

MySQL 复制选型

吴炳锡
知数堂联合创始人

吴炳锡

- 知数堂联合创始人及MySQL课程讲师
- 企业级MySQL架构设计服务顾问
- 曾任职飞信（新媒传信）数据库架构师，某彩票公司数据库DBA，10年+MySQL从业经验
- 专注MySQL优化，高可用，多IDC多活架构



复制选型

- MySQL复制技术的现状
- 三种binlog格式对复制的影响
- GTID复制的特点
- 传统复制及特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- 并行复制
- replication crash recover
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- group replication的实现
- 复制选择推荐

MySQL 复制技术的现状

- MySQL 4.0 开始支持复制 (2003年 statement)
- MySQL 5.1 基于row-based复制 (2008年 SBR,MBR,RBR)
- MySQL 5.5 semi-sync replication
- MySQL 5.6 GTID , delay replication , 基于库的并行复制
- MySQL 5.7 lossless replication, mutli-source repliation , group-commit, 基于事务的并行复制, group replication (2016 年)

MySQL 复制技术的现状

- 当前网搜文章关于复制绝大多数还是讲传统复制
- 实质上现在是GTID+ROW复制的天下
- 并行复制的在生产中大量的使用
- 多源复制在统计分析环境中使用比较多

复制现状

- 异步复制
 - 传统类型
- 同步复制
 - PXC, MySQL Group Replication
- 半同步复制
 - 增强半同步复制（推荐MySQL 5.7）
 - 半同步复制

三种Binlog格式对复制的影响

- SBR
 - binlog_format='statement'
 - 记录操作的原始格式，最容易造成主从数据不一致
- MIXED
 - binlog_format='mixed'
 - 一个过渡产品，从5.1里出现。如果你还在使用这个格式，请改成： row
- RBR
 - binlog_format=row
 - 现在主要使用的格式

三种Binlog格式对复制的影响

- SBR 模式
 - update,delete 如果没有合适的索引，容易造成从库延迟严重
 - 数据维护更新，删除行比较多的情况下使用
 - pt-table-checksum/pt-table-sync
- RBR
 - 现在使用的主流模式，也是趋势

三种Binlog格式对复制的影响

- RBR模式从库对索引的使用



大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- Replication Crash Recover
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

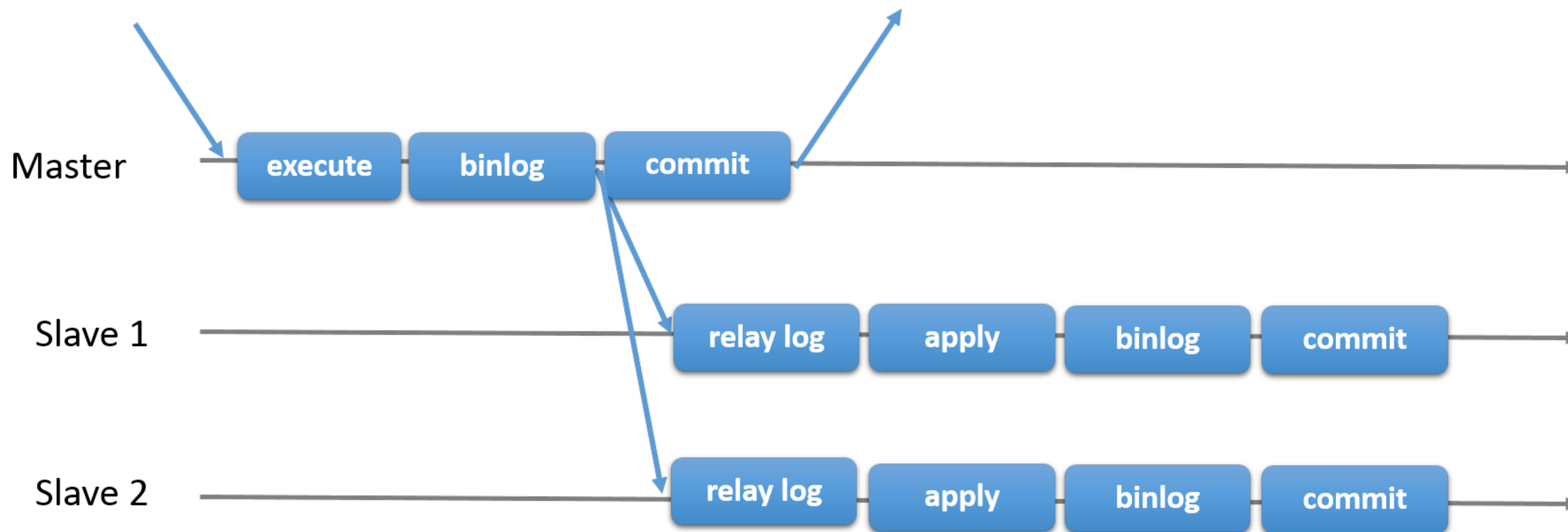
传统复制的特点

- 基于binlog模型
 - IO-Thread
 - master_log_file
 - read_master_log_pos
 - SQL-Thread
 - relay_master_log_file
 - execute_master_log_pos
- second_behind_master
- relay log中当前执行的timestamp值和最新获得的event的timestamp的差
- 同步完成：
 - master_log_file==relay_master_log_file &&
read_master_log_pos==execute_master_log_pos

