

binlog

目录

1 binlog 和主库性能

2 binlog 和主备同步性能

基础知识回顾：binlog 存储和格式

```
root@devops_db 15:30: [test]> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
binlog.000001	925			ee5328d0-9342-11ef-a02d-a0510b15e2c1:1-4

```
1 row in set (0.00 sec)
```

```
root@devops_db 16:39: [test]> select @@server_uuid;
```

@@server_uuid
ee5328d0-9342-11ef-a02d-a0510b15e2c1

基础知识回顾：binlog 存储和格式

row_format=statement

```
mysql> set binlog_format='statement';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> create table t1(id int primary key auto_increment, a int not null , b int,      c int  , index (a), index bc(b,c))engine=innodb;
Query OK, 0 rows affected (0.03 sec)

mysql> insert into t1(a,b,c) values(1,1,1),(2,2,2),(3,3,3),(4,4,4);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> insert  into t1(a,b,c) select a,b,c from t1;
Query OK, 4 rows affected, 1 warning (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 1
```

mysqlbinlog binlog.00000001

```
/*!80013 SET @@session.sql_require_primary_key=0*//*!*/;
create table t1(id int primary key auto_increment, a int not null , b int,      c int  , index (a), index bc(b,c))engine=innodb
/*!*/;
# at 449
```

```
SET TIMESTAMP=1723329354*//*!*/;
insert into t1(a,b,c) values(1,1,1),(2,2,2),(3,3,3),(4,4,4)
/*!*/;
# at 778
```


基础知识回顾：binlog 存储和格式

flush binary logs ;set binlog_format=row

```

/*!80013 SET @@session.sql_require_primary_key=0*//*!*/;
create table t1(id int primary key auto_increment, a int not null , b int,      c int  , index (a), index bc(b,c))engine=innodb
/*!*/;
# at 449

```

```

# at 603
#240811  6:42:28 server id 1  end_log_pos 654 CRC32 0xd9a3df9d  Table_map: `test`.`t1` mapped to number 174
# has_generated_invisible_primary_key=0
# at 654
#240811  6:42:28 server id 1  end_log_pos 757 CRC32 0x8f9d472f  Write_rows: table id 174 flags: STMT_END_F

```

```

BINLOG '
10y3ZhMBAAAAMwAAAI4CAAAAAK4AAAAAAAEABHRlc3QAAAnQxAAQDAwMDAAwBAQCd36PZ
10y3Zh4BAAAAZwAAAPUCAAAAAK4AAAAAAAEAAgAE/wABAAAAAQAAAAEAAAABAAAAAAIAAAACAAAA
AgAAAAIAAAAAAwAAAAMAAAADAAAAAwAAAAEAAAABAAAAAQAAAAEAAAAL0edjw==
'/*!*/;
# at 757
#240811  6:42:28 server id 1  end_log_pos 788 CRC32 0x6fe0ee13  Xid = 240
COMMIT/*!*/;

```

基础知识回顾：binlog 存储和格式

mysqlbinlog -vv master-bin.000002

```
BEGIN
/*!*/;
# at 942
#240811 6:42:28 server id 1  end_log_pos 993 CRC32 0x320922d7  Table_map: `test`.`t1` mapped to number 174
# has_generated_invisible_primary_key=0
# at 993
#240811 6:42:28 server id 1  end_log_pos 1096 CRC32 0xb3abe6ca          Write_rows: table id 174 flags: STMT_END_F

BINLOG '
10y3ZhMBAAAAmAAAA0EDAAAAAK4AAAAAAAEABHRLc3QAAAnQxAAQDAwMDAAwBAQDXIgky
10y3Zh4BAAAZwAAAEgEAAAAAK4AAAAAAAEAAgAE/wAFAAAAAQAAAAEAAAABAAAAAYAAAACAAAA
AgAAAAIAAAABwAAAAmAAAAADAAAAwAAAAIAAAABAAAAAQAAAAEAAAAyuarsw==
'/*!*/;
### INSERT INTO `test`.`t1`
### SET
###   @1=5 /* INT meta=0 nullable=0 is_null=0 */
###   @2=1 /* INT meta=0 nullable=0 is_null=0 */
###   @3=1 /* INT meta=0 nullable=1 is_null=0 */
###   @4=1 /* INT meta=0 nullable=1 is_null=0 */
### INSERT INTO `test`.`t1`
### SET
###   @1=6 /* INT meta=0 nullable=0 is_null=0 */
###   @2=2 /* INT meta=0 nullable=0 is_null=0 */
###   @3=2 /* INT meta=0 nullable=1 is_null=0 */
###   @4=2 /* INT meta=0 nullable=1 is_null=0 */
### INSERT INTO `test`.`t1`
### SET
###   @1=7 /* INT meta=0 nullable=0 is_null=0 */
###   @2=3 /* INT meta=0 nullable=0 is_null=0 */
###   @3=3 /* INT meta=0 nullable=1 is_null=0 */
###   @4=3 /* INT meta=0 nullable=1 is_null=0 */
### INSERT INTO `test`.`t1`
### SET
###   @1=8 /* INT meta=0 nullable=0 is_null=0 */
###   @2=4 /* INT meta=0 nullable=0 is_null=0 */
###   @3=4 /* INT meta=0 nullable=1 is_null=0 */
###   @4=4 /* INT meta=0 nullable=1 is_null=0 */
# at 1096
#240811 6:42:28 server id 1  end_log_pos 1127 CRC32 0x48377e4b          Xid = 241
COMMIT/*!*/;
```


基础知识回顾：binlog 存储和格式

课堂练习：binlog.999999 之后再 flush binary logs 会怎么样？

```
/**
 * Start writing to a new log file or reopen the old file.
 *
 * @param need_lock_log If true, this function acquires LOCK_log;
 * otherwise the caller should already have acquired it.
 *
 * @param extra_description_event The master's FDE to be written by the I/O
 * thread while creating a new relay log file. This should be NULL for
 * binary log files.
 *
 * @retval 0 success
 * @retval nonzero - error
 *
 * @note The new file name is stored last in the index file
 */
int MYSQL_BIN_LOG::new_file_impl(
    bool need_lock_log, Format_description_log_event *extra_description_event) {
    int error = 0;
    bool close_on_error = false;
    char new_name[FN_REFLen], *new_name_ptr = nullptr, *old_name, *file_to_open;
    const size_t ERR_CLOSE_MSG_LEN = 1024;
    char close_on_error_msg[ERR_CLOSE_MSG_LEN];
    memset(close_on_error_msg, 0, sizeof close_on_error_msg);
```

```
if (!error) {
    /* reopen the binary log file. */
    file_to_open = new_name_ptr;
    error = open_binlog(old_name, new_name_ptr, max_size,
        true /*null_created_arg=true*/,
        false /*need_lock_index=false*/,
        true /*need_sid_lock=true*/, extra_description_event);
}
```

基础知识回顾：binlog存储和格式

课堂练习：binlog.999999 之后再 flush binary logs 会怎么样？

