
数据技术嘉年华
// Data Technology Carnival
开源 · 融合 · 数智化 — 引领数据技术发展 释放数据要素价值
中国DBA联盟
All China DBA Union
墨天轮

数据技术嘉年华

MogDB中自治异步事务提交的设计与实现

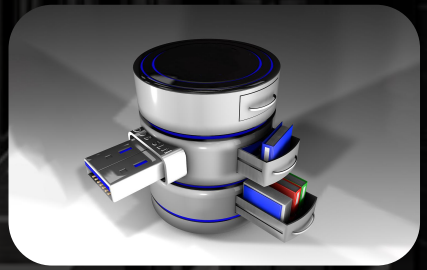
演讲人：王春玲 云和恩墨

中国DBA联盟 All China DBA Union

墨天轮

概述

在TP场景中，对高并发下事务处理的性能有较高的需求。然而openGauss中事务提交的过程中需要同步等待日志落盘，这一期间的工作线程处于空闲状态，并不能被利用去处理其他的事务。虽然在openGauss中有异步提交的实现，但该实现并不能保证数据库中途崩溃之后数据的完整性。MogDB中实现的自洽异步事务提交能够实现真正意义上的事务异步提交，在保证数据库可靠性的前提下更加充分利用CPU，提升高并发场景下事务处理处理能力，尤其在小查询的增删改操作上会有明显的体现。

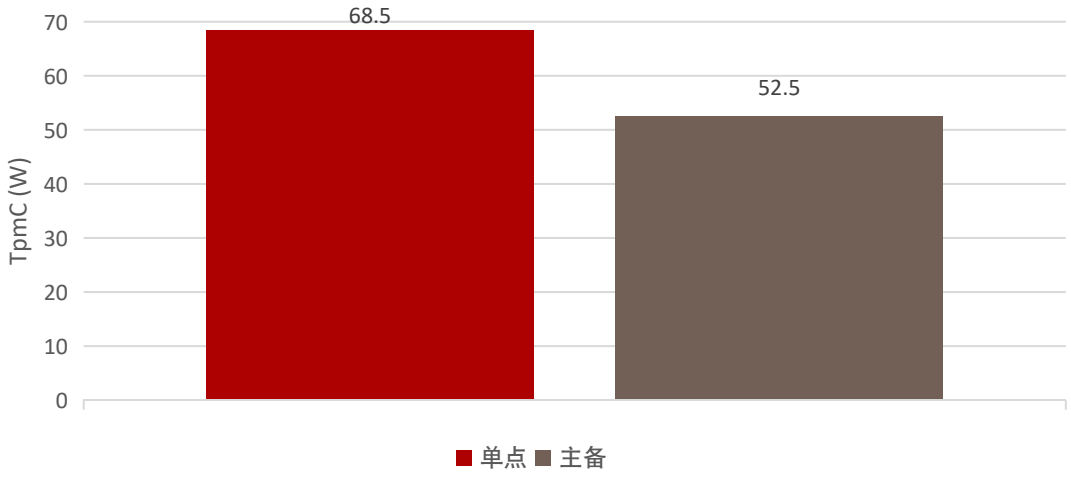


目录

CONTENTS

实现背景——问题发现

- 在TP场景中，对高并发下事务处理的性能有较高的需求。
- opengauss事务提交的时候需要同步等待日志落盘，主备模式下还需要等待日志被同步到备机，影响事务处理性能。
- 对于主备（写日志慢）场景，比单点性能下降明显：

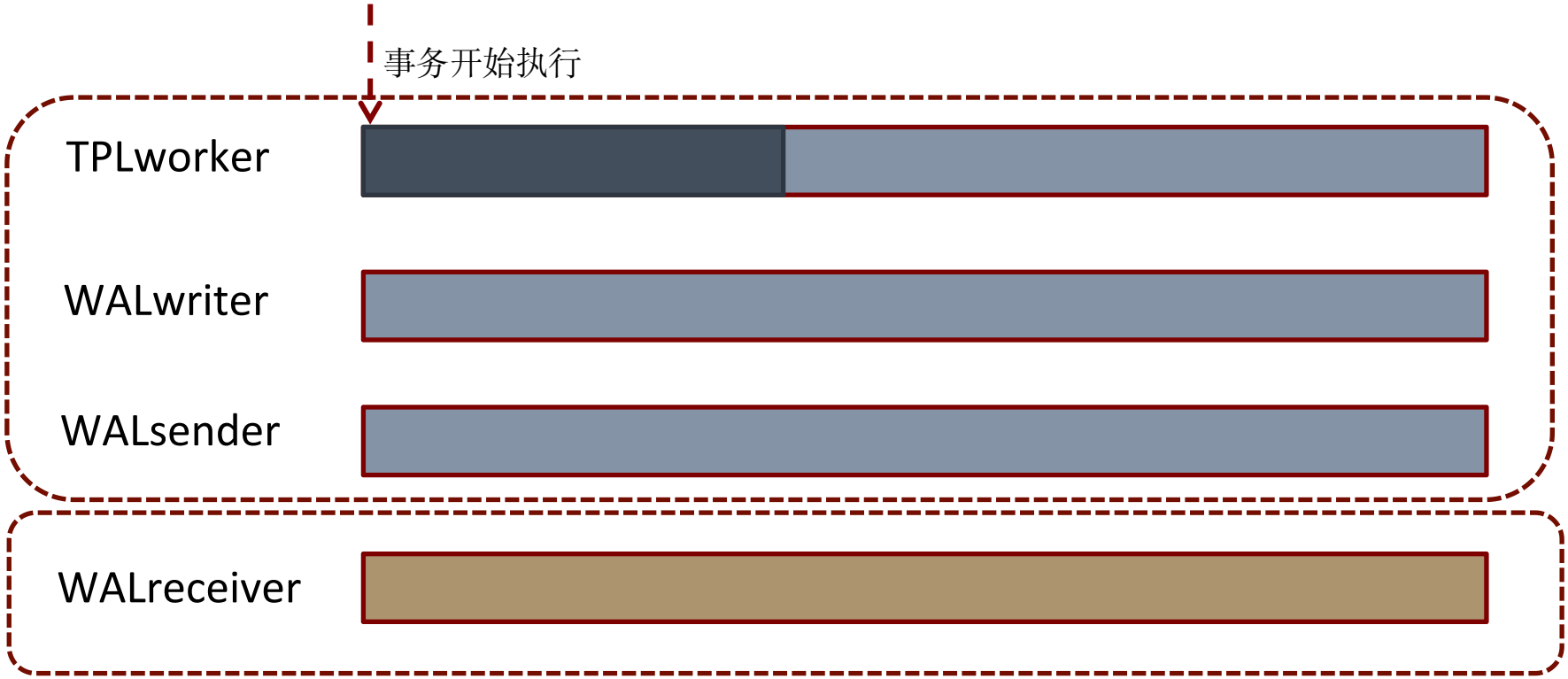


实现背景——基本概念

- 线程池
  - opengauss线程池技术的整体设计思想是线程资源池化、并且在不同连接之间复用。系统在启动之后会根据当前核数或者用户配置启动固定一批数量的**工作线程(TPLworker)**，一个工作线程会服务一到多个连接**会话(session)**，这样把会话和线程进行了解耦。因为工作线程数是固定的，因此在高并发下不会导致线程的频繁切换，而由数据库层来进行线程的调度管理。
- 举例：假设有3个CPU，Session1~Session10，分别执行TXN1~TXN10，在没有线程池的情况下要分别起10个Worker线程执行并抢占cpu；线程池模式下起3个TPLworker，就可以把10个会话分别绑定到TPLworker不用抢占cpu执行。

