Replication

Nous allons mettre en place une replication asynchrone.

Variables

innodb_flush_log_at_trx_commit

innodb_flush_log_at_trx_commit contrôle l'équilibre entre la conformité ACID stricte pour les opérations de validation et les performances supérieures possibles lorsque les opérations d'E / S associées à la validation sont réorganisées et effectuées par lots. Vous pouvez obtenir de meilleures performances en modifiant la valeur par défaut, mais vous pouvez ensuite perdre des transactions en cas de blocage. Le paramètre par défaut de 1 est requis pour la conformité ACID complète. Les journaux sont écrits et vidés sur le disque à chaque validation de transaction. Avec un réglage de 0, les journaux sont écrits et vidés sur le disque une fois par seconde. Les transactions pour lesquelles les journaux n'ont pas été vidés peuvent être perdues dans un crash. Avec un réglage de 2, les journaux sont écrits après chaque validation de transaction et vides les transactions sur le disque une fois par seconde. Les transactions pour lesquelles les journaux n'ont pas été vidés peuvent être perdues dans un crash. Pour les réglages 0 et 2, la propriété ACID n'est pas garantie

sync_binlog

sync_binlog contrôle la fréquence à laquelle le serveur MySQL synchronise le journal binaire sur le disque.

- sync_binlog = 0: désactive la synchronisation du journal binaire sur le disque par le serveur MySQL. À la place, le serveur MySQL s’appuie sur le système d’exploitation pour vider le journal binaire sur le disque de temps en temps, comme il le fait pour tout autre fichier. Ce paramètre offre les meilleures performances, mais en cas de panne d'alimentation ou de panne du système d'exploitation, il est possible que le serveur ait validé des transactions qui n'ont pas été synchronisées avec le journal binaire.
- sync_binlog = 1: active la synchronisation du journal binaire sur le disque avant que les transactions ne soient validées. Ce paramètre est le plus sûr, mais peut avoir un impact négatif sur les performances en raison du nombre accru d'écritures sur disque. En cas de panne d'alimentation ou de panne du système d'exploitation, les transactions manquantes dans le journal binaire sont uniquement dans un état préparé. Cela permet à

la routine de récupération automatique d'annuler les transactions, ce qui garantit qu'aucune transaction n'est perdue à partir du journal binaire.
- sync_binlog = N, où N est une valeur autre que 0 ou 1: le journal binaire est synchronisé sur le disque après la collecte de N groupes de validation de journal binaire. En cas de panne d'électricité ou de panne du système d'exploitation, il est possible que le serveur ait validé des transactions qui n'ont pas été vidées dans le journal binaire. Ce paramètre peut avoir un impact négatif sur les performances en raison du nombre accru d'écritures sur disque. Une valeur plus élevée améliore les performances, mais avec un risque accru de perte de données.

binlog-format

binlog-format contrôle la facon dont le format de log par statement, par données ou les deux.(ROW,STATEMENT,MIXED)

- Lors de l'utilisation de la journalisation binaire basée sur des instructions, le maître écrit des instructions SQL dans le journal binaire. La réplication du maître sur l'esclave fonctionne en exécutant les instructions SQL sur l'esclave. C'est ce qu'on appelle la réplication basée sur les instructions (qui peut être abrégée en SBR), ce qui correspond au format de journalisation binaire basé sur les instructions MySQL.

- Lors de l'utilisation de la journalisation basée sur les lignes, le maître écrit dans le journal binaire des événements indiquant comment les rangées individuelles d'une table sont modifiées. La réplication du maître sur l'esclave fonctionne en copiant les événements représentant les modifications apportées aux lignes du tableau vers l'esclave. C'est ce qu'on appelle la réplication basée sur les lignes (qui peut être abrégée en RBR).

log_slave_updates

Normalement, un esclave n'écrit aucune mise à jour reçue d'un serveur maître dans son propre journal binaire. Cette option amène l'esclave à écrire les mises à jour effectuées par son thread SQL dans son propre journal binaire.

master-info-file

Nom à utiliser pour le fichier dans lequel l’esclave enregistre des informations sur le maître

- La journalisation basée sur les lignes est la méthode par défaut.

- Vous pouvez également configurer MySQL pour utiliser une combinaison de consignation basée sur les instructions et basée sur les lignes, en fonction de l'option la plus appropriée pour la journalisation des modifications. C'est ce qu'on appelle la journalisation mixte. Lorsque vous utilisez une journalisation mixte, un journal basé sur des instructions est

utilisé par défaut. En fonction de certaines instructions, ainsi que du moteur de stockage utilisé, le journal est automatiquement basculé en cas de modification basée sur les lignes.