```
Thread 39 "connection" hit Breakpoint 1, MYSQL_BIN_LOG::open_binlog (this=0x867a700 <mysql_bin_log>, log_name=0x7fe85002c2c0 "binlog",
  new_name=0x7fe8906f4e80 "./binlog.000021", max_size_arg=1073741824, null_created_arg=true, need_lock_index=false, need_sid_lock=true,
  extra_description_event=0x0, new_index_number=0) at /root/dq/mysql-8.0.31/sql/binlog.cc:4842
4842      assert(need_sid_lock || !need_lock_index);
(gdb) p new_name[9]='9'
$1 = 57 '9'
(gdb) p new_name[10]='9'
$2 = 57 '9'
(gdb) p new_name[11]='9'
$3 = 57 '9'
(gdb) p new_name[12]='9'
$4 = 57 '9'
(gdb) p new_name[13]='9'
$5 = 57 '9'
(gdb) p new_name[14]='9'
$6 = 57 '9'
(gdb)
$7 = 57 '9'
(gdb) p new_name
$8 = 0x7fe8906f4e80 "./binlog.999999"
```


基础知识回顾：binlog存储和格式

课堂练习：binlog.999999 之后再 flush binary logs 会怎么样？

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
binlog.999999	157			

```
1 row in set (0.00 sec)
```

```
mysql> flush binary logs;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
binlog.1000000	157			

基础知识回顾：binlog 存储和格式

课堂练习：服务一直开着，binlog 会不会把磁盘打爆？

```
binlog_expire_logs_auto_purge  
binlog_expire_logs_seconds      | 2592000 |
```

