

## Administration de MySQL Configuration des journaux

### Partie I

---

#### 1. Le journal binaire

Point Rappel. Le journal binaire ou *binlog* est chargé de stocker, sous un format binaire, toutes les requêtes qui modifient les objets de la base de données (INSERT, UPDATE, DELETE, DROP, CREATE, ALTER...). C'est aussi l'élément central de la réplication MySQL.

##### Questions

- ⇒ Activez le journal binaire (*binlog*) et définissez son fichier d'index.
- ⇒ Quel est le fichier binaire courant
- ⇒ Combien de fichiers présents depuis la dernière purge ? affichez-les.
- ⇒ Désactivez la journalisation binaire de votre session
- ⇒ Pouvez-vous la désactiver de manière globale ?
- ⇒ Localisez le journal des logs.
- ⇒ Localiser tous les fichiers de MySQL (commande déjà vue lors de l'installation/configuration)
  
- ⇒ Purgez les fichiers binaires, en utilisant les deux méthodes possibles :
  - Méthode purge manuelle
  - Méthode 'date d'expiration' :
    - Configurer le fichier de configuration pour effacer tous les journaux binaires vieux de plus de 1 jour
  
- ⇒ Le journal des erreurs est un fichier qui grossit au fil du temps. En général, des règles bien définies permettent d'assurer automatiquement la purge. En attendant, nous vous demandons de le faire manuellement.
  - Fermez, via une commande, le fichier courant.
  - Indiquez dans quel cas de figures le fichier binaire est ouvert.
  
- ⇒ Réinitialiser la journalisation binaire

#### 2. Le journal des requêtes lentes

##### Questions

- ⇒ Activer le journal des requêtes lentes.
  - L'activation se fait dans le fichier de configuration et dans la section [mysqld]
  - Spécifier un chemin pour ce fichier.
- ⇒ Changer le fichier de configuration pour mettre un seuil faible pour les requêtes lentes (Par défaut, il est à 10ms).
- ⇒ Configurez l'interception de toutes les requêtes sans index.
- ⇒ Redémarrer le serveur
- ⇒ Tracer l'exécution d'une requête avec la commande EXPLAIN (une requête quelconque du TP1)

- ⇒ Analyser le fichier des logs des requêtes lentes et analyser les informations relatives à la requête de la question précédente
- ⇒ Quelles sont donc les différentes informations que l'on peut trouver dans le journal des requêtes lentes ?
- ⇒ Utiliser l'outil natif mysqldumpslow pour analyser la même information
  - Comment mysqldumpslow classe-t-il les informations du journal ?
  - Quel outil préférez-vous ?
- ⇒ Activer le journal des requêtes lentes avec une orientation vers une table systèmes (pensez au paramètre `log_output`).
- ⇒ Afficher les informations contenues dans cette table
- ⇒ Peut-on accéder à cette table en écriture

### 3. Le journal général des connexions et requêtes

Le journal général (general log) permet d'enregistrer toute l'activité du serveur. Il stocke les informations relatives à la connexion et à la déconnexion des clients ainsi que toutes les requêtes et toutes les commandes qui arrivent au processus mysqld, qu'elles soient valides ou non.

L'intérêt d'un tel journal se situe particulièrement dans les phases d'optimisation, pendant l'audit du serveur ou encore lorsqu'on souhaite identifier un problème venant de l'application. Ce fichier peut grossir très rapidement sur une machine qui subit une forte charge. Par conséquent, il peut, de par sa lourdeur, rapidement devenir inexploitable. En effet, rares sont les éditeurs de texte qui permettent de manipuler avec aisance des fichiers de plusieurs dizaines ou centaines de mégaoctets.

Un autre inconvénient, et non des moindres, vient du fait que l'activation du journal général représente une charge de travail supplémentaire pour la machine, essentiellement au niveau des E/S. Cette surcharge dépend bien évidemment de l'application et des caractéristiques de la machine. Néanmoins celle-ci peut être estimée, lorsque la journalisation s'effectue dans un fichier, à environ 30 %. Il est donc fortement conseillé de ne l'activer que de façon ponctuelle, dans les phases d'audit, d'optimisation, de débogage...

Question :

- ⇒ Activez le fichier des logs grâce aux commandes :
- ⇒ Indiquez le nom et le chemin du fichier grâce :
- ⇒ Analysez le fichier `mysql.log`
- ⇒ Désactivez ce journal pour votre session avec le paramètre `sql_log_off`.
- ⇒ Configurez votre fichier pour orienter les logs dans une table : système (`mysql.general_log`)

## Partie II

---

### Rotation des logs

Pour faciliter la gestion du journal des erreurs, vous pouvez utiliser l'outil de maintenance logrotate (<https://linux.die.net/man/8/logrotate>) et planifier des tâches (cron) pour éventuellement compresser et déplacer l'historique trop ancien.

Nous allons donc utiliser le `mysql-log-rotate`. Si vous avez installé MySQL à partir d'une distribution RPM, ce script aurait dû être installé automatiquement.

Ce script se trouve par exemple dans un environnement MacOS par ici : `/usr/local/Cellar/mysql/8.0.19/support-files`.

**Question : Les actions à spécifier dans votre fichier de rotation sont les suivantes :**

- ⇒ La rotation doit être effectuée tous les jours
- ⇒ `dateext` : au lieu d'avoir un nombre (1, 2, 3, ...), on demande d'utiliser une date (par défaut au format YYYYMMDD)
- ⇒ Le nombre de versions à conserver (ici 56 jours)
- ⇒ On demande de ne pas compresser les versions
- ⇒ Ne pas réduire le fichier à zéro après la copie
- ⇒ Faire la rotation même si le fichier est vide
- ⇒ Déclenche une erreur (non bloquante) si le fichier n'existe pas, ça peut être utile pour détecter une sauvegarde qui ne s'est pas effectuée
- ⇒ On ne recrée pas le fichier d'origine (`/backups/mysql/all-databases.sql.gz` dans mon exemple)
- ⇒ On ne déplace pas les versions dans un autre dossier
- ⇒ Ne pas envoyer les nouvelles versions par mail (ça risque de faire de gros messages sinon)
- ⇒ Conserver l'extension du fichier à la fin du nom (par défaut `fichier.toto` devient `fichier.toto-YYYYMMJJ`)