

Brenda-Blender

Une application pour créer une ferme de rendu chez Amazon Web Services (AWS).

Brenda utilise des "instances EC2 avec des files d'attente SQS et un support S3" chez Amazon pour mettre en œuvre une ferme de rendus distribués en utilisant des instances de travail EC2 à faible coût.

En utilisant Brenda, vous pouvez accélérer des tâches de rendu complexe en répartissant le travail sur des dizaines, des centaines, voire des milliers de machines virtuelles dans le Cloud.

Brenda est spécifiquement conçu pour tirer parti du marché des instances EC2 à moindre coût et avec une tolérance de pannes sur ces instances en cours de création et qui peuvent se terminer au hasard en cours de tâches, comme c'est souvent le cas avec la volatilité du marché à prix flottant.

Vous pouvez écouter James Yonan en parler à la Conférence Blender 2013:

http://www.youtube.com/watch?v=_Oqo383uviw

Les notes de conférence sont en [doc/Brenda-talk-blendercon-2013.pdf](#)

Brenda comprend cinq outils qui sont décrits ci-dessous.

Pour avoir une aide détaillée sur chaque outil, exécutez la commande avec l'option -h (ou --help).

1. **\$ brenda-work** : s'utilise pour créer et peupler une file d'attente SQS pour des tâches de rendu. Une tâche de rendu est un court (souvent une seule ligne) script de shell qui exécute Blender pour faire le rendu d'une seule image [ou frame] d'un projet. → Utilise l'API SQS.
2. **\$ brenda-run** : s'utilise pour démarrer des instances EC2 exécutant Brenda, soit comme des instances à la demande ou de repérer des cas à un prix maximum de l'offre donnée. → Utilise l'API EC2.
3. **\$ brenda-tool** : s'utilise pour surveiller le fonctionnement d'une ferme de rendu EC2. Il permet une commande ssh ou rsync à exécuter simultanément sur tous les nœuds de la batterie. Utilise API EC2 et

nécessite un shell standard Unix où `ssh` et `rsync` sont disponibles et peuvent être exécutés à partir de la ligne de commande.

4. **\$ brenda-ebs** : un outil simple qui crée un nouveau volume EBS d'une taille spécifiée et l'attache à une instance `t1.micro` d'EC2 venant de démarrer.

5. **\$ brenda-node** : script d'exécution qui effectue le travail sur les noeuds eux-mêmes de la ferme de rendu. Il lit les tâches d'un file d'attente SQS, exécute la tâche (généralement des opérations de rendus), et copie les résultats des tâches (comme une image en PNG) sur un support S3. La commande `brenda-node` n'est généralement pas exécuter directement par l'utilisateur, mais est instancié à distance par la commande `brenda-run`. → Utilise les API SQS & S3.

Plate-formes prises en charge

L'application Brenda s'exécute en ligne de commande et a actuellement été testée sur Mac OS X et Linux uniquement.

TUTORIAL

Ce tutoriel est destiné à être utilisé sur Mac OS X ou Linux.

Si vous n'avez pas de compte AWS (Amazon Web Services), inscrivez-vous pour en avoir un dès maintenant.

Tout d'abord, installer la bibliothèque python "boto". Cette bibliothèque est utilisée par Brenda pour interagir avec AWS.

Avant l'installation de Brenda, assurez vous de posséder la bibliothèque python Dev : "`libpython-dev`", sinon installez la

Ensuite, téléchargez et installez Brenda sur votre machine cliente.

```
$ Git clone http://github.com/jamesyonan/brenda.git
```

```
$ Cd Brenda
```

```
$ Python setup.py install
```

Vous aurez besoin d'une clé SSH à base de RSA pour accéder aux machines

virtuelles (VM) que nous louerons en utilisant le service EC2 d'AWS. Ces machines virtuelles sont souvent désignées comme des «instances», et nous en créeront beaucoup d'entre elles qui agiront en tant que nœuds de travaux dans notre ferme de rendu.

Si vous avez une paire de clés SSH existante dans

```
~/.ssh/id_rsa.pub
```

```
~/.ssh/id_rsa
```

le logiciel client Brenda va les utiliser. Sinon, le logiciel va générer une nouvelle paire de clés sur AWS et télécharger la clé privée dans

```
~/.ssh/id_rsa.brenda
```

Ensuite, il faut obtenir la "clé d'accès" AWS et la "clé secrète" de la console de gestion AWS. Celles-ci permettront à l'application Brenda d'accéder aux ressources AWS de votre compte.

Nous aurons également besoin d'un outil pour accéder au magasin de fichiers AWS S3, parce que nous allons utiliser le support S3 comme **emplacement de stockage**:

1. Pour notre projet Blender, pour permettre aux nœuds de la ferme d'accéder aux fichiers Blender.
2. Pour les images de rendues générées par la ferme de rendu.

Pour cela, téléchargez et installez l'outil "s3cmd". Vous devrez configurer s3cmd avec vos identifiants de compte AWS:

```
$ s3cmd --configure
```

À ce stade, nous allons copier nos projets Blender, actifs et autres fichiers de support dans le cloud AWS de sorte que la ferme de rendu puisse y accéder. Il existe deux méthodes pour ce faire, chacun avec leurs propres avantages et inconvénients:

1. Empaqueter le projet Blender et les fichiers de support dans un fichier zip ou tar et le copier sur le support S3 (S3 est un service de stockage de fichiers distribués hébergé par AWS).

