41 | 怎么最快地复制一张表？

初始化表语句

create database db1;

use db1;

create table t(id int primary key, a int, b int, index(a))engine=innodb;

delimiter ;;

create procedure idata()

begin

declare i int;

set i=1;

while(i<=1000)do

insert into t values(i,i,i);

set i=i+1;

end while;

end;;

delimiter ;

call idata();

create database db2;

create table db2.t like db1.t

假设，把db1.t里面大于a>900的数据行导出来，插入到db2.t中

**Mysqldump 方法**

一种方式是，使用mysqldump命令将数据导出成一组Insert语句，可以使用下面的命令

mysqldump -h$host -P$port -u$user --add-locks=0 --no-create-info --single-transaction --set-gtid-purged=OFF db1 t --where="a>900" --result-file=/client\_tmp/t.sql

把结果输出到临时文件

这个条命令，主要参数含义如下

1. -single-transaction的作用是，在导出数据的时候不需要对db1.t加表锁，而是使用STATR TRANSACTION WITH CONSISTENT SNAPSHOT的方法
2. -add-locks 设置为0，表示在输出文件结果里，不增加“LOCK TABLES t WRITE;”
3. -no-create-info的意思是，不需要导出表结构
4. -set-gtid-purged=off 表示的是，不输出跟GTID相关的信息
5. -result-file指定了输出文件的路径，其中client表示生成的文件是在客户端机器上的，

通过这条mysqldump命令生成的t.sql文件中包含了如图1所示的Insert语句

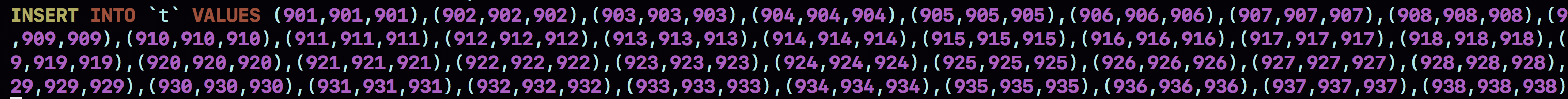


图 1mysqldump输出文件的部分结果

可以看到，一条INSERT语句里面包含了多个value对，这是为了后续用这个文件来写入数据时，执行速度可以更快。

如果希望生成的文件中一条INSERT语句只插入一行数据，可以在执行mysqldump命令时，加上参数-skip-extended-insert.

通过下面的命令，将这些INSERT语句放到db2库里去执行

mysql -h127.0.0.1 -P13000 -uroot db2 -e "source /client\_tmp/t.sql"

需要说明的是，source并不是一条SQL语句，而是一个客户端命令，mysql客户端执行这个命令的流程是这样的。

1. 打开文件，默认以分号为结尾读取一条条的SQL语句
2. 将SQL语句发送到服务端执行

服务端执行的并不是这个“source t.sql”语句，而是INSERT语句，所以不论是在慢查询日志（slow log）,还是在bInlog,记录的都是这些要被真正执行的INSERT语句。

**导出CSV文件**

另一种方法是直接将结果导出成.csv文件，MySQL提供了下面的语法，用来将查询结果导出到服务端本地目录

select \* from db1.t where a>900 into outfile '/server\_tmp/t.csv';

使用这条语句，需要注意一下几点

1. 这条语句将结果保存在服务端，如果执行命令的客户端和MySQL服务端不在同一个机器上，客户端机器的临时目录下是不会生成t.csv文件的
2. Into outfile指定了文件的生成位置（/server\_tmp/）,这个位置必须受参数secure\_file\_priv限制，参数secure\_file\_priv的可选值和作用分别是
   1. 如果设置empty,表示不限制文件的生成位置，这是不安全的设置
   2. 如果设置了一个表示路径的字符串，就要生成的文件只能放在这个指定的目录，或者他的子目录
   3. 如果设置为null,就表示禁止在这个MySQL实例上执行select...into outfile操作
3. 这条命令不会帮住覆盖文件，因此需要确保/server\_tmp/t.csv这个文件不存在，否则执行语句时就会因为有同名的文件的存在而报错
4. 这条命令生成的文件中，原则上一个数据行对应文本文件的一行。但是，如果字段中包含换行符，在生成的文件中也会有换行符。不过类似换行符，制表符这类符号，前面都会跟上“\”这个转义符，这样就可以跟字段之间、数据行之间的分隔符区分开

得到.csv导出文件后，可以用下面的load data命令将数据导入到目标表db2.t中Load data

load data infile '/server\_tmp/t.csv' into table db2.t;

