Système d'information et base de données

CM5 : Sauvegarde, réplication et sécurité

Mickaël Martin Nevot

V3.2.0

Système d'information et base de données

Sauvegarde MySQL

- Une sauvegarde MySQL peut être très simple :
  - Les bases de données sont des répertoires
  - Les tables sont des fichiers
- Commandes MySQL / système de *backup* :
  - `SELECT... INTO OUTFILE` / `LOAD DATA INFILE` (MySQL)
  - `mysqldump`
  - `mysqlhotcopy`
  - `rsync`

Méthodes de sauvegarde MySQL

- **Duplication de fichiers** (système d'exploitation) :
  - Copie directe des fichiers et des répertoires
  - `mysqlhotcopy`
  - `rsync`

- Création de **code** de LDD SQL :
  - `mysqldump`
- Sous forme de **fichier texte :**
  - `SELECT... INTO OUTFILE` / `LOAD DATA INFILE`

Critères de sauvegarde

- À chaud :
  - Serveur en marche
- À froid :
  - Serveur arrêté
- Complète :
  - Toutes les données
- Incrémentale :
  - Uniquement les modifications depuis la dernière sauvegarde

Types de sauvegardes MySQL

Sauvegarde	À chaud	À Froid	Complète	Incrémentale
SELECT... INTO OUTFILE / LOAD DATA INFILE	✓	✗	✓	✗
mysqldump	✓	✗	✓	✓
mysqlhotcopy	✓	✗	✓	✗
rsync	✗	✓	✓	✗

- Pour toutes sauvegardes **consistantes** à chaud :

```
mysql> LOCK TABLES;
mysql> FLUSH TABLES;
```

Verrouille l'utilisation des tables d'une base de donnée pour une copie sécurisée

SELECT … INTO OUTFILE

- **SELECT … INTO OUTFILE :**

Syntaxe :
```
SELECT … INTO OUTFILE 'file_name' [OPTIONS] FROM table_name
```

- **LOAD DATA INFILE :**

Syntaxe :
```
LOAD DATA INFILE ‘file_name' REPLACE ...
```

- Utile pour des ***dumps*** **rapides**

En informatique, un *dump* est une copie brute de l'état d'une mémoire ; le terme vient de l'anglais, *to dump*, qui signifie : déverser

Pour éviter les lignes dupliquées, il est nécessaire d’avoir une clef primaire ou une clef unique : on peut alors utiliser le mot clef `REPLACE`

mysqldump

- Sauvegarde de bases de données :

```
shell> mysqldump --opt database > backup-file.sql
```

  - Ou (plusieurs base) :

```
shell> mysqldump --databases database1 database2 > backup-file.sql
```

  - Ou (toutes les bases) :

```
shell> mysqldump --all-databases > backup-file.sql
```

Syntaxe :

```
mysqldump [OPTIONS] database [tables]
mysqldump [OPTIONS] --databases [OPTIONS] DB1 [DB2 DB3...]
mysqldump [OPTIONS] --all-databases [OPTIONS]
```

```
C:\>mysqldump mydatabase
mysqldump: Got error: 1045: Access denied for user 'ODBC'@'localhost' (using pas
sword: NO) when trying to connect

C:\>mysqldump -uusername -ppassword mydatabase
-- MySQL dump 10.12
--
-- Host: localhost    Database: mydatabase
-- ------------------------------------------------------
-- Server version       5.2.0-falcon-alpha-community-nt

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0
*/;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

mysqldump

- Copie de serveur à serveur :

```
shell> mysqldump --opt database | mysql ---host=remote-host -C database
```

- Restauration :

```
shell> mysql -u user -ppassword database < backup-file.sql
```

  - Ou :

```
shell> mysql -u user -ppassword -e "source /patch/backup-file.sql" database
```

- **Sauvegarde incrémentale** :
  - `mysqld` doit être démarré avec l'option : `--bin-log` (ou `--log-update` pour les versions antérieures à 4.1.3)
  - Chaque période entre deux copies constitue un incrément

mysqlhotcopy

- Sauvegarde d'une base de données avec toutes les tables dans `--datadir` (`/var/lib/mysql/` par défaut)
- Méthode **la plus rapide**
- Moyen **le plus sûr** de copie
- Mais ne fonctionne que sur la **machine locale**
- `LOCK TABLES` / `FLUSH TABLES` automatiques

Syntaxe :

