



Stratégie de sauvegarde EOLE 2.3

Benjamin Bohard

Cadoles

22 novembre 2011

Plan

- 1 Théorie de la sauvegarde
 - Objectif
 - Méthodes de sauvegarde
 - Méthodes de restauration
- 2 Solution utilisée
 - Logiciel
 - Infrastructure
 - Configuration
- 3 EOLE 2.2
 - Intégration
 - Inconvénients rencontrés
- 4 EOLE 2.3
 - Intégration
 - Améliorations
 - schedule

Plan

- 1 Théorie de la sauvegarde
 - Objectif
 - Méthodes de sauvegarde
 - Méthodes de restauration
- 2 Solution utilisée
 - Logiciel
 - Infrastructure
 - Configuration
- 3 EOLE 2.2
 - Intégration
 - Inconvénients rencontrés
- 4 EOLE 2.3
 - Intégration
 - Améliorations
 - schedule

L'objectif principal de la sauvegarde est la restauration

La sauvegarde totale consiste à sauvegarder la totalité des fichiers sélectionnés.

Avantage

La sauvegarde totale est la méthode la plus fiable puisqu'elle assure à elle seule l'intégrité des données sauvegardées.

Inconvénient

La sauvegarde totale est coûteuse en temps et en espace.

La sauvegarde incrémentale consiste à sauvegarder les fichiers modifiés ou créés depuis la dernière sauvegarde.

Avantage

Les volumes à sauvegarder sont moins importants que dans le cas d'une sauvegarde totale, d'où un gain de temps et d'espace.

Inconvénient

Les sauvegardes ne se suffisent pas à elles-mêmes pour restaurer les données.

La sauvegarde différentielle consiste à sauvegarder les fichiers modifiés ou créés depuis la dernière sauvegarde totale

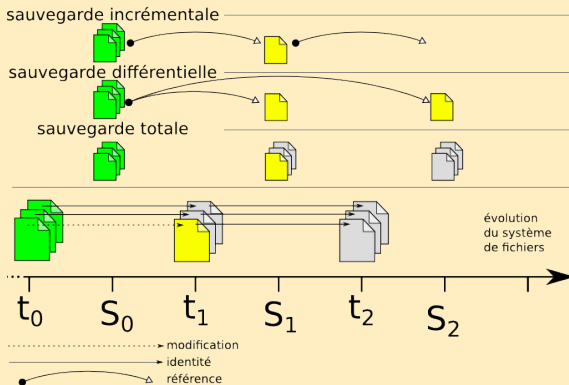
Avantage

Les volumes à sauvegarder sont moins importants que dans le cas d'une sauvegarde totale, d'où un gain de temps et d'espace.

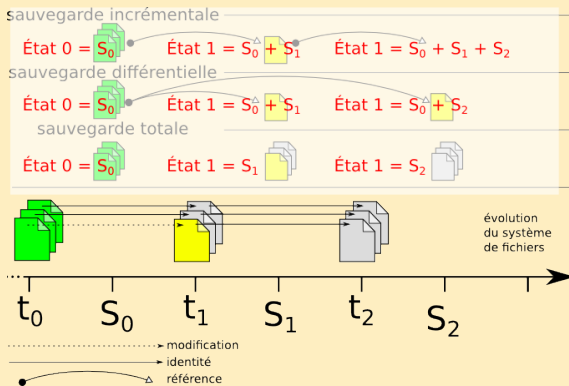
Inconvénient

Les sauvegardes ne se suffisent pas à elles-mêmes pour restaurer les données.

Comparaison des types de sauvegarde



Comparaison des méthodes de restauration



Deux méthodes de restauration sont possibles :

restauration totale Restauration de la totalité des fichiers sauvegardés. La restauration à partir d'une sauvegarde totale est plus rapide. La restauration à partir d'un autre type de sauvegarde nécessite la parcourir de la sauvegarde totale de référence puis de toutes les autres sauvegardes incrémentales ou de la dernière sauvegarde différentielle.

restauration partielle Restauration d'une partie des fichiers sauvegardés uniquement. Le bénéfice en terme de vitesse de la restauration depuis une sauvegarde totale n'est pas automatique. Il est dépendant du volume de données à parcourir pour trouver ces fichiers. Le volume est maximal mais contenu dans une seule archive dans le cas d'une sauvegarde totale. Il est variable et peut être contenu dans, au moins, deux archives dans les autres cas.



