数据库性能与故障恢复实验指导书

（基于 MySQL8，2022 学年编写）

【注意：以下操作的关键步骤，都需要抓屏，且要求定义的数据库对象包含有自己姓名首字母的字符串】

（1） 生成千万级数量的关键表记录

方案一：可参考 https://www.niepub.com/static/docs/exadmin/ngq_MySQL_sp_insertaccount_example.pdf 编写一个存储过程，可以增加任何多条记录到表中，比如：

【建议实验三在实验二的基础上进行，可以不再创建新表，这里创建此表只是为了方便快速演示】

```
CREATE TABLE ngqcolleges
(
   collegeid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   collegename varchar(50) NOT NULL,
   collegetype varchar(30),
   description varchar(255),
   create_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
   constraint ngqcolleges_unique unique(collegename)
) engine=InnoDB;

drop procedure insert_ngqcolleges;
delimiter $$
create procedure insert_ngqcolleges(p_rowcount bigint,p_tid int)
begin
      declare v_tid bigint;
      set v_tid = p_tid;
      set p_rowcount = p_rowcount + 1;
      while v_tid < p_rowcount do
          if (v_tid%10000 = 0) then
              commit;
          end if;
          insert into ngqcolleges (collegename) values ( concat('测试数据第',v_tid,'学院'));
          set v_tid = v_tid + 1;
      end while;
      commit;
```

```
end$$
delimiter ;
--call insert_ngqcolleges(1000,1);
--写入一千条记录到表 ngqcolleges
```

方案二，用 java 或 python 或 C#等编程语言编写代码，可以更加灵活地实现模拟数据记录的插入，并可以通过多线程或多进程技术同时实现下一步的并发操作测试

（2） 并发操作测试

方案一、采用 mysql 自带的 mysqlslap 作并发测试：

（a） 并发 insert：

【建议实验三在实验二的基础上进行，可以不再创建新表，这里创建此表只是为了方便快速演示】

```
CREATE TABLE ngqinsert
(
   id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   name varchar(50) NOT NULL,
   second int default 2400,
   score DECIMAL(5,2) default 1.5,
   memo1 varchar(10),
   memo2 varchar(20),
   memo3 varchar(30),
   memo4 varchar(40),
   memo5 varchar(50),
   updated_user varchar(30),
   create_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
   ) engine=InnoDB;
insert into ngqinsert (name) values ('北京 0');
select * from ngqinsert;
mysqlslap --no-defaults -uroot -p --create-schema='nietest' -q "insert into
ngqinsert (name) values ('北京 1')" -c?
--mysqlslap -udbuser -p --create-schema='ngqdb' -q "insert into ngqinsert (name)
values ('北京 1')" -c20
/*
    Average number of seconds to run all queries: 0.496 seconds
    Minimum number of seconds to run all queries: 0.496 seconds
    Maximum number of seconds to run all queries: 0.496 seconds
    Number of clients running queries: 20
*/
```

--mysqlslap -udbuser -p --create-schema='ngqdb' -q "insert into ngqinsert (name) values ('北京 1')" -c2000
/*
Benchmark
Average number of seconds to run all queries: 0.990 seconds
Minimum number of seconds to run all queries: 0.990 seconds
Maximum number of seconds to run all queries: 0.990 seconds
Number of clients running queries: 2000
Average number of queries per client: 1
有时会报错：mysqlslap: Error when connecting to server: 1040 Too many connections 多种因素都要考虑如：
Check more /etc/my.cnf
max_connections=3000
重启数据库试试
【mysql8：缺省 3000 并发可行】
还有问题，就需要考虑其它，比如文件打开数的限制
#ulimit -n 1024→65535
Check more /etc/security/limits.conf
root soft nofile 65535
root hard nofile 65535
* soft nofile 65535
* hard nofile 65535
[root@techdb pam.d]# more /etc/pam.d/login add
session required pam_limits.so
sysctl -p 或者 重新登录再试
*/

show full processlist ;