这条语句执行如下

1. 打开/server\_tmp/t.csv。以制表符（\t）作为字段键的分隔符，以换行符（\n）作为记录之间的分隔符，进行数据读取。
2. 启动事务
3. 判断每一行的字段数与表db2.t是否相同

·若不相同，则直接抱错，事务回滚

·若相同，则构造成一行，调用InnoDB引擎接口，写入到表中

1. 重复步骤3，直到/server\_tmp/t.csv整个文件读入完成，提交事务

疑问：**如果binlog\_format=statement，整个load语句记录到binlog里后，怎么在备库执行**

由于/server\_tmp/t.csv文件只保存在主库的机器上，如果只是把这条语句原文写到binlog中，在备库执行的时候，备库的本地机器上没有这文件，就会导致主备同步停止

所以，这条语句执行的完整流程，是下面这样的

1. 主库执行完，将/server\_tmp/t.csv文件的内容直接写到binlog文件中
2. 往binlog文件中写入语句load data local infile ’/tmp/SQL\_LOAD\_MB-1-0’ INTO TABLE ‘db2.t’
3. 把这个binlog日志传递备库
4. 备库的apply线程在执行这个事务日志时
5. 先将binlog中t.csv文件的内容读出来，写入到本地临时目录/tmp/SQL\_LOAD\_MB-1-0中
6. 在执行load data语句，往备库的db2.t表中插入跟主库相同的数据

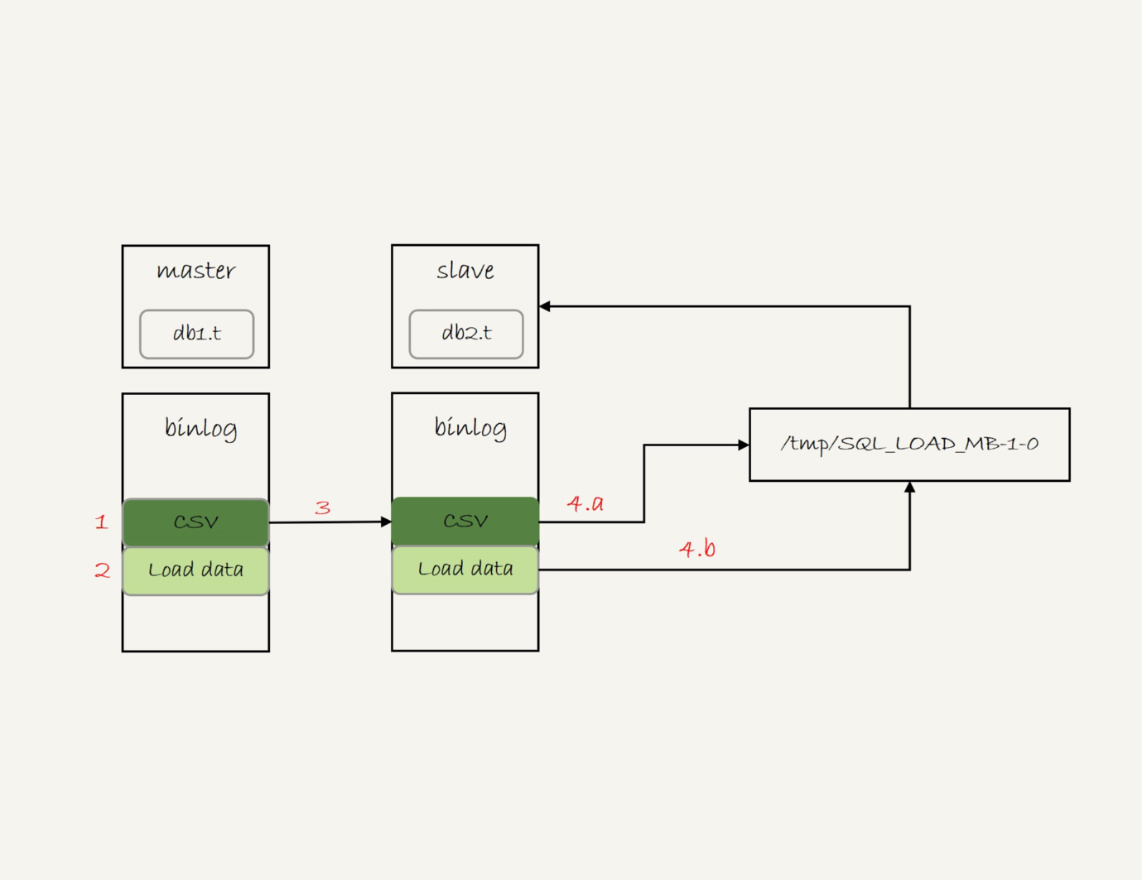


图2 load data 的同步流程

注意，这里备库执行load data语句里面，多了一个“local”,他的意思是“将执行这条命令的客户端所在机器的本地文件/tmp/SQL\_LOAD\_MB-1-0的内容，加载到目标表db2.t中”