课堂练习：到达 2592000 秒马上切吗？

一个 bug: extrabackup 导致 binlog 不

purge

LOCK INSTANCE FOR BACKUP

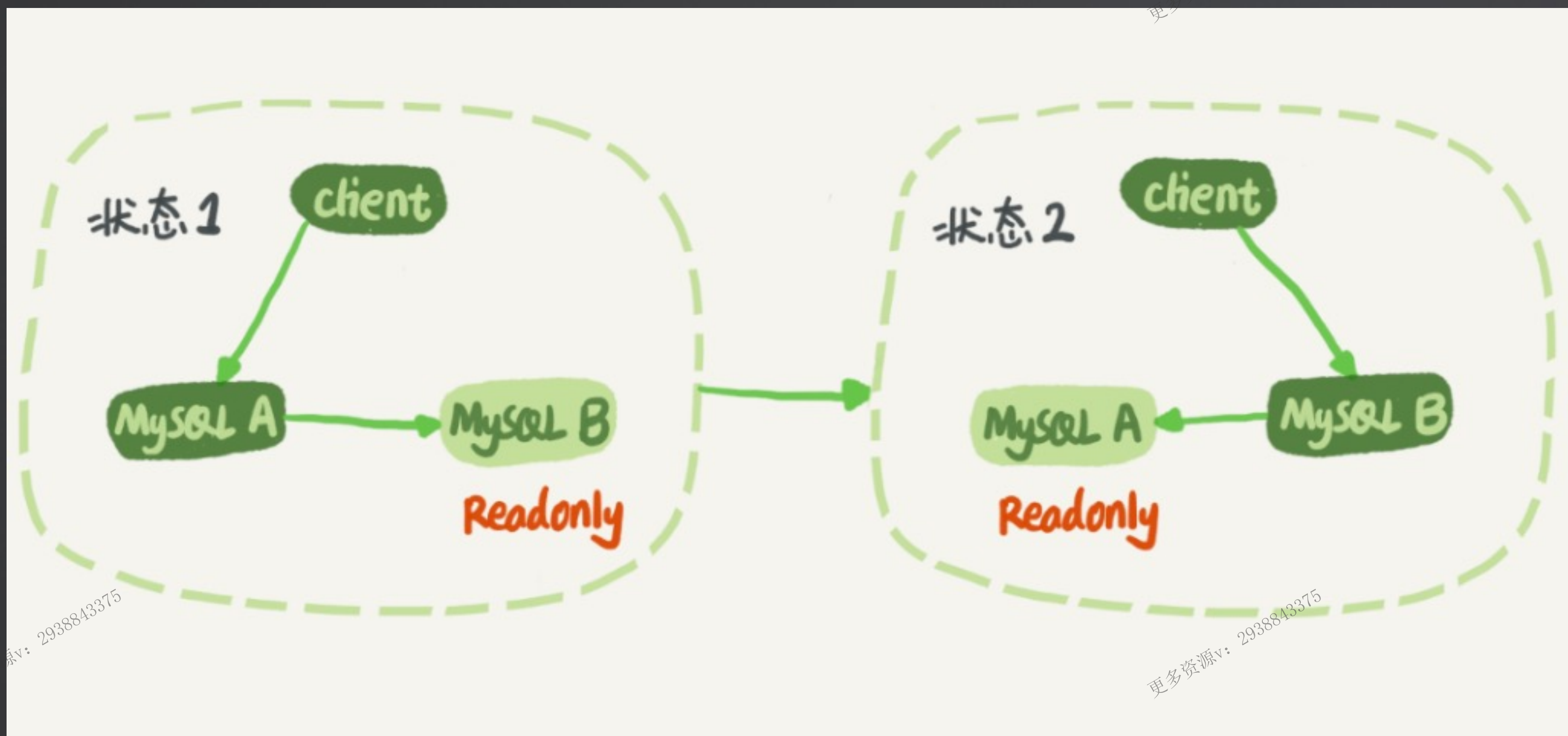
基础知识回顾：binlog 存储和格式

statement 格式要被官方弃用了，本单元只讨论 `binlog_format=row` 的情况。

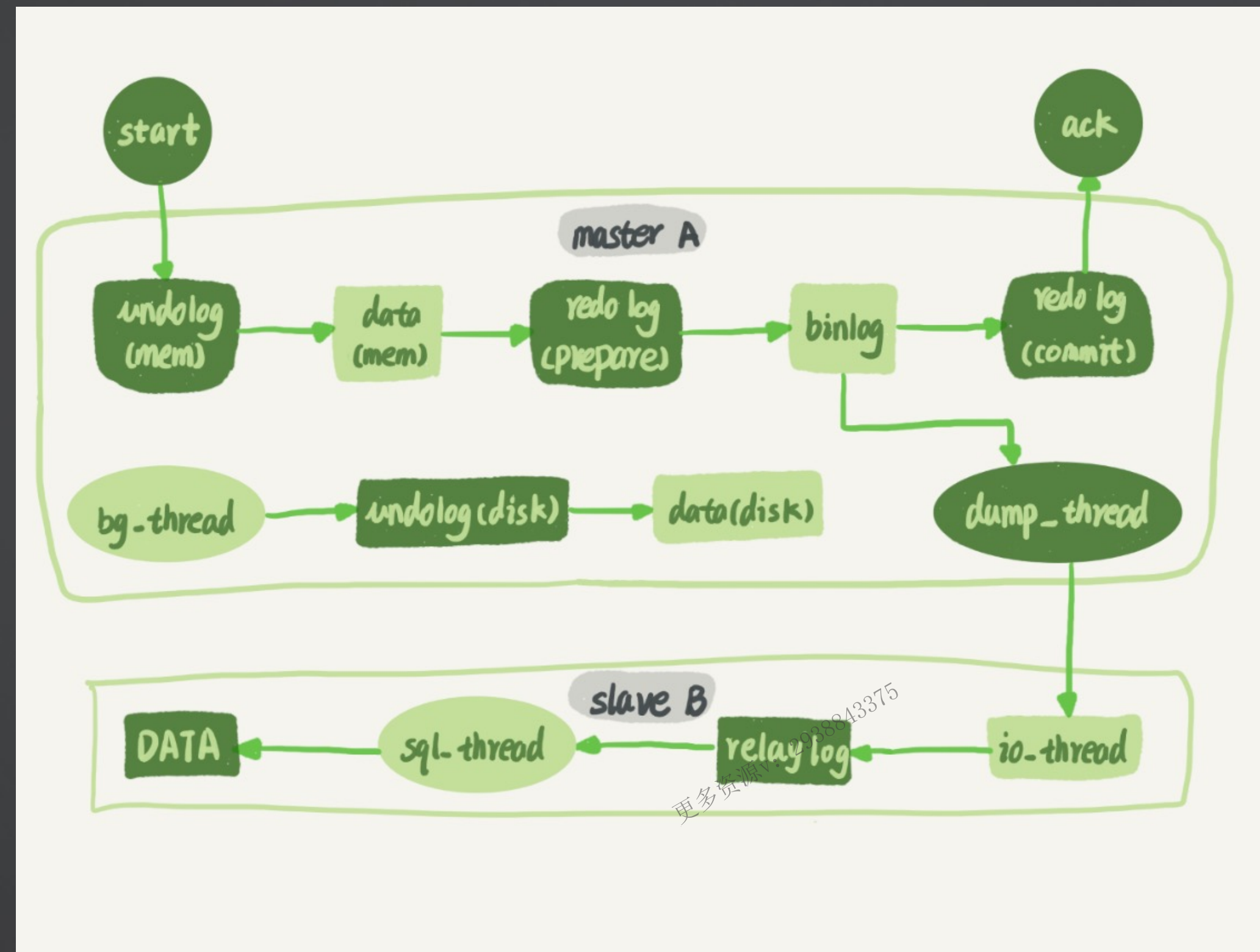
binlog 是让 MySQL 成为最流行开源数据库的重要原因



High Availability



主备流程图



影响主库性能的 binlog 参数

binlog_cache_size

binlog_checksum

binlog_group_commit_sync_delay

binlog_group_commit_sync_no_delay_count

binlog_order_commits

binlog_row_image

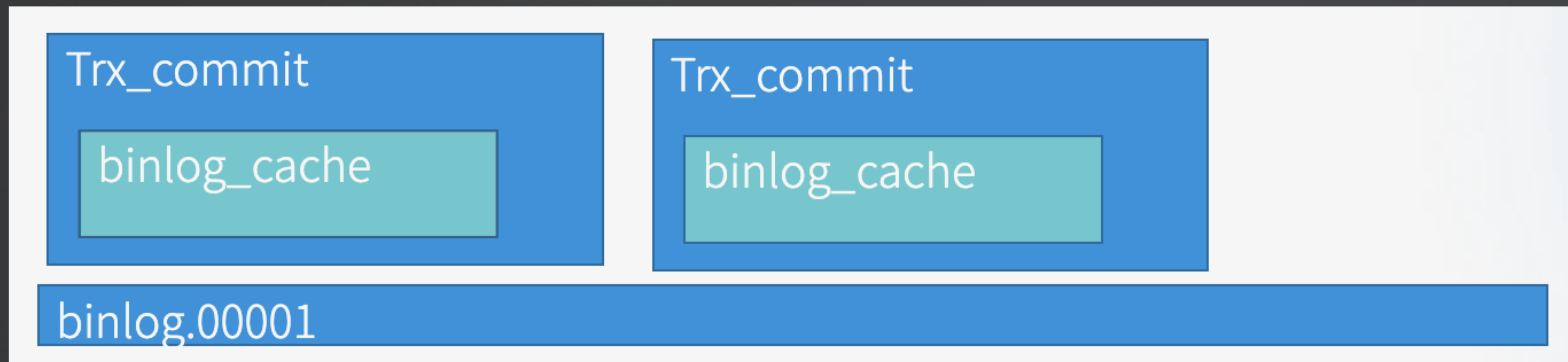
semi-sync...

影响主库性能的 binlog 参数 binlog_checksum

```
BEGIN
/*!*/;
# at 942
#240811 6:42:28 server id 1 end_log_pos 993 CRC32 0x320922d7 Table_map: `test`.`t1` mapped to number 174
# has_generated_invisible_primary_key=0
# at 993
#240811 6:42:28 server id 1 end_log_pos 1096 CRC32 0xb3abe6ca Write_rows: table id 174 flags: STMT_END_F

BINLOG '
10y3ZhMBAAAAMwAAA0EDAAAAAK4AAAAAAAEABHRLc3QAAAnQxAAQDAwMDAAwBAQDXIgky
10y3Zh4BAAAZwAAAEgEAAAAAK4AAAAAAAEAAgAE/wAFAAAAAQAAAAEAAAABAAAAAYAAAACAAAA
AgAAAAIAAAAABwAAAAMAAAADAAAAAwAAAAIAAAAABAAAAAQAAAAEAAAAYuarsw==
'/*!*/;
### INSERT INTO `test`.`t1`
### SET
### @1=5 /* INT meta=0 nullable=0 is_null=0 */
### @2=1 /* INT meta=0 nullable=0 is_null=0 */
### @3=1 /* INT meta=0 nullable=1 is_null=0 */
### @4=1 /* INT meta=0 nullable=1 is_null=0 */
### INSERT INTO `test`.`t1`
```

