

Passerelle - Development #32413

avoir un outil pour écrire de manière atomique

17 avril 2019 11:36 - Benjamin Dauvergne

Statut:	Fermé	Début:	17 avril 2019
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description voir code dans actesweb (git grep fsync dans passerelle).			
Demandes liées: Lié à Passerelle - Development #32091: cityweb: ne pas compresser les fichier... Fermé 08 avril 2019			

Révisions associées

Révision a52c914e - 17 avril 2019 16:42 - Benjamin Dauvergne

utils: add an atomic_write() context manager (#32413)

Révision 72489a17 - 19 avril 2019 11:19 - Benjamin Dauvergne

Revert "utils: add an atomic_write() context manager (#32413)"

This reverts commit a52c914e8f1a108049b9cc1586f2490825d04ce4.

Révision c778669e - 19 avril 2019 17:46 - Benjamin Dauvergne

utils: add an atomic_write() context manager (#32413)

Historique

#1 - 17 avril 2019 11:37 - Benjamin Dauvergne

Peut-être factorisé le code entre les deux et mettre ce qu'il faut dans passerelle.utils.

#2 - 17 avril 2019 12:06 - Benjamin Dauvergne

- Fichier 0001-utils-add-an-atomic_write-context-manager-32413.patch ajouté

- Patch proposed changé de Non à Oui

Ça pourra aider.

#3 - 17 avril 2019 12:22 - Thomas Noël

J'imposerais bien le "dir", parce qu'en fait il faut éviter de créer des "tmpxxx" dans le répertoire de travail (où un client FTP va venir chercher ses choses). Genre :

```
def atomic_write(filepath, tmpdir):
    '''Return a file descriptor to a temporary file using NamedTemporaryFile
    which will be atomically renamed to filepath if possible.

    Atomic renaming is only possible on the same filesystem, so the
    tmpdir must be in the same filesystem as filepath.
    '''
    fd = tempfile.NamedTemporaryFile(dir=tmpdir, delete=False)
    try:
        ...
```

#4 - 17 avril 2019 13:32 - Benjamin Dauvergne

Thomas Noël a écrit :

J'imposerais bien le "dir", parce qu'en fait il faut éviter de créer des "tmpxxx" dans le répertoire de travail (où un client FTP va venir chercher ses choses). Genre :

[...]

Dans ce cas j'imposerai bien un:

```
import errno

tmpdir = kwargs.get('dir')
if not tmpdir:
    tmpdir = os.path.join(settings.MEDIA_DIR, 'tmp')
    if not os.path.exists(tmpdir):
        try:
            os.makedirs(tmpdir)
        except OSError as e:
            if e.errno != errno.EEXIST:
                raise
```

dans l'idée que ce soit le plus simple à réutiliser possible (et si un jour on ne met pas tout /var/lib/passerelle/<domain>/media/ sur la même partoche, on se posera alors la question de faire autrement).

#5 - 17 avril 2019 13:33 - Benjamin Dauvergne

Et celui qui créera un connecteur nommé 'tmp' qui stocke des fichiers sera châtié.

#6 - 17 avril 2019 14:04 - Serghei Mihai

Pourquoi pas un .tmp?

Cela nous évitera à penser au chatiement.

#7 - 17 avril 2019 14:20 - Thomas Noël

Allons-y pour un classique et sans surprise "tmp" (je laisse l'attribution à Serghei mais je ne sais pas si Benjamin va s'en occuper (mini-congés))

#8 - 17 avril 2019 14:56 - Serghei Mihai

Benjamin va relire :)

#9 - 17 avril 2019 15:02 - Thomas Noël

Heu... Relire quoi ?... Au pire je peux (re)faire son patch, si tu es pressé et pas trop dispo

#10 - 17 avril 2019 15:06 - Serghei Mihai

- Assigné à changé de Serghei Mihai à Thomas Noël

Relire le patch appliqué au connecteur cityweb et actesweb. Passons.

Je veux bien un coup de main si tu es disponible.

