

# 扒下mysqldump和 xtrabackup的内裤

WOQU TECH  
.com

# 目录

- 01 mysqldump故障案例和解读
- 02 innobackupex故障案例和解读
- 03 解决方法以及如何避免类似备份故障

# 目录

01 mysqldump故障案例和解读

02 innobackupex故障案例和解读

03 解决方法以及如何避免类似备份故障

- 备份选项: `--single-transaction --master-data=2 --triggers --events --no-autocommit --hex-blob -R -A`

Query **FLUSH /\*!40101 LOCAL \*/ TABLES**

Query **FLUSH TABLES WITH READ LOCK**

Query **SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ**

Query **START TRANSACTION /\*!40100 WITH CONSISTENT SNAPSHOT \*/**

.....

Query **SELECT @@GLOBAL.GTID\_EXECUTED**

Query **SHOW MASTER STATUS**

.....

Query **UNLOCK TABLES**

.....

Init DB luoxiaobo

Query **SHOW CREATE DATABASE IF NOT EXISTS `luoxiaobo`**

Query **SAVEPOINT sp**

Query **show tables**

Query **show table status like 't\_xx'**

Query **show create table `t\_xx`**

Query **show fields from `t\_xx`**

Query **SELECT /\*!40001 SQL\_NO\_CACHE \*/ \* FROM `t\_xx`**

Query **SHOW TRIGGERS LIKE 't\_xx'**

Query **ROLLBACK TO SAVEPOINT sp**

