



GreatSQL 实战与高可用解决方案



关于GreatSQL ***01***

GreatSQL优势 ***02***

Others ***03***

高可用架构 ***03***

SQL优化 ***05***



01

GreatSQL简介

- GreatSQL是**适用于金融级应用的国内自主开源数据库**，具备**高性能、高可靠、高易用性、高安全**等多个核心特性，可以作为MySQL或Percona Server的可选替换，用于线上生产环境，且完全免费并兼容MySQL或Percona Server
- GreatSQL开源数据库由万里数据库创建并主导
- 源码、文档、下载等资源
 - 官网：<https://greatsql.cn>
 - 源码：<https://gitee.com/GreatSQL/GreatSQL>
 - 文档：<https://gitee.com/GreatSQL/GreatSQL-Doc>
 - 下载：<https://gitee.com/GreatSQL/GreatSQL/releases>



- **某大型股份制银行：**基于GreatSQL源码开发的数据库产品，已上线61个系统，包括：缴费平台、统一支付平台等。
- **某大型国有银行：**下属科技公司基于GreatSQL源码开发的数据库产品，已在该银行上线90个应用系统，包括：综合分账结算服务系统、进出口银行、银保监等外拓业务系统15个。
- **中信建投：**2022年加入GreatSQL社区，2023年2月向社区贡献相关代码。
- **梅州客商银行：**数据安全平台和征信指标变量衍生平台

金融行业

- **国家电网：**基于GreatSQL技术开发的思极有容数据库（SG-RDB），累计支撑国网公司100余个业务系统建设，包括国网全业务数据中心、S6000等应用覆盖总部、27家网省公司。

能源行业

- 恒生芸擎网络
- 福建靠谱云
- 华润网络
- 作业帮

互联网行业

- GreatSQL 8.0
 - GreatSQL 更新说明 8.0.32-24(2023-6-5)
 - GreatSQL 更新说明 8.0.25-17(2023-3-13)
 - GreatSQL 更新说明 8.0.25-16(2022-5-16)
 - GreatSQL 更新说明 8.0.25-15(2021-8-26)
- GreatSQL 5.7
 - GreatSQL 更新说明 5.7.36(2022-4-7)

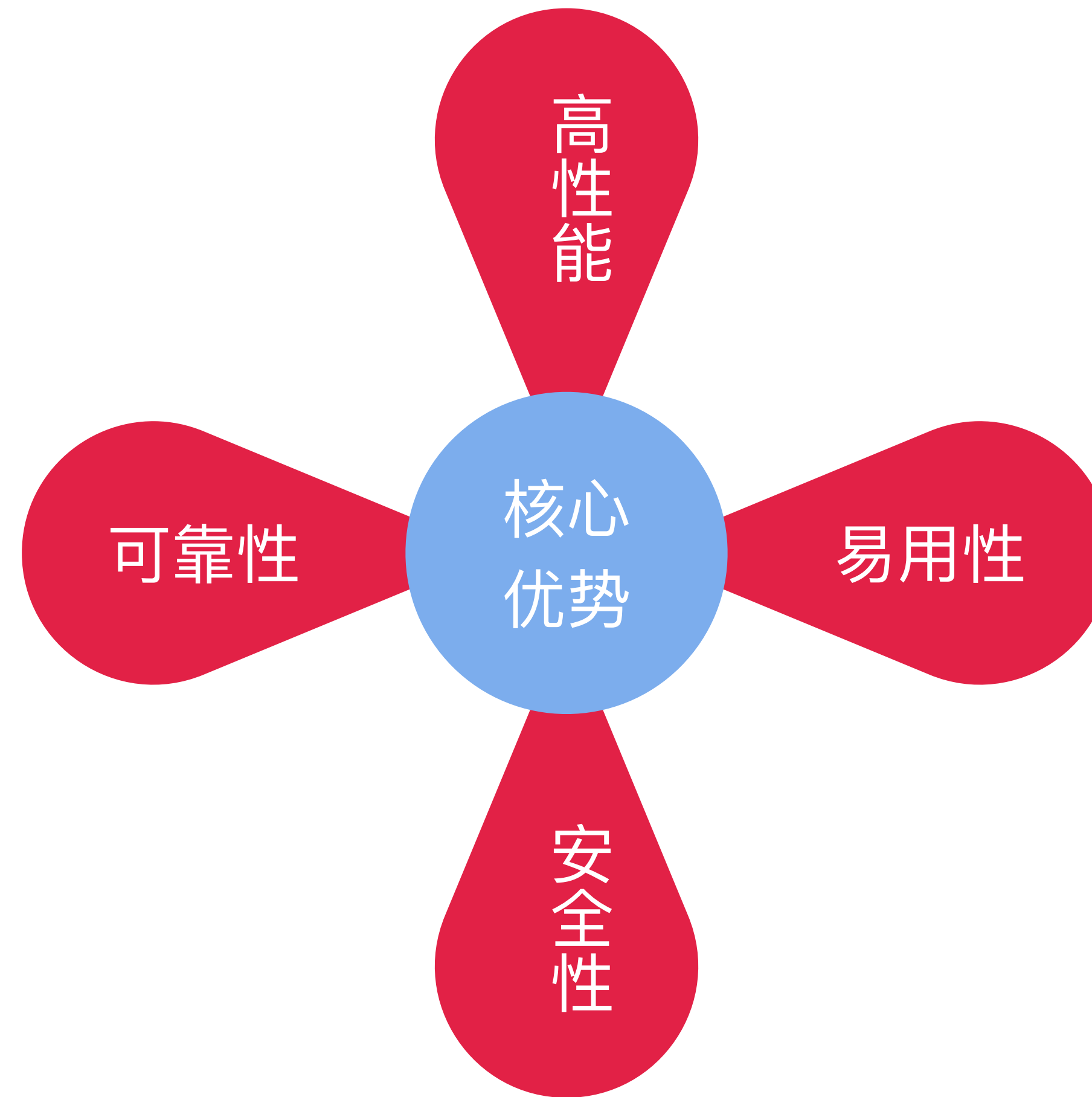
- 官网: <https://greatsql.cn>
 - 论坛: <https://greatsql.cn/forum.php>
 - FAQ: https://greatsql.cn/doc/#!/&v=51_19_0
 - 手册: <https://greatsql.cn/doc/>
 - 视频: https://greatsql.cn/smx_course-lesson.html?op=video
 - GCA认证: https://greatsql.cn/smx_course-lesson.html?op=video&ids=10
- 相关资源
 - QQ群: 533341697
 - 微信群: GreatSQL/MGR交流 (1-3) 群