```
mysqlhotcopy [OPTIONS] db_name_1 ... db_name_n /path/to/new_directory
```

```
[local-host]# /usr/bin/mysqlhotcopy -u root -p My2Secure$Password sugarcrm /home/backup/database --allowold --keepold
Locked 98 tables in 0 seconds.
Flushed tables (`sugarcrm`.`accounts`, `sugarcrm`.`accounts_audit`, `sugarcrm`.`accounts_bugs`) in 0 seconds.
Copying 295 files...
Copying indices for 0 files...
Unlocked tables.
mysqlhotcopy copied 98 tables (295 files) in 0 seconds (0 seconds overall).
```

Pour les sauvegardes à chaud

rsync

Syntaxe : `rsync source/ destination/`

- ***remote synchronization*** (synchronisation à distance)
- Hors du contrôle de `mysqld`
- Serveur **arrêté**
- Options :
  - `-a`, `--archive` : archive les fichiers (équivalent à `-rlptgoD`)
  - `-z` : compresse les fichiers
  - `-v` : verbosité
  - `-e ssh` : utilise SSH (très utile !)
  - `--delete-after` : supprime les fichiers à la fin de la copie
- Utilisation recommandée :

```
shell> rsync -avz /var/lib/mysql/mybase login@serveur.org:/destination/
```

Souvent utilisé pour cette utilisation mais pas obligatoirement

`-o` et `-D` uniquement avec les droit d'administrateur

!

Restaurations recommandées

- En premier (marche dans 99 % des cas) :

```
mysql> REPAIR TABLE … ;
```

  - Ou :

```
shell> myisamchk -r tbl_name
```

- Sinon :

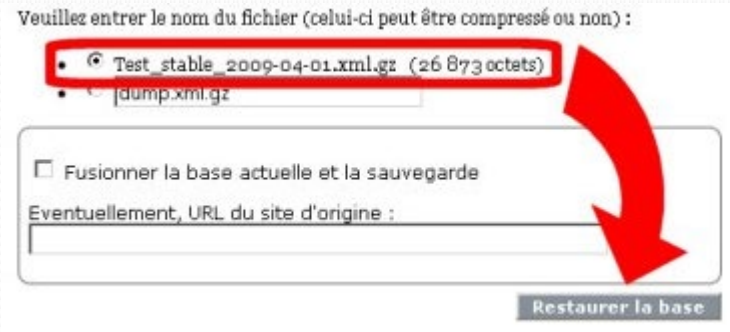
```
shell> mydsqldump …
```

- Remettre en marche les mises à jours dans le log binaire :

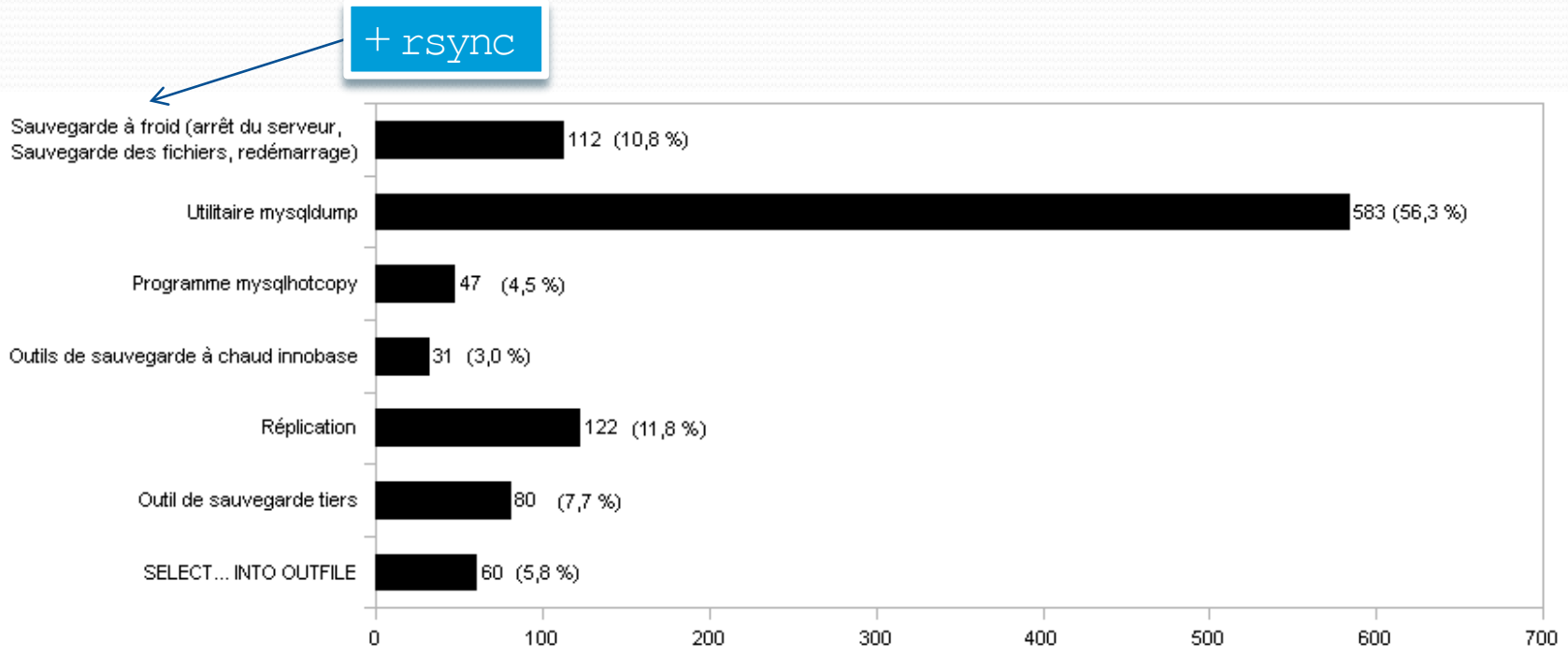
```
shell> mysqlbinlog hostname-bin.[0-9]* | mysql
```

- Ne pas oublier de débloquer des tables au besoin :