Query **show table status like 't\_xx2'**

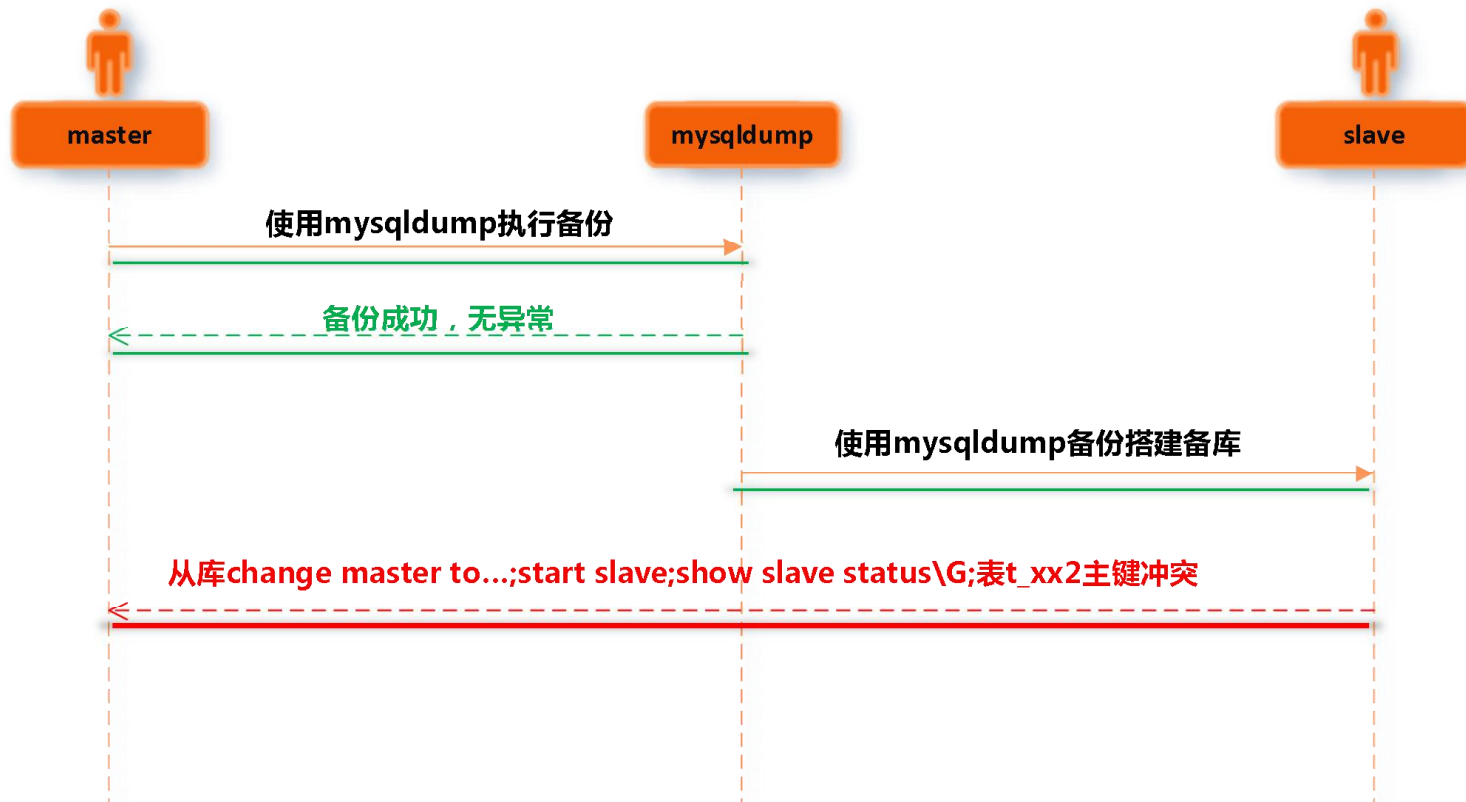
## 案例一

- 一主一从复制架构中，备库损坏需要重建，在主库上在线备份一份数据来恢复到备库

## 案例二

- 一主多从的复制架构中，在备库用于读负载均衡，读负载不够需要扩容备库，在已有备库上在线备份一份数据来克隆一个新的备库

# mysqldump案例一时序图



## 故障案例一

怀疑备份期间可能有myisam表发生DML操作

查看mysqldump备份选项和报错表的引擎

备库上执行show slave status\G;查看报错的binlog pos、冲突的主键值

查看备份文件中的binlog pos值以及从报错的主键值开始后续是否还有其他值

解析主库binlog，并使用复制报错位置的binlog pos和备份文件中的binlog pos查看对应的主键值是什么

重现：主库打开general\_log，使用mysqldump备份(备份选项不变)，并在备份期间持续对myisam表写入

Query SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ

Query insert into t\_xx2(test,datet\_time) values(11377,now())

Query START TRANSACTION /\*!40100 WITH CONSISTENT SNAPSHOT \*/

Query SHOW VARIABLES LIKE 'gtid\ \_mode'

Query SELECT @@GLOBAL.GTID\_EXECUTED

Query **SHOW MASTER STATUS**

Query **UNLOCK TABLES**

.....

Query insert into t\_xx2(test,datet\_time) values(11383,now())

.....

Query insert into t\_xx2(test,datet\_time) values(11380,now())

Init DB luoxiaobo

Query insert into t\_xx2(test,datet\_time) values(11382,now())

.....

