

Thème rsync et rsnapshot

I. Rsync

1. Historique

- rsync a été développé par **Andrew Tridgel** grâce à ses travaux sur rzip.
- Utilisation d'un algorithme de compression "longue distance".

2. Présentation

- Outil simple disponible sans installation particulière sous Ubuntu et existant pour divers systèmes d'exploitation.
- Outil qui permet le transfert de fichiers (souvent utilisé pour la sauvegarde).
- La grande particularité de **rsync** est d'optimiser la bande passante disponible.
- Il ne transfère uniquement que les données modifiées ou nouvelles.
- Possibilité d'avoir une interface graphique avec **Grsync**.

3. Utilisation de rsync

- Dans notre exemple, nous avons deux dossiers source et destination.

```
rsync -av source destination
rsync -av source/ destination
```

La première commande copiera « source » dans « destination » contrairement à la seconde qui copiera tout le contenu de « source » dans « destination ».

4. Faire tourner rsync sur SSH

L'utilisation de SSH par rsync se fait simplement en utilisant l'option "-e ssh", tout ce qui peut se passer ensuite est transparent pour l'utilisateur. (Voir exemple ci-dessous) :

@Ip serveur ici est : 192.168.10.1

```
root@client # rsync -av -e ssh jean@192.168.10.1::original /copie
```

5. Plus d'options

Description	Option rsync
Mode archive : cela veut dire que l'on veut la récursivité et tout préserver comme les droits, les propriétaires, les groupes etc ...	-a
Ceci dit à rsync de copier les répertoires récursivement	-r
Affiche les informations de la copies (vitesse de transfert, données envoyées)	-v
Affiche quelques statistiques de transfert de	--stats

fichiers	
Efface les fichiers qui n'existent pas chez l'émetteur	--del
Montrer la progression pendant le transfert	--progress
Transfert en compressant les données	-z

6. Références

<http://samba.anu.edu.au/rsync/>

<http://www.delafond.org/traducmanfr/man/man1/rsync.1.html>

II. rsnapshot

1. Présentation

rsnapshot est un outil de sauvegarde basé sur rsync. Grâce à rsnapshot, il est possible de prendre des snapshots de vos systèmes de fichiers à différents instants.

Utilisant les liens durs, rsnapshot crée l'illusion de l'existence de multiples sauvegardes complètes, tout en n'occupant que la place d'une seule plus les différences. Couplé avec ssh, il est également possible de prendre des snapshots de systèmes de fichiers distants.

2. Installation

Téléchargement du paquet sur le site rsnapshot. Ouvrir un terminal et ce mettre dans le dossier du paquet et taper les commandes ci-dessous :

```
./configure
make test
make intall
```

3. Configuration

Le fichier de configuration n'est ni créé, ni installé pendant le processus d'installation. Pour copier le fichier de configuration par défaut à l'endroit où rsnapshot cherchera le vrai fichier de configuration :

```
# cp rsnapshot.conf.default /etc/rsnapshot.conf
```

a. Les intervalles de sauvegardes

On commence par définir le répertoire où l'on va stocker nos backups :

```
snapshot_root /home/backups/ //on peut mettre ce que l'on veut
```

La section interval du fichier de configuration contient les entrées qui déterminent combien de sauvegardes seront créées et combien de copies seront gardés. Nous dé-commentons la ou les lignes que nous désirons en supprimant le # en début de ligne. Il faut que l'ordre des intervalles soit du plus fréquent au moins fréquent. Dans l'exemple ci-dessous, nous avons gardé les

intervalles quotidiens et hebdomadaires par défaut, en spécifiant que les 7 archives quotidiennes et les 4 archives hebdomadaires les plus récentes soient sauvegardées.

```
#interval hourly 6
interval daily 7
interval weekly 4
#interval monthly 6
```

b. Les fichiers à sauvegarder

Une autre section du fichier de configuration nous permet de savoir quels sont les fichiers à inclure ou à exclure de la sauvegarde. Dans l'exemple ci-dessous, nous excluons tous les fichiers se terminant par `~` qui sont en général des sauvegardes des fichiers.

```
include /home/jean/Desktop/download //par exemple
exclude *~
```

De la même façon on peut inclure ou exclure un fichier *spécifique* `include_file` « chemin fichier » ou `exclude_file` «...».

Voici la partie la plus importante du fichier de configuration. Elle définit ce qui doit être sauvegardé et où la sauvegarde devra être stockée. Dans l'exemple ci-dessous, **rsnapshot** va sauvegarder deux répertoires utilisateurs de ma machine.

```
backup /var/log/ localhost/
backup /etc/ localhost/
```

Dans l'exemple ci-dessus, les répertoires `log` et `etc` seront archivés dans le répertoire racine de l'utilisateur `localhost`.

c. Tester la configuration

Pour vérifier, nous lançons la commande suivante en ligne de commande. **Rsnapshot** nous indiquera les erreurs éventuelles qu'il pourra trouver ou nous dira que le fichier de configuration est OK.

```
rsnapshot configtest
```

d. Automatisons

La mise en place automatique de nos sauvegardes peut-être faite à l'aide **cron**. Pour rappel, **cron** permet d'exécuter des tâches à chaque minute, heure, jour, mois de l'année. Ainsi, en tant que `root`, nous lançons la commande **crontab -e** et rajoutons les deux lignes suivantes

```
0 23 * * * /usr/local/bin/rsnapshot daily
0 4 * * 7 /usr/local/bin/rsnapshot weekly
```

La sauvegarde quotidienne sera lancée à 23:00 tous les jours et l'hebdomadaire à 4:00 le septième jour de chaque semaine c'est-à-dire le dimanche.

e. Restaurer les sauvegardes

La solution la plus simple, la moins flexible aussi, est d'interdire l'accès à la snapshot root à tous les utilisateurs. Cependant, en interdisant simplement l'accès à tout le monde, l'utilisateur root sera le seul à pouvoir créer des sauvegardes. Pour monter le snapshot root en lecture seule :

```
mount /home/backups/
```

4. Référence

<http://www.rsnapshot.org/>