```
mysql> UNLOCK TABLES;
```



Utilisation des sauvegardes MySQL



Réplication

Réplication ≠ sauvegarde (dynamique ≠ statique)

- Système maître-esclave(s) (maître = source)
- Importante sécurité des données
- Types de réplication :
  - Active : calculs effectués par le maître et le(s) esclave(s)
  - Passive : calculs effectués par le maître et propagés ensuite
- Types de mise à jour :
  - Synchrone : synchronisation en temps réel
  - Asynchrone : stocke puis propage (gestion de priorité)

MySQL

Natif dans MySQL (pour les tables MyISAM !)

Configuration du maître

- Il faut avoir les droits `REPLICATION SLAVE` sur le **maître** (et sur le esclaves en cas de panne) :

```
mysql> GRANT REPLICATION SLAVE ON *.* TO replication@'%' IDENTIFIED BY 'replication';
```

- `my.cnf` du **maître** :

```
[mysqld]
# Activation des logs binaires
log-bin=/var/log/mysql/mysql-bin.log
# Définition de l'identifiant unique
server-id=1
```

  - Filtre de réplication :

```
# Réplique la base de données
replicate-do-db=database
# Réplique les requêtes multi-bases
replicate-wild-do-table=cdatabase.%
```

Attention : autoriser les connexions distantes si serveur(s) esclave(s) sur autre(s) machine(s) : `#bind-address=127.0.0.1`

L'activation des logs binaires est obligatoire car un serveur esclave va se connecter au maître et regarder sa position dans le fichier

Pour paramétrer les bases de données ou tables à répliquer (tout par défaut)

Configuration d'un esclave

- `my.cnf` d'un **esclave** :

```
[mysqld]
# Activation des logs binaires
log-bin=/var/log/mysql/mysql-bin.log
# Définition de l'identifiant unique
server-id=2
```

Important en cas de panne

Différent du maître !

- Lancer l'esclave :

```
mysql> CHANGE MASTER TO MASTER_HOST='master host name', MASTER_USER='replication',
MASTER_PASSWORD='replication', MASTER_LOG_FILE='log', MASTER_LOG_POS=offset;
```

Obtenable en faisant `SHOW MASTER STATUS` sur le maître

- Démarrer le processus esclave :

