Chapitre VII : Sauvegardes et protection des données

Eric Leclercq

Département IEM
http://ufrsciencestech.u-bourgogne.fr
http://ludique.u-bourgogne.fr/~leclercq

17 février 2017

Principes et outils

- Médias disponibles et leur durée de vie
- Sauvergardes totale, différentielles, incrémentales
- Différence entre sauvegarde et sécurité des données via les systèmes RAID ou la synchronisation
- Outils portables : `tar`, `cpio`
- Outils dépendant du système : dump, outils proriétaires
- Solution intégrées : Amanda Backup, Time Navigator etc.

La bande est un média à accès séquentiel, inutile de vouloir utiliser mount (périphérique en mode caractère)

Outils de manipulation de bandes

Le périphérique associé aux bandes (tape) est généralement sous Linux `/dev/st0`. Il existe différentes variantes de ce périphérique, par exemple, `/dev/st0n` ou `/dev/nt0` permettent de ne pas rembobiner la bande après chaque action.
Il est possible d'effectuer quelques actions sur les bandes (commande `mt`) qui peuvent par exemple être utilisées dans la programmation des tâches automatiques :

- `mt status` pour connaître le status du lecteur ;
- `mt -f /dev/st0 retension` permet de retendre la bande ;
- `mt -f /dev/st0 rewind` permet de rembobiner la bande dans le cas de l'utilisation de l'option de non rembobinage ;
- `mt -f /dev/st0 erase` permet l'effacement.

Outils de manipulation de bandes

- Si cela ne donne rien vérifiez que votre carte scsi est correctement reconnue avec : `dmesg | grep scsi`.
- Si vous avez compilé votre noyau vous même n'oubliez pas le *'scsi tape support'* dans la rubrique *'drivers scsi'*.
- Si vous compilez le noyau en utilisant des modules n'oubliez pas de charger le module (`st.o`) via la commande `insmod` ou `modprobe`.

tar et cpio

Les avantages de `cpio` par rapport à `tar` sont :

- possibilité de sauter les sections endommagées de l'archive et continuer les operations de restauration au lieu de renoncer completement ;
- possibilité de sauvegarder les fichiers speciaux tels que devices,liens,pipes,sockets et restaure les fichiers dans l'etat exact (meme heure d'acces,modifie,...) ;
- utilisation de l'espace sur la bande plus efficacement que le format tar ;
- archiver en differents formats.

Les options de cpio

La commande `cpio` sauvegarde sur la sortie standard les fichiers dont on saisit les noms sur l'entrée standard, par défaut le clavier et l'écran. On utilisera donc des redirections. Les options communes les plus courantes sont :

- `-v` : mode bavard pour avoir des informations détaillées ;
- `-c` : sauvegarde des attributs des fichiers sous forme ASCII pour l'échange entre divers OS) ;
- `-B` : augmente la vitesse d'exécution en utilisant une mémoire tampon (5120 octets soit 10 blocs).

Les options de `cpio`

Pour une sauvegarde on utilise généralement les options suivantes :

- `-o` : output, creation de la sauvegarde en sortie ;
- `-L` : sauve les fichiers liés et pas les liens symboliques.

Pour lister le contenu de l'archive on a les options suivantes :

- `-i` : lecture de l'archive en entrée ;
- `-t` : comme pour tar, liste le contenu de l'archive.

Les options de `cpio`

Pour une restauration on utilise les options suivantes :

- `-u` : restauration inconditionnelle, avec écrasement des fichiers qui existent déjà. Par défaut les fichiers ne sont pas restaurés si ceux présents sur le disque sont plus récents ou de même date ;
- `-m` : les fichiers restaurés conservent leur dernière date de modification ;
- `-d` : cpio reconstruit l'arborescence des répertoires et sous-répertoires manquants.

Exemples :

Sauvegarde de l'arborescence courante sur une disquette avec compression :

```
find . print | cpio -ocvB | compress > /dev/fd0
```

Restauration

```
cat /dev/fd0 | uncompress | cpio -iuvBd
```

Mise en place

Un fichier `sauvergarde.sh` incrit dans le service `cron` permet de lancer périodiquement les sauvegardes des différents systèmes de fichiers.

```
#!/bin/bash

#définition des variables du nom de fichiers
#avec date concatenée

index_home=‘date +/root/home_full%d%m%y.idx‘
index_var=‘date +/root/var_full%d%m%y.idx‘
index_opt=‘date +/root/opt_full%d%m%y.idx‘
index_usrlocal=‘date +/root/usrlocal_full%d%m%y.idx‘
index_slash=‘date +/root/slash_full%d%m%y.idx‘
```

Mise en place

Construction des catalogues de chaque sauvegarde utilisés pour simplifier les restauration :

```
cd /
#sauvegarde sur bande des systemes de fichiers et
#renvoie du contenu dans fichiers index

find home -print -depth | cpio -ov -B >
     /dev/nst0 2>$index_home \
find var -print -depth | cpio -ov -B > \
     /dev/nst0 2>$index_var
find opt -print -depth | cpio -ov -B > \
     /dev/nst0 2>$index_opt
find usr/local -print -depth | cpio -ov -B > \
     /dev/nst0 2>$index_usrlocal
find . -print -mount -depth | cpio -ov -B > \
     /dev/st0 2>$index_slash
```