Load data命令有两种用法

1. 不加“local”,是读取服务端的文件，这个文件必须在secure\_file\_priv指定的目录或子目录下
2. 加上“local”，读取是客户端的文件，只要mysql客户端 有访问这个文件的权限即可，此时，MySQL客户端会把本地文件传给服务端，然后执行上述的load data流程

另外需要注意的是，**select...into outfile 方法不会生成表结构文件，**所以导数据时还需要单独的命令得到表结构定义，mysqldump提供了一个-tab参数，可以同时导出表结构定义文件和csv数据文件，这条命令的使用方法如下

mysqldump -h$host -P$port -u$user ---single-transaction --set-gtid-purged=OFF db1 t --where="a>900" --tab=$secure\_file\_priv

这条命令会在$secure\_file\_priv 定义的目录下，创建一个t.sql文件保存建表语句，同时创建一个t.txt文件保存CSV数据。

**物理拷贝方法**

介绍mysqldump方法和导出CSV文件的方法，都是逻辑导数据的方法，也就是将数据从表db1.t中读出来，生成文本，然后再写目标db2.t中。

疑问，有物理导数据的方法吗？比如，直接把db1.t表的.frm文件和.ibd文件拷贝到db2目录下，是否可行？ 答案：不行

因为，一个InnoDB表，除了包含这两个物理文件外，还需要在数据字典中注册，直接拷贝这两个文件，因为数据字典没有db2.t这个表，系统时不会识别和接受他们的。

在MySQL5.6版本引入**可传输表空间（transportable tablespace）**的方法，可以通过导入+导入表空间的方式，实现物理拷贝表的功能

假设在目标的db1.t库下，复制一个表t相同的r,具体的中步骤如下：

1. 执行create table r like t,创建一个相同表结构的空表
2. 执行alter table r discard tablespace ,此时r.ibd文件被删除
3. 执行flush table t for export ,此时db1目录下生成一个t.cfg文件
4. 在db1目录下执行cp t.cfg r.cfg; cp t.ibd r.ibd；这两个命令，（需要注意的是，拷贝得到的两个文件，MySQL进程要有读写权限）
5. 执行unlock tables，此时t.cfg文件会被删除
6. 执行alter table r import tablespace 将这个r.ibd文件作为r的新的表空间，由于这个文件的数据内容和t.ibd是相同的，所以表r就有了表t相同的数据。

至此，拷贝数据的操作完成，这个流程如下，需要注意一下几点

1. 在第3步执行完成flush table命令，db1.t整个表处于只读状态，知道执行unlock tables命令才释放读锁
2. 在执行import tablespace 时，为了让文件的表空间id和数据字典中的一致，会修改r.ibd的表空间的id，而这个表空间id存在于每个数据页中，因此，如果是一个很大的文件（比如TB级别），每个数据页都需要修改，所以会看到import语句的执行是需要一些时间的，如果相比于逻辑导入的方法，import语句的耗时是非常短的。

总结

1. 物理拷贝的方法速度最快，尤其是对于大表拷贝来说是最快的方法，如果出现误删表的情况，用备份恢复出误删之前的临时库，然后再把临时库中的表拷贝到生产库上，是恢复数据最快的方法，但是，这种方法的使用也有一定的局限性；

·必须是全表拷贝，不能只拷贝部分数据

·需要到服务器上拷贝数据，在用户无法登陆数据主机的场景下无法使用

·由于是通过拷贝物理文件实现的，原表和目标表都是使用InnoDB引擎时才能使用

1. 用mysqldump生成包含INSERT语句文件的方法，可以在where 参数增加过滤条件，来实现只导出部分数据。这个方式的不足之一是，不能使用join这种比较复杂的where条件写法
2. 用select...into outfile的方法是最灵活的，支持所有的SQL写法，但，这个方法的缺点之一是，每次只能导出一张表的数据，而且表结构也需要另外的语句单独备份

后两种方式是逻辑备份方式，可以跨引擎使用的

**思考题**

前面介绍binlog\_format=statement时，binlog记录的load data命令是带local的。既然这条命令是发送到备库执行的，那么备库执行的时也是本地执行，为什么需要这个local呢？如果写到binlog中的命令不带local,又会出现什么问题呢？