Avantages : Relativement facile à mettre en place.

Inconvénients : commence à être extrêmement lents comme ensemble

de données pénètre dans la gamme multi-GB.

2. Créer « **un instantané EBS** » contenant le projet Blender et les fichiers. Un volume EBS est une sorte de disque dur virtuel auquel on peut attacher des instances EC2. Un *instantané EBS* est une sorte de copie en un point temporel d'un volume EBS auxquelles de nombreuses instances EC2 peuvent accéder simultanément.

Avantages : efficace et évolutif - les instantanés EBS peuvent être d'une taille jusqu'à 1 To, et Brenda peut soutenir jusqu'à 66 *instantanés EBS* pour chaque instance de ferme de rendu.

Moins : Plus compliqué à mettre en place.

Pour poursuivre le tutoriel, nous allons utiliser la première méthode S3, mais il faut lire la section "rendu de grands projets en utilisant des *instantanés EBS*" ci-dessous si vos données sont grandes, vous voudrez utiliser la deuxième méthode.

Ensuite, nous allons regrouper notre projet Blender et l'enregistrez sur un S3 pour que la ferme de rendu puisse y accéder (s'assurer que le projet Blender utilise des chemins relatifs pour l'accès aux fichiers de support car les instances de la ferme de rendu doivent pouvoir les suivre).

Pour ce faire, créer un dossier avec votre fichier .blend et tous les autres fichiers de support nécessaires pour rendre des images, puis compresser le en utilisant tar ou zip. Par exemple, en supposant que le répertoire du projet s'appelle "**monprojet**", exécutez:

```
$ Tar cfzv monprojet.tar.gz monprojet
```

Ensuite, créer un "bucket" S3 sur AWS pour stocker le fichier de **monprojet.tar.gz** que nous venons de créer ci-dessus. Un bucket S3 sont un peu comme des dossiers, mais il doit avoir un nom unique au monde, et il ne peut contenir que des fichiers plats, et pas de sous-dossiers. Vous pouvez ici choisir un nom pour le bucket du projet auquel nous nous référerons ensuite : PROJET_BUCKET.

```
S3cmd $ mb s3: // PROJET_BUCKET
```

Il est possible que le nom que vous avez choisi PROJET_BUCKET soit déjà utilisé par quelqu'un d'autre. Dans ce cas, vous verrez un message d'erreur, et pourrez recommencer la commande en utilisant un nom différent.

Nous devons également créer un autre seau pour la ferme de rendu puisse enregistrer les images rendues. Nous l'appellerons `IMAGE_BUCKET`. Tout comme `PROJET_BUCKET` ci-dessus, vous devez sélectionner un nom unique.

```
S3cmd $ mb s3: // IMAGE_BUCKET
```

Maintenant, nous allons copier notre fichier de projet Blender comprimé dans notre `PROJET_BUCKET` sur S3:

```
S3cmd $ put monprojet.tar.gz s3: // PROJET_BUCKET
```

Pour vérifier que le fichier a été copié, listez les fichiers dans le `PROJET_BUCKET`:

```
S3cmd $ ls s3: // PROJECT_BUCKET
```

Ensuite, nous allons créer un fichier de configuration Brenda. Les outils client Brenda vont chercher le fichier de configuration dans `~/.brenda.conf`

Créer `~/.brenda.conf` maintenant avec le contenu suivant, en veillant à remplacer `PROJET_BUCKET` et `IMAGE_BUCKET` avec les noms que vous avez choisi ci-dessus.

```
INSTANCE_TYPE=m3.xlarge
BLENDER_PROJECT=s3://PROJET_BUCKET/monprojet.tar.gz
WORK_QUEUE=sqs://IMAGE_BUCKET
RENDER_OUTPUT=s3://IMAGE_BUCKET
DONE=shutdown
```

Pour expliquer les paramètres de configuration ci-dessus dans le détail:

`INSTANCE_TYPE` décrit le type d'instance EC2 (c.a.d. machine virtuelle) qui fera la ferme de rendu. Différents types d'instance offrent différents niveaux de performance et de coût.

BLENDER_PROJECT est le nom de notre fichier de projet sur S3. Il peut être :

`s3: //..`

ou une url :

`file: //..`

WORK_QUEUE est le nom d'une file d'attente SQS que nous allons créer afin d'organiser la mise en scène et le séquençage des tâches dans notre rendu.

RENDER_OUTPUT est le nom d'un compartiment S3 qui contiendra nos images rendues.

DONE=shutdown raconte les instances de ferme de rendu qu'ils doivent automatiquement se arrête après le rendu est terminée.

A l'étape suivante, nous allons créer la *file d'attente de travail* pour notre ferme de rendu. Une *file d'attente de travail* est essentiellement une liste de nombreux petits scripts qui, lorsqu'ils seront exécutés ensemble, exécuteront le rendu de toutes les images de notre projet Blender.

L'objectif essentiel de Brenda est d'accélérer le processus de rendu en traitant simultanément notre *file d'attente de travail* en utilisant des dizaines, des centaines voire des milliers de machines virtuelles.

