

Tutoriel sur Unison

par [Karlesnine](#)

Date de publication : 14/11/2006

Dernière mise à jour :

Ce tutoriel vous accompagnera dans la mise en place de l'outil Unison, afin d'effectuer des sauvegardes entre machines linux mais aussi à partir de clients windows. Nous verrons de plus différents modes de connexion pour transférer les données.

- 1 - Introduction
- 2 - Installation sur différentes plateformes
- 3 - Première utilisation
- 4 - Méthode : De disque à disque
 - 4.1 - Sous linux avec NFS
 - 4.2 - Sous windows
- 5 - Sous Linux via SSH
 - 5.1 - Quels avantages ?
 - 5.2 - Serveur de fichiers
 - 5.3 - Serveur client
 - 5.3.1 - Fichier de configuration unison
 - 5.3.2 - Exemple d'un shell script de backup
 - 5.3.3 - Exemple d'un shell script de backup
- 6 - Remerciements et autres.

1 - Introduction

A la recherche d'une solution souple et adaptable pour effectuer des sauvegardes de données de stations de travail et de serveurs sur un gros serveur de fichiers distant avec une grappe raid6 de 1,2 To (8 disques), la logique me fit lancer un coup d'apt-get lorsque je tomba sur UNISON Le man unison me donna alors cette brève description : "a file-synchronization tool for Unix and Windows" Toutes mes informations sont sur le site web de unison : [cliquez ici](#) Il y a entre autre le [User Manual and Reference](#) qu'il est indispensable de lire. Il débute par quelques exercices faciles pour comprendre le principe de base. Faites vos tests dans un coin pour mieux appréhender l'outil."

2 - Installation sur différentes plateformes

Installation Linux Debian

L'installation sous debian est toujours aussi facile :

```
apt-get install unison
```

Installation sous Windows XP

Il y a différents binaires disponibles en téléchargement sur le site de UNISON mais je recommande de prendre la version stable en ligne de commande, facile à installer et à utiliser en batch avec l'ordonnateur de tâche. Comme cette version porte le nom de unison.win32-textui.exe je vous recommande de le renommer en unison.exe et de le placer dans c:/windows/system32/ afin qu'il soit disponible via les variables d'environnement (path).

Installation sous Aix

Unison est disponible sous IBM Aix également, vous pouvez le télécharger sur la page bullfreeware.com. Les instructions d'installation via smitty sont disponibles sur le même site.

3 - Première utilisation

Simple comme bonjour si vous avez au moins regardé le *User Manual and Reference* qu'il est indispensable de lire.

```
unison dir1 dir2
```

Par défaut unison travaille en mode interactif. Il va donc préparer une liste des fichiers et répertoires présents dans dir1 avec leur attributs (date, taille), puis il vous demandera pour chacun des différences avec dir2, la marche à suivre.

Par exemple pour recopier le fichier file-exemple de dir1 a dir2 voici le dialogue :

```
/dir1      /dir2  
new file-exemple ---->      / [f] f
```

la commande f pour flush indique de procéder.

A la suite de cela il vous demande de valider la globalité de la synchronisation, repondez y.

```
Proceed with propagating updates? []
```

Attention à l'ordre de présentation des répertoires à synchroniser ! Présenter dir2 avant dir1 implique que c'est le repertoire dir2 qui est le référent. Si celui-ci est vide, unison supprimera tout de dir1.

Regardez bien avant de valider.

```
/dir2      /dir1  
----> delete file-exemple      / [f] f
```

4 - Méthode : De disque à disque

Le plus facile reste encore de faire une synchronisation de disque à disque sur un même machine. Il est possible de le faire au travers d'un montage nfs également.

Voici deux exemples de scripts à placer dans la crontab ou dans l'ordonnateur de tâches.

4.1 - Sous linux avec NFS

```
#!/bin/sh

/usr/bin/unison /var/lib -path=mysql /mnt/serveur/host/ -batch -force /var/lib
/usr/bin/unison /var -path=www /mnt/serveur/host/ -batch
```

Ici en deux lignes je sauvegarde :

- mes bases de données présentes dans /var/lib/mysql
- mon site web dans /var/lib/www

j'ai pris soin de monter un partage nfs du serveur serveur dans /mnt/serveur. Pour cela ajoutez la ligne suivante à /etc/fstab et lancer la commande commande mount -a : serveur.karlesnine.com:/partagenfs /mnt/serveur nfs defaults 0 0

Les options utilisées :

- -patch=mysql pour ne prendre en compte qu'un repertoire ou fichier dans un dossier racine pouvant en présenter plusieurs comme dans le cas de /var/lib.
- -force /var/lib oblige unison à synchroniser dans un seul sens. ici ne prenant en compte que les modifications local force /var/lib et les imposant sur le partage nfs.
- -batch mode batch, toutes les modifications sont validées par défaut il n'y a donc de confirmation à donner.

