**1. 环境描述**

目的：当数据库中设置了binlog-do-db时，在不同的binlog\_format=statement | row | mixed 下对binlog的写入影响，这个在主从复制中会有一些坑，由于binlog的写入不完全，极有可能会导致主从不一致的情况的。

[blog地址：http://blog.csdn.net/hw\_libo/article/details/40476577](http://blog.csdn.net/hw_libo/article/details/40476577)

SuSE 11 sp1 x86\_64  +  MySQL 5.5.37

**参数设置：**

binlog-do-db = bosco1

**测试样例1：**

use bosco2;

create table bosco1.bosco1\_tb01(id int);

create table bosco2.bosco2\_tb01(id int);

insert into bosco1.bosco1\_tb01(id) values(1);

insert into bosco2.bosco2\_tb01(id) values(1);

**测试样例2：**

use bosco1;

create table bosco1.bosco1\_tb01(id int);

create table bosco2.bosco2\_tb01(id int);

insert into bosco1.bosco1\_tb01(id) values(1);

insert into bosco2.bosco2\_tb01(id) values(1);

**2. 测试1：use bosco2及SBR/RBR/MBR下**

binlog-do-db=bosco1;

MySQL [(none)]> use bosco2;

Database changed

MySQL [bosco2]> select @@tx\_isolation,@@binlog\_format;

+-----------------+-----------------+

| @@tx\_isolation | @@binlog\_format |

+-----------------+-----------------+

| REPEATABLE-READ | STATEMENT |

+-----------------+-----------------+

1 row in set (0.00 sec)

MySQL [bosco1]> flush logs;

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> create table bosco1.bosco1\_tb01(id int);

Query OK, 0 rows affected (0.01 sec)

MySQL [bosco1]> create table bosco2.bosco2\_tb01(id int);

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> insert into bosco1.bosco1\_tb01(id) values(1);

Query OK, 1 row affected (0.01 sec)

MySQL [bosco1]> insert into bosco2.bosco2\_tb01(id) values(1);

Query OK, 1 row affected (0.00 sec)

MySQL [bosco1]> flush logs;

Query OK, 0 rows affected (0.01 sec)

那么来查看一下上面的操作有没有写入binlog中：

# mysqlbinlog --verbose --base64-output=decode-rows mysql-bin.000013

/\*!50530 SET @@SESSION.PSEUDO\_SLAVE\_MODE=1\*/;

/\*!40019 SET @@session.max\_insert\_delayed\_threads=0\*/;

/\*!50003 SET @OLD\_COMPLETION\_TYPE=@@COMPLETION\_TYPE,COMPLETION\_TYPE=0\*/;

DELIMITER /\*!\*/;

# at 4

#141026 1:41:09 server id 1303308 end\_log\_pos 107 Start: binlog v 4, server v 5.5.37-log created 141026 1:41:09

# at 107

#141026 1:43:02 server id 1303308 end\_log\_pos 150 Rotate to mysql-bin.000014 pos: 4

DELIMITER ;

# End of log file

ROLLBACK /\* added by mysqlbinlog \*/;

/\*!50003 SET COMPLETION\_TYPE=@OLD\_COMPLETION\_TYPE\*/;

/\*!50530 SET @@SESSION.PSEUDO\_SLAVE\_MODE=0\*/;

可见，指定了binlog-do-db=bosco1，事务隔离级别RR + binlog\_format=statement或是row，在使用其他database（非bosco1数据库）下的所有操作都不会记录到binlogs中，即使是操作binlog-do-db=bosco1下的表；而且DDL也不会被记录。

**3. 测试2：use bosco1及RBR下**

binlog-do-db=bosco1;

MySQL [bosco2]> use bosco1;

MySQL [bosco1]> select @@tx\_isolation,@@binlog\_format;

+-----------------+-----------------+

| @@tx\_isolation | @@binlog\_format |

+-----------------+-----------------+

| REPEATABLE-READ | ROW |

+-----------------+-----------------+

1 row in set (0.00 sec)

MySQL [bosco1]> flush logs;

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> create table bosco1.bosco1\_tb01(id int);

Query OK, 0 rows affected (0.01 sec)

MySQL [bosco1]> create table bosco2.bosco2\_tb01(id int);

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> insert into bosco1.bosco1\_tb01(id) values(1);

Query OK, 1 row affected (0.01 sec)

MySQL [bosco1]> insert into bosco2.bosco2\_tb01(id) values(1);

Query OK, 1 row affected (0.00 sec)

MySQL [bosco1]> flush logs;

Query OK, 0 rows affected (0.01 sec)

那么来查看一下上面的操作有没有写入binlog中：

# mysqlbinlog --verbose --base64-output=decode-rows mysql-bin.000006

……

SET @@session.collation\_database=DEFAULT/\*!\*/;

create table bosco1.bosco1\_tb01(id int)

/\*!\*/;