Par exemple, l'un des scripts dans une file d'attente de travail pourrait ressembler à ça (pour rendre l'image 5 de notre projet):

```
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 5 5 1 -j -t 0 -a
```

Supposons que notre projet contienne 240 images. Alors la *file d'attente de travail* ressemblerait à ça, où chaque ligne est une tâche distincte dans la file d'attente de travail:

```
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 1 1 1 -j -t 0 -a
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 2 2 1 -j -t 0 -a
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 3 3 1 -j -t 0 -a
...
...
...
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 238 238 1 -j -t 0 -a
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 239 239 1 -j -t 0 -a
blender -b * .blend -F PNG -o $ OUTDIR / frame _ ##### -s -e 240 240 1 -j -t 0 -a
```

La première étape dans la création d'une *file d'attente de travail* est de commencer avec un **modèle de script**. Un modèle de script décrit comment exécuter Blender pour réaliser un seul travail. Supposons que nous voulions un seul travail pour faire le rendu d'une image PNG unique. Dans ce cas, notre modèle de script serait le suivant:

```
blender -b *.blend -F PNG -o $OUTDIR/frame_##### -s $START -e $END -j $STEP -t 0 -a
```

En utilisant un éditeur de texte, créez un fichier appelé "modele-image" qui contient la ligne ci-dessus. C'est un modèle simple qui est conçu pour faire le rendu d'une image à la fois (pour un tâche plus compliqué, il est également possible de créer un *modèle de rendu de sous-images* qui va découper la plus petite *unité de rendu* en travail pour une partie d'un image, cela peut être utilisé pour accélérer le rendu, ou pour réduire le temps passé à traiter chaque *unité de travail* dans les animations où chaque image prend de nombreuses heures de rendu informatiques).

Brenda inclut un outil appelé "**brenda-work**" qui nous permet de générer facilement une **file d'attente de travail**. Supposons que vous vouliez rendre

les 240 premières images de votre projet. Utilisez cette commande pour générer la file de travail en utilisant le fichier "modele-image" que nous avons créé ci-dessus:

```
$ brenda-work -T modele-image -e 240 push
```

Cela va créer une *file d'attente de travail* pour rendre les images de 1 à 240 de votre projet Blender.

Pour voir la taille actuelle de la file d'attente de travail, exécutez:

```
$ brenda-work status
```

Vous devriez voir une taille de file d'attente de 240.

Si vous faites une erreur et que vous voulez supprimer la file d'attente actuelle et recommencer:

```
$ brenda-work reset
```

Enfin, dans le cadre de notre configuration initiale, nous allons faire une initialisation ponctuelle d'un nouveau compte AWS pour créer un groupe de sécurité et des profils-clés ssh. Cela ne doit être fait qu'une fois par compte AWS:

```
$ brenda-run init
```

À ce stade, nous sommes prêts à commencer le rendu.

Une des caractéristiques les plus intéressantes d'AWS est le marché flottant d'instances EC2. Sur ce marché, nous pouvons louer des VM (c.a.d. Machines Virtuelles) à l'heure pour notre projet à un prix considérablement inférieur au taux de demande en cours. Bien que le marché offre un bon prix, l'inconvénient est que nos instances peuvent être résiliées à tout moment si le prix au comptant s'élève au dessus de notre prix d'offre maximale.

Nous allons vérifier les prix du *marché flottant* en cours:

```
$ brenda-run -i c1.xlarge price
```

Quand je lance la commande, je vois ceci:

```
Spot price data for instance c1.xlarge
us-east-1a 2013-10-24T07:32:32.000Z $0.141
us-east-1b 2013-10-24T03:55:49.000Z $0.07
us-east-1c 2013-10-24T02:05:53.000Z $0.07
```

Ça indique que l'instance **c1.xlarge** (une VM raisonnablement rapide avec 8 cœurs) se loue actuellement pour **0,07 \$ US** par heure.

AWS propose de nombreux types d'instances différentes qui offrent différents

niveaux de performance et/ou coût. Les instances appropriées pour une utilisation comme ferme de rendus distribués sont par exemple:

Exemple	Prix de base à partir de nov. 2013
c1.xlarge	0,07 dollars / heure
m2.xlarge (par défaut)	0,035 dollars / heure
m3.2xlarge	0,115 dollars / heure
m3.xlarge	0,0575 dollars / heure

Pour voir le prix en cours d'une instance EC2 selon plusieurs zones de disponibilité:

```
$ brenda-run -i INSTANCE_TYPE price
```

A certain moment, le nombre d'instance utilisées d'un certain type peut s'accroître. Dans ce cas, le prix du marché s'élève considérablement au dessus du prix de base. En général, il faut éviter de payer plus que le prix de base sinon votre rendu peut vite vous coûter cher. Si un type instance donnée est très utilisée, essayez d'utiliser un autre type d'instance dont le prix est encore à son prix de base. C'est la sacro-saint loi de l'offre et de la demande.

Pour un rendu sous GPU accéléré, AWS propose une instance GPU (**cg1.4xlarge**), mais son prix est beaucoup plus que les prix standards, ce qui en fait une proposition douteuse.