GTID复制的特点

- MySQL 5.6 在Binlog里引入
- 全局事务ID，每个事跟在提交时创建。
 - GTID=server-uuid:N
 - gtid_set -> gtid_executed
 - gtid_purge
- 特点： 这个事务是谁产生，产生多少事务，非常直接的标识出来

复制实现



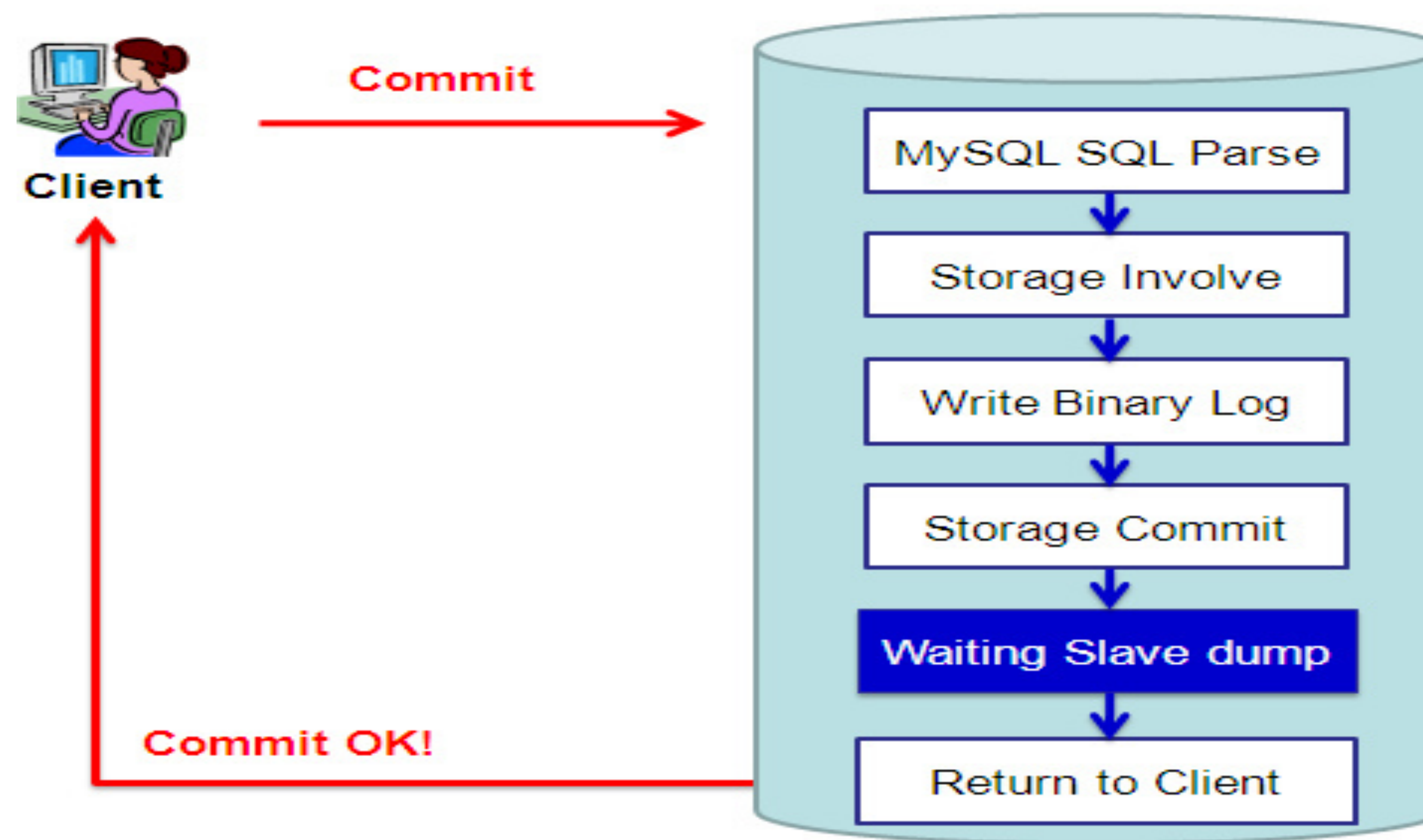
大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- 并行复制
- Replication Crash Recover
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