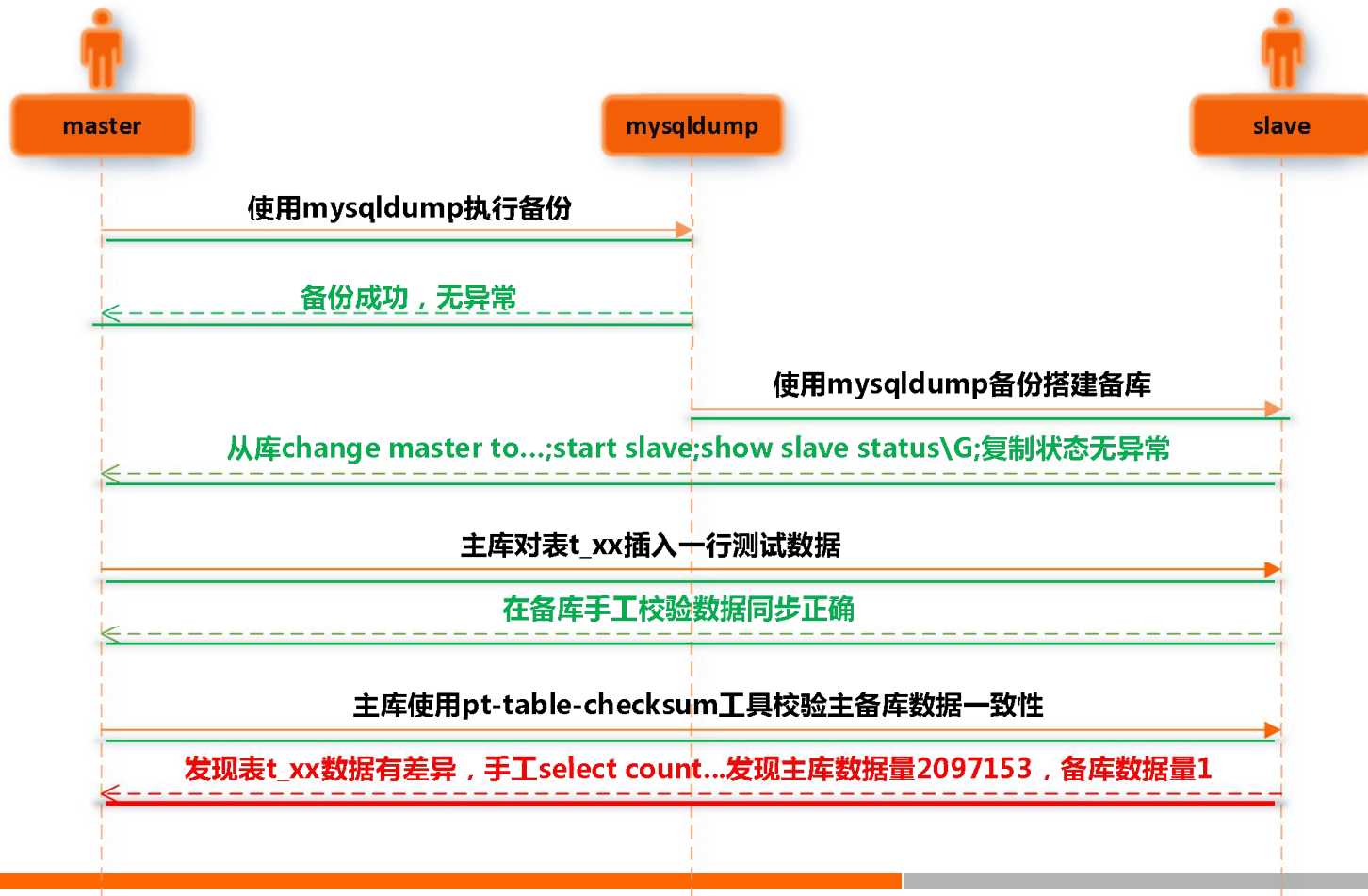
Query show fields from `t\_xx2`

Query show fields from `t\_xx2`

Query **SELECT /\*!40001 SQL\_NO\_CACHE \*/ \* FROM `t\_xx2`**



# mysqldump案例二时序图



## 故障案例二

怀疑备份期间可能有**innodb**事务表发生**DDL**操作

查看mysqldump**备份选项**和报错表的**引擎**

查看**备份文件**中的**binlog pos**值以及**数据有差异的表**备份出来的**数据情况**

**复现**：主库打开**general\_log**，使用mysqldump备份(备份选项不变)，并在备份期间**持续对innodb表执行DDL操作**

Query FLUSH /\*!40101 LOCAL \*/ TABLES  
Query FLUSH TABLES WITH READ LOCK  
Query SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ  
Query **START TRANSACTION /\*!40100 WITH CONSISTENT SNAPSHOT \*/**

.....  
Query SELECT @@GLOBAL.GTID\_EXECUTED  
Query **SHOW MASTER STATUS**

.....  
Query **UNLOCK TABLES**  
Query **alter table t\_xx add column test8 varchar(10)**

.....  
Init DB luoxiaobo  
Query SHOW CREATE DATABASE IF NOT EXISTS `luoxiaobo`  
Query SAVEPOINT sp  
Query show tables  
Query show table status like 't\_xx'  
Query **show create table `t\_xx`**  
Query **show fields from `t\_xx`**

# 这里原本应该是一句: **SELECT /\*!40001 SQL\_NO\_CACHE \*/ \* FROM `t\_xx`**, 但是却没有, 我们可以推理一下  
# **select**的时候报了表定义已经发生变化的错误(但mysqldump报不出来), 所以这句**select**并没有被记录到查询日志中来

Query SHOW TRIGGERS LIKE 't\_xx'  
Query ROLLBACK TO SAVEPOINT sp  
Query **show table status like 't\_xx2'**

思考？

- mysqldump备份过程中的如下语句有什么作用？
- **SAVEPOINT sp**
- **ROLLBACK TO SAVEPOINT sp**
- **RELEASE SAVEPOINT sp**

