36.MySQL-PT工具√

一.PT（Percona-Toolkits）工具的应用

1.PT工具的安装

软件包可直接从percona官网下载

```bash
yum install -y  percona-toolkit-3.1.0-2.el7.x86_64.rpm
```

2.常用工具使用介绍

2.1 pt-archiver（归档表）

使用场景

```
pt-archiver工具功能一：做归档表
做归档表的原因：
一张表数据量太大会影响索引树高度，所以要将大表分库分表，分区表 或者做归档表（转储表）

归档表的使用场景：
业务使用实时数据较多的场景，可以把之前的数据做成归档表存储

pt-archiver工具功能二：批量删除数据
批量删除数据时会造成大事务，mysql中不能并发运行，所以使用PT工具来解决。


面试题： 定期按照时间范围，进行归档表。
面试题： 亿级的大表，delete批量删除100w左右数据。
```

重要参数

```
-limit 100          每次取100行数据用pt-archive处理
--txn-size  100     设置100行为一个事务提交一次，
--where 'id<3000'   设置操作条件
--progress 5000     每处理5000行输出一次处理信息
--statistics        输出执行过程及最后的操作统计。（只要不加上--quiet，默认情况下
--charset=UTF8      指定字符集为UTF8—这个最后加上不然可能出现乱码。
--bulk-delete       批量删除source上的旧数据(例如每次1000行的批量删除操作) --归

注意： 需要归档表中至少有一个索引,做好是where条件列有索引
```

使用案例

```
一.模拟创建归档环境
1.首先模拟环境，上传一张大表(t100w.sql)到数据库中
登陆到要归档的数据库上传表到数据库中
source /opt/t100w.sql
2.登陆到归档的目标端，创建出需要归档表的表结构（整个空表），表结构要一致，归档表重新
2.1在源端数据库中查看建表语句
show create table test.t100w;
CREATE TABLE `t100w` (
  `id` int DEFAULT NULL,
  `num` int DEFAULT NULL,
  `k1` char(2) DEFAULT NULL,
  `k2` char(4) DEFAULT NULL,
  `dt` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
2.2 在目标端执行
mysql> create database test;
Query OK, 1 row affected (0.01 sec)

mysql> use test ;
Database changed
mysql> CREATE TABLE `test1` (
    ->   `id` int DEFAULT NULL,
    ->   `num` int DEFAULT NULL,
    ->   `k1` char(2) DEFAULT NULL,
    ->   `k2` char(4) DEFAULT NULL,
    ->   `dt` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CU
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_
    -> ;
3.在源端上对要归档的t100w表进行创建索引，我们使用工具时要以索引列作为操作的条件。
alter table test.t100w add index idx(id);

二.
1.开始对大表做归档表操作
pt-archiver --source h=10.0.0.51,P=3307,D=test,t=t100w,u=root,p=123 --
2.因为源端的数据已经归档到目标端，所以可以将源端归档的数据进行删除
pt-archiver --source h=10.0.0.51,P=3307,D=test,t=t100w,u=root,p=123 --
3.只把数据导出到外部文件，但是不删除源表里的数据（使用场景迁移数据）
pt-archiver --source h=10.0.0.51,D=world,t=city,u=root,p=123 --where '1

```

2.2 pt-osc(类似工具gh-ost)

使用场景

```
mysql8.0版本之前在线（alter语句）修改表结构 属性、索引创建删除等
线上要使用alter语句就使用PT-osc工具，pt-osc不能加快执行速度，但是能减少对业务的影响

pt-osc的弊端：在主从延时严重时，可能会执行失败，可替换gh-ost工具
```

面试题-pt-osc工作流程 六大步骤🧠

```
前提要创建原来表的两倍磁盘空间，因为中间会拷贝原表数据

1、检查更改表是否有主键或唯一索引，是否有触发器
2、检查修改表的表结构，创建一个临时表，在新表上执行ALTER TABLE语句
create table  bak like t1;
alter table bak add telnum char(11) not null; 在没有数据的表上加索引，所以很
3、在源表上创建三个触发器分别对于INSERT UPDATE DELETE操作
因为是在线上修改，业务没有中止，所以会将DML操作结果分别记录在触发器中，用于追加数据在
4、从源表拷贝数据到临时表，在拷贝过程中，对源表的更新操作会写入到新建表中
5、将临时表和源表rename（需要元数据修改锁，需要短时间锁表）
6、删除源表和触发器，完成表结构的修改。
```