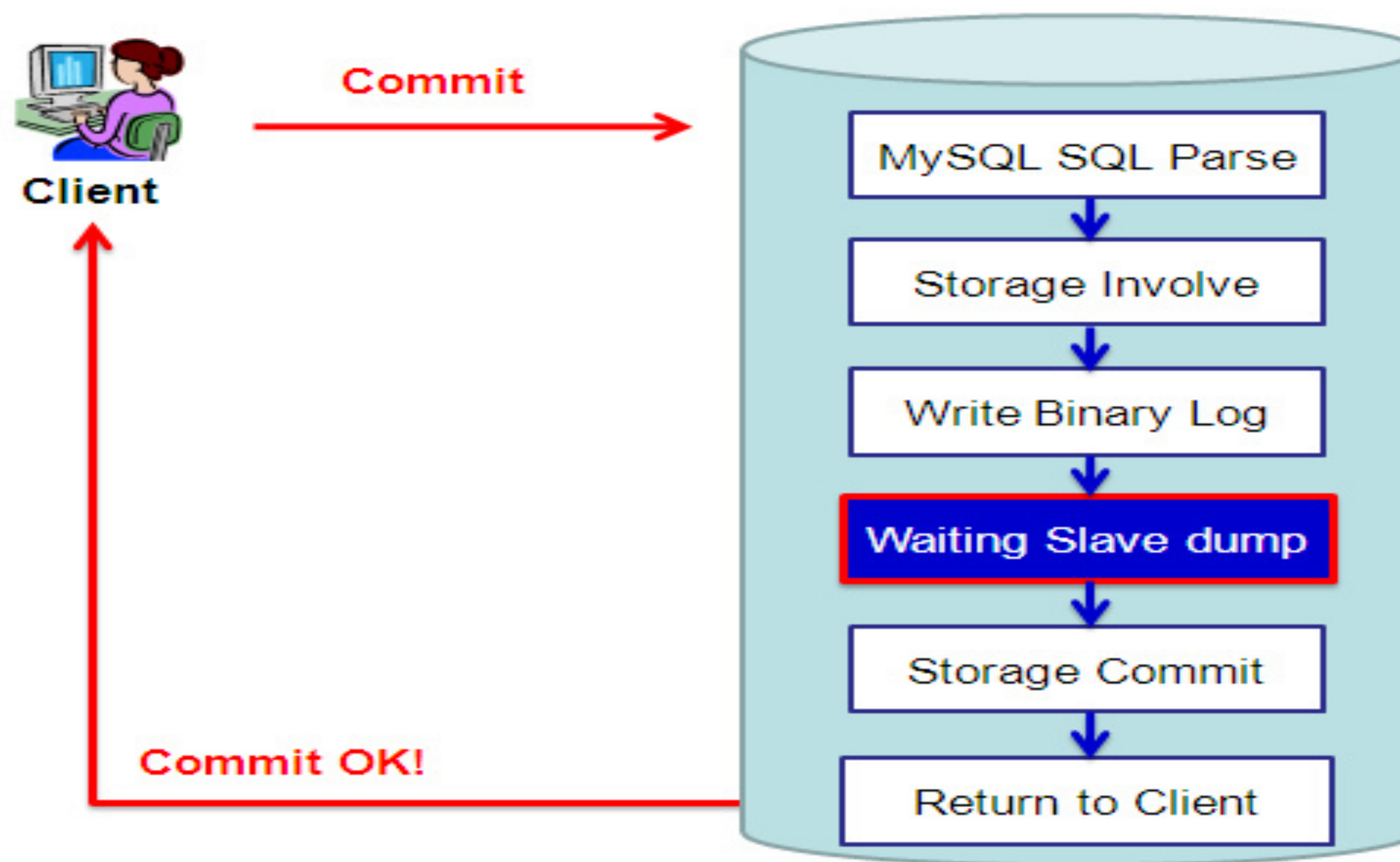
Binlog & 引擎写入

- Binlog/Innodb XA
 - Innodb Prepare;
 - write/sync Binlog;
 - Innodb Commit (commit后释放prepare_commit_mutex)