Pour plus d'informations sur les types d'instances EC2 voir ici:

<http://aws.amazon.com/ec2/instance-types/>

Pour poursuivre ce tutoriel, nous allons choisir quatre instance de type **c1.xlarge** à 0,07 dollars/heure. Notez que lorsque vous exécutez la commande suivante, vous acceptez la somme de 0,28 dollars/heure au total pour louer 4 machines virtuelles (c'est à dire : 4 instances x 0,07 dollars US).

```
$ brenda-run -P -i c1.xlarge -N 4 -p 0.07 spot
```

Notez que nous avons utilisé l'option **-P** pour créer nos demandes d'instances. Cette option indique à AWS de rendre nos demandes "persistante", ce qui signifie que si la fluctuation du prix d'une instance **c1.xlarge** dépasse 0,07 dollars/heure, ce qui clorait le travail de nos instances, la demande resterait active, ce qui signifierait que dès que le prix revient à 0,07 dollars/heure, nos instances seraient automatiquement réactivées afin qu'elles puissent continuer le travail de rendu.

Si la commande ci-dessus réussit, vous verrez une liste d'informations dont le

script avec lequel Brenda va lancer les futurs rendus et une liste de quatre objets « *SpotInstanceRequest* ».

Dans certains cas, pendant les heures de pointe dans les Data-center d'AWS, les instances ponctuelles peuvent ne pas être disponibles pour un coût raisonnable. Dans ce cas, il est possible d'utiliser des instances « *à la demande* » à la place (Notez cependant que ces demandes sont beaucoup plus chères que les instances ponctuelles).

```
$ brenda-run -i c1.xlarge -N 4 demand
```

Pour voir l'état actuel de vos instances ponctuelles:

```
$ brenda-run status
```

Si vous avez utilisé la commande "**brenda-run ... spot**" ci-dessus, vous devriez voir un aperçu de vos quatre instances en attente.

Spot Requests

```
sir-54b91c35 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: pending-evaluation>
sir-97762e35 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: pending-evaluation>
sir-aa98a235 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: pending-evaluation>
sir-db7e5a35 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: pending-evaluation>
```

Ces instances ne sont pas encore actives, mais sont plutôt dans un état d'"attente-évaluation". Après plusieurs minutes, si votre prix demandé correspond aux conditions du marché du prix en cours, les instances seront activées. Lorsque cela se produira, le status affichera quelque chose comme ceci:

```
$ brenda-run status
```

Active Instances

```
ami-e7c4988e 0:01:51 ec2-107-20-36-70.compute-1.amazonaws.com
ami-e7c4988e 0:01:51 ec2-50-16-67-202.compute-1.amazonaws.com
ami-e7c4988e 0:01:51 ec2-54-205-52-227.compute-1.amazonaws.com
ami-e7c4988e 0:01:50 ec2-54-224-220-141.compute-1.amazonaws.com
```

Spot Requests

```
sir-54b91c35 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: fulfilled>
sir-97762e35 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: fulfilled>
sir-aa98a235 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: fulfilled>
sir-db7e5a35 RegionInfo:us-east-1 one-time 2013-10-24T05:32:16.000Z $0.07 <Status: fulfilled>
```

D'autre part, si vous avez utilisé la commande "**brenda-run ... demand**" ci-dessus, vos instances devraient commencer immédiatement:

```
$ brenda-run status
```

Active Instances

```
ami-e7c4988e 0:00:23 ec2-107-20-72-84.compute-1.amazonaws.com
ami-e7c4988e 0:00:23 ec2-54-211-251-85.compute-1.amazonaws.com
```

```
ami-e7c4988e 0:00:23 ec2-54-226-31-134.compute-1.amazonaws.com
```

```
ami-e7c4988e 0:00:23 ec2-54-234-204-209.compute-1.amazonaws.com
```

À ce stade, le travail de rendu est en cours d'exécution. Il existe plusieurs méthodes que vous pouvez utiliser pour suivre ses progrès.

Utilisez cette commande pour afficher le nombre d'images en attente dans la file d'attente de travail:

```
$ brenda-work status
```

Brenda inclut un outil d'usage général pour exécuter des commandes sur le rendu en cours. Par exemple, pour voir à la fin les fichiers journaux pour chaque instance de rendu:

```
$ brenda-tool ssh tail log
```

Ou voir la charge de CPU sur chaque cas de rendu:

```
$ brenda-tool ssh uptime
```

Pour voir le nombre de tâches (c.a.d. d'images) exécutées par chaque instance de rendu:

```
$ brenda-tool ssh cat task_count
```

Un des risques de la gestion de groupe d'instances, c'est que vos instances peuvent être résiliées sans avertissement si le prix au comptant s'élève au dessus de votre prix d'offre maximale. Dans ce cas, le statue des interrogations peut ressembler à ceci:

```
$ brenda-run status
```

```
Spot Requests
```

```
sir-5dbba434 RegionInfo:us-east-1 one-time 2013-10-23T09:12:30.000Z $0.07 <Status: instance-terminated-by-price>
```

```
sir-64daca34 RegionInfo:us-east-1 one-time 2013-10-23T09:12:30.000Z $0.07 <Status: instance-terminated-by-price>
```

```
sir-9cf0b834 RegionInfo:us-east-1 one-time 2013-10-23T09:12:30.000Z $0.07 <Status: instance-terminated-by-price>
```

```
sir-af53be34 RegionInfo:us-east-1 one-time 2013-10-23T09:12:30.000Z $0.07 <Status: instance-terminated-by-price>
```