影响主库性能的 binlog 参数 binlog_cache_size



影响主库性能的 binlog 参数 sync_binlog

sync_binlog=0 的时候，表示每次提交事务都只 write，不 fsync；

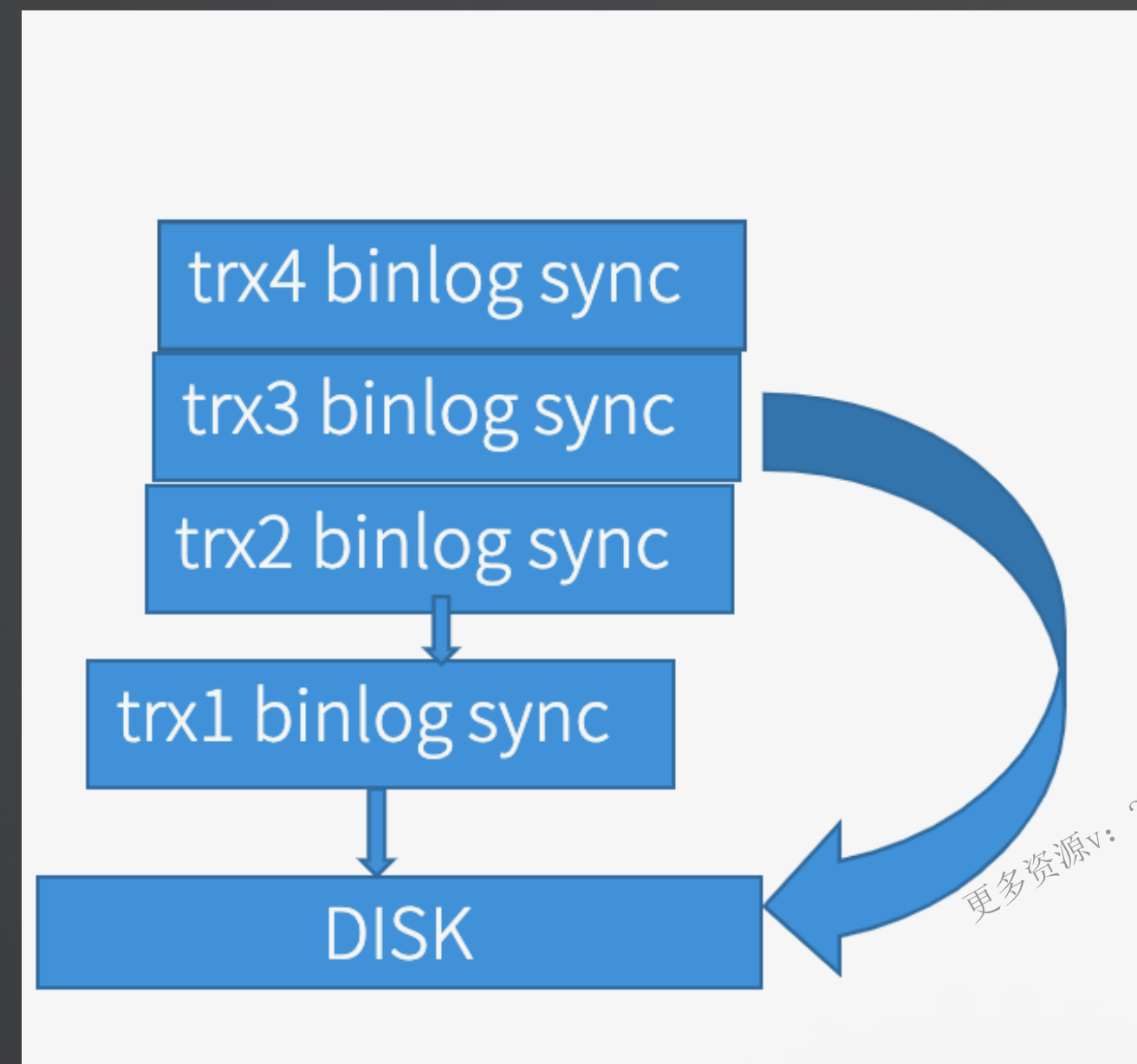
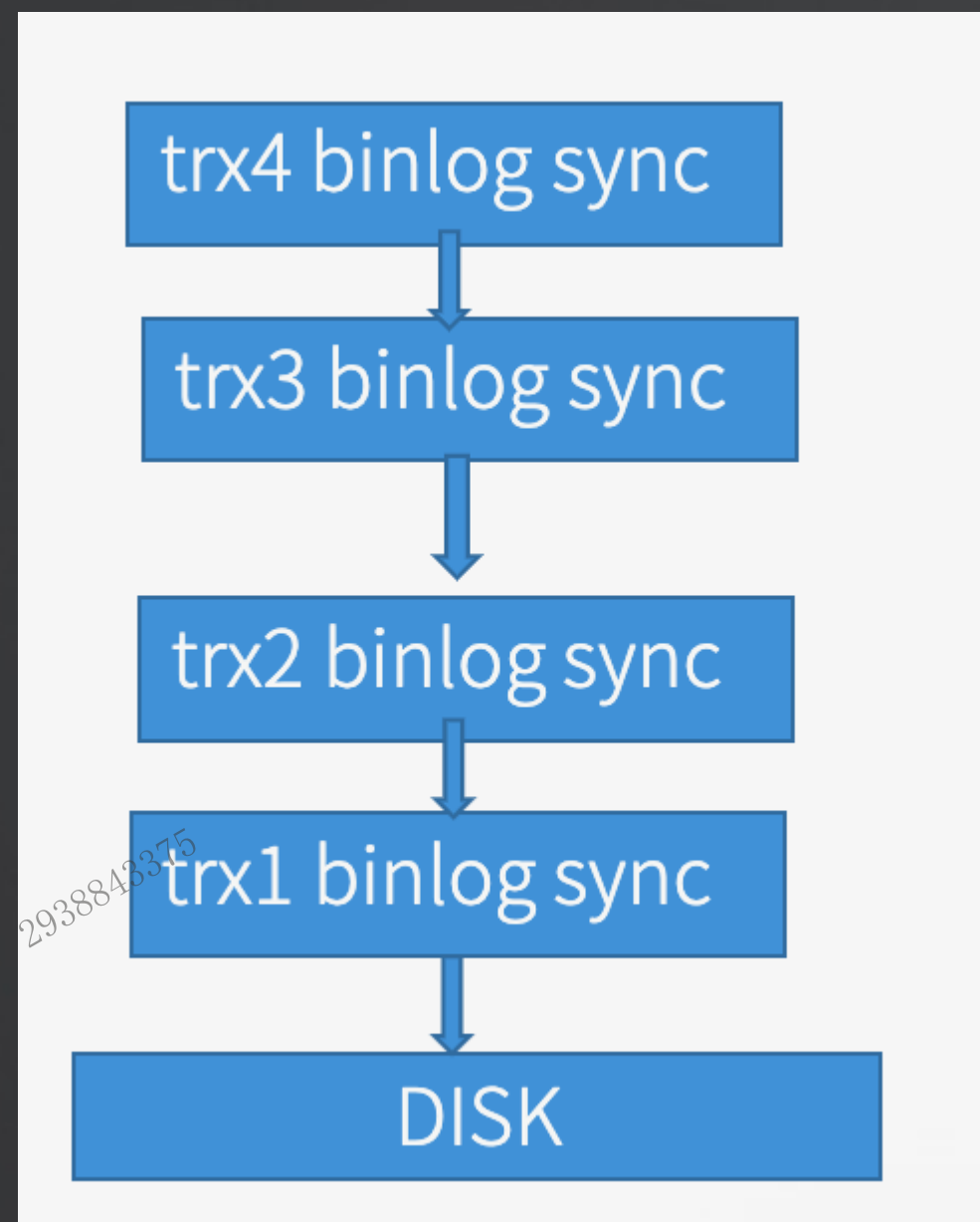
sync_binlog=1 的时候，表示每次提交事务都会执行 fsync；

sync_binlog=N(N>1) 的时候，表示每次提交事务都 write，但累积 N 个事务后才 fsync。

影响主库性能的 binlog 参数 group_commit

binlog_group_commit_sync_delay /

binlog_group_commit_sync_no_delay_count



影响主库性能的 binlog 参数 group_commit

binlog_group_commit_sync_delay /

binlog_group_commit_sync_no_delay_count

```
mysql> set global binlog_group_commit_sync_delay=1000000;
Query OK, 0 rows affected (0.00 sec)

mysql> create table t1(id int primary key auto_increment, a int not null , b int, c int , index (a), index bc(b,c));
Query OK, 0 rows affected (1.03 sec)

mysql> insert into t1(a,b,c) values(1,1,1),(2,2,2),(3,3,3),(4,4,4);
Query OK, 4 rows affected (1.01 sec)
Records: 4 Duplicates: 0 Warnings: 0
```