Configuration replication asynchrone

Dans cette replication, mysqld1 va servir de maître et mysqld2 seras un esclave. Dans le groupe mysqld1, nous allons associé un id de serveur ainsi que les bonne valeurs de réplication

Configuration du master:

```
[mysqld]
port = 3306
server-id               = 1
socket = /tmp/mysql.sock1
skip-external-locking
key_buffer_size = 16K
max_allowed_packet = 1M
table_open_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
net_buffer_length = 2K
thread_stack = 128K
table_open_cache=500
datadir = /home/pilou/Formation/replication/master-slave/master/data
pid-file = /home/pilou/Formation/replication/master-slave/master/mysqlmaster.pid
log-error=/home/pilou/Formation/replication/master-slave/master/logerror.err
expire_logs_days = 10
max_binlog_size = 100M
innodb_flush_log_at_trx_commit = 1
sync_binlog = 1
binlog-format = ROW
```

et dans le serveur esclave (en readonly) nous associons <source lang='ini'>

```
datadir = /home/pilou/mysql80/datadir/mysql2
pid-file = /home/pilou/mysql80/datadir/mysql2/mysql2.pid
server-id = 2
log_bin = /home/pilou/mysql80/datadir/mysql2/mysql-bin.log
log_error = /home/pilou/mysql80/datadir/mysql2/error_slave.log
relay-log = /home/pilou/mysql80/datadir/mysql2/relay-bin
relay-log-index = /home/pilou/mysql80/datadir/mysql2/relay-bin.index
master-info-file = /home/pilou/mysql80/datadir/mysql2/master.info
relay-log-info-file = /home/pilou/mysql80/datadir/mysql2/relay-log.info
read_only = 1
```

Création d'un utilisateur de replication.

Il est nécéssaire d'avoir une connectivité du slave vers le master pour faire la replication.

Sur le master faire:

```
CREATE USER 'slave'@'%'  WITH mysql_native_password IDENTIFIED  BY 'password';
GRANT REPLICATION SLAVE ON *.* TO 'slave'@'%';
```

Puis regarder la position dans les journaux

```
mysql> SHOW MASTER STATUS;
+---------------+----------+--------------+------------------+-------------------+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+---------------+----------+--------------+------------------+-------------------+
| binlog.000004 |      690 |              |                  |                   |
+---------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)
```

Une autre solution est de faire un rattrapage avec position des journaux:

Nous allons faire un dump de la base du master via mysqldump

```
 ./bin/mysqldump -u root -p --host=127.0.0.1 --port=3306 --all-databases --master-data=2 >
replicationdump.sql
```

L'utilisation du flag master-data permet d'inclure les information du master, et en particulier la position des log dans le fichier de log

```
- MySQL dump 10.13 Distrib 8.0.13, for linux-glibc2.12 (x86_64) --

-- Host: 127.0.0.1 Database: -- ------------------------------------------------------

-- Server version 8.0.13

/*!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 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!50606 SET @OLD_INNODB_STATS_AUTO_RECALC=@@INNODB_STATS_AUTO_RECALC */;

/*!50606 SET GLOBAL INNODB_STATS_AUTO_RECALC=OFF */;

/*!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 */;

-- -- Position to start replication or point-in-time recovery from --

-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000004', MASTER_LOG_POS=690;

-- -- Current Database: `mysql` --
```

Sur l'esclave, connecter reinjecter les données:

mysql -uroot -p --host=127.0.0.1 --port=3307 < replicationdump.sql

Puis connecter vous sur l'esclave CHANGE MASTER TO MASTER_HOST='127.0.0.1', MASTER_USER='slave', MASTER_PASSWORD='password', MASTER_LOG_FILE='binlog.000004', MASTER_LOG_POS=973;

Mise en place de la replication

Faire un start slave et vérifier le statut de l'esclave

```
mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 127.0.0.1
                  Master_User: slave
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: binlog.000004
          Read_Master_Log_Pos: 973
               Relay_Log_File: relay-bin.000002
                Relay_Log_Pos: 321
        Relay_Master_Log_File: binlog.000004
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 973
              Relay_Log_Space: 524
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
```

```
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
                  Master_UUID: 631f7e58-9421-11ea-b477-080027193107
             Master_Info_File: mysql.slave_master_info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
           Master_Retry_Count: 86400
                  Master_Bind:
      Last_IO_Error_Timestamp:
     Last_SQL_Error_Timestamp:
               Master_SSL_Crl:
           Master_SSL_Crlpath:
           Retrieved_Gtid_Set:
            Executed_Gtid_Set:
                Auto_Position: 0
         Replicate_Rewrite_DB:
                 Channel_Name:
           Master_TLS_Version:
       Master_public_key_path:
        Get_master_public_key: 0
            Network_Namespace:
1 row in set (0.00 sec)
```