Brenda a effectivement été conçu pour récupérer facilement cet exemple de résiliation et peut gérer le cas où les instances sont créés et détruites au hasard au cours du rendu. Il est capable de faire cela parce qu'aucune tâche n'est définitivement retirée de la file d'attente jusqu'à ce que les résultats de la tâche (les images rendues) soient enregistrées dans *le magasin de fichiers Rendu* sur le support S3. En particulier,

1. Toute instance qui se termine renverra toutes les tâches inachevées en retour à la file d'attente de travail.

2. **Brenda-run** peut être utilisé pour créer des instances "**persistantes**"

sur place, où même si une instance est tuée parce que la hausse des prix dépasse l'enchère maximale désirée, une autre sera redémarrée automatiquement lorsque le prix retombera en dessous de l'offre maximale que vous avez spécifié avec la commande "**brenda-run spot**". Pour créer un groupe d'instances persistantes, utiliser l'option **-P (comme Persistante)** lors de l'exécution de la commande "**brenda-run spot**".

Alors que le travail de rendu progresse, vous pouvez voir les images qui ont été rendus à ce jour et copiées dans *le magasin de fichiers rendu* du support S3:

```
$ s3cmd ls s3://IMAGE_BUCKET
```

Parce que nous avons dit aux instances de rendu d'arrêter immédiatement lorsque le rendu est terminée, la commande "**brenda-work status**" affichera : « *aucun instances actives en achèvement de rendu* ». À ce stade, la commande "**brenda-work status**" devrait montrer une taille de fichier en attente de 0.

Pour télécharger les images rendues, préparer un répertoire localement pour accueillir les images:

```
$ mkdir images
$ s3cmd get -r s3://IMAGE_BUCKET images
```

Si quelque chose se passe mal pendant le rendu, Brenda ne fermera pas les instances normalement, pour vous donner une chance de télécharger les fichiers « *log* » (journaux):

```
$ mkdir logs
$ brenda-tool rsync log logs/HOST.log
```

Si vous souhaitez annuler le rendu et arrêter les instances, il existe des commandes pour cela aussi:

Annuler les tâches en attente qui n'ont pas encore été traitées:

```
$ brenda-run cancel
```

Terminez toutes les instances actives:

```
$ brenda-run -T stop
```

I - L'arrêt de la ferme de rendu.

La façon la plus simple d'arrêter la ferme de rendu, après le rendu de toutes les images, est de régler la variable de config "DONE = shutdown" dans votre fichier `~/.brenda.conf` avant de commencer une tâche.

Ou en forçant l'arrêt de l'ensemble de l'exploitation maintenant:

```
$ brenda-run -T stop
```

Cependant, il y a des variantes utiles à la procédure d'arrêt, comme dans le cas où vous avez un grand nombre d'instances sur le point de transition dans leur prochain heures facturables, mais la file d'attente des tâches de votre ferme de rendu est presque vide (mais pas tout à fait).

Pour ce cas, Brenda possède une capacité "prune" pour réduire le nombre d'instances en fonctionnement à une valeur spécifiée. Par exemple, pour réduire le nombre d'instances s'exécutant à 4:

```
$ brenda-tool -T prune 4
```

La capacité « prune » arrête intelligemment les instances, cad qu'elle arrête les instances qui ont terminée le plus récemment une tâche, afin de minimiser la quantité de travail perdu, c'est à dire les tâches stopées avant leur fin.

Si vous voulez seulement empêcher l'arrêt des instances proches de leur transition à leur prochaine heure facturable, vous pouvez utiliser le paramètre de seuil (-T).

Par exemple, si vous voulez seulement empêcher l'arrêt des instances ayant dépassées leur valeur de disponibilité à ≥ 55 minutes, utilisez une valeur de seuil de 55:

```
$ brenda-tool -T -t 55 prune 4
```

Pour **tester la commande** mais sans mettre fin effectivement à toutes les instances, ajouter l'option -d (→ fonctionnement à vide).

2 - Evaluation de la performance.

AWS a de nombreux types d'instances à différents niveaux de prix et de performance.

Comment faire pour déterminer le type d'instance optimale pour l'utilisation d'une ferme de rendu?

Pour répondre à cette question, Brenda contient un outil qui permettra d'évaluer la performance d'une ferme de rendu en termes d'images/dollars US.

Premièrement, mettre en place une exécution à l'aide de `"brenda-work ... push"` comme indiqué dans le tutoriel ci-dessus, mais aussi ajouter l'option `"-R"` pour rendre aléatoire la file d'attente des tâches. Cela aidera à normaliser la moyenne des temps de rendu par image de notre analyse.

Cette commande va ajouter les images **1 à 500** à la file d'attente des tâches en utilisant le script `"image-modele"` présenté ci-dessus dans le tutoriel:

```
$ brenda-work -T image-modele -e 500 -r push
```

Ensuite, lancez le fonctionnement d'une ferme de rendus en utilisant un ensemble d'instances AWS appropriées pour une utilisation en ferme de rendu. Ce script lancera des requêtes d'instance pour tous ces types d'instances AWS à leur prix de base. Un plus grand nombre d'instances pour les types d'instances plus lentes seront utilisés pour équilibrer les échantillons de données afin que les résultats pour ces types d'instances soient statistiquement significatifs.

