

Passerelle - Development #39668

connection pooling

09 février 2020 11:26 - Frédéric Péters

Statut:	Fermé	Début:	09 février 2020
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Je faisais des tests avec [#39387](#) et comme ça va être utilisé pour de l'autocomplétion, je regardais un peu les performances et il y a un coût évident à passer par passerelle, plutôt que directement interroger geo.api.gouv.fr, mais peut-être que ce coût peut être réduit.

À regarder très grossièrement, le coût principal c'est la requête HTTP, et une idée pour réduire ça ce serait de profiter des possibilités de connection pooling et keep alive HTTP de requests (via urllib3).

De la manière la plus brutale qui soit ce serait de faire ça sur tout Passerelle,

```
--- a/passerelle/utils/__init__.py
+++ b/passerelle/utils/__init__.py
@@ -24,6 +24,7 @@ import warnings

from requests import Session as RequestSession, Response as RequestResponse
from requests.structures import CaseInsensitiveDict
+from requests.adapters import HTTPAdapter
from urllib3.exceptions import InsecureRequestWarning

from django.conf import settings
@@ -201,6 +202,10 @@ def log_http_request(logger, request, response=None, exception=None, error_lo
g=T
    log_function(message, extra=extra)

+# connection pooling and HTTP keep-alive support
+http_adapter = HTTPAdapter()
+https_adapter = HTTPAdapter()
+
# Wrapper around requests.Session
# - log input and output data
# - use HTTP Basic auth if resource.basic_auth_username and resource.basic_auth_password exist
@@ -214,6 +219,8 @@ class Request(RequestSession):
    self.logger = kwargs.pop('logger')
    self.resource = kwargs.pop('resource', None)
    super(Request, self).__init__(*args, **kwargs)
+    self.mount('https://', https_adapter)
+    self.mount('http://', http_adapter)

    def request(self, method, url, **kwargs):
        cache_duration = kwargs.pop('cache_duration', None)

mais on pourrait aussi se dire qu'il suffit de faire ça au niveau du connecteur, que ça aura en plus l'avantage de ne pas éjecter geo.api.gouv.fr du pool de connexions, ou d'encombrer le pool avec des connexions où la persistance n'a pas d'intérêt,
```

```
    super(Request, self).__init__(*args, **kwargs)
+    if hasattr(self.resource, 'https_adapter'):
+        self.mount('https://', self.resource.https_adapter)
+    if hasattr(self.resource, 'http_adapter'):
+        self.mount('http://', self.resource.http_adapter)
```

Pour mesurer un peu le bénéfice, j'ai fait des tests, script attaché, graphe attaché.

Révisions associées

Révision 8cbbd99a - 15 février 2020 10:01 - Frédéric Péters

general: use connection pooling and HTTP Keep-Alive (#39668)

Historique

#1 - 09 février 2020 11:27 - Frédéric Péters

- Fichier 0001-base-adresse-use-connection-pooling-and-HTTP-Keep-A.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

#2 - 09 février 2020 11:28 - Frédéric Péters

- Fichier Figure_1.png supprimé

#3 - 09 février 2020 11:29 - Frédéric Péters

- Fichier Figure_1.png ajouté

#4 - 09 février 2020 15:15 - Benjamin Dauvergne

Je ne vois que des bénéfiques, mais comme je ne vois pas de cas où on voudrait un adapter différents ce celui-ci je rendrai le support moins générique, du style :

```
ADAPTER_REGISTRY = {}

if getattr(self.resource, 'use_keep_alive', False):
    adapter = ADAPTER_REGISTRY.setdefault(type(self.resource), HttpAdapter())
    self.mounts('https://', adapter)
    self.mounts('http://', adapter)
```

Sauf tu vois un besoin proche de différencier http: et https: ou de mutualiser l'adapter entre plus d'un type de connecteur.

#5 - 10 février 2020 10:54 - Frédéric Péters

```
if getattr(self.resource, 'use_keep_alive', False):
```

Mais limite on pourrait se dire qu'on le fait de manière générale (par connecteur), sans demander d'opt-in du connecteur ?

#6 - 10 février 2020 11:47 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Oui ça me va aussi.

#7 - 10 février 2020 11:58 - Frédéric Péters

- Statut changé de Solution validée à En cours

Ok je vais écrire le patch.

#8 - 14 février 2020 08:41 - Frédéric Péters

- Fichier 0001-general-use-connection-pooling-and-HTTP-Keep-Alive-3.patch ajouté
- Statut changé de En cours à Solution proposée

#9 - 14 février 2020 17:22 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#10 - 15 février 2020 10:02 - Frédéric Péters

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

```
commit 8cbbd99afc9d52cc4fcb6e2ade9be03a4f75b0a3
Author: Frédéric Péters <fpeters@entrouvert.com>
```

Date: Sun Feb 9 11:25:02 2020 +0100

general: use connection pooling and HTTP Keep-Alive (#39668)

#11 - 19 février 2020 18:15 - Frédéric Péters

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

Fichiers

test-pooling.py	1,1 ko	09 février 2020	Frédéric Péters
0001-base-adresse-use-connection-pooling-and-HTTP-Keep-A.patch	1,87 ko	09 février 2020	Frédéric Péters
Figure_1.png	56,8 ko	09 février 2020	Frédéric Péters
0001-general-use-connection-pooling-and-HTTP-Keep-Alive-3.patch	1,57 ko	14 février 2020	Frédéric Péters