Configuration replication semi-synchrone

Sur le master il faut:

- Activer le plugin de replication semi synchrone INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';
- Modifier le fichier de configuration en rajoutant rpl_semi_sync_master_enabled = 1

Une fois le redemarrage du serveur on observe:

```
mysql> show status like "rpl_semi_sync%";
+--------------------------------------------+-------+
| Variable_name                              | Value |
+--------------------------------------------+-------+
| Rpl_semi_sync_master_clients               | 0     |
| Rpl_semi_sync_master_net_avg_wait_time     | 0     |
| Rpl_semi_sync_master_net_wait_time         | 0     |
| Rpl_semi_sync_master_net_waits             | 0     |
| Rpl_semi_sync_master_no_times              | 0     |
| Rpl_semi_sync_master_no_tx                 | 0     |
| Rpl_semi_sync_master_status                | ON    |
| Rpl_semi_sync_master_timefunc_failures     | 0     |
| Rpl_semi_sync_master_tx_avg_wait_time      | 0     |
| Rpl_semi_sync_master_tx_wait_time          | 0     |
| Rpl_semi_sync_master_tx_waits              | 0     |
| Rpl_semi_sync_master_wait_pos_backtraverse | 0     |
| Rpl_semi_sync_master_wait_sessions         | 0     |
| Rpl_semi_sync_master_yes_tx                | 0     |
+--------------------------------------------+-------+
14 rows in set (0.00 sec)
```

La replication est pour l'instant asynchrone car aucun client semi synchrone n'est connecté

```
ysql>  SHOW VARIABLES LIKE "rpl_semi_sync_master_timeout";
+-------------------------------+-------+
| Variable_name                 | Value |
+-------------------------------+-------+
| rpl_semi_sync_master_timeout | 10000 |
+-------------------------------+-------+
1 row in set (0.02 sec)
```

Sur le slave

Installer le plugin tel que sur le master mais dans sa version salve

```
INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';
```

Puis dans le fichier my.ini

```
rpl_semi_sync_slave_enabled = 1
```

Enfin il faut vérifier le statut de la replication:

```
mysql> SHOW VARIABLES LIKE "Rpl_semi_sync_slave_%";
+---------------------------------+-------+
| Variable_name                   | Value |
+---------------------------------+-------+
| rpl_semi_sync_slave_enabled     | ON    |
| rpl_semi_sync_slave_trace_level | 32    |
+---------------------------------+-------+
2 rows in set (0.01 sec)
```

Sur le master, il est maintenant indiqué qu'il y a un client:

```
mysql> show status like "rpl_semi_sync%";
+--------------------------------------------+-------+
| Variable_name                              | Value |
+--------------------------------------------+-------+
| Rpl_semi_sync_master_clients               | 1     |
| Rpl_semi_sync_master_net_avg_wait_time     | 0     |
| Rpl_semi_sync_master_net_wait_time         | 0     |
| Rpl_semi_sync_master_net_waits             | 0     |
| Rpl_semi_sync_master_no_times              | 0     |
| Rpl_semi_sync_master_no_tx                 | 0     |
| Rpl_semi_sync_master_status                | ON    |
| Rpl_semi_sync_master_timefunc_failures     | 0     |
| Rpl_semi_sync_master_tx_avg_wait_time      | 0     |
| Rpl_semi_sync_master_tx_wait_time          | 0     |
| Rpl_semi_sync_master_tx_waits              | 0     |
| Rpl_semi_sync_master_wait_pos_backtraverse | 0     |
| Rpl_semi_sync_master_wait_sessions         | 0     |
| Rpl_semi_sync_master_yes_tx                | 0     |
+--------------------------------------------+-------+
14 rows in set (0.01 sec)
```

Configuration GTID

Sur le master

```
server-id = 1
log-bin = mysql-bin
binlog_format = row
gtid-mode=ON
enforce-gtid-consistency
log-slave-updates
```

Sur le slave

```
server-id = 2
log-bin = mysql-bin
relay-log = relay-log-server
relay-log = relay-log-server
read-only = ON
gtid-mode=ON
enforce-gtid-consistency
log-slave-updates
```

Puis

```
CHANGE MASTER TO
MASTER_HOST = '54.89.xx.xx',
MASTER_PORT = 3306,
MASTER_USER = 'repl_user',
MASTER_PASSWORD = 'XXXXXXXXX',
MASTER_AUTO_POSITION = 1;
```

Revision #7
Created 11 May 2020 18:23:24 by ggpilou2
Updated 26 November 2021 07:12:06 by ggpilou2