MySQL Optimisation

Objectifs

- tirer le maximum du matériel disponible
- tirer le maximum de l'application qui utilise la base de données
- répondre à l'accroissement du volume des données
- répondre à l'accroissement de la charge : nombre de connexions, nombre d'utilisateurs
- éviter de recoder l'application par manque de performances → trop cher
- satisfaire le client

Optimiser la structure de la base de données

- choisir les bons types de données : optimisation du stockage, facilité de manipulations
- choisir la bonne taille des données
- éliminer les données redondantes
- optimiser les jointures entre les tables
- stocker des données calculées ou faire les calculs par requêtes ?
- ajouter des colonnes intelligentes : auto_increment, valeurs par défaut, valeurs encadrées (de x à y)…
- coder correctement les requêtes (pas de SELECT * …)

Choix du moteur

- un moteur de stockage est un ensemble d'algorithmes utilisés pour stocker les informations et y accéder au moyen de requêtes SQL
- MySQL et son fork MariaDB permettent de choisir pour chaque table quel moteur on veut utiliser
- plusieurs moteurs peuvent coexister dans la même base
- c'est choix de conception avec avantages et inconvénients
- néanmoins, très souvent un seul moteur est retenu par base

InnoDB

- InnoDB est un moteur transactionnel : il permet de gérer des transactions et des retours en arrière (rollbacks)
- InnoDB est un moteur relationnel : on peut créer des relations entre les données de plusieurs tables
- modifier des données génère des modifications en cascade aux tables liées par des instructions simples : ON UPDATE CASCADE, ON DELETE CASCADE
- InnoDB gère l'intégrité référentielle de la base
- il gère le verrouillage des données au niveau de chaque enregistrement : l'écriture d'un nouvel enregistrement dans une table bloque seulement cet enregistrement pour les autres utilisateurs, pas les autres enregistrements de la table

MyISAM

- MyISAM est un moteur non transactionnel
- il est rapide en écriture
- il est très rapide en lecture
- il ne gère pas les relations ni les transactions
- il perd donc en sécurité
- il gère le verrouillage au niveau de la table entière, pas au niveau de chaque enregistrement
- il est peu gourmand en ressources
- il permet de créer des index sur les champs de type TEXT
  - efficace pour les recherches avancées dans les textes
  - beaucoup plus efficaces qu'avec l'instruction LIKE

Tableau comparatif

Critères	InnoDB	MyISAM
Transactions	Oui	Non
Relations	Oui	Non
Cascade	Oui	Non
Intégrité référentielle	Oui	Non
Verrouillage	Enregistrement	Table
Utilisation des ressources	Correcte	Peu gourmand
Vitesse	Correcte	Maximale
Gestion des textes	Correcte	Optimisée

Pistes d'optimisation

La structure

Les tables

- éviter la redondance d'informations : réaliser une conception de qualité
- limiter le nombre d'indexes : réagir aux retours des utilisateurs
- réorganisation des données
- réorganisation des indexes
- conséquences des réorganisations :
  - optimisation de l'espace utilisé par la base
  - influe sur le stockage permanent
  - influe sur les temps de sauvegarde et restauration

Pistes d'optimisation

Le code

Les requêtes SQL

- bien sélectionner les données à manipuler à ce qui est nécessaire (demandé par le client)
- optimiser les jointures : chaque jointure est coûteuse en ressources, il faut donc limiter les jointures en optimisant la conception de la base de données
- limiter :
  - les filtres (clauses WHERE, HAVING)
  - l'usage des GROUP BY
  - l'usage des ORDER BY
  - l'usage des sous-requêtes

Pistes d'optimisation

Le serveur MySQL

Les disques

Les disques

Liens symboliques

On peut déplacer des bases de données ou des tables vers d'autres emplacements et les remplacer par des liens symboliques vers ces nouveaux emplacements

Exemple :

- j'ai 2 disques durs, ou 1 disque dur avec 2 partitions
- ma base est implantée sur le 1er disque (ou partition) qui est presque plein(e)
- pour ne rien changer au logiciel ni au paramétrage du serveur MySQL qui s'adresse à la base sur le 1er disque :
  - je copie la base sur le 2ème disque qui a plus d'espace libre
  - je crée un lien symbolique sur le 1er disque qui permet d'accéder à la base qui est désormais sur le 2ème

Pistes d'optimisation

Le serveur MySQL

Les disques

Les disques

Lorsque l'on utilise plusieurs disques, notamment pour gérer plusieurs bases de données sur le même serveur, on peut segmenter

Striping (segmentation)

- si vous avez plusieurs disques, segmenter signifie répartir les données sur ces disques :
  - base par base si vous avez plusieurs bases sur le même serveur MySQL
  - table par table d'une seule base
  - enregistrement par enregistrement d'une seule table

Pistes d'optimisation

Le serveur MySQL

La mémoire

La mémoire

- MySQL alloue des buffers et des caches pour améliorer les performances des opérations de base de données
- à noter : la configuration par défaut est conçue pour permettre à un serveur MySQL de démarrer sur une machine virtuelle disposant de seulement 512 Mo de RAM
- on peut améliorer les performances de MySQL en augmentant les valeurs de certaines variables système liées au cache et aux buffers
- tout dépend de l'utilisation de la mémoire par vos requêtes
- par exemple, le fichier de données d'une table est ouvert (et donc sollicite la mémoire) une fois par requête qui l'utilise
- le fichier d'index est aussi ouvert une fois par requête (donc encore une fois, attention au nombre d'indexes)
- pour chaque requête, on sollicite un buffer égal à 3 fois taille d'un enregistrement de la table