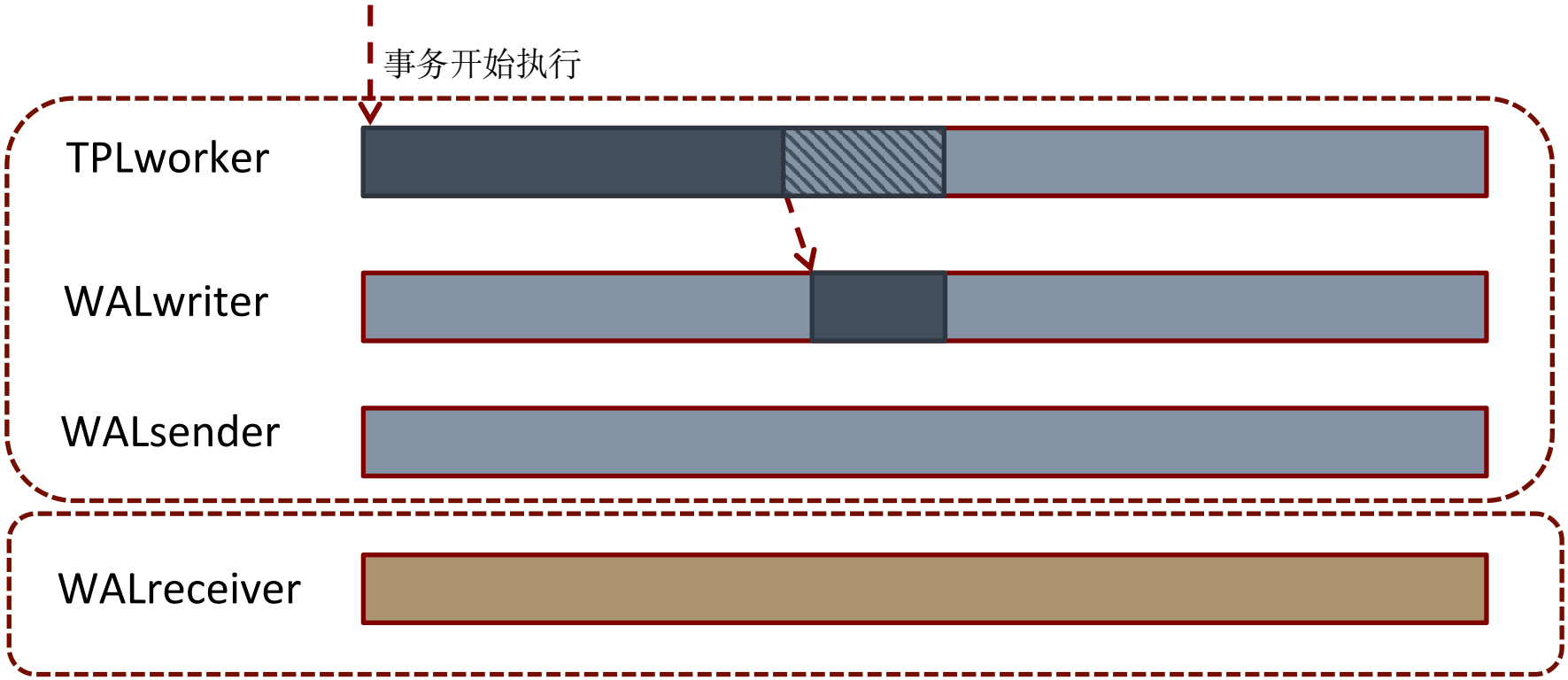
实现背景——基本概念

- opengauss同步事务提交流程



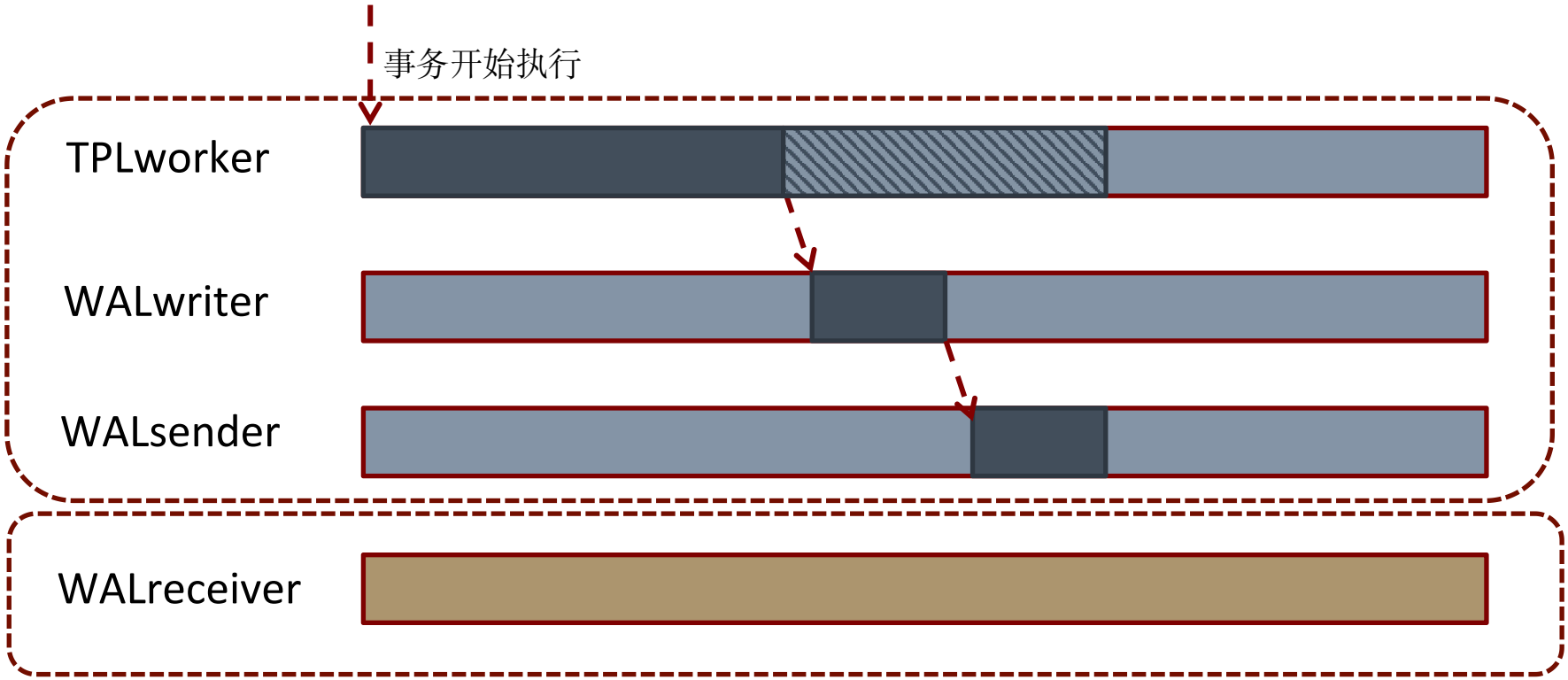
实现背景——基本概念

- opengauss同步事务提交流程



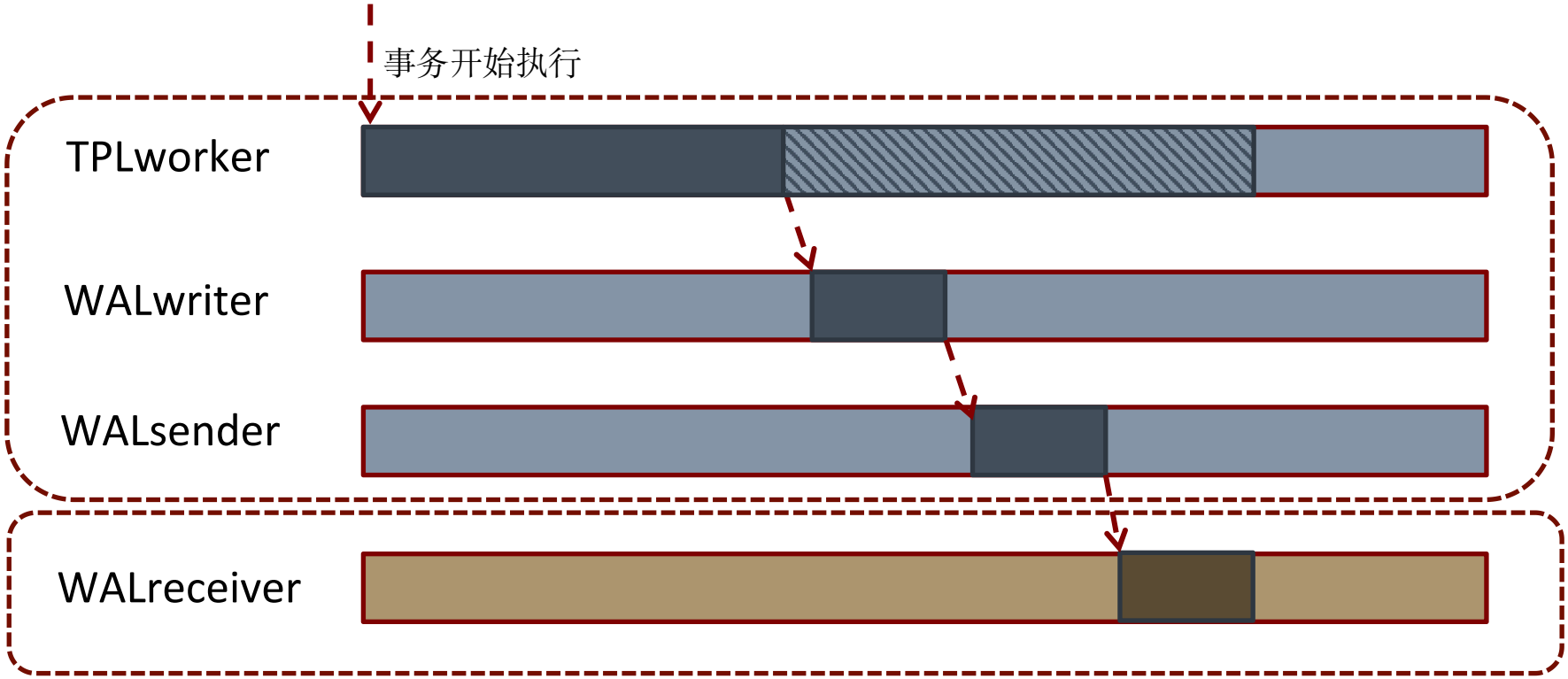
实现背景——基本概念

- opengauss同步事务提交流程