#11 - 17 avril 2019 15:43 - Thomas Noël

- Assigné à changé de Thomas Noël à Serghei Mihai

Discuté avec Serghei : on attend de savoir si les clients cityweb peuvent consommer du XML non-zipé pour savoir s'il y aura quelque chose à factoriser. (J'attribue à Serghei qui a demandé aux clients concernés)

#12 - 17 avril 2019 16:23 - Serghei Mihai

- Lié à Development #32091: cityweb: ne pas compresser les fichiers avec les demandes ajouté

#13 - 17 avril 2019 17:40 - Benjamin Dauvergne

Thomas Noël a écrit :

Discuté avec Serghei : on attend de savoir si les clients cityweb peuvent consommer du XML non-zipé pour savoir s'il y aura quelque chose à factoriser. (J'attribue à Serghei qui a demandé aux clients concernés)

C'est quoi le rapport ? Zip ou XML il faut les créer atomiquement.

#14 - 17 avril 2019 17:48 - Serghei Mihai

Quel intérêt d'écrire atomiquement un XML puis le ZIP dans lequel on le place?

Pour moi, on ajoute la fonction `atomic_write` et on écrit des XML avec cette fonction.

#15 - 17 avril 2019 17:55 - Serghei Mihai

Je me dis qu'on devrait utiliser le `atomic_write` au niveau du `save` du `default_storage`. Ainsi toutes les écritures de fichiers dans Publik gagnent l'atomicité.

#16 - 17 avril 2019 18:09 - Serghei Mihai

- Fichier `0001-utils-add-an-atomic_write-context-manager-32413.patch` ajouté

- Fichier `0002-cityweb-do-not-compress-demand-files-32091.patch` ajouté

- Tracker changé de Support à Bug

- Statut changé de Nouveau à Solution proposée

En lisant le code d'écriture des fichiers par les storages de Django je vais me calmer.

Patch initial de Benjamin repris pour s'assurer que les sous-répertoires du fichier existent. Et cityweb qui écrit le XML de manière atomique.

#17 - 17 avril 2019 21:19 - Benjamin Dauvergne

Euh..

```
target_dir = kwargs.get('dir') or os.path.dirname(filepath)
if not os.path.exists(target_dir):
    os.makedirs(target_dir)
```

C'est le merge de Frankenstein.

#18 - 17 avril 2019 21:19 - Benjamin Dauvergne

- Fichier `0001-utils-add-an-atomic_write-context-manager-32413.patch` ajouté

#19 - 17 avril 2019 21:21 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Euh..

[...]

C'est le merge de Frankenstein.

Pas compris que tu ne reprenais pas tout mon code.

#20 - 17 avril 2019 21:23 - Benjamin Dauvergne

Aussi le deuxième patch n'a pas de rapport avec ce ticket...

#21 - 17 avril 2019 23:33 - Serghei Mihai

Benjamin Dauvergne a écrit :

Pas compris que tu ne reprenais pas tout mon code.

J'ai appliqué ton patch en local: il y a 2 fichiers. Ou je rate un truc?

#22 - 18 avril 2019 10:53 - Benjamin Dauvergne

Serghei Mihai a écrit :

Benjamin Dauvergne a écrit :

Pas compris que tu ne reprenais pas tout mon code.

J'ai appliqué ton patch en local: il y a 2 fichiers. Ou je rate un truc?

Non mais j'avais mis en commentaire une modification du code (mais pas testé, d'où l'erreur `MEDIA_DIR/MEDIA_ROOT`) pour répondre au commentaire de Thomas, j'espérais que tu l'intégrerais, je l'ai fait dans le dernier patch que j'ai soumis.

#23 - 18 avril 2019 10:55 - Benjamin Dauvergne

- Sujet changé de cityweb: écrire les demandes de manière atomique à avoir un outil pour écrire de manière atomique
- Assigné à changé de Serghei Mihai à Benjamin Dauvergne

Je renomme le ticket pour ne pas avoir à faire le boulot dans cityweb.

#24 - 18 avril 2019 10:56 - Serghei Mihai

- Fichier 0002-cityweb-do-not-compress-demand-files-32091.patch supprimé

#25 - 18 avril 2019 11:33 - Benjamin Dauvergne

- Fichier 0002-settings-add-default-value-for-FILE_UPLOAD_DIRECTORY.patch ajouté
- Fichier 0003-actesweb-use-atomic_write-32413.patch ajouté
- Fichier 0001-utils-add-an-atomic_write-context-manager-32413.patch ajouté
- Tracker changé de Bug à Development

Voilà j'ai intégré avec actesweb aussi, j'ai revu le calcul et tests des droits parce que Josué avait confondu écriture sur le fichier et écriture sur le répertoire (pour déplacer/supprimer le fichier).