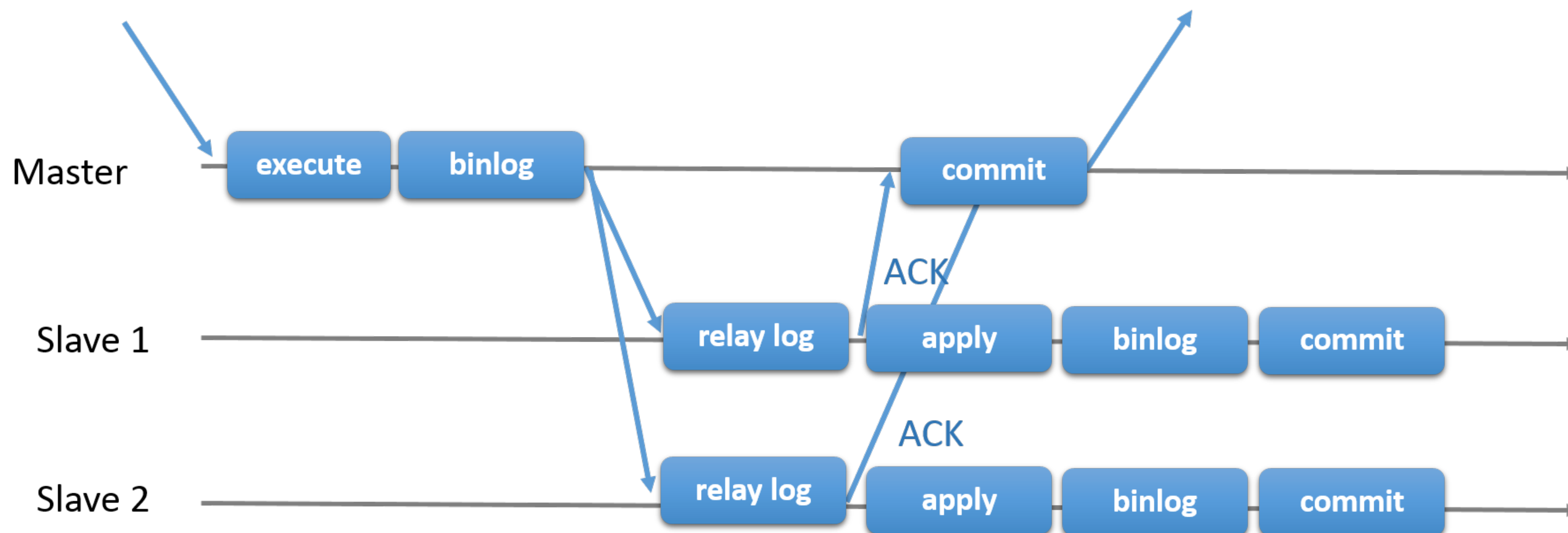
半同步复制



增强半同步



增强半同步



增强半同步

- 增强半同步 Master Crash Recovery
 - 扫描最后一个Binlog 提取其中Xid
 - 扫描InnoDB redo维持状态在Prepare的事务链表，和Binlog中的Xid比较，如果Binlog中存在，则提交，否回滚事务。
- **提交到redo中处于Prepare，并且写入Binlog的，就可以从崩溃中恢复事务**

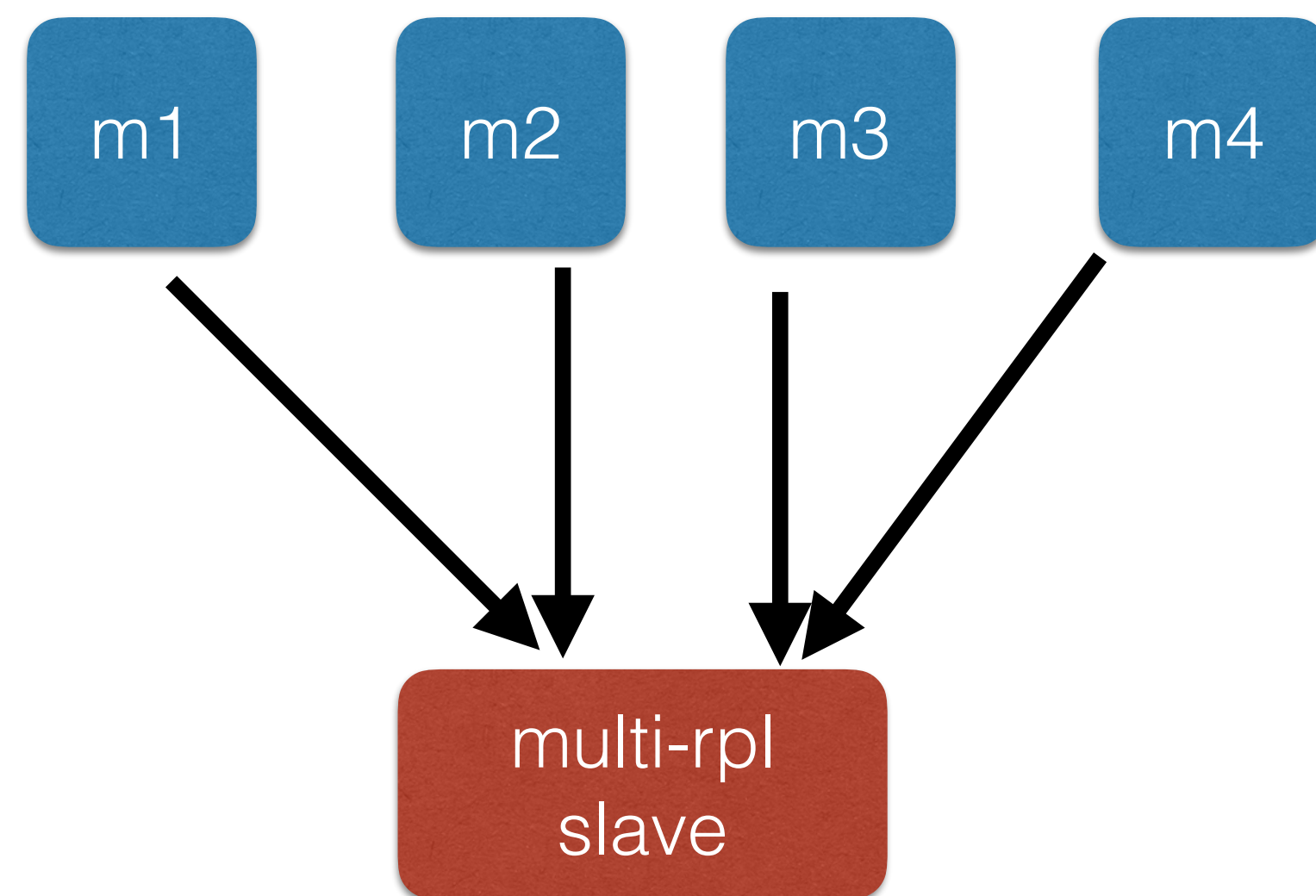
半同步配置参数

- `rpl_semi_sync_master_wait_point=AFTER_SYNC|AFTER_COMMIT`
 - AFTER_SYNC
 - AFTER_COMMIT
- `rpl_semi_sync_master_timeout=N`
 - 该值设置需要讨论
 - 重要生产环境中，建议无穷大
 - 用于故障恢复的从库，或利用备份数据新创建的从库，建议以异步方式追上同步后在改成半同步