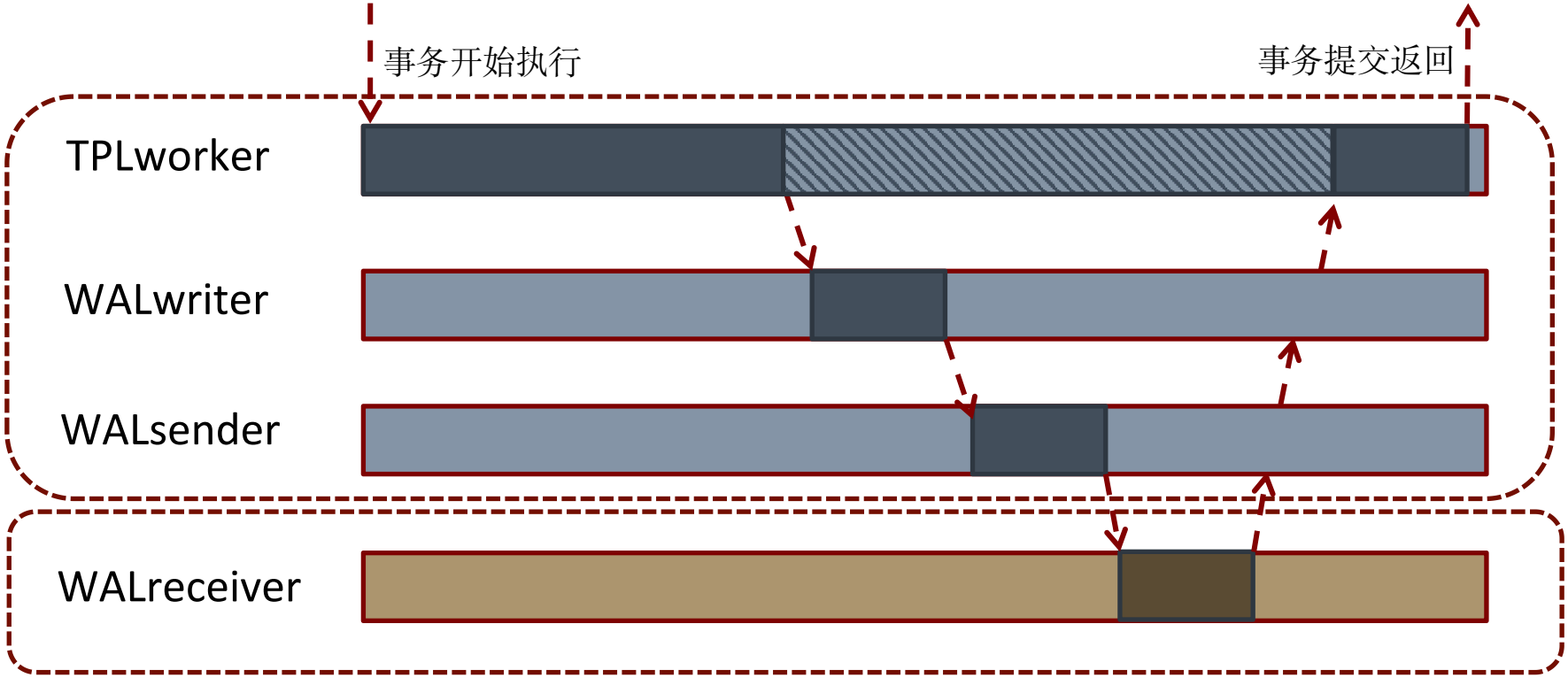
实现背景——基本概念

- opengauss同步事务提交流程



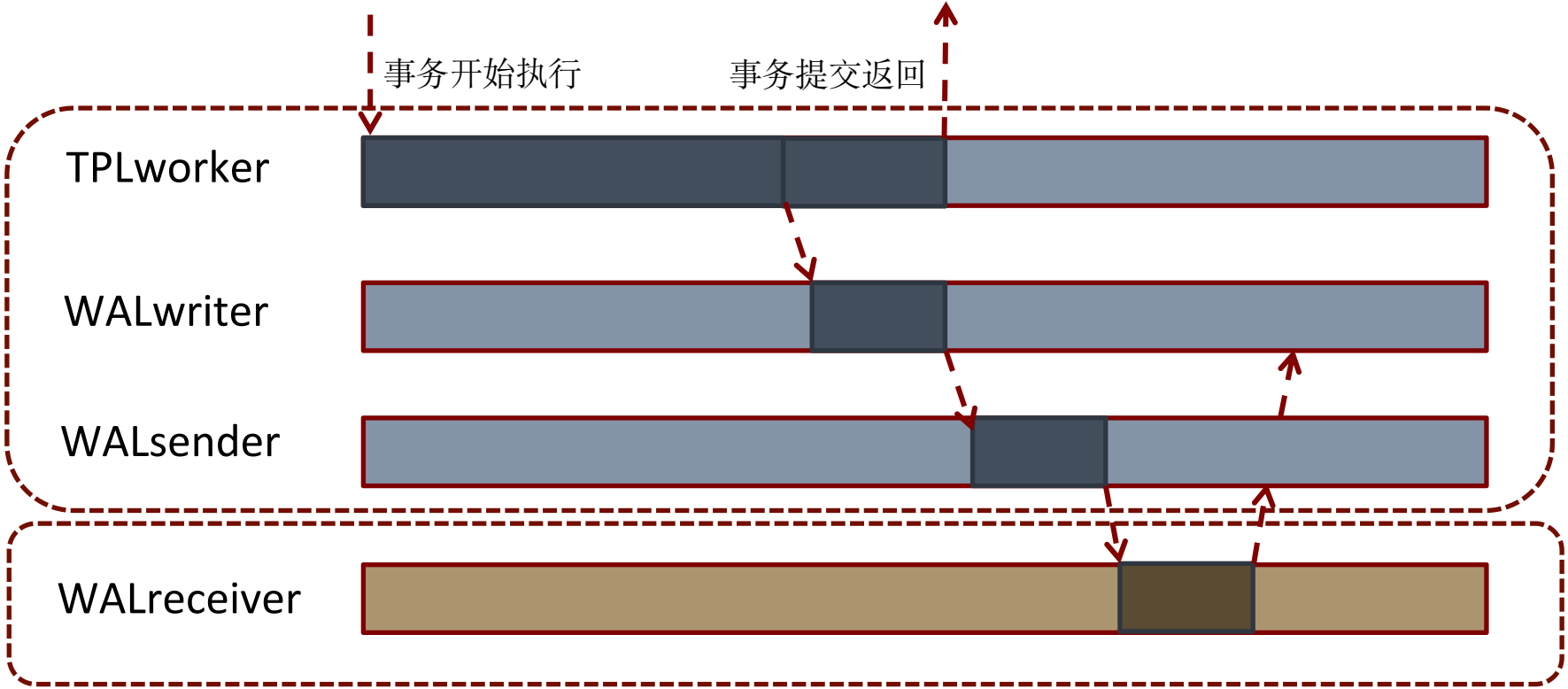
实现背景——基本概念

- opengauss同步事务提交流程



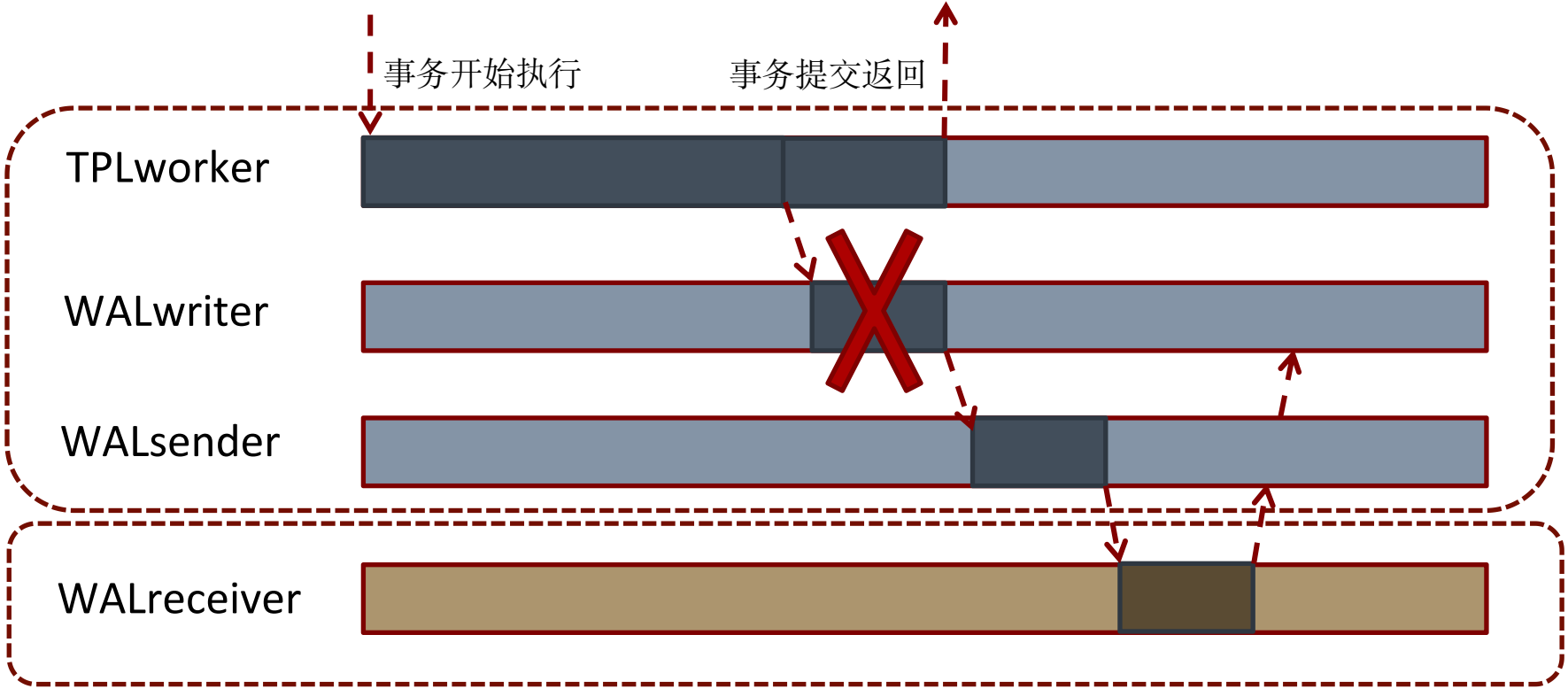
实现背景——基本概念