课堂练习

`binlog_group_commit_sync_delay=10ms`

`binlog_group_commit_sync_no_delay_count=10`

与

`sync_binlog=10`

的区别

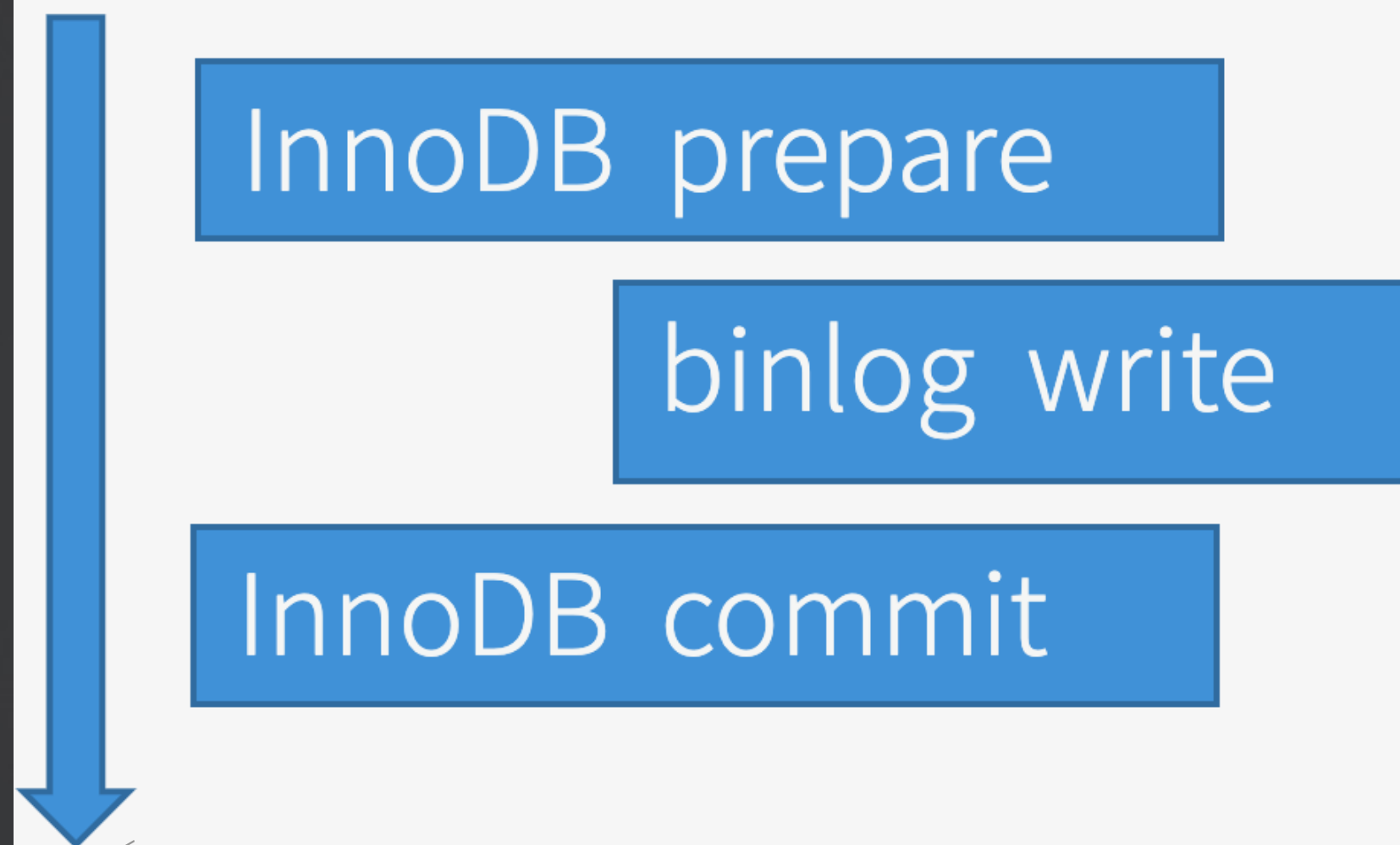
更多资源v: 2938843375

更多资源v: 2938843375

影响主库性能的 binlog 参数 binlog_order_commits

相关概念：

两阶段提交



影响主库性能的 binlog 参数 binlog_row_image

full (Log all columns)

minimal (Log only changed columns, and columns needed to identify rows)