# 目录

01 mysqldump故障案例和解读

02 innobackupex故障案例和解读

03 解决方法以及如何避免类似备份故障



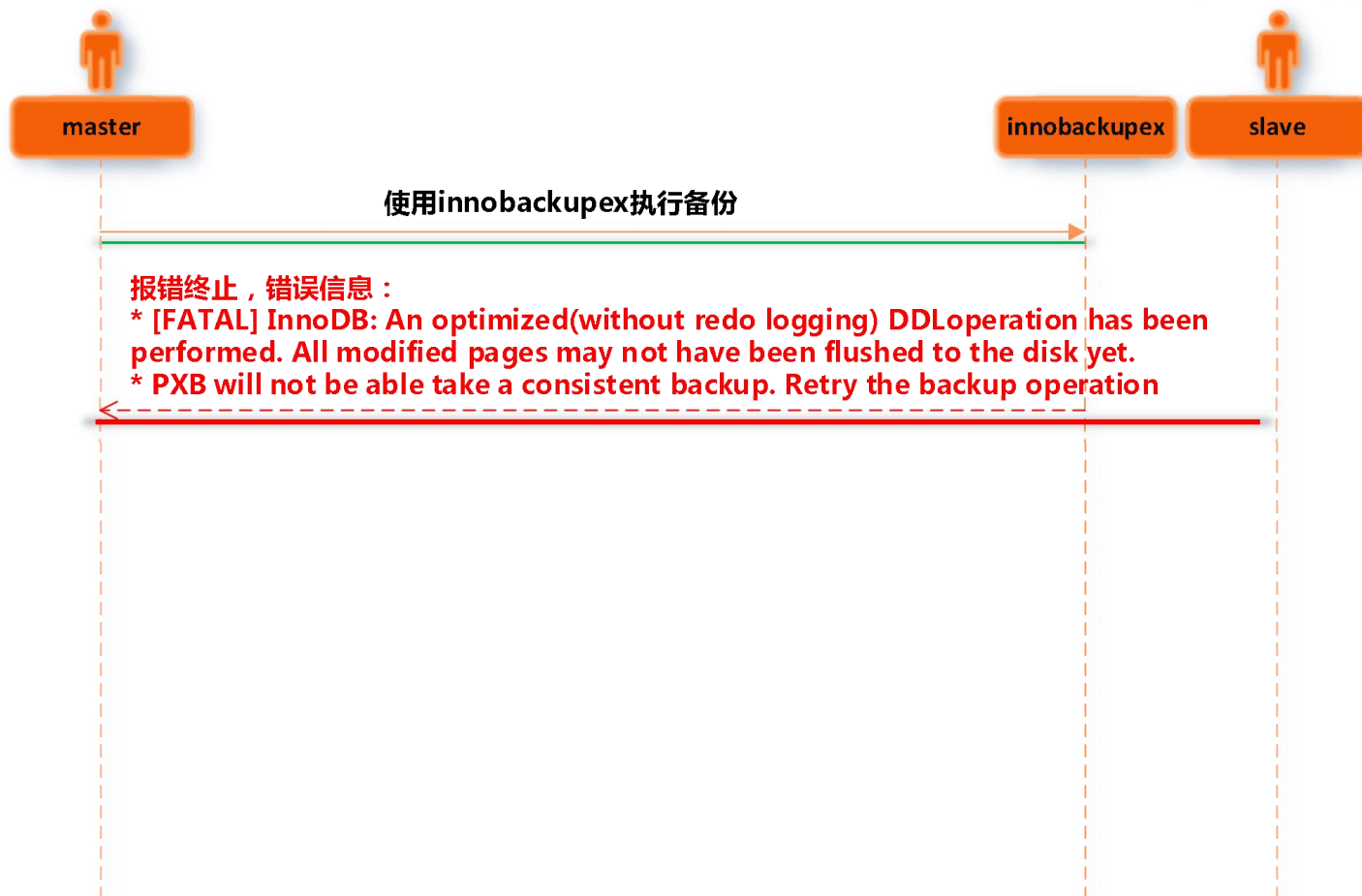
```
Query set autocommit=1
Query SET SESSION wait_timeout=2147483
Query SELECT CONCAT(@@hostname, @@port)
Query SET SESSION wait_timeout=2147483
Query SHOW VARIABLES
Query SHOW ENGINE INNODB STATUS
Query SET SESSION lock_wait_timeout=31536000
Query FLUSH NO_WRITE_TO_BINLOG TABLES
Query FLUSH TABLES WITH READ LOCK
Query SHOW SLAVE STATUS
Query SHOW MASTER STATUS
Query SHOW VARIABLES
Query FLUSH NO_WRITE_TO_BINLOG ENGINE LOGS
Query UNLOCK TABLES
Query SELECT UUID()
Query SELECT VERSION()
Quit
```