- opengauss异步事务提交流程



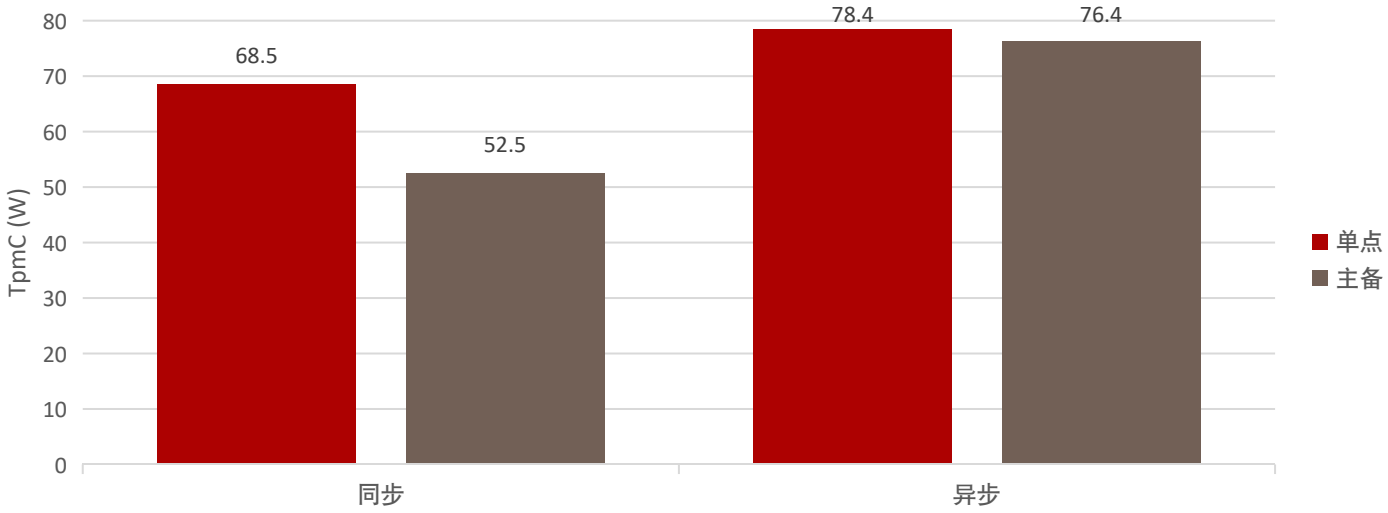
实现背景——基本概念

- opengauss异步事务提交流程



实现背景——预实验

- opengauss现有实现异步事务提交（设置GUC synchronous_commit=off时生效），不等待日志落盘即返回事务提交成功。
- 优点：性能好，不受日志落盘速度影响
- 缺点：一旦中途崩溃之后，不能保证数据库完整性。