#26 - 18 avril 2019 11:58 - Serghei Mihai

Dans 0001, tu ne vérifies pas que le(s) répertoire(s) du fichier cible existe (c'est ce que fait default_storage.save).

Donc la fonction créera le répertoire temporaire mais pas celui ou le fichier doit être ensuite déplacé.

#27 - 18 avril 2019 12:11 - Serghei Mihai

IMO, la fonction atomic_write doit faire:

```
# create target file directory if it doesn't exist
dirname = os.path.dirname(filepath)
if not os.path.exists(dirname):
    os.makedirs(dirname)
```

#28 - 18 avril 2019 12:13 - Benjamin Dauvergne

- Fichier 0002-settings-add-default-value-for-FILE_UPLOAD_DIRECTORY.patch ajouté
- Fichier 0003-actesweb-use-atomic_write-32413.patch ajouté
- Fichier 0001-utils-add-an-atomic_write-context-manager-32413.patch ajouté

Deux modifications :

- atomic_write s'occupe de créer le répertoire temporaire dans tous les cas (si media/tmp/ est utilisé ou si un répertoire spécifique est désigné)
- actesweb continuera à créer ses fichiers temporaires dans media/actesweb/<slug>/tmp/ pour ne pas avoir à changer le déploiement en production pour Dreux

#29 - 18 avril 2019 12:13 - Benjamin Dauvergne

Serghei Mihai a écrit :

Dans 0001, tu ne vérifies pas que le(s) répertoire(s) du fichier cible existe (c'est ce que fait default_storage.save).

Ce n'est pas le job d'atomic_write, de toute façon comme vu dans actesweb les modules souhaitent des autorisations spécifiques, si on déporte la création du répertoire cela va compliquer les choses sur ce point.

Donc la fonction créera le répertoire temporaire mais pas celui ou le fichier doit être ensuite déplacé.

Ce n'est pas grave, c'est le boulot de l'appelant de savoir où il veut écrire et comment (le but c'est juste d'améliorer la robustesse d'un simple with open(..., 'w') as fd: fd.write(...).

Par contre je me rends compte que vu comment actesweb est déployé à Dreux je dois absolument avoir le répertoire 'tmp' dans le répertoire de destination (actesweb est un lien vers /srv/nfs/chroot/ville-de-dreux/home/ville-de-dreux/actesweb et donc le tmp dans "media" ne marchera pas).

Je trouve un peu dommage ce fonctionnement (de même que les masques de droit qui sont dans le code n'ont rien à faire ici, c'est au déploiement qu'on doit faire des liens vers /srv et mettre les droits d'écriture pour le groupe sur les répertoires et pas sur les fichiers, normalement

FILE_UPLOAD_PERMISSIONS doit suffire à s'assurer que les groupes ont les droits en lecture).

#30 - 18 avril 2019 17:21 - Serghei Mihai

Benjamin Dauvergne a écrit :

Serghei Mihai a écrit :

Dans 0001, tu ne vérifies pas que le(s) répertoire(s) du fichier cible existe (c'est ce que fait default_storage.save).

Ce n'est pas le job d'atomic_write, de toute façon comme vu dans actesweb les modules souhaitent des autorisations spécifiques, si on déporte la création du répertoire cela va compliquer les choses sur ce point.

Ok, donc chaque connecteur devra s'assurer d'avoir son répertoire avant d'appeler atomic_write. Je vais adapter mon patch dans [#32091](#).

Par contre je me rends compte que vu comment actesweb est déployé à Dreux je dois absolument avoir le répertoire 'tmp' dans le répertoire de destination (actesweb est un lien vers /srv/nfs/chroot/ville-de-dreux/home/ville-de-dreux/actesweb et donc le tmp dans "media" ne marchera pas).

C'est fait pour donner accès ssh (pour faire SFTP ou rsync) aux clients. Depuis leur \$HOME on fait un lien symbolique vers le répertoire du connecteur.

