

## Отчет о проведении тестирования.

Время проведения: 3.06.2020

### 1. Цель теста

Бизнес-цель теста – *получение навыков работы с репликацией БД.*

Техническая цель теста – оценка производительности с использованием *wrk* и *СУБД MySQL 5.7.30* на кластере *Kubernetes 1.14.10-gke.36*.

*В испытании участвовало 3 операций.*

*Таблица 1. Модель нагрузки.*

<i>1</i>	<i>Настройка репликации БД MySQL.</i>
<i>2</i>	<i>Нагрузка на запись в БД MySQL. На стороне, которой нагружаем считаем, сколько строк мы успешно записали.</i>
<i>3</i>	<i>После падения master. Выбираем slave и переключаем на него второй слейв.</i>

### 2. Настройки, создание кластера master/slave.

1. Для настройки репликации *master/slave* использовались возможности кластера *Kubernetes*.
2. Развертывание и клонирование БД осуществлялось контейнерами *init-mysql* и *clone-mysql* с использованием программного продукта *xtrabackup*.
3. Настройка требуемых параметров (row-based, GTID) репликации производилось в конфигурационной единице *ConfigMap*:

```
log-bin
log-slave-updates = true
gtid-mode = on
enforce-gtid-consistency = true
binlog_format = ROW
```

5. Настройка полусинхронной репликации и *slave* серверов производилась с помощью вспомогательного контейнера *xtrabackup* и конфигурационного скрипта:

```
[[ `hostname` =~ -([0-9]+)$ ]] || exit 1
ordinal=${BASH_REMATCH[1]}
# Copy appropriate conf.d files from config-map to emptyDir.
if [[ $ordinal -eq 0 ]]; then
    echo "Waiting for mysqld to be ready (accepting connections)"
    until mysql -h 127.0.0.1 -u root -p$MYSQL_ROOT_PASSWORD -e "SELECT 1"; do sleep 1; done
    echo "Configure semi-sync replication"
    if ! mysql -h 127.0.0.1 \
        -u root -p$MYSQL_ROOT_PASSWORD \
        -e "INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';
            SET GLOBAL rpl_semi_sync_master_enabled = 1;
            SET GLOBAL rpl_semi_sync_master_timeout = 10000;
            INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';" ; then
        echo "Error code: $? of SQL: SET GLOBAL rpl_semi_sync_master_enabled = 1;"
    else
        echo "Query OK"
    fi
else
    echo "Waiting for mysqld to be ready (accepting connections)"
    until mysql -h 127.0.0.1 -u root -p$MYSQL_ROOT_PASSWORD -e "SELECT 1"; do sleep 1; done
    echo "Configure semi-sync replication"
    if ! mysql -h 127.0.0.1 \
        -u root -p$MYSQL_ROOT_PASSWORD \
        -e "INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';"
    then
        echo "error code: $? of sql: INSTALL PLUGIN rpl_semi_sync_slave"
    else
        echo "query ok"
    fi
    echo "Enable semi-sync replication"
    if ! mysql -h 127.0.0.1 \
        -u root -p$MYSQL_ROOT_PASSWORD \
        -e "SET GLOBAL rpl_semi_sync_slave_enabled = 1;
            STOP SLAVE IO_THREAD;
            START SLAVE IO_THREAD;"
    then
        echo "error code: $? of sql: SET GLOBAL rpl_semi_sync_slave_enabled = 1;"
    else
        echo "query ok"
    fi
fi
# Check if we need to complete a clone by starting replication.
if [[ -f change_master_to.sql.in ]]; then
    echo "Waiting for mysqld to be ready (accepting connections)"
    until mysql -h 127.0.0.1 -u root -p$MYSQL_ROOT_PASSWORD -e "SELECT 1"; do sleep 1; done
    echo "Initializing replication from clone position"
    mysql -h 127.0.0.1 \
        -u root -p$MYSQL_ROOT_PASSWORD \
        -e "STOP SLAVE; CHANGE MASTER TO \
            MASTER_HOST='mysql-0.mysql.default.svc.cluster.local', \
            MASTER_USER='root', \
            MASTER_PASSWORD='$MYSQL_ROOT_PASSWORD', \
            MASTER_CONNECT_RETRY=10, \
            MASTER_AUTO_POSITION = 1; \
            START SLAVE;" || exit 1
    # In case of container restart, attempt this at-most-once.
    mv change_master_to.sql.in change_master_to.sql.orig
fi
```

6. Файл описания *StatefulSet*-а: [mysql-statefulset.yaml](#),  
файл описания сервисов: [mysql-services.yaml](#),  
файл описания дополнительной конфигурации: [mysql-configmap.yaml](#).
7. Балансировка нагрузки на чтение, также осуществлялась при помощи кластера *Kubernetes*.  
С использованием сервиса *Kubernetes*. Файл описания [mysql-reads.loadBalancer.yaml](#).
8. При проведении нагрузочного тестирования наиболее ресурсоемкие запросы в одной транзакции:  

```
SELECT benchmark(
  (SELECT FLOOR(
    RAND((SELECT UNIX_TIMESTAMP(NOW()))
      * (4999999 - 99999 + 1) + 99999)),
    md5('when will it end?')
  )
  и
  INSERT INTO user_log (user_profile_id, date_time) VALUES (?, ?)
```
9. Вывод проверки настроек репликации в файлах [gdit.txt](#), [row-based.txt](#), [semi\\_sync.txt](#).

### 3. Выводы

1. Первичное тестирование: [44](#) - транзакции порядковые номера со стороны тестирования записывались в файл [ok.count.txt](#). Все записи в файле [ok.mysql.txt](#). Вывод созданных записей на клиента в файле [ok.wrk-2020-06-03\\_16\\_49\\_46.log.gz](#).
2. Тестирование с отключением master-сервера:  
[32](#) - транзакции порядковые номера со стороны тестирования записывались в файл [disaster.count.txt](#). Все записи в файле [disaster.mysql.txt](#). Вывод созданных записей на клиента в файле [disaster.wrk-2020-06-03\\_16\\_53\\_33.log.gz](#).
3. Тестирование после переключения записи на бывший slave:  
[32](#) - транзакции. Все записи в файле [change\\_master.mysql.txt](#). Вывод созданных записей на клиента в файле [change\\_master.wrk-2020-06-03\\_19\\_11\\_40.log.gz](#).
4. Команда переключения slave mysql-2 на бывший slave в файле [change\\_master.mysql.txt](#).
5. В результате сбоя потерь в транзакция не было [32](#) транзакции зафиксировано на тестирующей стороне, [32](#) записи было добавлено в БД.