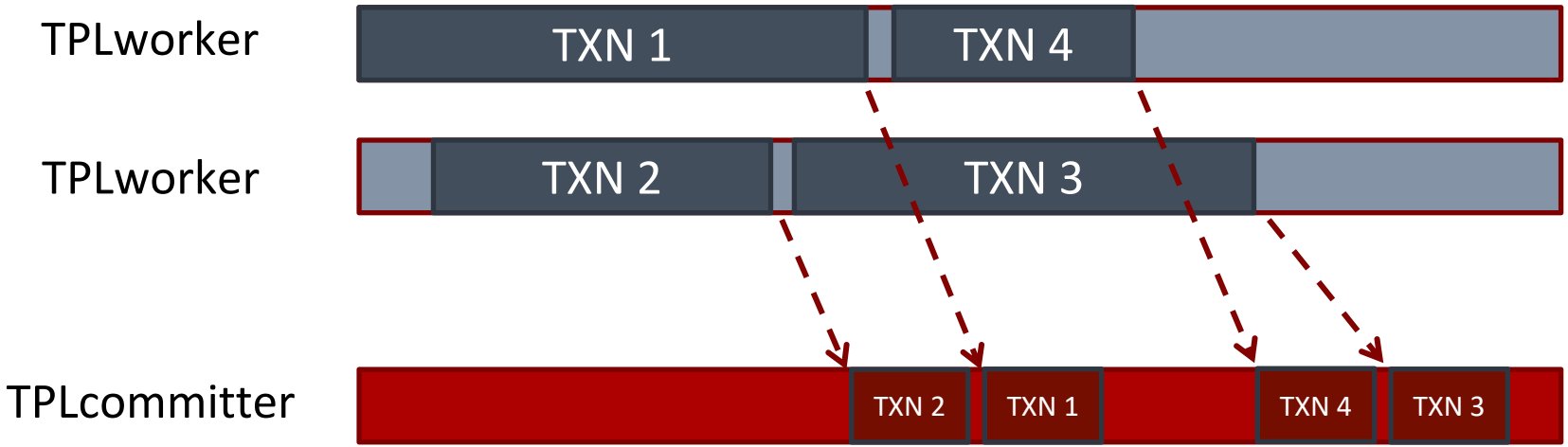


设计思路

- MogDB增加TPLcommiter线程异步进行事务提交操作，TPLworker可以继续处理其他待处理的session，实现事务处理和提交的流水线化。

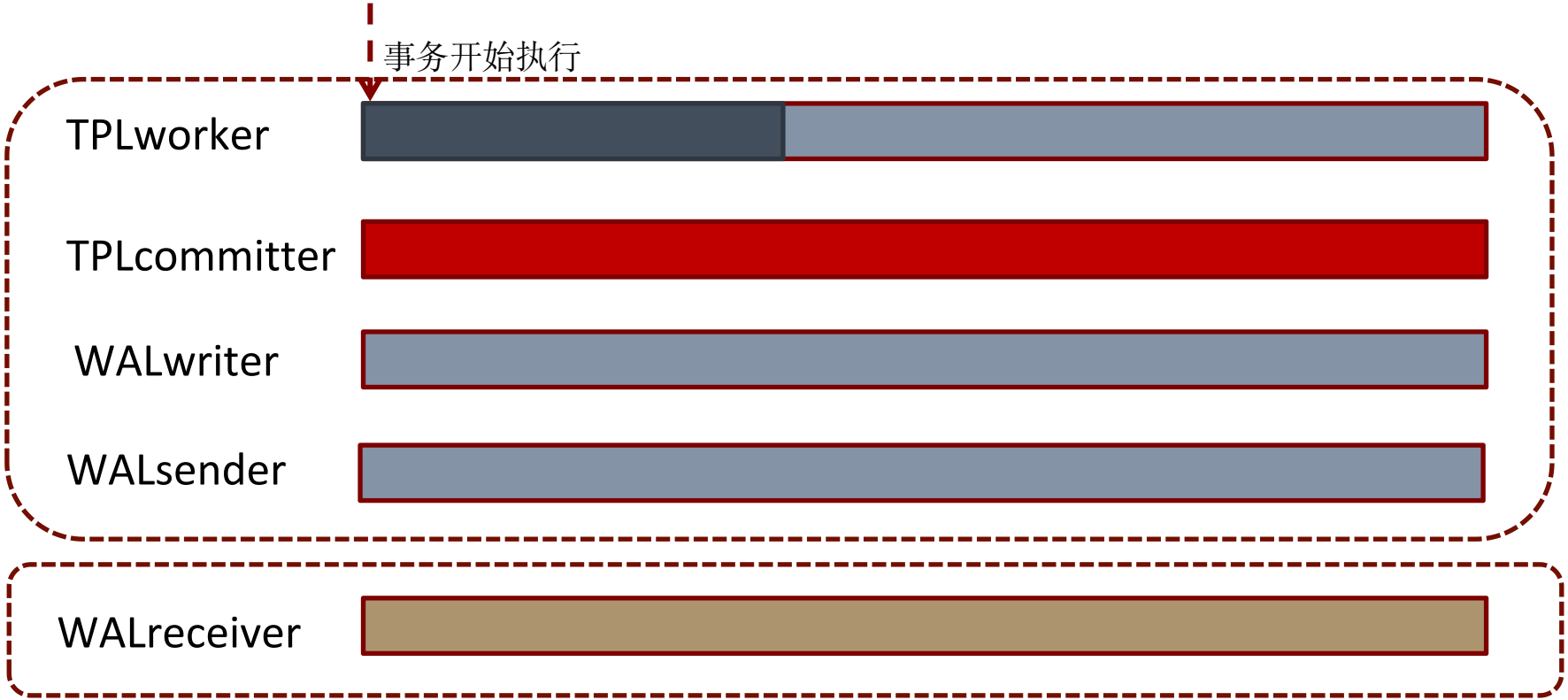
设计思路

- MogDB增加TPLcommiter线程异步进行事务提交操作，TPLworker可以继续处理其他待处理的session，实现事务处理和提交的流水线化。



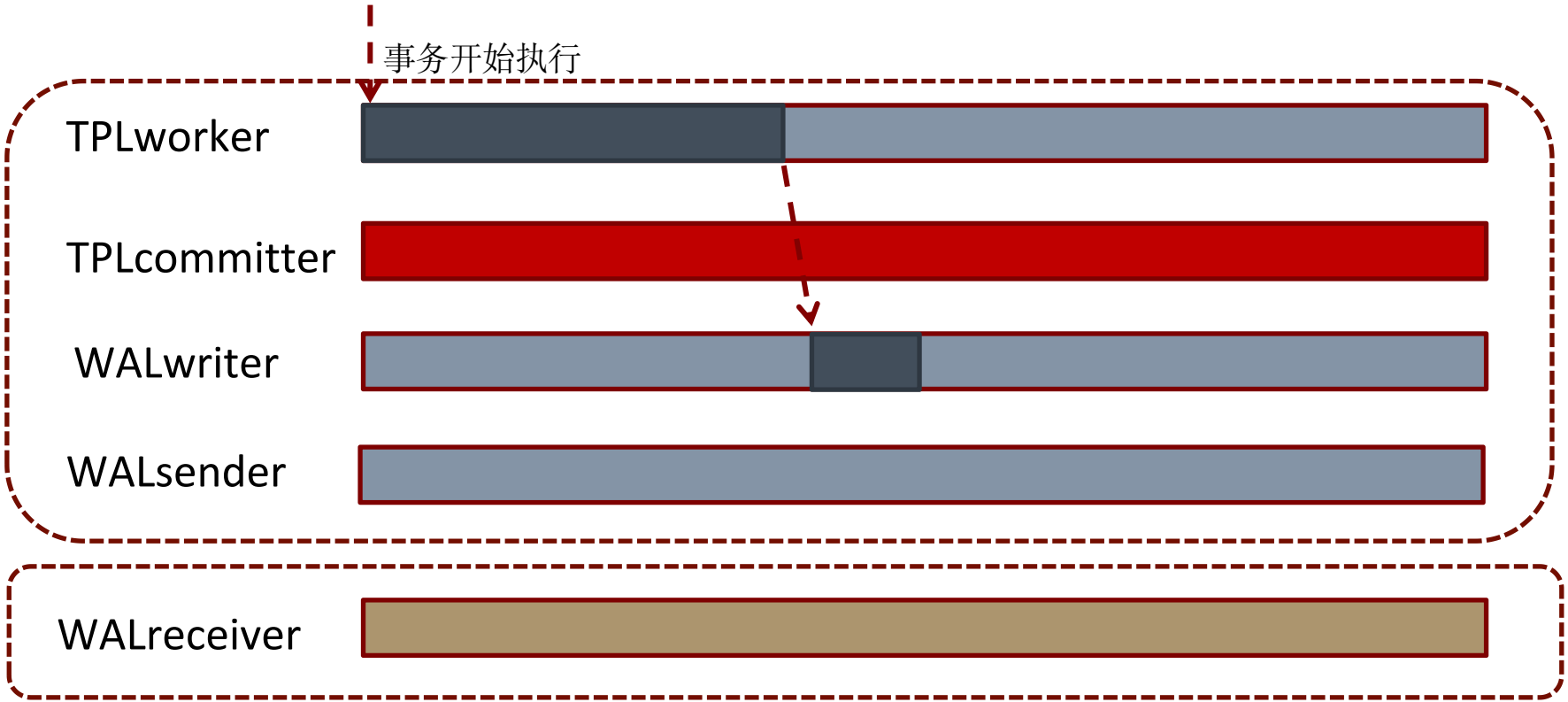
设计实现——自治异步提交流程

- MogDB自治异步事务提交流程



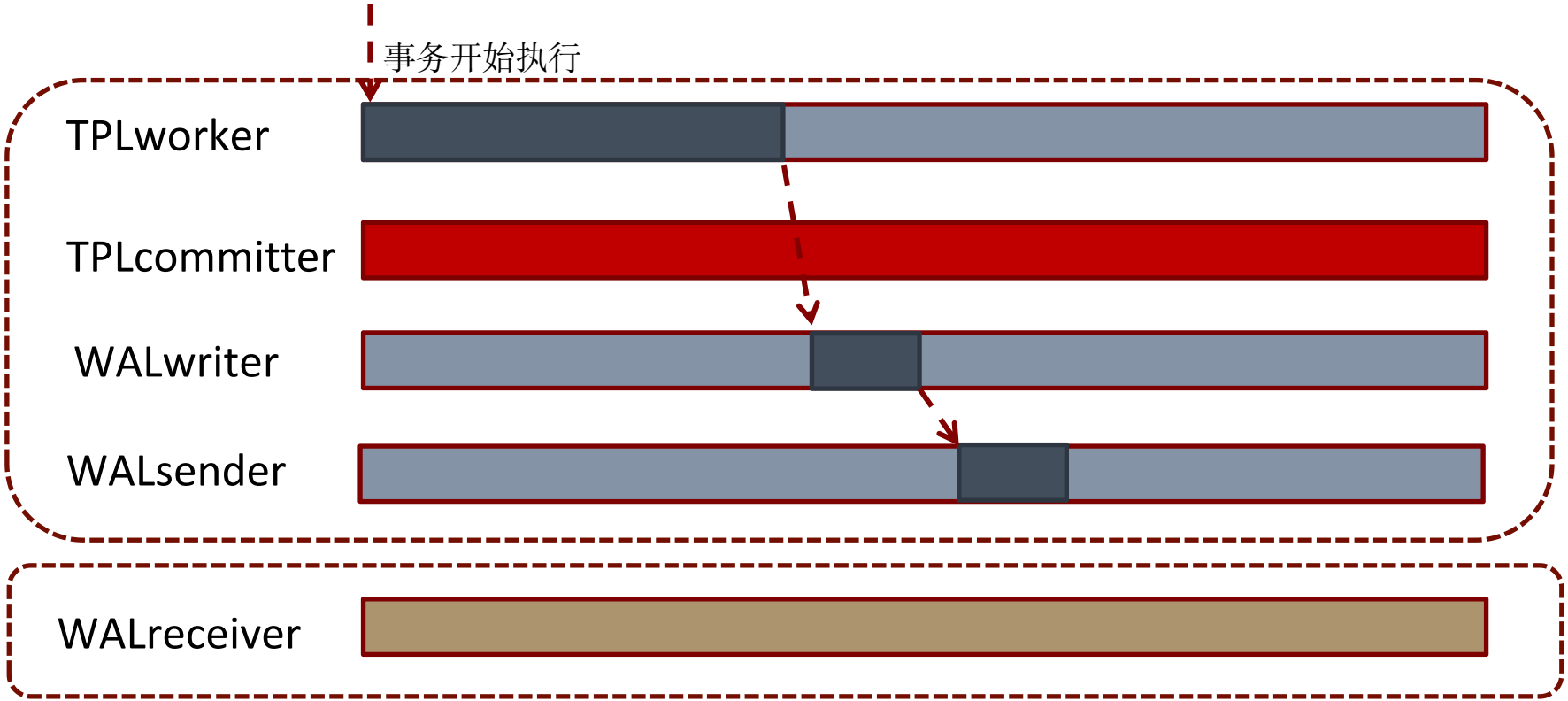
设计实现——自治异步提交流程