C'est le cas pour actesweb mais aussi pour Cityweb et MDEL. Il faudrait alors revoir notre façon de faire.

#31 - 18 avril 2019 18:38 - Benjamin Dauvergne

Serghei Mihai a écrit :

Benjamin Dauvergne a écrit :

Serghei Mihai a écrit :

Dans 0001, tu ne vérifies pas que le(s) répertoire(s) du fichier cible existe (c'est ce que fait default_storage.save).

Ce n'est pas le job d'atomic_write, de toute façon comme vu dans actesweb les modules souhaitent des autorisations spécifiques, si on déporte la création du répertoire cela va compliquer les choses sur ce point.

Ok, donc chaque connecteur devra s'assurer d'avoir son répertoire avant d'appeler atomic_write. Je vais adapter mon patch dans [#32091](#).

Oui comme avant quoi, sinon on est parti pour réécrire FileStorage.

Par contre je me rends compte que vu comment actesweb est déployé à Dreux je dois absolument avoir le répertoire 'tmp' dans le répertoire de destination (actesweb est un lien vers /srv/nfs/chroot/ville-de-dreux/home/ville-de-dreux/actesweb et donc le tmp dans "media" ne marchera pas).

C'est fait pour donner accès ssh (pour faire SFTP ou rsync) aux clients. Depuis leur \$HOME on fait un lien symbolique vers le répertoire du connecteur.

C'est l'inverse qui est fait il y a un lien symbolique du répertoire du connecteur vers le sftp.

C'est le cas pour actesweb mais aussi pour Cityweb et MDEL. Il faudrait alors revoir notre façon de faire.

Le fait est que ces histoires n'ont rien à faire dans le code de passerelle et pour l'instant y sont, un chmod + mount -bind au niveau du serveur sftp aurait suffi.

#32 - 19 avril 2019 11:30 - Serghei Mihai

Benjamin Dauvergne a écrit :

C'est l'inverse qui est fait il y a un lien symbolique du répertoire du connecteur vers le sftp.

Alors on peut faire le atomic_write sans soucis.

Si pour ActesWeb on a fait n'importe quoi on réparera dans un ticket à part.

Frileux, je suis d'avis de pousser uniquement la fonction, puis l'usage de celle-ci dans cityweb ([#32091](#)).

#33 - 19 avril 2019 14:50 - Benjamin Dauvergne

Ok mais faut quand même me valider le ticket :)

#34 - 19 avril 2019 15:14 - Serghei Mihai

- Statut changé de Solution proposée à Solution validée

Go pour le premier patch!

#35 - 19 avril 2019 17:45 - Benjamin Dauvergne

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

```
commit 94b6eb213b71dd8428b4a68ea38703dac4481f6f
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Wed Apr 17 12:06:18 2019 +0200
```

```
utils: add an atomic_write() context manager (#32413)
```

#36 - 23 avril 2019 11:48 - Serghei Mihai

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

Fichiers

0001-utils-add-an-atomic_write-context-manager-32413.patch	3,62 ko	17 avril 2019	Benjamin Dauvergne
0001-utils-add-an-atomic_write-context-manager-32413.patch	4,23 ko	17 avril 2019	Serghei Mihai
0001-utils-add-an-atomic_write-context-manager-32413.patch	4,39 ko	17 avril 2019	Benjamin Dauvergne
0002-settings-add-default-value-for-FILE_UPLOAD_DIRECTORY.patch	7,62 octets	18 avril 2019	Benjamin Dauvergne
0003-actesweb-use-atomic_write-32413.patch	4,72 ko	18 avril 2019	Benjamin Dauvergne
0001-utils-add-an-atomic_write-context-manager-32413.patch	4,39 ko	18 avril 2019	Benjamin Dauvergne
0002-settings-add-default-value-for-FILE_UPLOAD_DIRECTORY.patch	7,62 octets	18 avril 2019	Benjamin Dauvergne
0003-actesweb-use-atomic_write-32413.patch	4,79 ko	18 avril 2019	Benjamin Dauvergne
0001-utils-add-an-atomic_write-context-manager-32413.patch	4,17 ko	18 avril 2019	Benjamin Dauvergne