Notez que ces instances si elles sont instanciées avec succès, coûteront 1,11 dollars US/heure.

```
#!/bin/bash
brenda-run -N 8 -i m1.medium -p 0.013 -P spot
brenda-run -N 4 -i m1.large -p 0.026 -P spot
brenda-run -N 2 -i m1.xlarge -p 0.052 -P spot
brenda-run -N 2 -i m3.xlarge -p 0.0575 -P spot
brenda-run -N 1 -i m3.2xlarge -p 0.115 -P spot
brenda-run -N 4 -i c1.medium -p 0.018 -P spot
brenda-run -N 1 -i c1.xlarge -p 0.07 -P spot
brenda-run -N 4 -i m2.xlarge -p 0.035 -P spot
brenda-run -N 2 -i m2.2xlarge -p 0.071 -P spot
brenda-run -N 1 -i m2.4xlarge -p 0.14 -P spot
```

Une fois que les instances sont en place et en cours d'exécution, utilisez la commande suivante pour analyser la performance du fonctionnement:

```
$ brenda-tool perf
```

La code ci-dessous est une analyse d'un rendu réel d'une simulation de fluide en 500 images, rendu avec Cycles en 1920x1080 px, en utilisant un fluide pré-rendu.

L'impression terminal montre la performance de la ferme de rendu grâce aux deux "Tasks per hour" et "Tasks per US\$". Notez que "Tâches" est généralement équivalent de "Image" si vous configurez votre file de travail de manière standard, où chaque tâche fait le rendu d'une seule image.

```
Tasks per hour (137.90)
```

```
m2.4xlarge 28.15
m3.2xlarge 27.75
c1.xlarge 24.35
m2.2xlarge 15.52
m3.xlarge 13.18
m1.xlarge 11.05
m2.xlarge 8.49
m1.large 5.07
m1.medium 2.77
```

```
Tasks per US$
```

```
c1.xlarge 347.81
m2.xlarge 242.56
m3.2xlarge 241.29
m3.xlarge 229.22
m2.2xlarge 218.53
m1.medium 213.37
m1.xlarge 212.46
m2.4xlarge 201.07
m1.large 194.91
```

Les résultats de cette analyse sont que le type d'instance "c1.xlarge" est le «gagnant», en faisant un travail de **347,81 images par US \$**.

Notez que ces données de performance sont subordonnées au type d'instance particulière étant disponible à votre prix de base. Si un type d'instance particulière a augmenté, les requêtes d'instances ne seront pas instanciés, provoquant l'arrêt des données disponibles pour ce type d'instance. Si vous augmentez votre prix d'offre pour égaler ou dépasser le prix du marché et ainsi augmenter le nombre de ces types d'instances (cad pour les forcer à travailler), ce serait fausser les résultats de ces types d'instances, en les avançant dans le classement des «Tâches par \$ US».

3 - Rendu de sous-images.

Normalement, la plus petite unité de travail avec Brenda est « **l'image** ». Bien que cela soit souvent suffisant, il est parfois avantageux d'utiliser une plus petite unité de travail, par exemple en divisant chaque image en une grille de tuiles (« tile » en anglais), chaque tuile étant une unité séparée de travail.

1. Par exemple, le **rendu des tuiles** peut accélérer le rendu d'images fixes. Normalement, un cliché (parce qu'il s'agit d'une seule image) ne peut pas profiter de l'accélération de rendu d'une ferme conçue pour l'animation. Cependant, le rendu de sous-images permet de subdiviser le rendu d'une seule image au travers de nombreuses instances, en permettant d'accélérer le rendu d'une seule image.

2. Lors du rendu d'une animation où chaque image prend de nombreuses heures informatiques, il y a toujours le risque que l'instance puisse être résilié avant la fin, ce qui provoque la perte du temps d'ordinateur déjà investis dans l'image inachevée. Pour atténuer ce risque, l'animation de sous-images peut être utilisée pour réduire le temps nécessaire pour rendre une unité de travail.

Pour faire un rendu de sous-images, nous devons créer un modèle de tâche différent:

```
cat >subframe.py <<EOF
import bpy
bpy.context.scene.render.border_min_x = $SF_MIN_X
bpy.context.scene.render.border_max_x = $SF_MAX_X
bpy.context.scene.render.border_min_y = $SF_MIN_Y
bpy.context.scene.render.border_max_y = $SF_MAX_Y
bpy.context.scene.render.use_border = True
EOF
blender -b *.blend -P subframe.py -F PNG -o $OUTDIR/frame_#####_X-$SF_MIN_X-$SF_MAX_X-Y-$SF_MIN_Y-$SF_MAX_Y -s $START -e $END -j $STEP -t 0 -a
```

Enregistrer ce modèle dans le fichier "**sous-image-modele**", puis générer la file d'attente de tâche comme suit : En supposant que nous voulions diviser chaque image en 64 tuiles de 8x8, utilisez la commande suivante pour générer une file de tâches pour les 240 premières images:

```
brenda-work -T sous-image-modele -e 240 -X 8 -Y 8 -d push
```

4 - Rendu Multiframe.