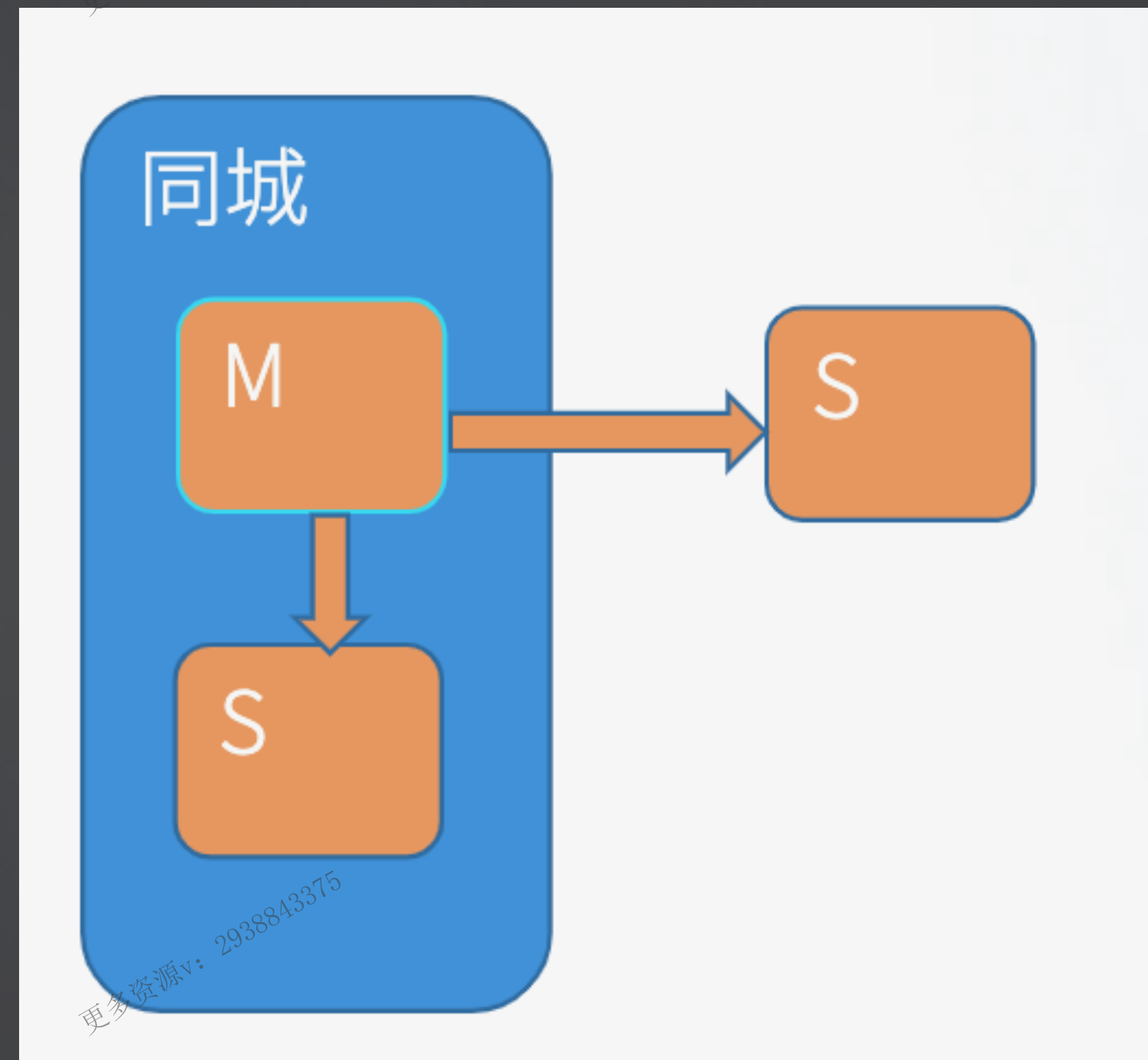
noblob (Log all columns, except for unneeded BLOB and TEXT columns)

```
BINLOG '
2Qy4ZhMBAAAAMwAAANACAAAAAKoAAAAAAAEABHRlc3QAAAnQxAAQDAwMDAA4BAQDP3rLI
2Qy4Zh8BAAAAOAAAAAgDAAAAAKoAAAAAAEEAgAEAQgAAQAAAAADAAAAAIAAAAABgAAABl446Y=
'/*!*/;
### UPDATE `test`.`t1`
### WHERE
###   @1=1 /* INT meta=0 nullable=0 is_null=0 */
### SET
###   @4=3 /* INT meta=0 nullable=1 is_null=0 */
### UPDATE `test`.`t1`
### WHERE
###   @1=2 /* INT meta=0 nullable=0 is_null=0 */
### SET
###   @4=6 /* INT meta=0 nullable=1 is_null=0 */
# at 776
#240811  8:59:05 server id 1  end_log_pos 807 CRC32 0x21d773e6  Xid = 232
COMMIT/*!*/;
```


影响主库性能的 binlog 参数 binlog_row_image

```
BINLOG '
84S4ZhMBAAAMwAAADIFAAAAKoAAAAAAAEABHRlc3QAAnQxAAQDAwMDAA4BAQCfx+Pe
84S4Zh8BAAAANAEEAGYGAAAAAKoAAAAAAEEAagAE//8AAQAAAAEAAAABAAAAQAAAAABAAAAQAA
AAEAAAACAAAAAAIAAAACAAAAAGAAAAIAAAAAAGAAAAIAAAACAAAAAwAAAAADAAAAwAAAAAMAAAD
AAAAAMAAAAADAAAAAwAAAAQAAAAABAAAAQAAAAEAAAABAAAAAAEAAAABAAAAAQAAAAFAAAAAUA
AAABAAAAAQAAAAEAAAABQAAAAEAAAABAAAAAgAAAAAGAAAAAgAAAAIAAAACAAAAAAAYAAACAAAA
AgAAAAAMAAAAABwAAAAAMAAADAAAAAwAAAAHAAAAAwAAAAAMAAAEAAAAAGAAAAEAAAABAAAAQA
AAAACAAAAAQAAAAEAAAABQAAAPI1MmM=
'/*!*/;
### UPDATE `test`.`t1`
### WHERE
###   @1=1 /* INT meta=0 nullable=0 is_null=0 */
###   @2=1 /* INT meta=0 nullable=1 is_null=0 */
###   @3=1 /* INT meta=0 nullable=1 is_null=0 */
###   @4=1 /* INT meta=0 nullable=1 is_null=0 */
### SET
###   @1=1 /* INT meta=0 nullable=0 is_null=0 */
###   @2=1 /* INT meta=0 nullable=1 is_null=0 */
###   @3=1 /* INT meta=0 nullable=1 is_null=0 */
###   @4=2 /* INT meta=0 nullable=1 is_null=0 */
### UPDATE `test`.`t1`
### WHERE
###   @1=2 /* INT meta=0 nullable=0 is_null=0 */
###   @2=2 /* INT meta=0 nullable=1 is_null=0 */
###   @3=2 /* INT meta=0 nullable=1 is_null=0 */
###   @4=2 /* INT meta=0 nullable=1 is_null=0 */
### SET
###   @1=2 /* INT meta=0 nullable=0 is_null=0 */
###   @2=2 /* INT meta=0 nullable=1 is_null=0 */
###   @3=2 /* INT meta=0 nullable=1 is_null=0 */
###   @4=3 /* INT meta=0 nullable=1 is null=0 */
```

影响主库性能的 binlog 参数 semi-sync



课堂练习

在主库单线程执行纯 insert 语句的情况下，开启下面这些参数，对性能（TPS）的影响，从高到低是？

1. binlog_checksum (on vs off)
2. sync_binlog (1vs 0)
3. binlog_group_commit_sync_delay (1000ms vs 0)
4. binlog_order_commits (on vs off)
5. binlog_row_image (full vs minimal)
6. semi-sync (on vs off)

