

Étude de plusieurs moteurs de stockage sous MySQL

Par [Jérémy BEDDES](#) • Publié le 31/07/2015 à 12:04:18 • Noter cet article: ★★★★★ (16 votes)
Avis favorable du comité de lecture



Jérémy BEDDES

Tout d'abord, définissons ce qu'est un moteur de stockage. Un moteur de stockage est constitué d'algorithmes qui seront utilisés par le Système de Gestion de Base de Données Relationnelles pour enregistrer les données et les récupérer au moyen de requêtes SQL qui seront exécutées sur le système, elles-mêmes intégrées dans des transactions. Il faut savoir que MySQL propose plusieurs moteurs de stockages, ce qui diverge d'un grand nombre de concurrents qui n'en proposent qu'un seul. Ainsi, dans MySQL, les administrateurs système et les développeurs ont la possibilité de faire le choix du moteur qui sera le plus performant selon eux pour une action donnée.

Voici les principaux moteurs de stockage actuels de MySQL :

- MyISAM
- InnoDB
- MERGE
- MEMORY
- BLACKHOLE
- ARCHIVE
- FEDERATED



- [MyISAM](#)
- [InnoDB](#)
- [MERGE](#)
- [MEMORY](#)
- [BLACKHOLE](#)
- [ARCHIVE](#)
- [FEDERATED](#)
- [Conclusion](#)

Bien sûr, il en existe une multitude d'autres, notamment car il est possible de créer son propre moteur de stockage. Cependant ceux qui ont été cités précédemment restent les plus utilisés et répondent à la majorité des besoins. Il existe deux autres moteurs qui ont une grande importance :

- **EXAMPLE** : moteur non utilisable en l'état mais permettant aux informaticiens qui veulent créer leur propre moteur de stockage d'avoir un exemple sur lequel ils peuvent s'appuyer
- **NDBCluster** : moteur de stockage spécifique à la version clusterisée de MySQL. Nous aborderons ce moteur dans la dernière partie de ce livre consacrée à la mise en place d'un cluster

A tout moment, il est possible de récupérer la liste des moteurs de stockage disponibles. Pour cela, il faut exécuter la commande suivante dans MySQL lorsque que vous êtes connectés :

```
mysql> SHOW ENGINES;
+-----+-----+-----+-----+-----+-----+
| Engine | Support | Comment | Transactions | XA | Savepoints |
+-----+-----+-----+-----+-----+-----+
| FEDERATED | NO | Federated MySQL storage engine | NULL | NULL | NULL |
| CSV | YES | CSV storage engine | NO | NO | NO |
| MyISAM | YES | MyISAM storage engine | NO | NO | NO |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it disappears) | NO | NO | NO |
| MRG_MYISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| PERFORMANCE_SCHEMA | YES | Performance Schema | NO | NO | NO |
| ARCHIVE | YES | Archive storage engine | NO | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables | NO | NO | NO |
| InnoDB | DEFAULT | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

qu'InnoDB est le moteur par défaut à partir de MySQL 5.5 et que MyISAM l'est pour les versions inférieures.

Le moteur de stockage s'applique à une table lors de sa création. Prenons la création de table suivante, avec spécification :

```
CREATE TABLE utilisateur(  
  id          Int(11) Auto_Increment NOT NULL ,  
  username    Varchar(50) NOT NULL ,  
  password     Varchar(255) NOT NULL ,  
  PRIMARY KEY (id)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

On remarque bien qu'à la dernière ligne, le moteur de stockage est précisé. Il faut aussi savoir qu'il est possible de changer le moteur de stockage par défaut. Par exemple dans le cas de notre table utilisateur, la requête à utiliser est :

```
ALTER TABLE utilisateur ENGINE=MyISAM;
```

Nous allons maintenant étudier les familles de moteurs. Il existe deux familles :

- Les moteurs transactionnels
- Les moteurs non-transactionnels

Souvenez-vous de la commande « SHOW ENGINES ; » : nous y avons constaté qu'InnoDB était le seul moteur de stockage transactionnel. Tout d'abord, il faut savoir qu'un moteur transactionnel est sécurisant, c'est-à-dire qu'il va vérifier que l'opération sur la base de données a été effectuée, et ce, du début jusqu'à la fin. Si un problème était survenu, l'opération aurait simplement été annulée. Un mécanisme de validation et de récupération complexe est en place, ainsi les données d'une transaction envoyée durant une coupure électrique pourront être récupérées.

Ensuite, les moteurs non-transactionnels ont eux aussi leurs avantages. Ce sont des avantages en termes de ressources matérielles. En effet, moins de mémoire est consommée pour la même opération, moins d'espace disque. Notez aussi qu'ils sont de ce fait plus rapides que les moteurs transactionnels. Ce qui est tout à fait normal compte tenu du fait que les transactions ne sont pas vérifiées de la même manière.

Nous allons maintenant faire un point sur chacun des moteurs cités au début de ce chapitre.