Deux méthodes de restauration sont possibles :

restauration totale Restauration de la totalité des fichiers sauvegardés. La restauration à partir d'une sauvegarde totale est plus rapide. La restauration à partir d'un autre type de sauvegarde nécessite la parcourir de la sauvegarde totale de référence puis de toutes les autres sauvegardes incrémentales ou de la dernière sauvegarde différentielle.

restauration partielle Restauration d'une partie des fichiers sauvegardés uniquement. Le bénéfice en terme de vitesse de la restauration depuis une sauvegarde totale n'est pas automatique. Il est dépendant du volume de données à parcourir pour trouver ces fichiers. Le volume est maximal mais contenu dans une seule archive dans le cas d'une sauvegarde totale. Il est variable et peut être contenu dans, au moins, deux archives dans les autres cas.



Deux méthodes de restauration sont possibles :

restauration totale Restauration de la totalité des fichiers sauvegardés. La restauration à partir d'une sauvegarde totale est plus rapide. La restauration à partir d'un autre type de sauvegarde nécessite la parcourir de la sauvegarde totale de référence puis de toutes les autres sauvegardes incrémentales ou de la dernière sauvegarde différentielle.

restauration partielle Restauration d'une partie des fichiers sauvegardés uniquement. Le bénéfice en terme de vitesse de la restauration depuis une sauvegarde totale n'est pas automatique. Il est dépendant du volume de données à parcourir pour trouver ces fichiers. Le volume est maximal mais contenu dans une seule archive dans le cas d'une sauvegarde totale. Il est variable et peut être contenu dans, au moins, deux archives dans les autres cas.



Plan

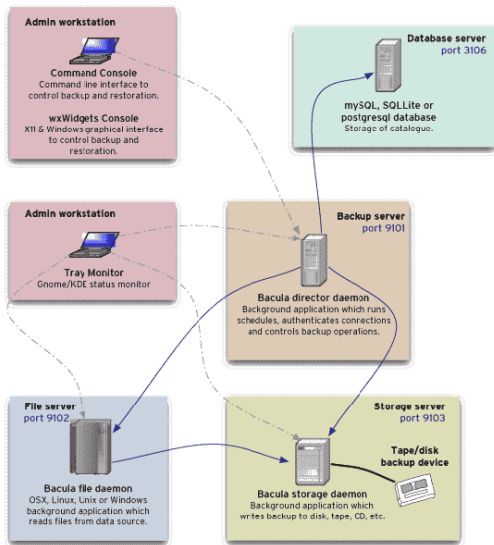
- 1 Théorie de la sauvegarde
 - Objectif
 - Méthodes de sauvegarde
 - Méthodes de restauration
- 2 **Solution utilisée**
 - Logiciel
 - Infrastructure
 - Configuration
- 3 EOLE 2.2
 - Intégration
 - Inconvénients rencontrés
- 4 EOLE 2.3
 - Intégration
 - Améliorations
 - schedule

Bacula

« *Un ensemble de programmes qui vous permet de gérer vos sauvegardes, restaurations ou vérifications de données d'un ordinateur sur un réseau hétérogène.* »

logiciel sous licences :

- GPL
- LGPL
- Public domain



© Aristedes Maniatis

Configuration par trois fichiers principaux :

`bacula-dir.conf` configuration du directeur, des tâches, des groupes de fichiers à sauvegarder, des ensembles de supports de sauvegarde.

