

Hobo - Development #13617

lancer les tâches de l'agent en parallèle (?)

16 octobre 2016 19:39 - Frédéric Péters

| | | | |
|---|--------|----------------------|-----------------|
| Statut: | Fermé | Début: | 16 octobre 2016 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | |
| Patch proposed: | Oui | | |
| Description | | | |
| Aujourd'hui l'agent reçoit un message et lance hobo_notify pour les différents services concernés; dans notre archi SaaS avec un service par VM ça passe inaperçu, dans une archi avec tous les services sur le VM ça peut se sentir; il me semble qu'on pourrait utiliser multiprocessing pour faire digérer le message à plusieurs services en même temps. (pool.imap_unordered(...)) | | | |

Révisions associées

Révision f2a9d917 - 16 janvier 2017 10:25 - Frédéric Péters

worker: run hobo processes in parallel (#13617)

Historique

#1 - 17 octobre 2016 10:59 - Benjamin Dauvergne

À mon avis on peut juste utiliser des threads, tout ce que fait l'agent c'est des fork (et je pense qu'on peut forker dans un thread).

#2 - 14 janvier 2017 15:14 - Frédéric Péters

- Fichier 0001-worker-run-hobo-processes-in-parallel-13617.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Le gain est significatif en local (même si plutôt $\times 2$ que $\times 4$, pour avoir l'ensemble des deploy/notify exécutés).

#3 - 15 janvier 2017 01:34 - Thomas Noël

a priori processes=multiprocessing.cpu_count() c'est la valeur par défaut (et ne rien mettre évitera un crash si NotImplementedError).

#4 - 15 janvier 2017 16:11 - Frédéric Péters

- Fichier 0001-worker-run-hobo-processes-in-parallel-13617.patch ajouté

En effet, "if processes is None then the number returned by cpu_count() is used".

#5 - 16 janvier 2017 09:38 - Thomas Noël

Une nuit plus tard, y'a encore une raison qui m'échappe dans l'utilisation de « pool.imap_unordered » au lieu de « pool.map ».

#6 - 16 janvier 2017 09:45 - Frédéric Péters

Sans connaître les détails d'implémentation derrière, signifier explicitement qu'on se fout de l'ordre des résultats me paraissait une indication qui pouvait être utile.

#7 - 16 janvier 2017 09:48 - Thomas Noël

C'est bien possible (et je vais pas lire les implémentations non plus). Ack.

#8 - 16 janvier 2017 10:25 - Frédéric Péters

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

```
commit f2a9d9170f208f249fcc873a66759916e0ca271f
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sat Jan 14 15:10:51 2017 +0100
```

```
worker: run hobo processes in parallel (#13617)
```

#9 - 16 janvier 2017 10:30 - Benjamin Dauvergne

La différence c'est qu'`imap_unordered` renvoie je pense un générateur avec les résultats dans l'ordre où ils sont produits alors que `map` renvoie une liste qui respecte l'ordre initial et donc ça attend le dernier résultat avant de renvoyer quelque chose. Si on fait un pipeline de tâches (i.e. plusieurs `map` en cascade qui se nourrissent les uns les autres), chaque `map()` va bloquer alors que `imap_ordered()` permettra au pipeline d'avancer en parallèle.

#10 - 23 décembre 2018 16:02 - Frédéric Péters

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

Fichiers

| | | | |
|--|---------|-----------------|-----------------|
| 0001-worker-run-hobo-processes-in-parallel-13617.patch | 1,8 ko | 14 janvier 2017 | Frédéric Péters |
| 0001-worker-run-hobo-processes-in-parallel-13617.patch | 1,72 ko | 15 janvier 2017 | Frédéric Péters |