MyISAM

MyISAM est, comme nous l'avons dit précédemment, le moteur par défaut dans les versions de MySQL inférieures à 5.5 et ce, depuis la version 3.23. MyISAM est un moteur non-transactionnel ce qui explique qu'il est rapide (surtout en lecture). Il faut noter qu'il ne gère ni les relations, ni les transactions. C'est un moteur qui perd énormément en intégrité à cause des points précédents, en revanche il gagne de la vitesse.

Ensuite, il faut savoir qu'une table utilisant MyISAM est verrouillée en lecture et en écriture si un utilisateur est en train d'écrire des données. Ainsi, il faudra attendre que l'écriture soit terminée pour pouvoir écrire de nouveau.

MyISAM est également un moteur gérant l'indexation des contenus, et est le seul qui permet de créer des index de type FULLTEXT sur des champs de type TEXT. Cela a pour but de rendre une recherche plus rapide et efficace que nos recherches habituelles avec « LIKE ». Ainsi si vous souhaitez rechercher des données, c'est un moteur à privilégier.

Voici un exemple d'utilisation de celui-ci :

```
CREATE TABLE livre(  
  id          BIGINT NOT NULL PRIMARY KEY Auto_Increment,  
  titre_chapitre Varchar(200),  
  texte       TEXT,  
  FULLTEXT(titre_chapitre, texte)  
)ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

InnoDB

InnoDB quant à lui est un moteur transactionnel. Ainsi l'utilisateur est certain que les relations entre les tables sont cohérentes et que les changements effectués sur les données contenues dans ces tables seront transmis aux tables qui y sont liées.

Il faut noter qu'InnoDB est moins rapide que MyISAM (même si cet écart de vitesse a été réduit lors de la version 5.5 de MySQL). Cependant, il offre une bonne sécurité en ce qui concerne la validité des données et est donc un moteur fortement conseillé pour stocker des données relationnelles ou nécessitant des transactions classiques.

Voici un exemple d'utilisation d'InnoDB :

```
CREATE TABLE ordinateur(  
  id_ordinateur BIGINT NOT NULL PRIMARY KEY Auto_Increment,  
  nom_ordinateur Varchar(255) NOT NULL  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE utilisateur(  
  id_utilisateur BIGINT NOT NULL PRIMARY KEY Auto_Increment,  
  username        Varchar(50) NOT NULL,  
  password        Varchar(255) NOT NULL,
```

MERGE est un moteur de stockage qui regroupe plusieurs tables MyISAM de façon totalement transparente par le biais d'une fusion virtuelle. Grâce à cela, il offre une interface unique pour accéder à toutes les tables en même temps, réparties physiquement sur plusieurs bases de données, en lecture. Pour l'écriture, l'administrateur de la base de données aura la possibilité de fixer des règles pour accéder en simultané à une ou plusieurs tables ; dans tous les cas ce sont ces dernières qui seront modifiées.

Contrairement à MyISAM, MERGE ne prend pas en charge les index FULLTEXT.

Pour fusionner plusieurs tables avec MERGE, il faut respecter quelques points. Tout d'abord, les tables doivent être stockées localement sur le même serveur, avoir des noms et des types de champs identiques, avoir les mêmes index déclarés dans le même ordre.

Ainsi, pour utiliser MERGE pour fusionner deux tables, il faudra insérer la ligne suivante lors de la création de la table :

```
ENGINE=MERGE UNION(bdd1.table1, bdd2.table2) INSERT_METHOD=LAST;
```

Pour finir, MERGE a d'autres défauts : il est très consommateur en ressources matérielles, les recherches sur les tables sont plus lentes que sur des tables MyISAM et il faut faire très attention aux structures et aux déclarations des index des tables fusionnées.

MEMORY

MEMORY est un moteur de stockage assez différent des trois premiers que nous venons d'étudier. En effet, contrairement à MyISAM, InnoDB et MERGE qui stockent les données sur le disque dur de la machine, MEMORY les stocke dans la mémoire vive. Cela en fait le moteur de stockage le plus rapide de MySQL. Cependant, c'est aussi le plus dangereux puisque les données seront effacées de la mémoire vive après un simple redémarrage ou la moindre panne de la machine.

Il faut savoir que MEMORY ne permet pas de gérer les relations et les transactions puisque c'est un moteur non-transactionnel. Ensuite, les champs BLOB et TEXT ne sont pas pris en compte puisqu'ils prennent trop de place et pourraient influencer sur la rapidité de traitement des données stockées en mémoire vive.

Ainsi, l'utilisation de MEMORY est idéale pour des données temporaires qui nécessitent un traitement le plus rapide possible et qui peuvent être aisément reconstruites en cas d'incidents.

Prenons un cas concret d'utilisation. Lorsque vous commandez un produit sur un site de vente en ligne, vous devez vous inscrire. Pour finaliser votre inscription, un courriel vous parvient et permet grâce à une URL particulière de valider votre inscription. Cette URL est donc conservée en mémoire dans la base de données et lorsque vous cliquez sur le lien pour y accéder, votre inscription se termine et va créer votre compte dans une autre table, classique en disque cette fois, du site web en question.

Les informations temporaires d'inscription n'ont pas besoin d'être stockées sur le disque dur. Il est très simple par ailleurs de réinitialiser une procédure d'inscription.

Voici donc un exemple d'utilisation de MEMORY :