```
mysql> START SLAVE;
```

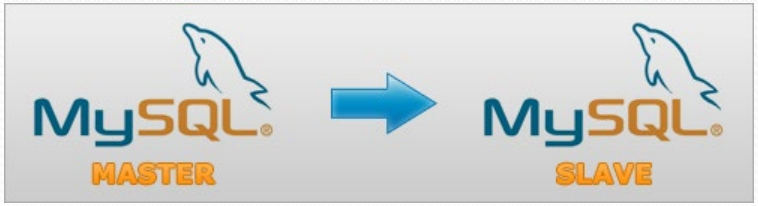
L'esclave vérifiera en « continu » s'il est synchronisé avec son maître

Il est possible de faire des réplications en chaîne : un serveur maitre peut ainsi être esclave d'un autre serveur, etc.

Réplication : gestion de panne

- Panne esclave : **invisible**
- Panne maître : service qui peut être ininterrompu grâce à(aux) esclave(s) et remplacement **sans interruption** :
  - Uniquement si `log-bin` est activé dans esclave
  - Arrêter le processus esclave (et le transformer en maître) :
    ```
    mysql> STOP SLAVE;
    ```
  - Transformer le processus esclave en maître :
    ```
    mysql> RESET MASTER;
    ```

Il peut être intéressant d'écrire un script vérifiant l'état du master et d'utiliser un DNS modifiable dynamiquement grâce à la commande `nsupdate` en cas d'utilisation de BIND



Réplication : gestion de panne

- L'ancien maître devient esclave :

```
mysql> CHANGE MASTER TO
       MASTER_HOST='slave_server',
       MASTER_PORT=3306,
       MASTER_USER='replication',
       MASTER_PASSWORD='replication';
```

- Réaffecter l'ancien maître restauré :

```
mysql> RESET MASTER;
```

Côté (ancien) maître

- L'ancien esclave redevient esclave :

```
mysql> CHANGE MASTER TO
       MASTER_HOST='master_server',
       MASTER_PORT=3306,
       MASTER_USER='replication',
       MASTER_PASSWORD='replication';
```

- Démarrer le processus esclave :

```
mysql> START SLAVE;
```

Côté (ancien) esclave

Sécurité (générale)

- Mettre un **mot de passe** à `root` (aucun par défaut)
- **Supprimer le second compte administrateur** qui se connecte depuis `hostname` car inutile :

```
mysql> DELETE FROM user WHERE Host='hostname' AND user='root';
```

- Interdire les **connexions non identifiées** :

```
mysql> DELETE FROM user WHERE Password='';
```

- Supprimer la **base** `test` et toutes les bases commençant par `test_` car elles sont accessibles par tout le monde :

```
mysql> DELETE FROM db WHERE db='test' OR db='test\_%';
```

- Ne donner que les **privilèges nécessaires** aux utilisateurs (avec une attention particulière au privilège `GRANT`)

Sécurité (fichier my.cnf)

- Interdire les connexions distantes (si possible) :

```
bind-address = 127.0.0.1
```

- Limiter le nombre de connexions simultanées au serveur :

```
# 100 connexions maximum au total
max_connections=100
# 1 utilisateur a le droit à 50 connexions au maximum
max_user_connections=50
```

Sécurité (avancée)

- Sensibilisation générale (avec une **matrice de sensibilité**), notamment face à l'**ingénierie sociale** et pour les développeurs contre les risques d'**injection SQL**
- **Restriction réseau** (limiter les connexions au port)
- Virtualisation :
  - ***chrooting*** (racine virtuelle dans un dossier)
  - **Noyau en espace utilisateur** (plusieurs noyaux virtuels)
  - **Machine virtuelle** (émulation d'une machine complète)
  - **Paravirtualisation** :
    - Noyau en espace utilisateur + machine virtuelle

Liens

- Documents électroniques :
  - http://www.elliptic.fr/doc/mysql
  - http://dev.mysql.com/doc/refman/5.5/en
  - http://greg.rubyfr.net/pub/?page_id=26
  - http://www.pariscyber.com/security/secure_mysql.php
  - http://www.cgsecurity.org/Articles/mysql.html
  - http://jgrondin.developpez.com/article/MySQL/Replication_MySQL
- Documents classiques :
  - Cours :
    - Maurice Libes. *Administration et exploitation du SGBDR MySQL*.
    - Bertrand Liaudet. *Base de données, administration*.
  - Livre :
    - Robin Schumacher. *Un aperçu de la nouvelle sauvegarde sous MySQL 6*.

Crédits

Auteur

Mickaël Martin Nevot
mmartin.nevot@gmail.com

Carte de visite électronique

Relecteurs

- Christophe Delagarde

Cours en ligne sur : www.mickael-martin-nevot.com