

Tuning MySQL

innodb_file_per_table

Cette option permet d'avoir 1 fichier par table InnoDB au lieu d'un seul fichier

```
innodb_file_per_table=ON
```

innodb_stats_on_metadata

Lorsque l'option est définie sur ON, les statistiques d'index InnoDB sont mises à jour lors de l'exécution de `SHOW TABLE STATUS`, de `SHOW INDEX` ou de la requête `d'INFORMATION_SCHEMA.TABLES` ou `d'INFORMATION_SCHEMA.STATISTICS`. Ces statistiques incluent la cardinalité et le nombre d'entrées. Elles sont utilisées par l'optimiseur pour trouver un plan d'exécution optimal.

Ainsi, même si les instructions `SELECT` ne peuvent pas modifier les statistiques réelles, MySQL met à jour les statistiques des tables InnoDB. C'est contre-intuitif.

Est-ce utile? Pas vraiment, car InnoDB calculera toujours des statistiques lorsque vous ouvrez une table pour la première fois et lorsque des parties importantes de la table ont été modifiées (et lorsque vous exécutez `ANALYZE TABLE`).

Maintenant, pourquoi avons-nous eu une charge de lecture aussi élevée lorsque `innodb_stats_on_metadata` a été défini sur ON? Pour InnoDB, les statistiques sont estimées à partir de plongées à index aléatoire, ce qui se traduit par des lectures aléatoires.

innodb_buffer_pool_instances

Diviser le pool mémoire en plusieurs "régions" peut permettre d'améliorer l'accès aux données de façon concurrente, de diminuer les conflits issus des threads lors d'opérations de lecture ou d'écriture sur des pages verrouillées.

Chaque pool utilise alors sa propre liste d'accès aux données, sa liste de "flush", son propre mécanisme de LRU et surtout son propre buffer "Mutex".

Tous ces éléments rendent chaque région indépendante et permettent de diminuer les phénomènes de contention.

Ce paramètre n'est efficace que si la définition mémoire du `innodb_buffer_pool_size` est supérieure à 1 Go.

innodb_change_buffer_max_size

La variable `innodb_change_buffer_max_size` permet de configurer la taille maximale de la mémoire tampon de changement sous forme de pourcentage de la taille totale du pool de mémoire tampon. Par défaut, `innodb_change_buffer_max_size` est défini sur 25. Le paramètre maximal est 50.

Pensez à augmenter `innodb_change_buffer_max_size` sur un serveur MySQL avec une activité importante d'insertion, de mise à jour et de suppression, où la fusion de tampons de modification ne suit pas le rythme des nouvelles entrées de tampon de modifications, ce qui lui permet d'atteindre sa taille maximale.

Envisagez de réduire `innodb_change_buffer_max_size` sur un serveur MySQL avec des données statiques utilisées pour la génération de rapports, ou si la mémoire tampon de modification utilise trop d'espace mémoire partagé avec le pool de mémoire tampon, ce qui entraîne la disparition plus rapide des pages du pool de mémoire tampon.

query_cache_type

Sert de cache au requete coté serveur. A desactiver, les performances sont mauvaise.

Sinon mettre le `query_cache_type` à 2

```
query_cache_type = 1  
query_cache_size = 256M
```

`query_cache_type` est le type de cache que l'on va adopter:

0 = pas de cache

1 = met en cache toutes les requetes sauf celles qui ont le flag "SELECT S_NO_CACHE"

2 = met en cache seulement les requetes qui comportent le flag "SELECT SQL_CACHE"

innodb_flush_method

La bonne valeur pour éviter un double buffering est O_DIRECT

Revision #2

Created 23 November 2019 14:55:04 by Admin

Updated 26 November 2019 19:13:15 by Admin