La problématique de la sauvegarde des bases de données

- Quels fichiers sont sensibles ?
- Le mécanisme de transactions et son lien avec les sauvegardes
- Problématique spécifique de la sauvegarde base ouverte

Les solutions envisagées :

- Arrêt de la base
- Sauvegarde d'une extration (transactionnelle) en mode texte
- Logiciels spécifiques
- Mise en place d'un replica qui peut être arrêté pour effectuer les sauvegardes

Scénario de sauvegarde

Un scénario de sauvegarde est défini par :

- le type de sauvegarde totale, différentielle ou incrémentale
- des caractéristiques temporelles :
  - la fréquence de sauvegarde (heures, jours, mois)
  - la fenêtre d'ouverture (période pendant laquelle les sauvegardes sont réalisables)
  - la durée de validité des sauvegardes
- l'espace à sauvegarder qui est défini par une liste d'inclusion de fichiers ou de répertoires et une liste d'exclusion de fichiers ou de répertoires.

Le scénario de sauvegarde est fonction du type d'utilisation de la machine à sauvegarder et en particulier de certains critères comme le taux de modification des fichiers, l'importance des fichiers, le volume etc.

Le service cron

Les systèmes de type Unix possèdent un outil (plus exactement un démon) permettant de réaliser le lancement de tâches périodiques :`cron`.

- `cron` utilise une table référençant les tâches à lancer ainsi que l'année, le mois, le jour, l'heure et la minute à laquelle l'exécuter
- le démon `crond`, lance automatiquement les tâches en fonction de la table
- la commande `crontab` permet d'éditer la table des tâches périodiques planifiées

La commande `crontab -e` édite un fichier propre à chaque l'utilisateur qui l'exécute. Ce fichier se situe dans : `/var/spool/cron/crontabs/login`

Format de la contab

La commande `crontab -e` permet de définir ou modifier la table au moyen d'un éditeur de texte défini par la variable `EDITOR`
Le format général d'une entrée de la crontab est :

```
mm hh jj MM JJJ commande > log
```

- mm représente les minutes (de 0 à 59)
- hh représente l'heure (de 0 à 23)
- jj représente le numéro du jour du mois (de 1 à 31)
- MM représente le numéro du mois (de 1 à 12) ou l'abréviation du nom du mois sur 3 lettres (jan, feb, mar, apr, etc.)
- J représente l'abréviation du nom du jour ou le chiffre correspondant au jour de la semaine (0=dimanche, 1=lundi, etc.)
- `log` est le nom d'un fichier dans lequel stocker le journal des opérations. Si la clause `> log` n'est pas spécifiée, cron enverra automatiquement un email de confirmation, pour l'éviter il suffit de spécifier `> /dev/null`

Format de la contab

Pour chaque unité de temps (minute, heure, jour, mois, jour de la semaine) les notations acceptées sont :

- `*` : a chaque unité de temps
- `a-b` : les unités de temps (par exemple 2-5 = 2,3,4,5)
- `*/x` : toutes les x unités de temps (par exemple */5 = 0,5,10,etc.)
- `a,b` : les unités de temps (par exemple 5,8 = 5 et 8)

Les utilisateurs ne peuvent pas définir de crontab par défaut : utiliser les fichier `cron.allow` et `cron.deny` pour contrôler la fonction. Avec certaines implémentation il faut aussi appartenir au groupe cron.

Comportement hérité

cron définit un comportement par défaut /etc/crontab

contab

```
# Global variables
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
# check scripts in cron.hourly, cron.daily, cron.weekly and cron.monthly
0   *   * * *     root    rm -f /var/spool/cron/lastrun/cron.hourly
1   3   * * *     root    rm -f /var/spool/cron/lastrun/cron.daily
15  4   * * 6     root    rm -f /var/spool/cron/lastrun/cron.weekly
30  5   1 * *     root    rm -f /var/spool/cron/lastrun/cron.monthly
*/10   *   * * *  root    test -x /usr/sbin/run-crons && /usr/sbin/run-crons
```

Dans /etc on retrouve des répertoires contenant les scripts à lancer.

Utilisations de RSYNC pour les sauvegardes

rsync est (remote synchronization) est un outil de synchronisation unidirectionnelle de fichiers multi-plateformes :

- fonctionnement en mode démon
- fonctionnement au travers de ssh
- fonctionnement local entre file systèmes
- permet les sauvergardes incrémentales

contab

```
rsync -rvaz  --exclude '.DS*' -e ssh Personnel \
leclercq@192.168.0.11:/home1/leclercq/DONNEES
```

Utilisations de RSYNC pour les sauvegardes

contab

```
DATE=‘date␣"+%Y-%m-%d-%H:%M:%S"␣‘
rsync -prvaz  --exclude ’.DS*’  \
-b --backup-dir=/home1/leclercq/DONNEES/BACKUP/$DATE \
-e ssh Travail leclercq@192.168.0.11:/home1/leclercq/DONNEES/
```