- 
- 按照固定**每日备份策略**进行备份



# innobackupex故障案例时序图



## 故障案例

怀疑备份期间可能有**innodb**表发生**DDL**操作

查看**备份选项**是什么，备份打印日志中**报错时在备份什么表**

查看备份目录下的**文件有哪些**

**复现**：主库打开**general\_log**，使用**innobackupex**备份(备份选项不变)，并在备份期间**持续对innodb表执行online DDL加字段操作**，最好加个**strace -f**命令

```
.....
Query      SELECT DATABASE()
Init DB    luoxlaobo
Query      alter table t_luoxlaobo add column test1 varchar(10)
.....
Query      SET SESSION wait_timeout=2147483
Query      SELECT CONCAT(@@hostname, @@port)
.....
Query      alter table t_luoxlaobo add column test4 varchar(10)
Query      SHOW VARIABLES
.....
Query      SET SESSION wait_timeout=2147483
Query      SHOW VARIABLES
Query      SHOW ENGINE INNODB STATUS
# 对比innobackupex工具输出的时间信息，错误发生在执行'\
SHOW ENGINE INNODB STATUS'这句SQL之后，且后续没有任何输出了
.....
Quit
```

## 思考？

- 我们通过复现步骤中的general\_log和innobackupex工具打印的日志确认了故障案例中的报错是由于备份过程中的online DDL操作导致的，但是：
- 什么是Online DDL？
- 为什么Online DDL操作会导致innobackupex备份报错终止呢？

# 为什么Online DDL操作会导致innobackupex备份报错终止

- alter table tb\_name add column test1 varchar(10);
- 使用online ddl添加字段会使用~~inplace rebuild~~方式，该方式虽然不会阻塞DML操作，但是会内部重建一张新表，与原表tb\_name进行交换-----也就是说：table\_id、tablespace ID 发生了变化

- innobackupex在启动redo log拷贝之后，先打开一次ibdata1读取如table\_id、tablespace ID 等这些元数据信息，后续如果发现表的table\_id、tablespace ID 等元数据发生了变化，那么在开始备份ibdata1时再次打开就会发现元数据发生了变化而导致备份失败，针对这个场景，percona-xtrabackup2.4.1手册中有一段描述：
- Executing DDL operations, TRUNCATE TABLE, DROP/CREATE the \_same\_ table or ALTER statements on InnoDB tables while taking a backup could lead to a xtrabackup failure due to a tablespace ID mismatch when using per-table tablespaces

# 目录

01 mysqldump故障案例和解读

02 innobackupex故障案例和解读

03 解决方法以及如何避免类似备份故障

## mysqldump 故障案例一

- **解决方法：**
- 1、如果备份数据要求**强一致性**场景，如用于搭建备库，那么建议**去掉--single-transaction**选项，但备份过程会**全程加锁**，建议在业务**低峰期**进行操作
- 2、如果必须即时处理的场景，建议使用innobackupex工具备份

## mysqldump 故障案例二

- **解决方法：**
- 1、可以从**规范**上入手，DDL操作**避开**备份时间段
- 2、可以不使用--single-transaction选项，让其全程加锁

## innobackupex 故障案例

- **解决方法：**
- 1、没有办法全程加锁，只能从**规范**上入手，DDL操作**避开**备份时间段

1

- 一定要保持一种高度**谨慎**的态度，在数据库备份方案选型时，一定要根据自己的业务场景**充分测试**，**校验**，尽可能地把可能出现的深坑**挖出来**，提前进行**规避**

2

- 除了寻找**适合自己的**，可行的**备份方案**之外，更应该做好**备份校验**（备份是否成功完成、备份文件是否损坏）、**备份恢复演练**（备份文件是否可以正常恢复数据），以备不时之需

3

- 对生产库的**DDL**操作、**大事务**、或者长时间**锁表**的操作，一定要**避开**备份时间

- **PS:**
- 很多人喜欢在备份前先**flush binary logs**一把，其实在有**大事务**对数据进行修改时，一不小心可能会出现数据库**hang**死，所以**不建议**这么做
- **innobackupex**备份期间，在数据库中创建的**连接不要误杀**，否则**备份失败**





# Let Data Drive!

WOQU TECH all rights reserved.

杭州沃趣科技股份有限公司  
Hangzhou WOQU Technology Co., Ltd.

WOQU TECH all rights reserved.