- MogDB自治异步事务提交流程



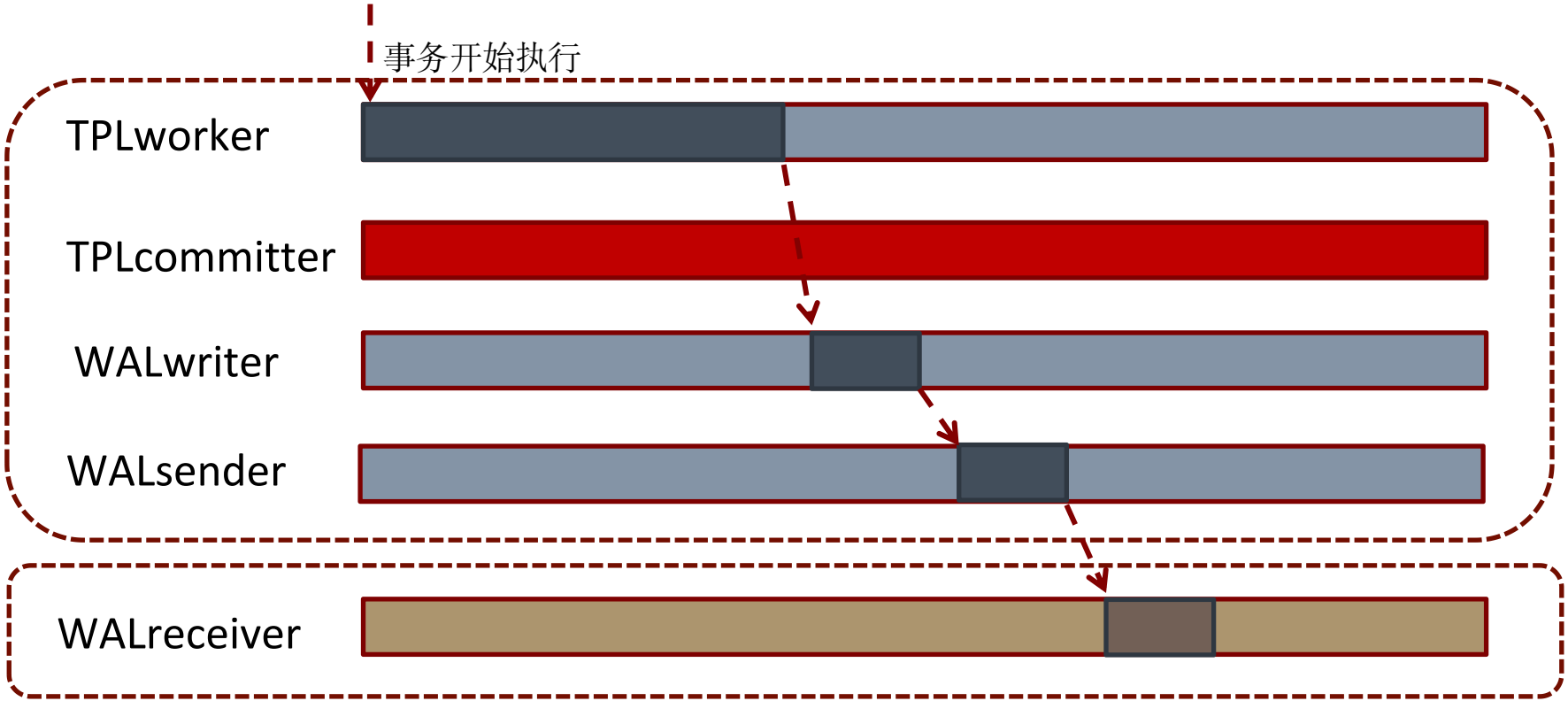
设计实现——自治异步提交流程

- MogDB自治异步事务提交流程



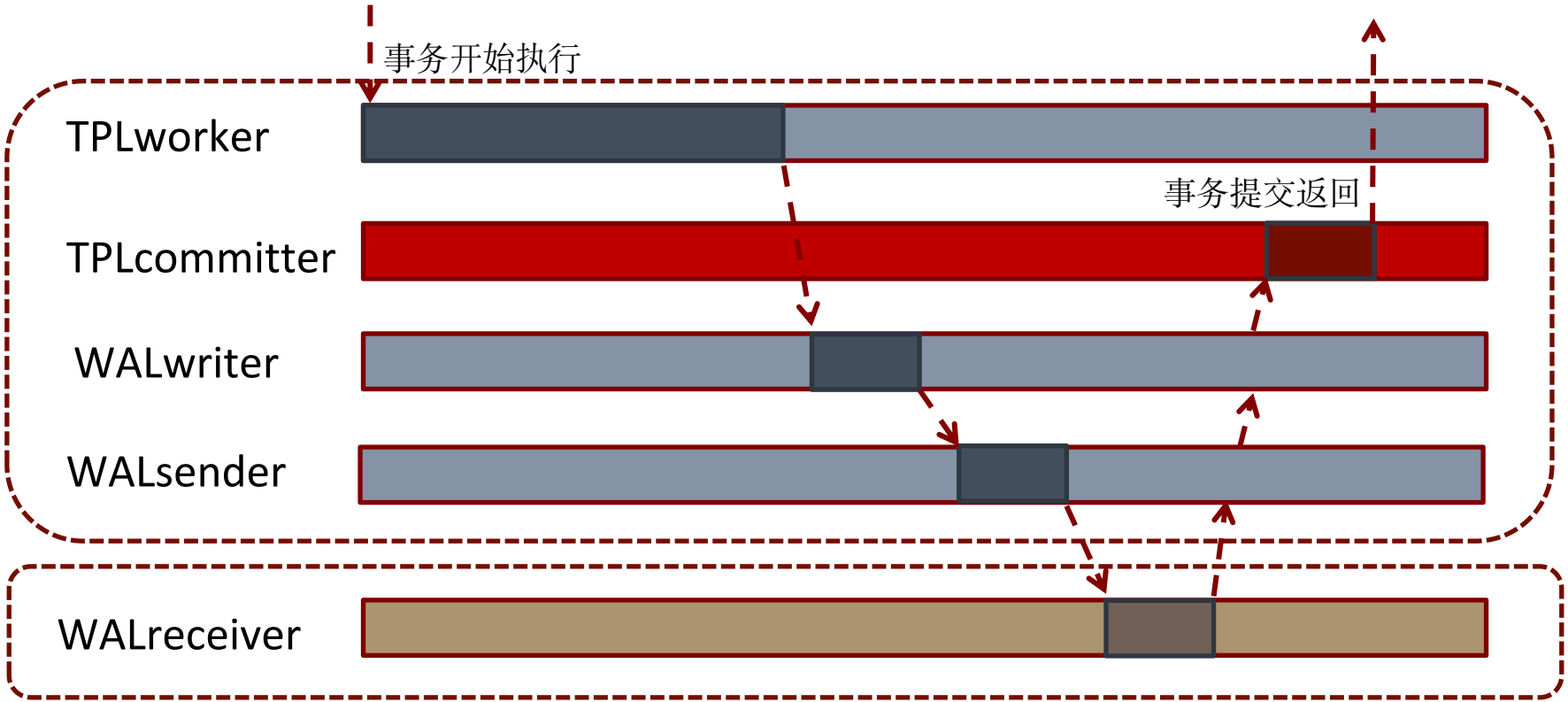
设计实现——自治异步提交流程

- MogDB自治异步事务提交流程



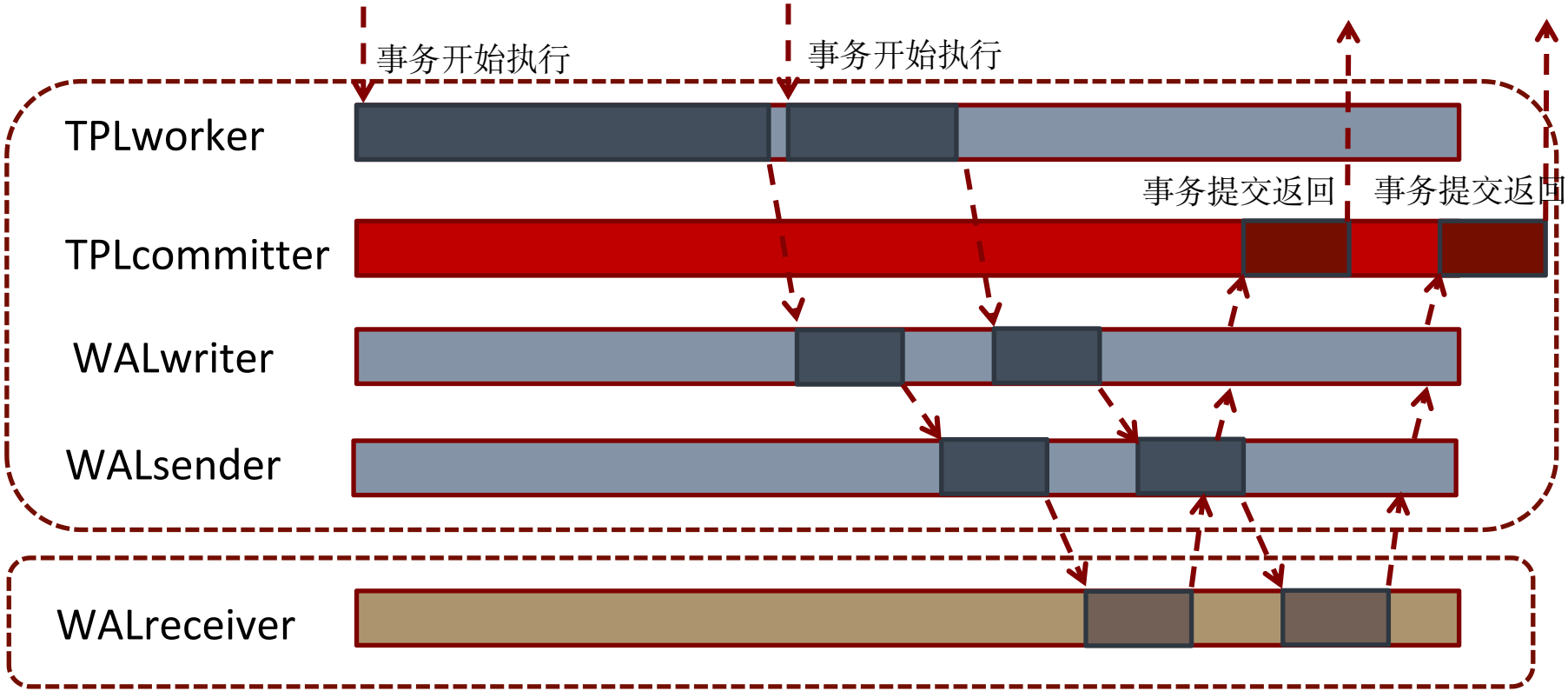
设计实现——自治异步提交流程

- MogDB自治异步事务提交流程



设计实现——自治异步提交流程

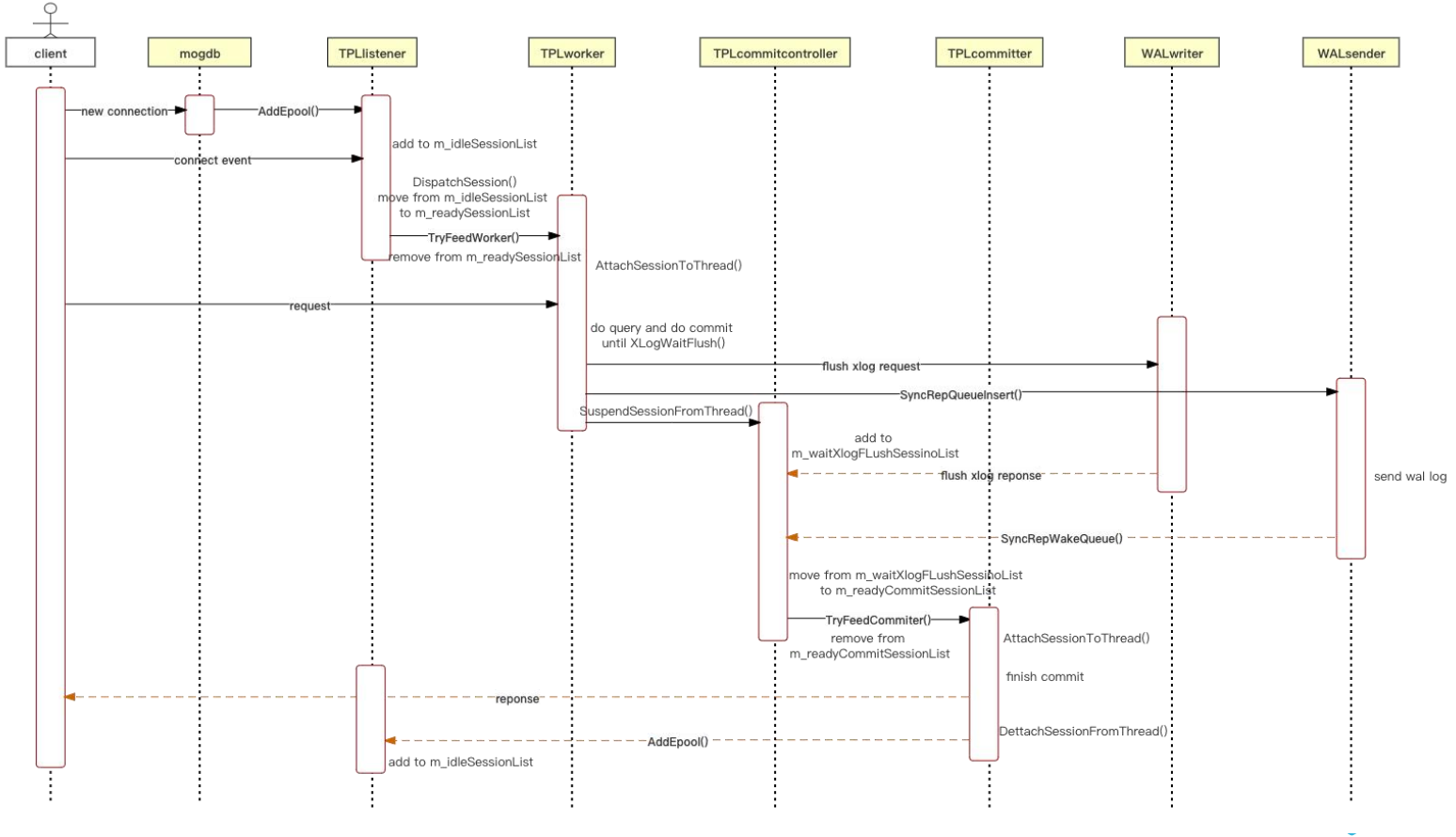
- MogDB自治异步事务提交流程



设计实现——实现细节

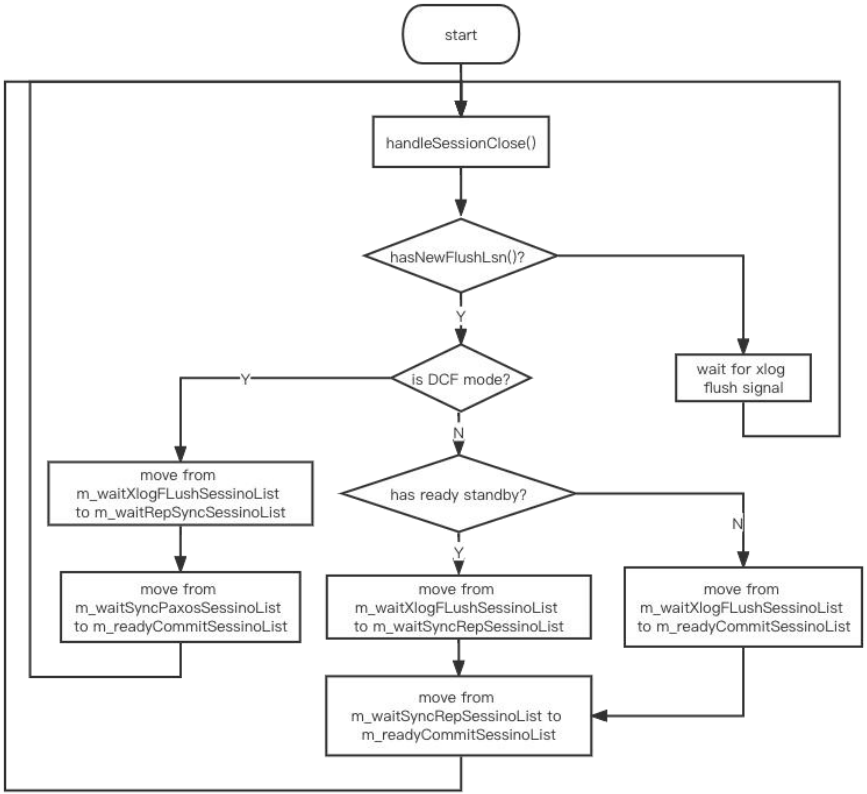
- 每一个线程组中新增一组TPLcommiter和一个TPLcommitterCtl线程。
  - 当事务提交需要等待日志落盘（XlogWaitFlush()）时，将该事务所属的会话挂起，加入该线程组的等待刷盘会话队列（m_waitXlogFlushSessinoLis）中。
  - 等待日志落盘（多点时还要等待日志同步）操作完成之后，会通知TPLcommitterCtl线程将其挪入待提交会话队列（m_readyCommitSessinoList）中。
  - 对于ThreadPoolCommitter线程会持续从待提交会话队列（m_readyCommitSessinoList）中选取会话完成该会话上事务剩余的事务提交工作。
  - 而会话原本所在ThreadPoolWorker线程在会话被挂起后，就可以去ThreadPoolListener中选取新的会话继续工作。