4.2 - Sous windows

Sous windows, il est possible de faire de même via un partage de fichier smb / samba préalablement activé. Je ne détaille pas ici cette manipulation propre a windows.

```
unison "C:\Documents and Settings\Nat\Bureau" "Z:\Nat\Bureau" -batch -force "C:\Documents and Settings\Nat\Bureau"
unison "C:\Documents and Settings\Nat\Mes documents" "Z:\Nat\Mes documents" -batch -force "C:\Documents and Settings\Nat\Mes documents"
unison "C:\Documents and Settings\Nat\Application Data\Thunderbird" "Z:\Nat\Thunderbird" -batch -force "C:\Documents and Settings\Nat\Application Data\Thunderbird"
```

Sous windows la syntaxe reste la même, seule change la présentation des chemins de repertoire. Cependant, faites attention aux doubles quotes.

5 - Sous Linux via SSH

5.1 - Quels avantages ?

Sauvegarder via un connexion ssh offre l'avantage de ne pas utiliser de nfs. On évite donc l'administration de montages non sécurisés par l'utilisation d'un tunnel crypté avec identification par clef.

Pas de NFS pour éviter :

- les accès root sur un serveur de fichiers distant
- les problèmes de montage
- la simple indication donnée par la présence ou le parcours d'une arborescence
- Palier à la faible sécurité de nfs V3.

Ssh pour :

- Avoir des connexions ponctuelles, automatisées et masquées
- Limiter les droits de connexion
- Crypter les tunnels de connexion.
- Avoir une compression minimum du transfert de donnée
- Pouvoir révoquer les droits de connexion de façon centralisé

Unison pour : - La synchronisation bi-directionnelle

- Le filtrage sur les noms de fichiers
- L'intégration avec ssh

5.2 - Serveur de fichiers

Sur le serveur de fichiers, copiez la clef ssh publique créée spécialement pour cette connexion de backup en ajoutant l'option no-pty dans le fichier \$HOME/.ssh/authorized_keys du compte choisi pour recevoir la connexion . Ceci, pour bloquer l'ouverture d'un shell (apportant un petit plus de sécurité) comme dans cette exemple :

```
no-pty ssh-dss AAAAB3NzaC... / ...qxX/27ZZ9X5vAVA== backup
```

Attention à bien choisir le compte utilisé et à configurer les droits d'écriture sur l'arborescence de destination choisie. Dans mon architecture j'ai pris le compte admin qui est qu'un simple compte utilisateur lambda. Ce fichier de configuration apporte le filtrage sur différents noms de fichiers qu'il est trop difficile à sauvegarder ou inutile. -

ignore = Name pour exclure de la synchronisation des fichiers trop souvent modifiés ou volumineux. - ignore = Path .unison pour le pas synchroniser le repertoire de unison présent dans le compte utilisateur qui lance la synchronisation. En effet unison stocke ici ses tables de mise à jour qu'il modifie durant chaque synchronisation. - Logfile Pour que unison sauvegarde les logs ailleurs que dans l'environnement utilisateur.

5.3 - Serveur client

Ajouter dans le répertoire .ssh du compte servant à faire le backup, ici root pour parcourir l'arborescence quelque soit le propriétaire des répertoires, le fichier de configuration particulier nommé config suivant

```
Host backup
Compression yes
ForwardAgent no
Cipher blowfish
User admin
IdentityFile ~/.ssh/backup
HostName serveur.karlesnine.com
```

Ce fichier apporte l'utilisation d'un alias backup pour un ensemble de paramètres. En voici l'explication dans l'ordre :

- La compression
- Le blocage du forward de clé
- Une cryptographie légère blowfish
- Le compte utilisateur cible.
- la définition de la clé
- Le serveur de destination

Je ne peux que vous conseiller de vous reporter au site officiel Openssh pour plus de documentation.