大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- Replication Crash Recover
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

多源复制



- 限制
- 基于GTID
- RBR
- 对于添加节点同步，只能用逻辑备份及恢复
- 多源复制中的DB，用于恢复线上库，最好的办法也是逻辑备份及恢复

延迟复制

- change master to master_delay=N;
- binlog传输到从库中的relay_log，但需要延迟N秒执行。默认是0
- 使用场景：
 - 特别大的库备份
 - 给自己一个拯救世界的机会
 - 测试，估计让从库延迟多少秒，让连接不请求过来

大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- 并行复制
- Replication Crash Safe
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

binlog group commit

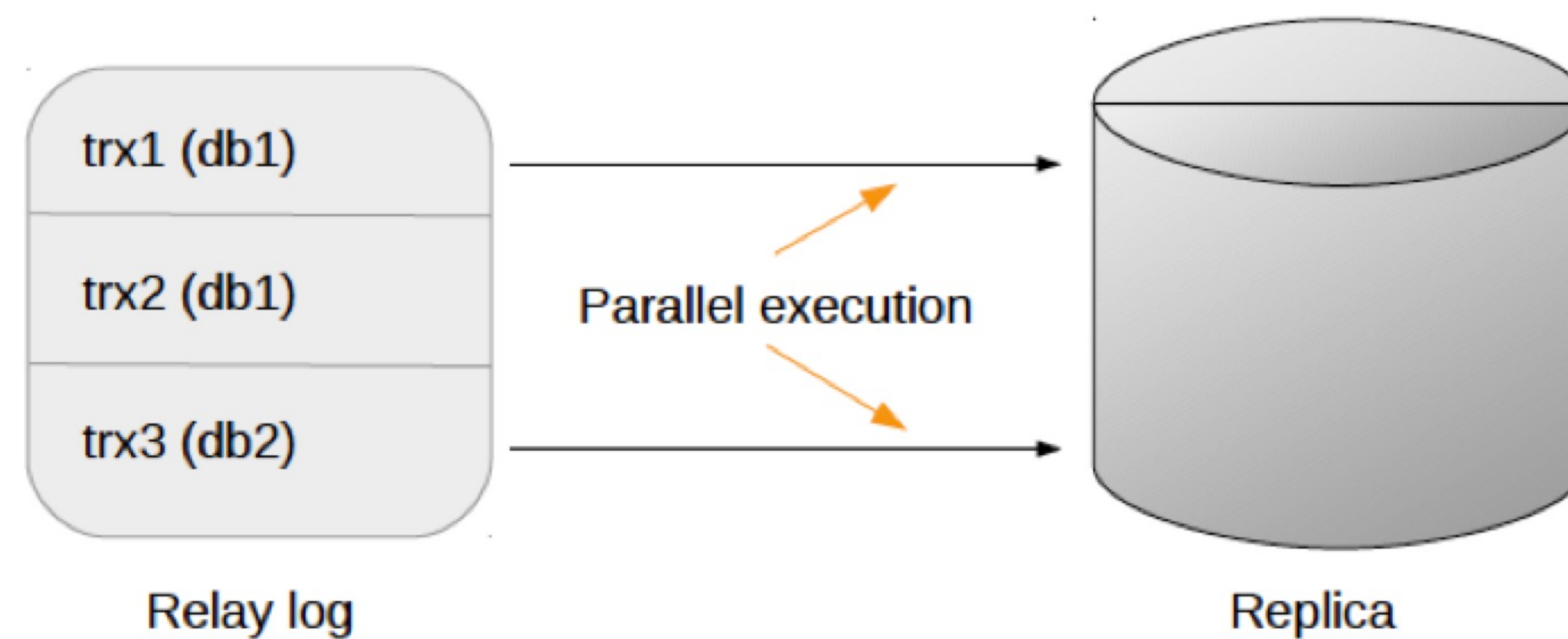
- `binlog_group_commit_sync_delay=0 -> 1000000 microseconds`
- `binlog_group_commit_no_delay_count= 0 - 1000000` ↑
- 启用：
 - `innodb_flush_log_at_trx_commit=1`
 - `sync_binlog=1`
 - `binlog_order_commits = On`

Replication Crash Safe

- 参数
 - relay_log_info_repository=Table
 - sync_relay_log_info=1
 - relay_log_recovery=ON
- 作用：
 - 从库故障启动后，利用sql_thread执行到位置恢复IO_thread的相关从信息，从主库上重新获取新的日志