（b）由并发 update 导致的死锁实验：
SELECT * FROM information_schema.INNODB_TRX;
select trx_mysql_thread_id,p.* from information_schema.innodb_trx p;
--是否有正在锁定的事务线程，看看 ID 是否在 show full processlist 里面的 sleep 线程中，如果是，就证明这个 sleep 的线程事务一直没有 commit 或者 rollback 而是卡住了，我们需要手动 kill 掉。
--mysql>kill trx_mysql_thread_id
select @@autocommit;
--set global autocommit=0;
begin; update ngqinsert set score=1.6 where name = '北京 1';commit;
begin;
update ngqinsert set score = 1.8 where id = 73;
commit;

方案二、用 java 或 python 或 C#等的多线程技术编写代码实现并发操作测试。

（3） SQL 优化实验

在关键表的记录数达到千万级别后，重新运行实验二的 SQL 脚本分析出银行的优质客户并对客户标星（1-5 星），并对其相关 SQL 做 explain，分析其解释计划，并通过添加索引等方式观察是否起到优化的作用。

```
explain format = json | tree
select courseid, studentno from students_score where studentno like '%319%';
```

（4） 数据库备份恢复或主从架构故障切换实验

方案一、对自己的数据库做备份，并做恢复实验

备份脚本(有建库命令，可手工修改库名)：

```
mysqldump -uroot -p -B ngqdb > ngqdbbackup20210419.sql
```

修改脚本中的 ngqdb-->ngqdb1

```
mysql -uroot -p < ngqdbbackup20210419.sql
```

【或 source< ngqdbbackup20210419.sql】

```
select count(*) from ngqdb1.ngq_account;
```

方案二、基于 InnoDB Cluster（或 ClusterSet 或 ReplicaSet）集群拓扑结构做故障切换实验：

先参考文档

https://www.niepub.com/static/docs/exadmin/MySQL8_InnoDB_Cluster_Installation_Guide.pdf

安装 MySQL InnoDB Cluster 集群服务器环境。

集群安装好后，重点测试故障转移及单活路由：

故障转移测试

```
mysqlsh -uroot -p -h127.0.0.1 -P6446
cluster = dba.getCluster();
cluster.status()
\sql
select * from performance_schema.replication_group_members\G
\js
#pgrep mysqld -fla
#kill -9    primaryInstancePID
\sql
select * from performance_schema.replication_group_members\G
\js
cluster = dba.getCluster();
cluster.status()
```

恢复就主实例

```
mysqlsh -uroot -p -h127.0.0.1 -P6446
dba.startSandboxInstance(3307);
cluster = dba.getCluster();
cluster.status()
```

--重启整个集群服务器

用 stopSandboxInstance and startSandboxInstance 重启所有实例之后

```
var cluster = dba.rebootClusterFromCompleteOutage()
```

or

Mysqlsh 先启动主节点，并运行：

```
mysql -uroot -p -h127.0.0.1 -P3307
SET GLOBAL group_replication_bootstrap_group=ON;
start group_replication;
SET GLOBAL group_replication_bootstrap_group=OFF;
SELECT * FROM performance_schema.replication_group_members;
```

然后再依次启动所有其它节点(GTID 由大到小)：

5. 多活与单活的转换测试

cluster.switchToMultiPrimaryMode() 多主模式:所有实例都成为主要实例。

```
mysql -uroot -p -h127.0.0.1 -P3309|3307|3308
cluster.switchToSinglePrimaryMode('root@127.0.0.1:3309') 切换到单主模式
```

重启 router：./stop.sh ./start.sh

```
mysql -uroot -p -h127.0.0.1 -P6446|6447
/*
SELECT @@PORT,@@hostname;
create database ngqdb;
create table ngq_tab (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
, name varchar(100) not null
, type    varchar(20)
, create_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
, last_update timestamp default current_timestamp on update current_timestamp);
insert into ngq_tab (name) values (concat('test:port',@@port));
select * from ngq_tab;
select user();
*/
mysql -uroot -p -P6447 -h127.0.0.1
SELECT @@PORT;
mysql -uroot -p -h127.0.0.1 -P6447 -se"select @@port\G"
```

连续连接 6447 三次看看

```
mysql -uroot -p -h127.0.0.1 -P3308 -se"select @@port\G select count(*) from ngq_tab;"
```