目录

1 binlog 和主库性能

2 binlog 和主备同步性能

3 binlog 和数据恢复速度

更多资源v: 2938843375

更多资源v: 2938843375

binlog 和 备库应用速度 - 并行复制策略

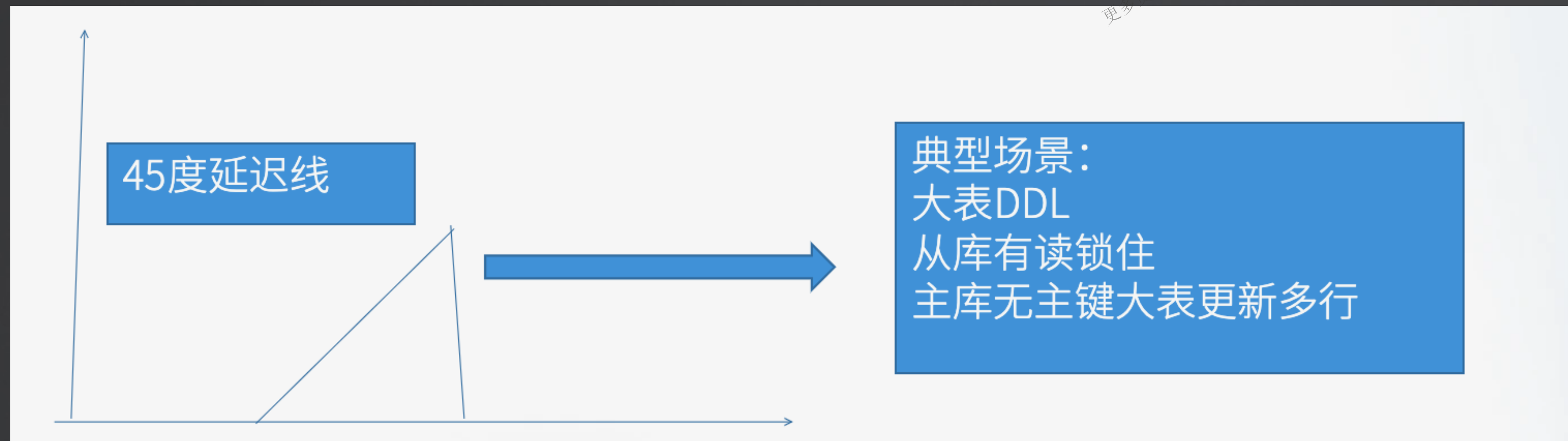
replica_sql_verify_checksum

semi_sync

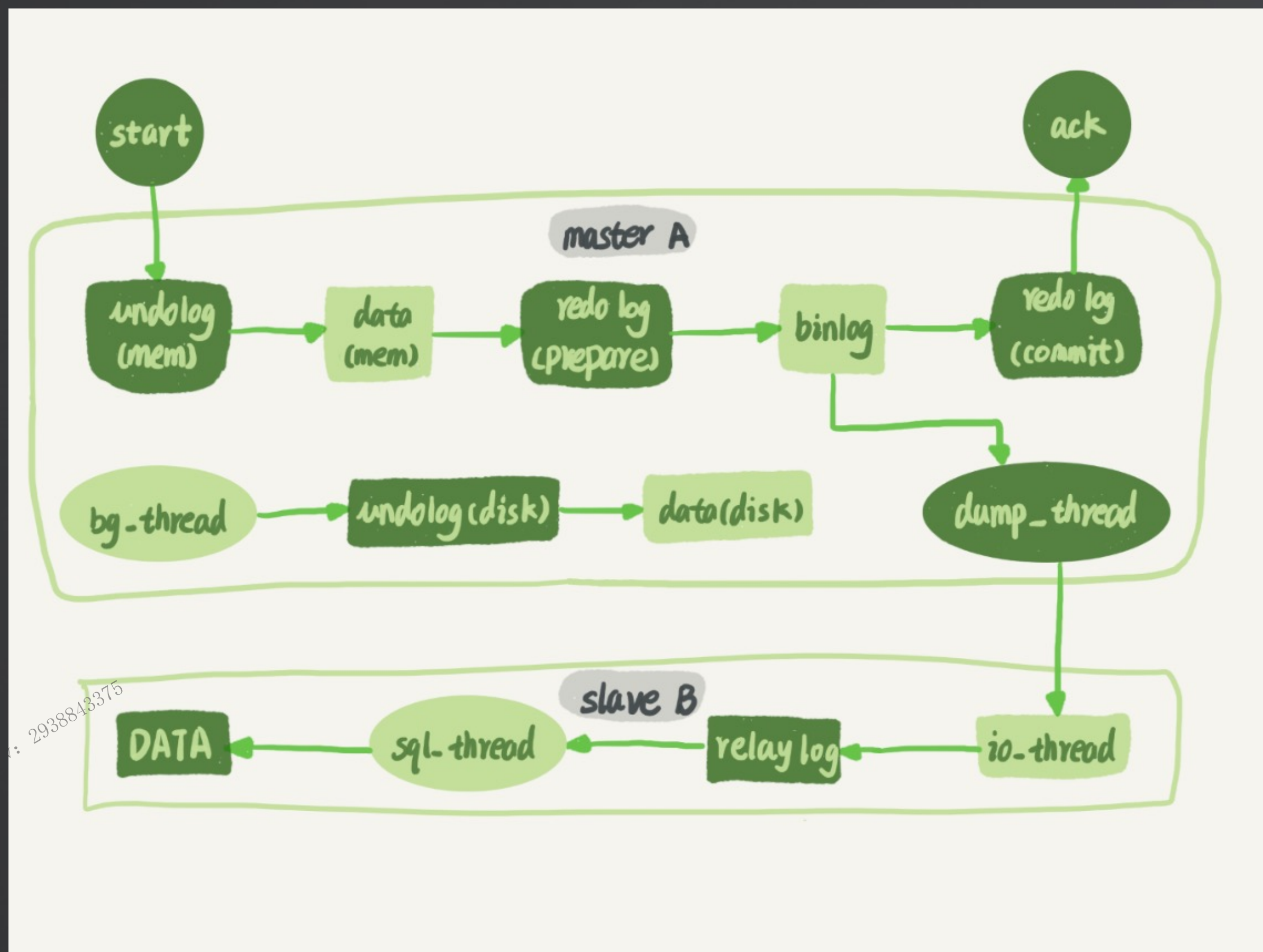
主备延迟常见原因：

1. 机器负载
2. 备库读压力
3. 大事务
4. 并发度