并行复制

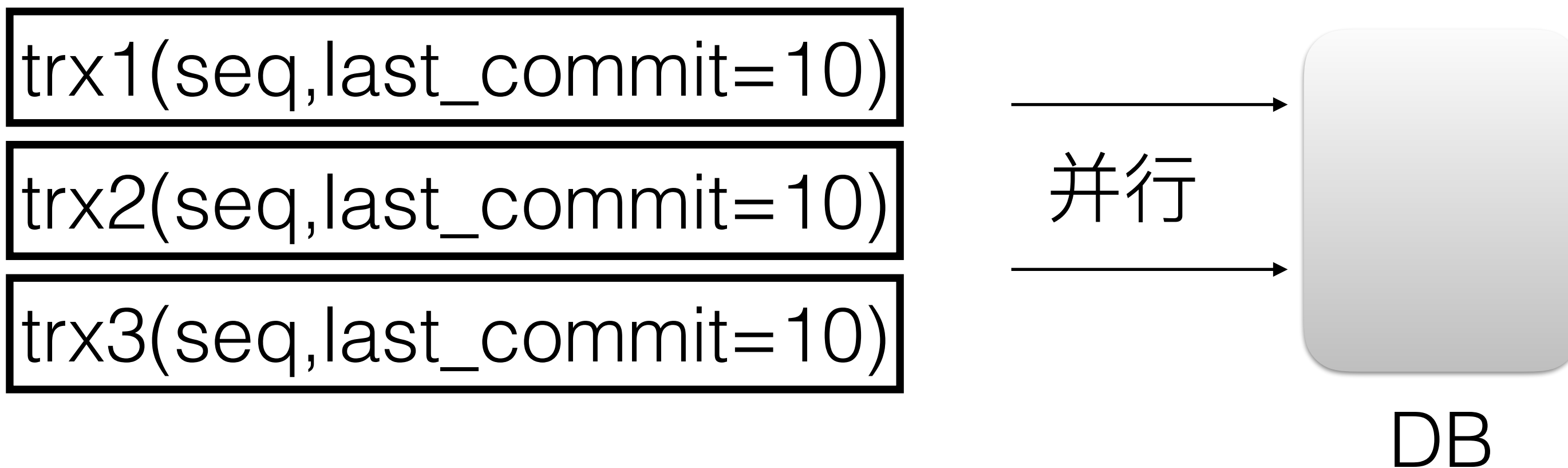
- 基于库级别的



并行复制

- 基于事务级别

基于binlog group commit,在主库上同一时间操作的SQL, 在从库也进行并发



Replication Crash Safe

- show slave status输出和slave_relay_log_info内容不一致?

sync_relay_log_info	relay_log_info_repository		
	FILE	TABLE	
		Transactional	Nontransactional
$N > 0$	The slave synchronizes its <code>relay-log.info</code> file to disk (using <code>fdatasync()</code>) after every N transactions.	The table is updated after each transaction. (N is effectively ignored.)	The table is updated after every N events.
0	The MySQL server performs no synchronization of the <code>relay-log.info</code> file to disk; instead, the server relies on the operating system to flush its contents periodically as with any other file.		The table is never updated.

大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- Replication Crash Safe
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

复制过滤规则

- MySQL 5.7 可以动态的更改从库的过滤规则

- change replication filter

| REPLICATE_DO_DB = (**db_list**)

| REPLICATE_IGNORE_DB = (**db_list**)

| REPLICATE_DO_TABLE = (**tbl_list**)

| REPLICATE_IGNORE_TABLE = (**tbl_list**)

| REPLICATE_WILD_DO_TABLE = (**wild_tbl_list**)

| REPLICATE_WILD_IGNORE_TABLE = (**wild_tbl_list**)

| REPLICATE_REWRITE_DB = (**db_pair_list**)

建议：

- binlog_format=row
- 过滤规则在从库设置

大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- Replication Crash Safe
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

影响复制一致性因素及解决方案

- 前提使用GTID+Row格式
- 对于复制一致性要求的高的： 增强半同步
- 出现不一致的环境：
 - 除了增强半同步外，在异常重启、意外掉电时造成不一致
 - 异步复制无法保证主从数据一致性