Le rendu "**Multiframe**" (ou "Multi-image" en français) signifie que chaque unité de travail traitées par la ferme de rendu comprend plusieurs images (dans un sens, c'est le contraire du rendu de sous-images). Cela peut être utile lorsque le temps de rendu par image est petite et que vous voulez rendre plusieurs images pour chaque instance de l'exécutable Blender, ou quand il y a un coût de démarrage associé pour faire le rendu d'une série d'images, et que vous voulez amortir le coût par plusieurs images.

Lors de la génération de la file de tâches, spécifier l'option **-S** et le nombre d'images que vous souhaitez regrouper dans chaque unité de travail.

L'exemple ci-dessous va générer une file de tâches pour le rendu de 240 images en unités de 10 images chacunes.

```
$ brenda-work -T image-template -S 10 -e 240 push
blender -b *.blend -F PNG -o $OUTDIR/image_##### -s 1 -e 10 -j 1 -t 0 -a
blender -b *.blend -F PNG -o $OUTDIR/image_##### -s 11 -e 20 -j 1 -t 0 -a
blender -b *.blend -F PNG -o $OUTDIR/image_##### -s 21 -e 30 -j 1 -t 0 -a
....
....
....
blender -b *.blend -F PNG -o $OUTDIR/image_##### -s 221 -e 230 -j 1 -t 0 -a
blender -b *.blend -F PNG -o $OUTDIR/image_##### -s 231 -e 240 -j 1 -t 0 -a
```


5 - Rendu de grands projets en utilisant des "clichés" EBS.

Brenda prend en charge les supports S3 d'AWS et les "clichés" EBS comme moyen pour fournir des données 3D à votre ferme de rendu.

Alors que les petits projets peuvent être fournis par le fichier .blend et d'autres fichiers du projet en partant d'un support S3, chaque instance de rendu doit télécharger un fichier de plusieurs Go à partir du S3 peut être extrêmement lent.

Une alternative est d'utiliser des "**volumes EBS**" (Elastic Block Store ou stock de blocs élastiques) d'AWS au lieu du support S3 comme plate-forme de stockage de vos données de projet en cours, à la disposition des instances de ferme de rendu. Les volumes EBS peuvent aller jusqu'à 1 To et Brenda permet actuellement jusqu'à 66 volumes EBS connectés par instance (d'une ferme de rendu).

Un "**volumes EBS**" peut être considéré comme une sorte de disque dur virtuel. Vous pouvez créer une instance EC2 et joindre un "**volumes EBS**" nouvellement créé à cette instance existante, puis copier vos fichiers de projet et d'assets sur le "volumes EBS". Votre fichier .blend doit être dans le haut niveau du volume EBS, afin que Brenda puisse le localiser. Les "volumes EBS" sont persistants, afin que vous puissiez maintenir un espace qui contient l'état actuel de votre projet, tout en ajoutant, supprimant ou modifiant des fichiers au besoin.

Brenda a un support limité pour créer et gérer les "volumes EBS". Par exemple, pour créer un nouveau "volumes EBS" d'une taille de 4 Go, et attacher une instance nouvelle **t1.micro** pour fonctionner sur **/mnt**, utilisez cette commande:

```
$ brenda-ebs -m -s 4 new
```

Une fois que l'instance est en place, vous pouvez copier vos fichiers de projet, vos textures bakées, et vos données en utilisant **rsync**. Par exemple, cette commande permet de copier *fluid.blend* et *cache_fluid* dans le répertoire de haut niveau du volume EBS:

```
$ brenda-tool rsync -av fluid.blend cache_fluid HOST:/mnt/
```

Lorsque vous avez terminé de copier les fichiers sur le "volumes EBS", démonter et arrêter l'instance qui lui est attachée:

```
$ brenda-tool ssh umount /mnt
```

```
$ brenda-run stop
```

Un "volumes EBS", par lui-même, ne peut être connecté à une seule instance à la fois, donc avant de commencer un rendu, nous devons faire un «cliché» du "volumes EBS" afin que nous puissions en faire une copie à la disposition de toutes les instances de la ferme de rendu. La console Web d'AWS fournit un ensemble complet d'outils pour créer des "volumes EBS" et en faire des clichés.

Si vous avez créé votre volume avec la commande ci-dessus "**brenda-ebs**", vous pouvez en faire un cliché en allant à **EC2/Volumes** dans la console de gestion d'AWS, un clic droit sur le volume que vous venez de créer (vous pouvez l'identifier par sa capacité), sélectionnez "**Create Snapshot**", et sélectionnez un nom pour le cliché. La création de clichés n'est pas instantanée, aller dans le panneau **Snapshot** pour vérifier que le cliché est terminée. À ce stade, vous pouvez nettoyer en mettant fin à l'instance **t1.micro**, nous allons d'abord accéder au volume associé (voir le panneau **Instances**) et ensuite la supprimer (voir panneau **Volume**).

Une fois que vous avez votre cliché EBS prêt, vous pouvez l'indiquer dans votre fichier `~/.brenda.conf` en utilisant une adresse `ebs://URL` :

```
BLENDER_PROJECT=ebs://EBS_SNAPSHOT_NAME
```

EBS_SNAPSHOT_NAME peut être un ID instantané telles que le "**snap-ab957cc2**" ou un nom de haut niveau que vous avez attribué au cliché comme "**mon projet**".