pt-osc工具限制

```
1、源表必须有主键或唯一索引，如果没有工具将停止工作
2、如果线上的复制环境过滤器操作过于复杂，工具将无法工作  ---->延时过高，过滤复制场景
3、如果开启复制延迟检查，但主从延迟时，工具将暂停数据拷贝工作
4、如果开启主服务器负载检查，但主服务器负载较高时，工具将暂停操作
5、当表使用外键时，如果未使用--alter-foreign-keys-method参数，工具将无法执行
6、只支持Innodb存储引擎表，且要求服务器上有该表1倍以上的空闲空间。
```

pt-osc之alter语句限制

```
1、不需要包含alter table关键字，可以包含多个修改操作，使用逗号分开，如"drop clolu
2、不支持rename语句来对表进行重命名操作
3、不支持对索引进行重命名操作
4、如果删除外键，需要对外键名加下划线，如删除外键fk_uid，修改语句为"DROP FOREIGN
```

pt-osc之命令模板

```
## --execute表示执行
## --dry-run表示只进行模拟测试
## 表名只能使用参数t来设置，没有长参数

pt-online-schema-change \
--host="127.0.0.1" \
--port=3307 \
--user="root" \
--password="123" \
--charset="utf8" \
--max-lag=10 \
--check-salve-lag='xxx.xxx.xxx.xxx' \
--recursion-method="hosts" \
--check-interval=2 \
--database="test" \
  t="tb001" \
--alter="add column c4 int" \
--execute
```

举例说明

```
例子：
加列：
[root@db01 data]# pt-online-schema-change  --alter "add column state int


加索引：
pt-online-schema-change  --alter "add index idx(num)" h=10.0.0.51,P=3307

```

2.3 pt-table-checksum/ pt-table-sync 一套工具

使用场景

```
校验主从数据一致性
```

应用

```
1.创建专门存储校验信息的库
Create database pt CHARACTER SET utf8;
2.
```

2.4 pt-table-sync

```
主要参数介绍
--replicate ：指定通过pt-table-checksum得到的表.
--databases ：指定执行同步的数据库。
--tables ：指定执行同步的表，多个用逗号隔开。
--sync-to-master ：指定一个DSN，即从的IP，他会通过show processlist或show slave status 去自动的找主。
h= ：服务器地址，命令里有2个ip，第一次出现的是Master的地址，第2次是Slave的地址。
u= ：帐号。
p= ：密码。
--print ：打印，但不执行命令。
--execute ：执行命令。
pt-table-sync --replicate=pt.checksums --databases test  --tables t1 h=10.0.0.51,u=checksum,p=checksum,P=3307 h=10.0.0.51,u=checksum,p=checksum,P=3307 --print
pt-table-sync --replicate=pt.checksums --databases test  --tables t1 h=10.0.0.51,u=checksum,p=checksum,P=3307 h=10.0.0.51,u=checksum,p=checksum,P=3307 --execute
```

2.5 显示主从结构：pt-slave-find（了解下ORCH）

[root@db01 tmp]# pt-slave-find -h10.0.0.51 -P3307 -uchecksum -pchecksum

2.6 监控主从延时

==========

自带判定方法:

show slave status \G

方法一:

second_behind_master ===> 从库SQL回放的TS-主从执行是的TS - 主从的时间差异(去皮)

为什么不准确?

网络延时,IO线程无法及时拿到日志.

IO线程down,binlog_dump线程down了

如果在构建主从时,没有NTP,主从节点如果此时有时间差,在从库连接主库时,会自动记录差异值,在计算second_behind_master,会加入到公式里.

如果,后期配置了NTP的话,就会导致second_behind_master值不准确.

经典案例 :

show slave status\G 快速执行,出现一会有值一会没有.

网络抖动可能性比较大.

exec_master_log_pos ====> MTS(SQL并发)可能会记录不准确.

==========

pt-heartbeat

主库:

pt-heartbeat --user=root --ask-pass --host=10.0.0.51 -P3307 --create-table -D test --interval=1 --update --replace --daemonize

从库:

pt-heartbeat --user=root --ask-pass --host=10.0.0.52 -P3307 -D test --table=heartbeat --monitor

==========

2.7. pt-show-grants

作用：用户和权限信息迁移。

pt-show-grants -h10.0.0.51 -P3307 -uchecksum -pchecksum

<https://service.weibo.com/share/share.p
Toolkits%EF%BC%89%E5%B7%A5%E5%85%B7%E7%9A%84%E5%BA%94%E7%94%A81.PT%E5%B7%A5%E5%85%B7%E7%9A%84%E5%A
y%20%20percona-toolkit-3.1.0-2.el7.x86_64.rpm2.%E5%B8%B8%E7%94%A8%E5%B7%A5%E5%85%B7%