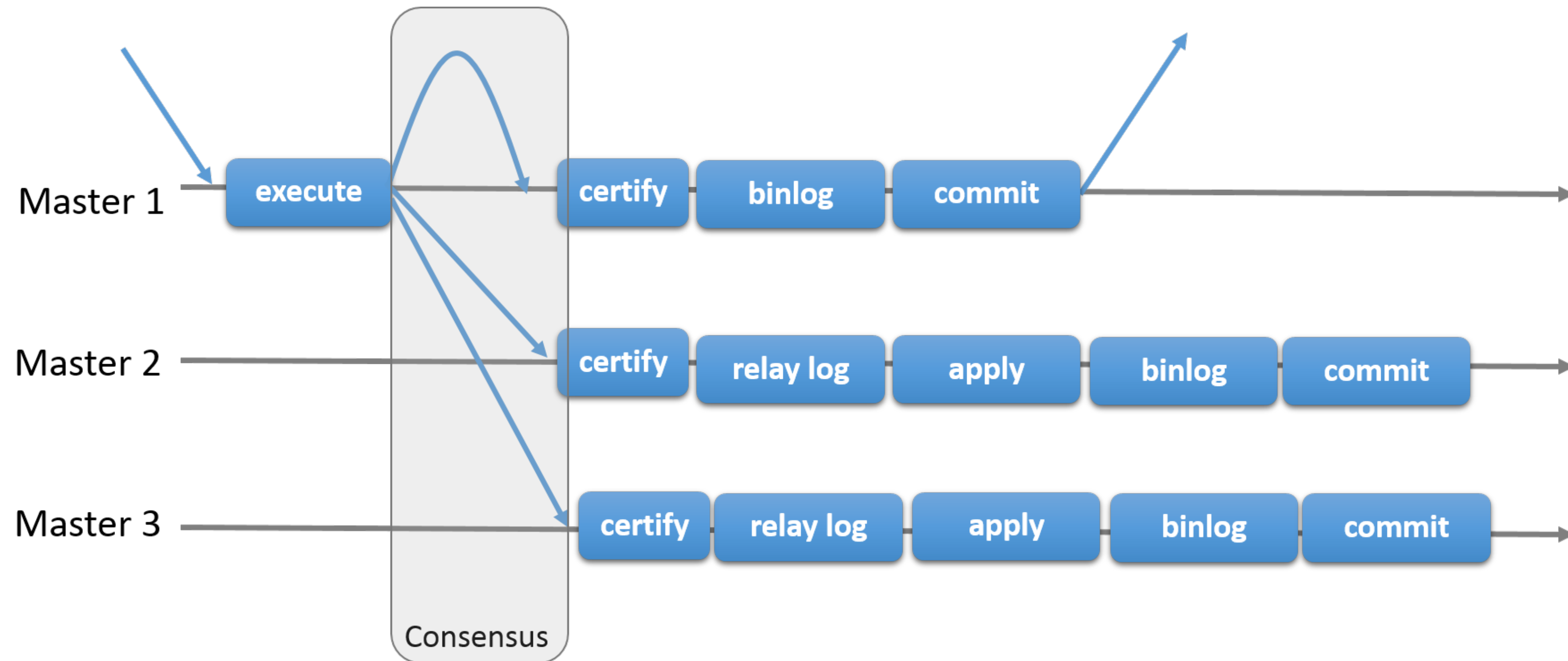
影响复制一致性因素及解决方案

- 解决方案
 - 主库： 双1设置
 - 从库： replication crash safe、repo设置为table
- 发生异常后、并定期检测&修复数据
 - pt-table-checksum
 - pt-table-sync

大纲

- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- Replication Crash Safe
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

group replication



group replication

- Group Replication实质是：
 - 二进制日志
 - 基于Row格式+全局事务标识
 - 一个通信框架+事务排序控制(atomic message delivery & total ordering of message)
- 增强半同步 & 并行复制

group replication

- Group Replication 是一个趋势，但也有不足之处
- 备份不好搞定
 - mysqldump不支持一致性备份
 - xtrabackup 备份，造成性能损失比较严重
- 新的节点加入过程对原有集群性能有影响