```
CREATE TABLE utilisateur(  
  id BIGINT NOT NULL PRIMARY KEY Auto_Increment,  
  username Varchar(255) NOT NULL,  
  password Varchar(255) NOT NULL,  
  email_code Varchar(10) NOT NULL  
)ENGINE=MEMORY DEFAULT CHARSET=utf8;
```

BLACKHOLE

« Black Hole » signifie « Trou noir » en français. C'est un moteur de stockage qui porte bien son nom. Toutes les données qui vont être insérées dans ce moteur ne seront pas enregistrées sur le disque dur de la machine et seront détruites.

Cependant, comme pour toutes les tables, MySQL journalisera l'ensemble des actions qui seront effectuées. Un chapitre plus précis de ce livre est consacré aux journaux MySQL, vous y trouverez de plus amples informations.

Un administrateur de base de données pourra donc utiliser ce moteur comme un moteur de test pour simuler des actions sur des tables réelles et déceler des éventuels problèmes.

Voici un exemple de table utilisant BLACKHOLE :

```
CREATE TABLE utilisateur(  
  id BIGINT NOT NULL PRIMARY KEY Auto_Increment,  
  username Varchar(255) NOT NULL,  
  password Varchar(255) NOT NULL  
)ENGINE=BLACKHOLE DEFAULT CHARSET=utf8;
```

ARCHIVE

ARCHIVE est un moteur de stockage non-transactionnel permettant de stocker de très grosses quantités de données en les compressant lors de leur insertion et en utilisant le moins de place possible sur le périphérique de stockage. Avec ce type de moteur, il n'est pas

FEDERATED quant à lui est un moteur qui permet de créer une table dont les données ne sont pas stockées sur le serveur MySQL local. Il s'agit en fait d'un moteur de stockage permettant l'hébergement des données sur un serveur MySQL distant. Les données de ce serveur distant sont reconstruites dans la table locale à l'aide d'une requête SELECT sur la table distante.

C'est un moteur contraignant. En effet, il n'est pas démarré par défaut sous MySQL. Pour pouvoir l'utiliser, le serveur MySQL devra être démarré avec le paramètre « --federated ».

Avec ce moteur, le traitement des requêtes est assez long et il ne sera pas possible de modifier la table locale. Si une modification au niveau de la table doit être apportée, il faudra qu'elle le soit au niveau de la table distante.

En outre, il faudra bien entendu que les deux serveurs MySQL puissent communiquer ensemble. Il faudra donc veiller à autoriser les connexions distantes sur le port 3306 (port par défaut).

Voici un exemple d'utilisation de FEDERATED :

```
# TABLE MyISAM sur le premier serveur
CREATE TABLE utilisateur(
  id          Int(20) NOT NULL Auto_Increment,
  username    Varchar(255) NOT NULL,
  password    Varchar(255) NOT NULL,
  PRIMARY KEY (id),
  KEY username (username),
  KEY password_key (password)
)ENGINE=MyISAM DEFAULT CHARSET=utf8;

# TABLE FEDERATED sur le second serveur
CREATE TABLE federated_utilisateur(
  id          Int(20) NOT NULL Auto_Increment,
  username    Varchar(255) NOT NULL,
  password    Varchar(255) NOT NULL,
  PRIMARY KEY (id),
  KEY username (username),
  KEY password_key (password)
)
ENGINE=FEDERATED
COMMENT='mysql://user@mysql.domain.com:9306/federated/utilisateur'
DEFAULT CHARSET=utf8;
```

Conclusion

Nous venons d'étudier les principaux moteurs de stockages disponibles sous MySQL. Vous devez certainement vous demander quel moteur est à privilégier. La réponse est simple : cela dépend du cas dans lequel vous vous trouvez...

Par exemple, si vous avez besoin d'une sécurité maximale au niveau des données et que vos tables ont des contraintes d'intégrité référentielles entre-elles, privilégiez InnoDB. Si au contraire vous avez besoin de performance et de rapidité et que vos données n'ont pas besoin d'être stockées sur le disque dur, utilisez MEMORY. Globalement, la majorité des besoins trouveront une réponse avec MyISAM et InnoDB.

Chaque cas est unique et c'est maintenant à vous de choisir le moteur qui sera le plus pertinent à utiliser.

A propos de SUPINFO | Contacts & adresses | Enseigner à SUPINFO | Presse | Conditions d'utilisation & Copyright | Respect de la vie privée | Investir



SUPINFO International University
 Ecole d'Informatique - IT School
 École Supérieure d'Informatique de Paris, leader en France
 La Grande École de l'informatique, du numérique et du management
 Fondée en 1965, reconnue par l'État. Titre Bac+5 certifié au niveau I.
 SUPINFO International University is globally operated by EDUCINVEST Belgium - Avenue Louise, 534 - 1050 Brussels

Contact @SUPINFO

ACCUEIL CURSUS COURS ADMISSIONS SUPINFO International University is globally operated by EDUCINVEST Belgium - Avenue Louise, 534 - 1050 Brussels

Naviguer sur la page