

Mandaye - Development #14913

La gestion du Timeout générale ne marche pas sur toutes les applis

08 février 2017 09:29 - Josué Kouka

Statut:	Fermé	Début:	08 février 2017
Priorité:	Normal	Echéance:	
Assigné à:	Josué Kouka	% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Dans le cas de Hermes (Archimed) il y'a des moments où l'on a un timeout alors que le processus phantomjs a terminé son job.			

Révisions associées

Révision ae988f51 - 09 février 2017 15:11 - Josué Kouka

improve generic timeout handling (#14913)

Historique

#1 - 08 février 2017 09:29 - Josué Kouka

- Fichier 0001-improve-generic-timeout-handling-14913.patch ajouté
- Patch proposed changé de Non à Oui

#2 - 08 février 2017 09:41 - Josué Kouka

Dans tests/settings.py le PHANTOM_JS_TIMEOUT est changé parce que signal.alarm ne prend que des entiers.

#3 - 08 février 2017 09:55 - Frédéric Péters

Ce serait bien de noter où et pourquoi ça ne fonctionnait pas.

#4 - 08 février 2017 16:12 - Josué Kouka

- Fichier 0001-improve-generic-timeout-handling-14913.patch ajouté
- Description mis à jour

Frédéric Péters a écrit :

Ce serait bien de noter où et pourquoi ça ne fonctionnait pas.

En cherchant mieux, les raisons pour lesquelles ça ne fonctionne pas sont les suivantes (principalement liées à mon implémentation) :

- deadlock (<https://docs.python.org/2.7/library/multiprocessing.html#all-platforms>)

```
process.start()
process.join(settings.PHANTOM_JS_TIMEOUT)
....
result = recv_end.recv()
```

Bizarrement, quand des infos d'une petite taille sont envoyées via le pipe, il n'y a aucun dysfonctionnement. C'est ce qui m'a induit en erreur.

- dans le run, je ne ferme pas le bout du Pipe (send_end) qui envoie le message (<https://docs.python.org/2.7/library/multiprocessing.html#exchanging-objects-between-processes>)

Note that data in a pipe may become corrupted if two processes (or threads) try to read from or write to the same end of the pipe at the same time. Of course there is no risk of corruption from processes using different ends of the pipe at the same time.

#5 - 08 février 2017 16:53 - Josué Kouka

- Fichier 0001-improve-generic-timeout-handling-14913.patch ajouté

- Utilisation de connection.poll pour "timeouter" le sender vu qu'un process.join() n'a pas d'effet sur un Pipe

#6 - 08 février 2017 17:09 - Josué Kouka

- Fichier 0001-improve-generic-timeout-handling-14913.patch ajouté

Sans modification des tests.

#7 - 08 février 2017 17:31 - Josué Kouka

- Fichier 0001-improve-generic-timeout-handling-14913.patch ajouté

rebase on master

#8 - 08 février 2017 22:18 - Serghei Mihai

Testé avec Ermes et Teamnet et ça fonctionne même si pour Ermes le timeout de 10 secondes souvent n'est pas suffisant.
Ack.

#9 - 09 février 2017 17:05 - Josué Kouka

- Fichier 0001-improve-generic-timeout-handling-14913.patch ajouté

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

- % réalisé changé de 0 à 100

Ok

#10 - 04 décembre 2018 20:07 - Frédéric Péters

- Statut changé de *Résolu* (à déployer) à *Fermé*

Fichiers

0001-improve-generic-timeout-handling-14913.patch	4,08 ko	08 février 2017	Josué Kouka
0001-improve-generic-timeout-handling-14913.patch	3,27 ko	08 février 2017	Josué Kouka
0001-improve-generic-timeout-handling-14913.patch	4,59 ko	08 février 2017	Josué Kouka
0001-improve-generic-timeout-handling-14913.patch	3,4 ko	08 février 2017	Josué Kouka
0001-improve-generic-timeout-handling-14913.patch	3,3 ko	08 février 2017	Josué Kouka
0001-improve-generic-timeout-handling-14913.patch	3,38 ko	09 février 2017	Josué Kouka