5.3.1 - Fichier de configuration unison

La configuration de unison se fait avec un fichier default.prf placé dans le répertoire \$HOME/.unison du compte servant à faire le backup.

```
# Unison preferences file
ignore = Name {core,*.pyc,*.dat,*.xcu, *.dvi, *.log, *.pid, *.out, unison.log}
ignore = Path .unison
ignore = Name toto
ignore = Name tutu
ignore = Name tmp
ignore = Name temp
logfile = /var/log/unison.log
```

5.3.2 - Exemple d'un shell script de backup

```
#!/bin/sh
```

```

#
# Backup avec Unison
# http://www.cis.upenn.edu/~bcpierce/unison/
# Charles-Christian CROIX
# 15/01/2005
# 15/07/2005

UNISON="/usr/bin/unison
MYSQLDUMP="/usr/bin/mysqldump"

$UNISON /etc ssh://backup//home/backupbyhost/$HOSTNAME/etc -silent -backups -maxbackups 3 -batch
-force /etc
$UNISON /root ssh://backup//home/backupbyhost/$HOSTNAME/root -silent -backups -maxbackups 3 -batch
-force /root

$MYSQLDUMP -u root --all-databases | gzip >
/var/lib/mysql/dump/FullDatabasesBackup_$(hostname).sql.gz
$UNISON /var/lib/mysql/dump ssh://backup//home/backupbyhost/$HOSTNAME/var/lib/mysql/dump/ -silent
-batch -force /var/lib/mysql/dump

cat /var/log/unison.log | mail admin@webank.fr -s "[$(hostname)] Backup"
cat /dev/null > /var/log/unison.log

```

Option de unison :

- ssh ://backup//home/backupbyhost/\$HOSTNAME/etc dans cette ligne nous voyons l'alias ssh backup utilisé.
- -silent pour éviter que le cron utilisé pour lancer ce script ne vous spamme
- -batch pour éviter l'interactivité
- -force pour forcer la mise à jour du serveur de backup même si celui-ci semble être plus récent.
- -backups pour demander de sauvegarder la version précédente d'un fichier devant être synchronisé avant son écrasement. Sur le serveur distant un fichier test.txt sera ainsi recopié en .test.txt.0.unison.bak avant que la nouvelle version de test.txt ne soit écrite.
- -maxbackups 3 Nombre maximum de sauvegarde avec l'option -backups. Ici trois versions comme dans cette exemple : .test.txt.0.unison.bak .test.txt.1.unison.bak .test.txt.2.unison.bak

Concernant l'utilisation de mysqldump :

Dans cette exemple j'ai fait le choix d'extraire l'intégralité des informations des bases de données (data et structure) dans un fichier .sql que le gzip par la suite. C'est ce fichier qui est placé dans un répertoire devant être synchronisé. Plus facilement il est possible de sauvegarder l'arborescence de mysql en entier afin de disposer des fichiers base par base de données et table par table. Attention de stopper mysql si besoin pour accéder au fichier en dehors de toute écriture.

5.3.3 - Exemple d'un shell script aix de backup

Voici le même script que précédemment mais adapté a l'environnement Aix qui ne comporte pas les mêmes variables d'environnement (\$HOSTNAME) ni les même emplacements pour unison. En outre la commande Mail ne répond pas à la même syntaxe.

```

#!/bin/sh
#

```

```
# Backup avec Unison
# http://www.cis.upenn.edu/~bcpierce/unison/
# Charles-Christian CROIX
# 15/01/2005
# 15/07/2005
# 04/08/2005 Aix version

HOSTNAME="aix3"
UNISON="/usr/local/bin/unison"

cat /dev/null > /root/.unison/unison.log
date > /root/.unison/unison.log

for I in `ls -l /home | grep ^d | grep -v lost+found | awk '{print $9}'`;
do /usr/local/bin/unison /home/$I ssh://backup//home/backupbyhost/$HOSTNAME/home/$I -silent
-backups -maxbackups 2 -batch -force /home/$I;
done;

$UNISON /etc ssh://backup//home/backupbyhost/$HOSTNAME/etc -silent -backups -maxbackups 3 -batch
-force /etc
$UNISON /root ssh://backup//home/backupbyhost/$HOSTNAME/root -silent -backups -maxbackups 3 -batch
-force /root

mail -s "[Aix3] Backup" admin@mon_mail.fr < /root/.unison/unison.log
```

La boucle for en shell est destinée à obtenir la liste des seuls répertoires présents dans /home (donc à l'exclusion des fichiers) et à filtrer ceux-ci par un grep -V pattern en fonction de l'environnement particulier de chaque machine aix, puis pour chaque élément de cette liste, réaliser une synchronisation.

6 - Remerciements et autres.

Adapté par [Katyucha](#)

Merci à [Pharaonix](#) pour sa relecture.