设计实现——实现细节



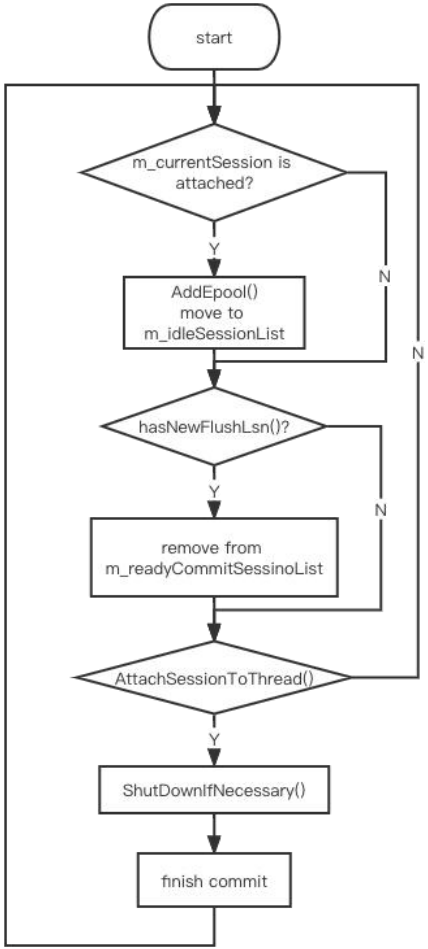
设计实现——实现细节

- ThreadPoolCommitterCtl线程
  - 监控会话的工作状态
    - 判断会话的日志落盘操作（HA、DCF模式下还有日志同步操作）是否完成
  - 处理会话关闭



设计实现——实现细节

- ThreadPoolCommitter线程
  - 获取得完成日志落盘（同步）的会话，继续完成事务提交剩余的部分（包括资源释放、cache invalidation、返回消息给客户端）



设计实现——GUC及统计信息视图

- 新增GUC：async_submit
  - 参数描述：可以会话级别控制是否使用自治异步事务提交。该开关仅在“enable_threadpool = on”和“synchronous_commit”不为“off”时有效。
  - 取值范围：Boolean，默认值为off。
  - on：表示打开事务异步提交，该会话上所有事务提交将异步完成。
  - off：该会话上所有事务按照原有的逻辑提交。

设计实现——GUC及统计信息视图

- 更改dbe_perf.local_threadpool_status视图结构
  - 用于查看当前状态下TPLcommitter线程状态

名称	类型	描述
node_name	text	数据库进程名称。
group_id	integer	线程池组ID。
bind_numa_id	integer	该线程池组绑定的NUMA ID。
bind_cpu_number	integer	该线程池组绑定的CPU信息。如果未绑定CPU，该值为NULL。
listener	integer	该线程池组的Listener线程数量。
worker_info	text	线程池中worker线程相关信息，包括以下信息: - default: 该线程池组中的初始线程数量。- new: 该线程池组中新增线程的数量。 - expect: 该线程池组中预期线程的数量。- actual: 该线程池组中实际线程的数量。 - idle: 该线程池组中空闲线程的数量。- pending: 该线程池组中等待线程的数量。
session_info	text	线程池中会话相关信息，包括以下信息: - total: 该线程池组中所有的会话数量。- waiting: 该线程池组中等待调度的会话数量。 - running: 该线程池中正在执行的会话数量。- idle: 该线程池组中空闲的会话数量。
stream_info	text	线程池中stream线程相关信息，包括以下信息： - total: 该线程池组中所有的stream线程数量。- running: 该线程池中正在执行的stream线程数量。 - idle: 该线程池组中空闲的stream线程数量
committer_info	text	线程池中committer线程相关信息，包括以下信息： - total: 该线程池组中所有的committer线程数量。- running: 该线程池中正在执行的committer线程数量。 - idle: 该线程池组中空闲的committer线程数量

设计实现——GUC及统计信息视图

- 新增view dbe_perf.gs_async_submit_sessions_status
  - 用于查看使用异步提交的会话状态（当前和历史）

名称	类型	描述
nodename	text	节点名
groupid	integer	线程组号
waiting_xlog_flush_session_num	integer	等待日志刷盘的session数
waiting_sync_rep_receive_session_num	integer	等待备份同步日志receive的session数
waiting_sync_rep_write_session_num	integer	等待备份同步日志write的session数
waiting_sync_rep_flush_session_num	integer	等待备份同步日志flush的session数
waiting_sync_rep_apply_session_num	integer	等待备份同步日志apply的session数
waiting_sync_paxos_session_num	integer	等待dcf同步日志的session数
waiting_commit_session_num	integer	已经完成日志刷盘（和备份同步）但还没有完成异步提交的session数
finished_commit_session_num	bigint	已经完成异步提交的session数

性能测试

- **硬件配置**

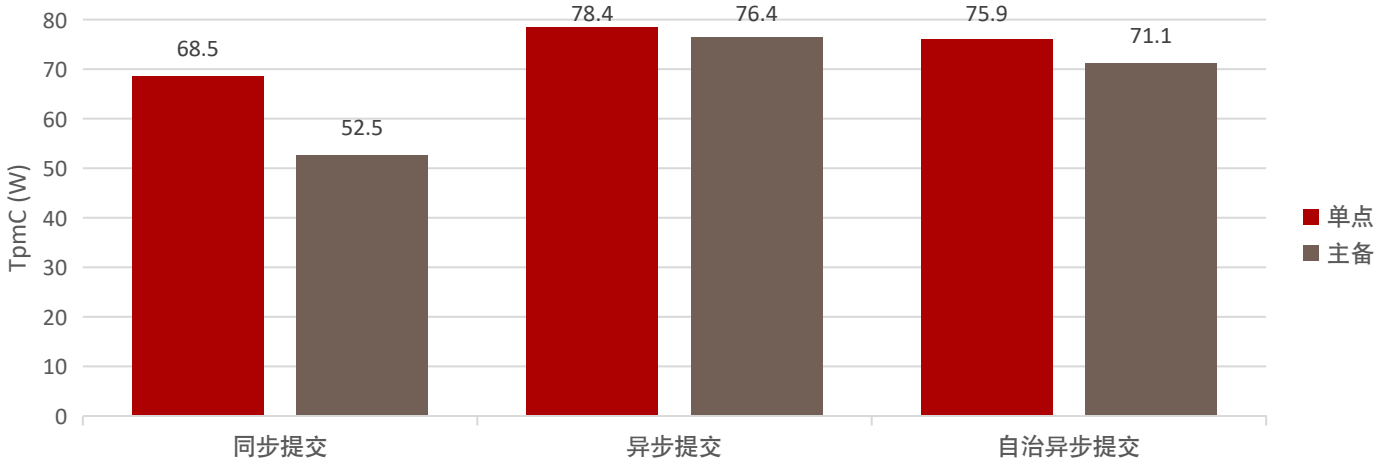
CPU	Kunpeng-920 * 64
内存	501GB
硬盘	SATA SSD 1.75T * 4
OS	openEuler-22.03-LTS Linux version 5.10.0-60.18.0.50.oe2203.aarch64
文件系统	XFS
网卡	10Gib/s

- **数据集**

TPC-C，1000个warehouse，约100G数据，600并发

性能测试

- TPCC
  - 单点：打开自治异步提交，相对不打开提升 10.8%
  - 主备：打开自治异步提交，相对不打开提升 36.2%
  - 打开自治异步提交，主备比单点性能下降 6.3%



- 总结

通过事务处理和提交的流水线化，充分利用CPU资源，实现性能提升。

- 优势场景

高并发、小query、写负载高、低速IO设备/HA模式/DCF模式。

谢谢观看

THANKS FOR WATCHING