大纲

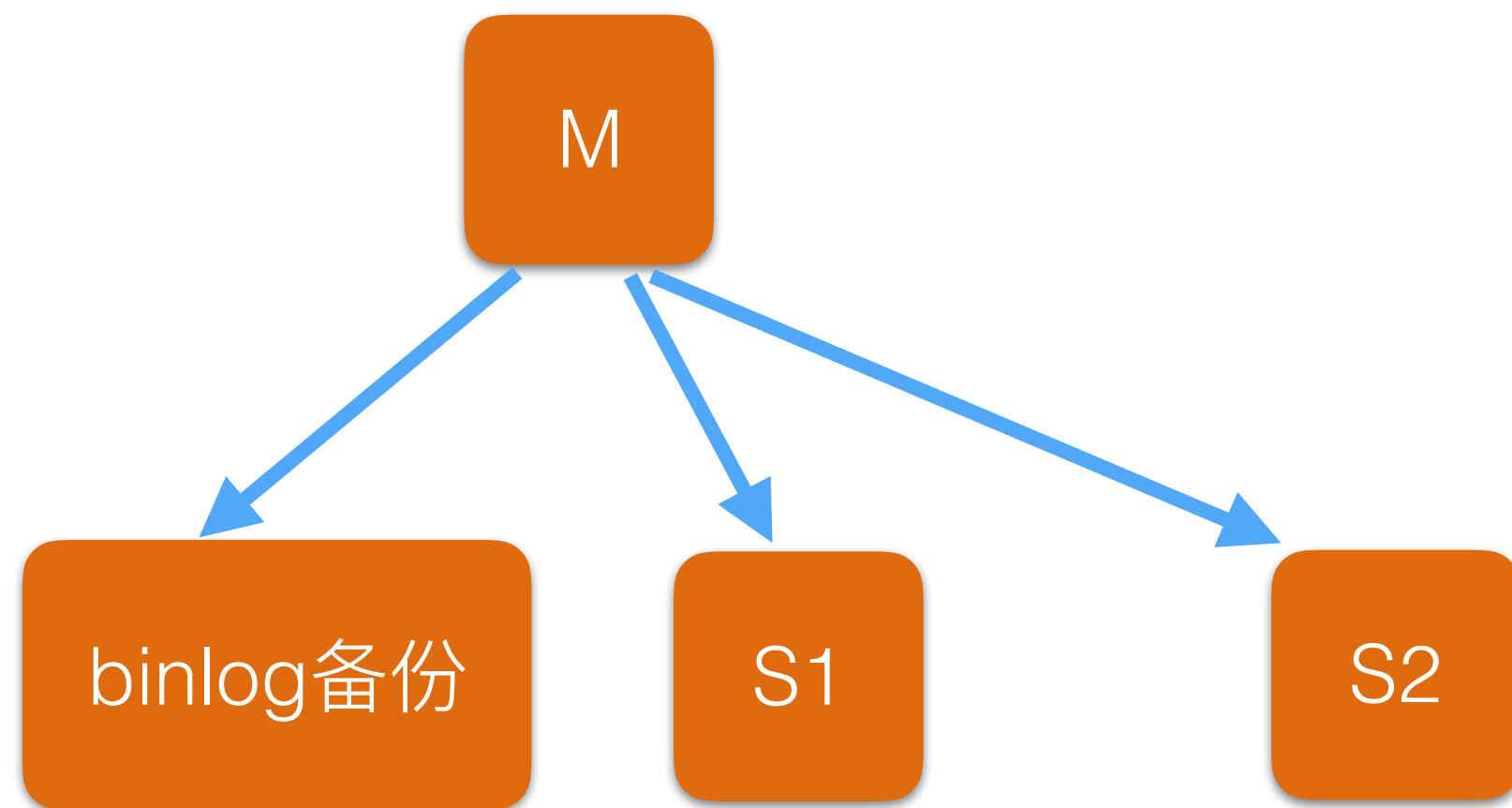
- MySQL复制技术的现状
- 三种Binlog格式对复制的影响
- 传统复制及特点
- GTID复制的特点
- 半同步复制的特点
- 增强半同步复制的特点
- 多源复制的实现
- 延迟复制
- binlog group commit
- Replication Crash Safe
- 复制过滤规则选择
- 影响复制一致性的因素及解决方案
- Group Replication的实现
- 复制选择推荐

复制选择推荐

- 推荐

- 最基础的: row+gtid
- 增强半同步/半同步/异步
- 基于多源复制
 - 基于多源复制高可用实现
- 同步复制

复制选择推荐

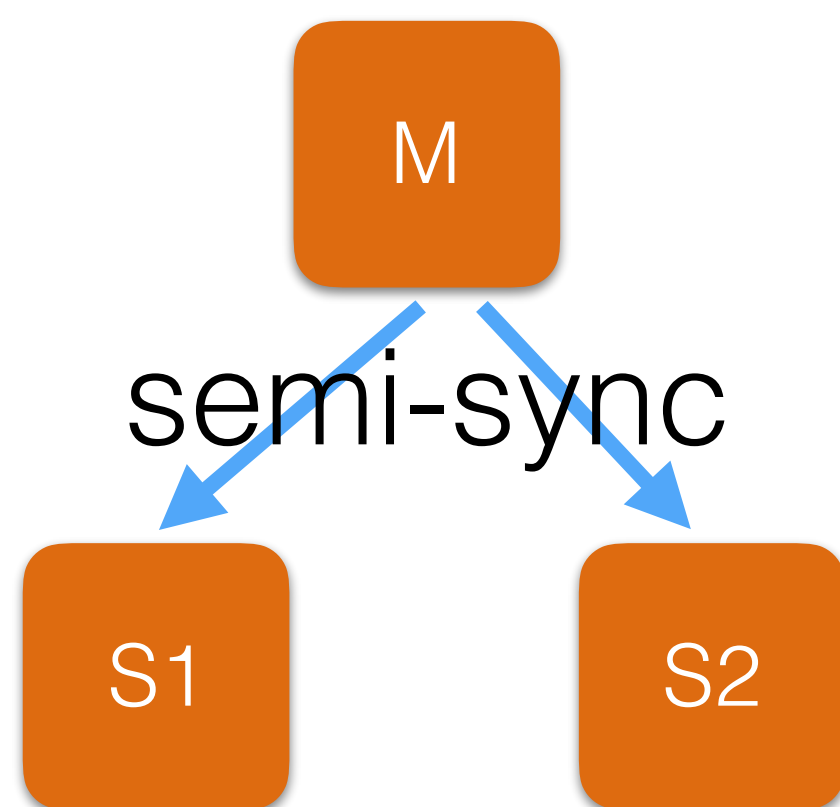


- 在该结构中如果binlog备份能保持半同步，整个结构安全度会增加很多
- 另外要考虑复制结构中单库的约束
- 整个实例的大小

复制选择推荐

- 金融交易

- 推荐MySQL 5.7的半同步



约束:

- row+gtid+半同步协议

常见问题:

- 网络不好, 切换到异步复制
- 从库网络、IO问题导致主库性能下降
- 从库故障, 利用备份还原后加入集群中, 把主库拖垮

谢谢， 希望有所帮助