# at 211

#141026 1:37:44 server id 1303308 end\_log\_pos 315 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258664/\*!\*/;

create table bosco2.bosco2\_tb01(id int)

/\*!\*/;

# at 315

#141026 1:37:44 server id 1303308 end\_log\_pos 385 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258664/\*!\*/;

BEGIN

/\*!\*/;

# at 385

# at 437

#141026 1:37:44 server id 1303308 end\_log\_pos 437 Table\_map: `bosco1`.`bosco1\_tb01` mapped to number 49

#141026 1:37:44 server id 1303308 end\_log\_pos 471 Write\_rows: table id 49 flags: STMT\_END\_F

### INSERT INTO `bosco1`.`bosco1\_tb01`

### SET

### @1=1

# at 471

#141026 1:37:44 server id 1303308 end\_log\_pos 498 Xid = 200

COMMIT/\*!\*/;

# at 498

#141026 1:37:49 server id 1303308 end\_log\_pos 541 Rotate to mysql-bin.000011 pos: 4

DELIMITER ;

# End of log file

ROLLBACK /\* added by mysqlbinlog \*/;

/\*!50003 SET COMPLETION\_TYPE=@OLD\_COMPLETION\_TYPE\*/;

/\*!50530 SET @@SESSION.PSEUDO\_SLAVE\_MODE=0\*/;

可见，指定了binlog-do-db=bosco1，事务隔离级别RR + binlog\_format=row：  
在使用指定的database（bosco1数据库）下操作本身库中的表所有DDL/DML操作都会记录到binlogs中，而操作其他库中的表时，只有DDL操作被记录下来，DML操作都不会记录。

**4. 测试3：use bosco1及SBR/MBR下**

binlog-do-db=bosco1;

MySQL [bosco2]> use bosco1;

MySQL [bosco1]> select @@tx\_isolation,@@binlog\_format;

+-----------------+-----------------+

| @@tx\_isolation | @@binlog\_format |

+-----------------+-----------------+

| REPEATABLE-READ | STATEMENT |

+-----------------+-----------------+

1 row in set (0.00 sec)

MySQL [bosco1]> flush logs;

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> create table bosco1.bosco1\_tb01(id int);

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> create table bosco2.bosco2\_tb01(id int);

Query OK, 0 rows affected (0.00 sec)

MySQL [bosco1]> insert into bosco1.bosco1\_tb01(id) values(1);

Query OK, 1 row affected (0.00 sec)

MySQL [bosco1]> insert into bosco2.bosco2\_tb01(id) values(1);

Query OK, 1 row affected (0.00 sec)

MySQL [bosco1]> flush logs;

Query OK, 0 rows affected (0.00 sec)

那么来查看一下上面的操作有没有写入binlog中：

# mysqlbinlog --verbose --base64-output=decode-rows mysql-bin.000008

……

SET @@session.collation\_database=DEFAULT/\*!\*/;

create table bosco1.bosco1\_tb01(id int)

/\*!\*/;

# at 211

#141026 1:33:43 server id 1303308 end\_log\_pos 315 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258423/\*!\*/;

create table bosco2.bosco2\_tb01(id int)

/\*!\*/;

# at 315

#141026 1:33:48 server id 1303308 end\_log\_pos 385 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258428/\*!\*/;

BEGIN

/\*!\*/;

# at 385

#141026 1:33:48 server id 1303308 end\_log\_pos 494 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258428/\*!\*/;

insert into bosco1.bosco1\_tb01(id) values(1)

/\*!\*/;

# at 494

#141026 1:33:48 server id 1303308 end\_log\_pos 521 Xid = 188

COMMIT/\*!\*/;

# at 521

#141026 1:33:50 server id 1303308 end\_log\_pos 591 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258430/\*!\*/;

BEGIN

/\*!\*/;

# at 591

#141026 1:33:50 server id 1303308 end\_log\_pos 700 Query thread\_id=14 exec\_time=0 error\_code=0

SET TIMESTAMP=1414258430/\*!\*/;

insert into bosco2.bosco2\_tb01(id) values(1)

/\*!\*/;

# at 700

#141026 1:33:50 server id 1303308 end\_log\_pos 727 Xid = 189

COMMIT/\*!\*/;

# at 727

#141026 1:33:58 server id 1303308 end\_log\_pos 770 Rotate to mysql-bin.000009 pos: 4

DELIMITER ;

# End of log file

ROLLBACK /\* added by mysqlbinlog \*/;

/\*!50003 SET COMPLETION\_TYPE=@OLD\_COMPLETION\_TYPE\*/;

/\*!50530 SET @@SESSION.PSEUDO\_SLAVE\_MODE=0\*/;

可见，指定了binlog-do-db=bosco1，事务隔离级别RR + binlog\_format=statement，在使用指定的database（bosco1数据库）下操作所有数据库下的表中的所有操作DML都会记录到binlogs中，即使是操作非binlog-do-db=bosco1指定数据库下的表；而且DDL也会被记录。另外在binlog\_format=mixed下也是一样的结果。