 18:47:42 2018 +0100
```

```
don't start a new cron if locked (#18519)
```

#13 - 23 décembre 2018 14:46 - Frédéric Péters

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

Fichiers

0001-don-t-start-a-new-cron-if-locked-18519.patch	2,08 ko	07 septembre 2017	Thomas Noël
0001-cron-fork-each-job-in-each-tenant-with-a-lock-18519.patch	6,48 ko	21 mars 2018	Thomas Noël
0001-don-t-start-a-new-cron-if-locked-18519.patch	4,45 ko	23 mars 2018	Thomas Noël
0001-don-t-start-a-new-cron-if-locked-18519.patch	4,23 ko	26 mars 2018	Thomas Noël