`bacula-fd.conf` configuration du client où se trouvent les fichiers à sauvegarder

`bacula-sd.conf` configuration du serveur de stockage

Plan

- 1 Théorie de la sauvegarde
 - Objectif
 - Méthodes de sauvegarde
 - Méthodes de restauration
- 2 Solution utilisée
 - Logiciel
 - Infrastructure
 - Configuration
- 3 **EOLE 2.2**
 - Intégration
 - Inconvénients rencontrés
- 4 EOLE 2.3
 - Intégration
 - Améliorations
 - schedule

Un ensemble de scripts dans le répertoire /usr/share/eole/bacula :

```
    baculaeole.py
    baculasupport.py
restore_quotas.py
    baculamessage.py
    envoiemessage.py
    baculaservices.sh
bacula_recherche.sh
    gen_catalog.sh
    regen_catalog.sh
    montage.sh
    test_montage.sh
restauration-bacula.sh
sauvegarde-bacula.sh
```

L'Eolisation version 2.2 du service de sauvegarde pose quelques problèmes :

- pas d'inclusions possibles dans la définition des groupes de fichiers à sauvegarder, donc un manque de flexibilité pour adapter ces groupes de fichiers aux services présents sur le serveur,
- pas de séparation physique du serveur de sauvegarde,
- pas de gestion de la sauvegarde différentielle,
- pas de programmation mensuelle,
- conflits des tâches cron et bacula mal gérés.

Plan

- 1 Théorie de la sauvegarde
 - Objectif
 - Méthodes de sauvegarde
 - Méthodes de restauration
- 2 Solution utilisée
 - Logiciel
 - Infrastructure
 - Configuration
- 3 EOLE 2.2
 - Intégration
 - Inconvénients rencontrés
- 4 EOLE 2.3
 - Intégration
 - Améliorations
 - schedule

Un ensemble de scripts dans le répertoire `/usr/share/eole/bacula` :

- `baculaconfig.py` gère la programmation des sauvegardes
- `baculamount.py` gère la gestion des périphériques de stockage
- `bacularestore.py` gère la restauration des données

liste des principales options de `baculaconfig.py` :

- `-h, -help` affiche la liste des options
- `-d, -display` affiche la configuration en cours, support de sauvegarde, adresse mail pour les notifications, sauvegardes programmées.
- `-a, -apply` applique les configurations
- `-n, -now` lance une sauvegarde immédiate
- `-s, -support` permet la configuration du support de sauvegarde
- `-m, -mail` permet la configuration de l'adresse mail de notification
- `-j, -j` programme une sauvegarde
- `-lock, -unlock` créé et supprime un verrou de sauvegarde

Exemple d'utilisation du front-end

Lancer une sauvegarde totale immédiate

```
baculaconfig.py -n -level=Full
```

Programmer une sauvegarde incrémentielle quotidienne à minuit

```
baculaconfig.py -j daily -job_hour=0
```

```
-job_level=Incremential
```

L'Eolisation version 2.3 du service de sauvegarde a apporté beaucoup d'améliorations :

- les groupes de fichiers sont définis dans le répertoire `/etc/bacula/baculafichiers.d` et importés dans le fichier de configuration de bacula au besoin,
- le serveur de stockage peut être déporté sur une autre machine, donc dans un autre lieu,
- la configuration est unifiée entre les différentes interfaces (EAD, `gen_config`, `baculaconfig.py`). L'interface en ligne de commande est unifiée et simplifiée.

Eole-schedule a pour but d'organiser l'exécution des deux processus de programmation d'événements, cron et bacula :

- Les paramètres de base de programmation des tâches (heure, jour, minute) sont tirés au sort lors de l'instanciation.
- Les tâches peuvent avoir quatre fréquences différentes (jour, semaine, mois, une seule fois).
- En cas de conflit entre des plages d'exécution des tâches, Eole-schedule permet de donner la main à bacula.

Le script important :

`/usr/share/eole/schedule/manage_schedule.py`. Ce script permet la gestion des tâches :

`pre`, `post` liste des tâches programmées
`add`, `del` ajout, suppression de tâches

Contrôler le déroulement des tâches

L'exécution peut être suivie dans le fichier de log
`/var/log/schedule.log`

Principes des traitements pre :

- exécution toutes les nuits,
- exécution à l'heure programmée, à l'heure d'une sauvegarde programmée ou juste avant une sauvegarde immédiate,
- exécution des scripts d'extraction des bases de données (MySQL, LDAP, ...) dans l'ordre alphabétique.

Principes des traitements post

- exécution à l'heure tirée au sort lors de l'instanciation (gestion par cron) ou après la sauvegarde (gestion par bacula) si les plages horaires sont en conflit,
- mises à jour (blacklist, système, ...).

Merci