Remarque : Lorsque vous utilisez un cliché EBS en tant que **BLENDER_PROJECT**, assurez-vous que votre fichier `.blend` est dans le haut niveau du volume EBS, afin que Brenda puisse le localiser. Assurez-vous également que tous les chemins utilisés dans votre fichier `.blend` soient relatifs, afin que les instances de votre ferme de rendu soient en mesure de les suivre.

Brenda prend également en charge le montage des clichés EBS supplémentaires (jusqu'à 65) qui apparaissent comme sous-dossiers dans le dossier de votre projet, et peuvent servir pour fournir des biens et des données supplémentaires pour le processus de rendu.

```
ADDITIONAL_EBS_0="ebs://EBS_SNAPSHOT_NAME,DIRECTORY"
```

```
ADDITIONAL_EBS_1="ebs://EBS_SNAPSHOT_NAME,DIRECTORY"
```

```
.  
. .  
.
```

Le cliché : **EBS_SNAPSHOT_NAME** sera monté comme un sous-dossier nommé **DIRECTORY** dans le même dossier que votre projet `.blend`.

6 - Installer vos données de projet dans le cloud AWS.

Pour utiliser Brenda, vous devez d'abord télécharger vos données de projet sur le Cloud AWS afin que les données réside sur un support S3 ou un volume EBS.

Quand vos données de projet sont importantes, il existe différentes méthodes qui peuvent être utilisées pour optimiser ce processus:

1. Envisager de générer vos « bakings » et vos caches sur le cloud lui-même de sorte que les données soient déjà là et vous n'avez pas besoin de les uploader. Cela peut se faire par le démarrage d'une instance EC2 de Windows, par l'installation de Blender et de vos fichiers de projet sur elle, d'y générer vos fichiers de bakings et/ou de caches, et ensuite de les copier sur un volume EBS.

2. Si vos données et caches de projet sont de grande taille (des centaines de Go ou même de To), il est parfois impossible de les uploader sur Internet. L'alternative est le module **Import/Export** d'AWS:

<http://aws.amazon.com/importexport/>

Vous pouvez expédier physiquement un périphérique de stockage (comme un disque dur) à un centre de données AWS, et ils vous copieront eux-même les données directement sur un cliché EBS, que vous pourrez ensuite consulter grâce à votre application Brenda. Voir la section ci-dessus "**Rendu de grands projets en utilisant des clichés EBS**» pour plus d'informations sur créer votre cliché EBS pour votre ferme de rendu.

7 - Comment créer une AMI pour Brenda.

(**AMI** pour **A** **M** **A** **M** **I** pour **A** **M** **A** **Z** **O** **N** **M** **A** **C** **H** **I** **N** **E** **S** **I** **M** **A** **G** **E** **S**).

Alors que Brenda a déjà un lien vers une AMI existante qui possède Blender & Brenda pré-installés, vous pouvez construire votre propre AMI en utilisant la procédure suivante :

- Pour créer une **AMI** EC2 capable d'agir comme un noeud Brenda, utilisez une **AMI sous Ubuntu 12.04 x64** comme point de départ, puis exécuter les commandes en tant qu'administrateur **root**.

```
$ perl -p -i.bak -e 's/^disable_root: 1/disable_root: 0/' /etc/cloud/cloud.cfg
$ perl -p -i.bak -e 's/.*ssh-rsa/ssh-rsa/' /root/.ssh/authorized_keys
$ add-apt-repository -y ppa:irie/blender
$ apt-get update
$ apt-get install -y blender python-pip gcc python-dev libcurl4-openssl-dev git unzip
$ pip install -U boto
$ pip install -U s3cmd
```

Ensuite, téléchargez et installez Brenda:

```
$ git clone http://github.com/jamesyonan/brenda.git
$ cd brenda
$ python setup.py install
```

Maintenant sauvegardez l'**AMI**. L'image disque résultante aura toutes les dépendances nécessaires pour exécuter Blender & Brenda.

Si vous avez l'intention de faire une **AMI** publique, n'oubliez pas de nettoyer les fichiers de sécurité du système de fichiers de l'instance avant de faire votre cliché:

```
$ rm -rf /root/.bash_history /home/ubuntu/.bash_history
$ rm -rf /root/.cache /home/ubuntu/.sudo_as_admin_successful /home/ubuntu/.cache
/var/log/auth.log /var/log/lastlog
$ rm -rf /root/.ssh/authorized_keys /home/ubuntu/.ssh/authorized_keys
/root/.ssh/authorized_keys.bak /home/ubuntu/.ssh/authorized_keys.bak
```

Remarque: Veillez à ne pas supprimer **/root/.ssh/authorized_keys** jusqu'à ce que vous utilisiez votre instance, parce que cela va vous empêcher de vous reconnecter à l'instance sous **ssh**.

Enfin, si vous voulez rendre publique votre **AMI**, utilisez la commande suivante (nécessite des outils **EC2** en ligne de commande):

```
$ ec2-modify-image-attribute MY_AMI --launch-permission --add all
```

Fin.