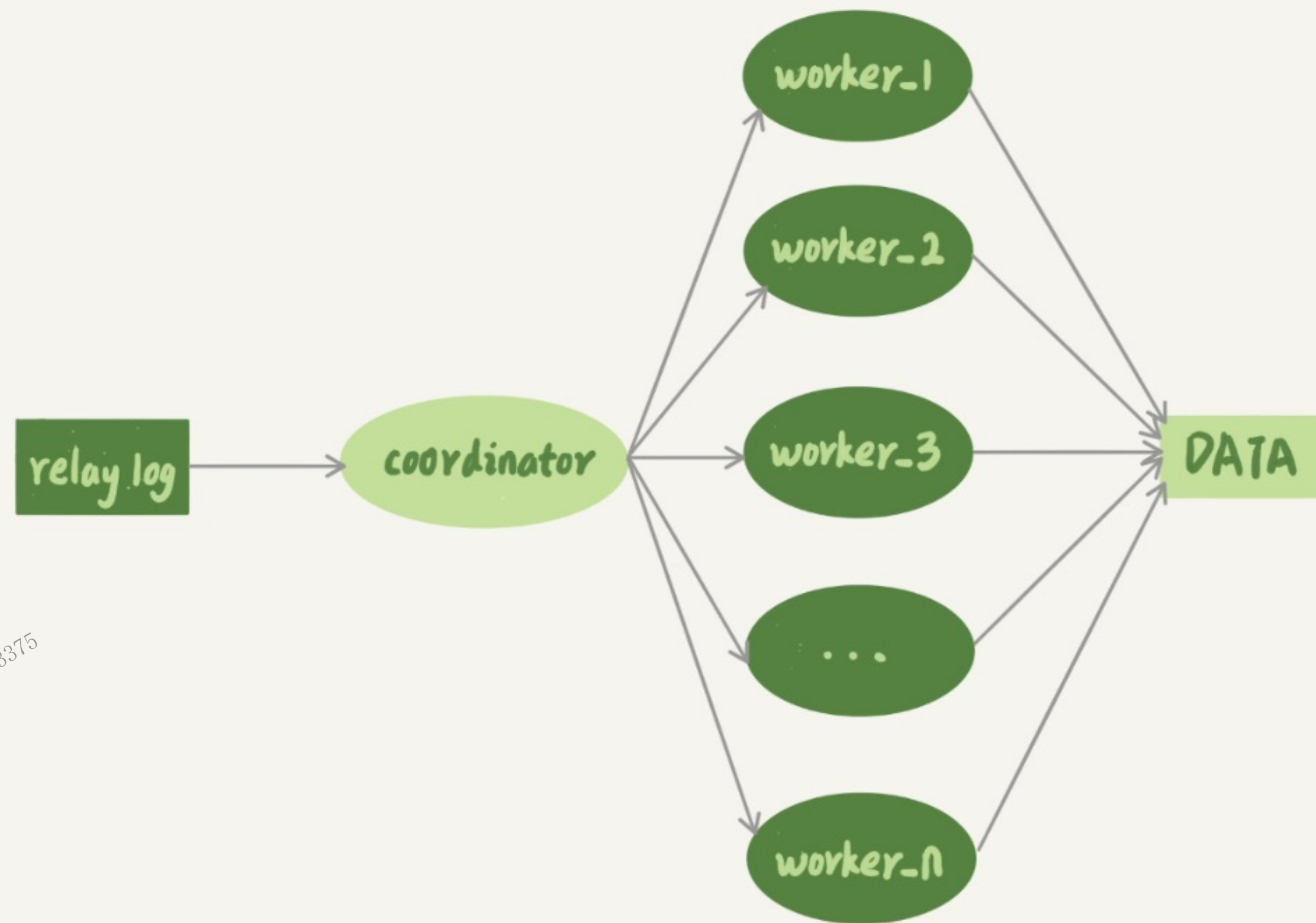
binlog 和 备库应用速度 - 大事务



binlog 和 备库应用速度 - 主备流程图



binlog 和 备库应用速度 - 并行复制策略



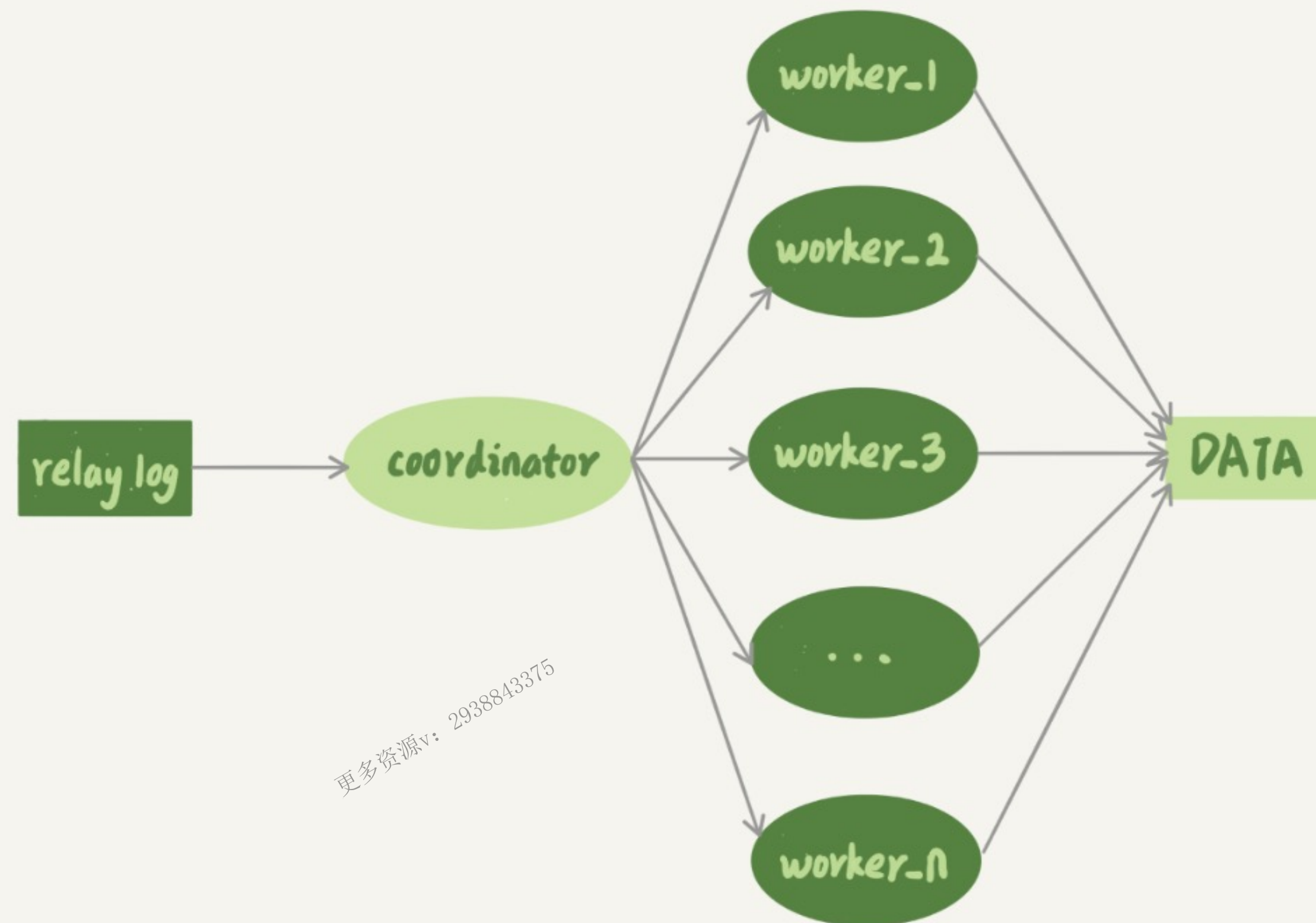
binlog 和 备库应用速度 - 并行复制策略

5.5 按表
按行

5.6 按库

5.7 LOGICAL_CLOCK

5.7.22 WRISSETSET



更多资源v: 2938843375

Q&A

更多资源v: 2938843375

THANKS