02 GreatSQL特性

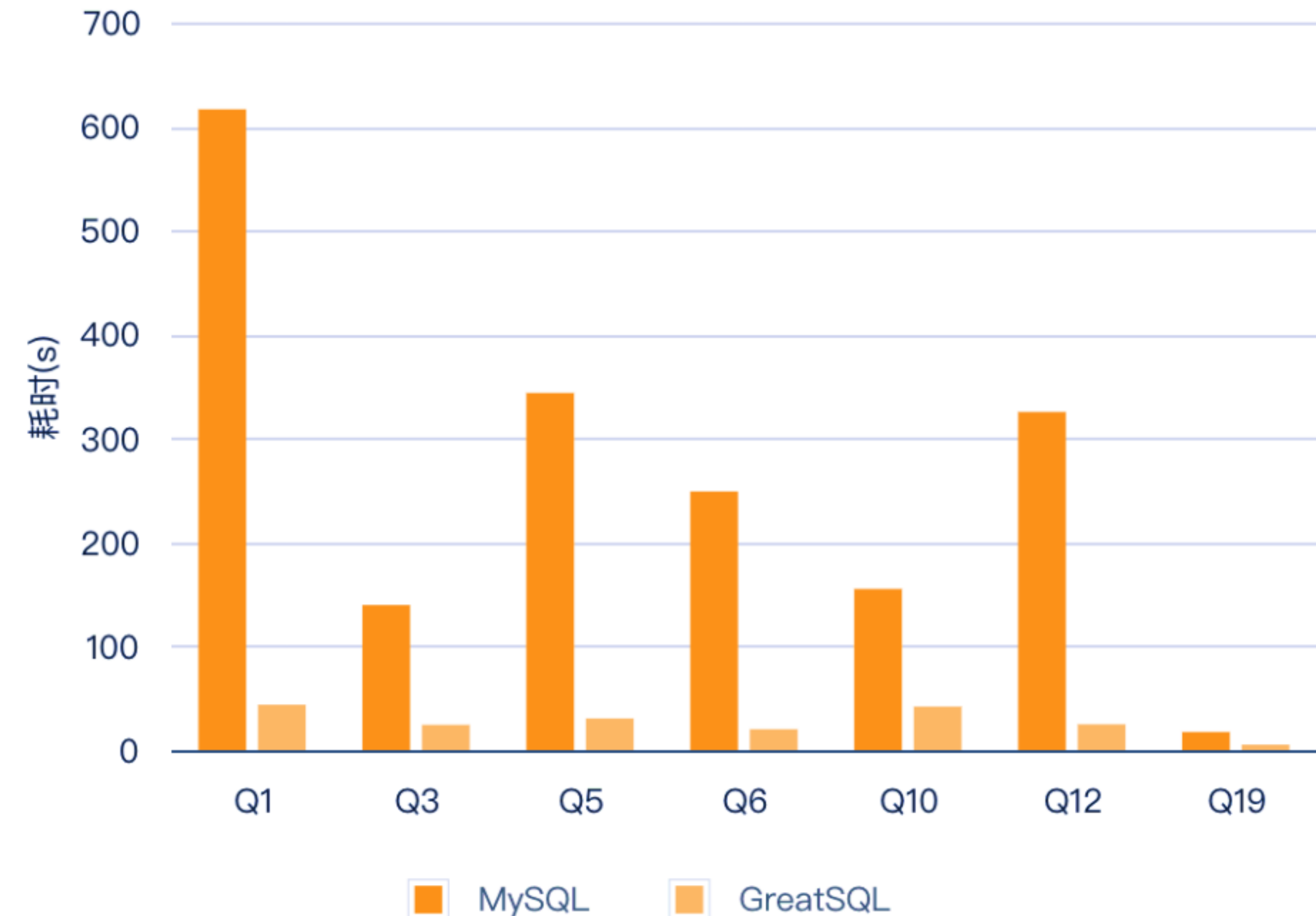


- GreatSQL开源数据库适用于金融级应用场景，具备以下几点优势

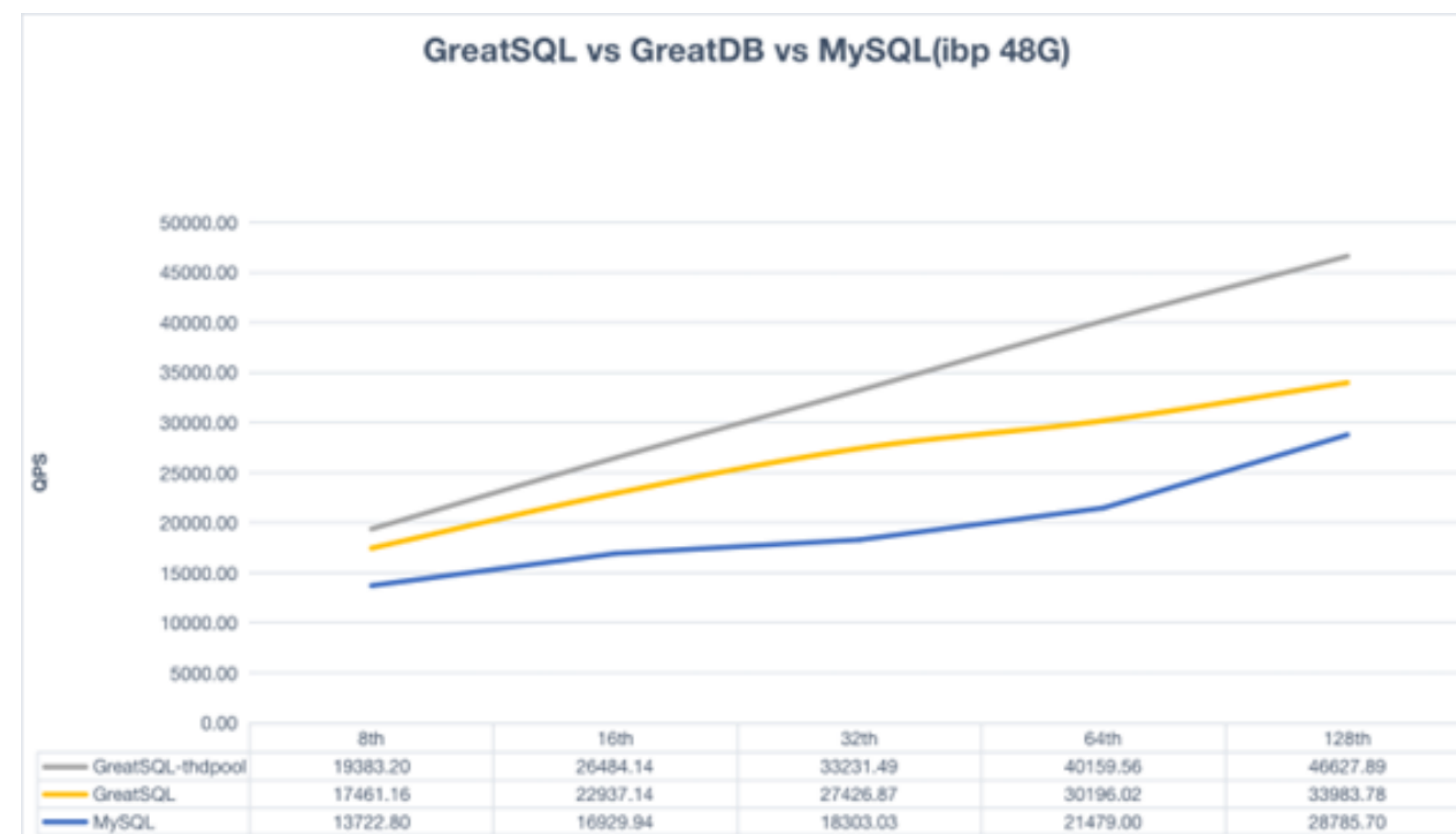


- Parallel Query
 - 充分利用多核CPU
 - 将单线程计划，转化为多线程并行计算
 - TPC-H测试平均提升15倍，最高40+倍
- Parallel Load
 - 充分利用多核CPU
 - 主线程解析数据文件
 - 多个工作线程并发load数据
 - 导入性能和工作线程数正比

TPC-H



- Thread pool
 - 降低了线程创建和销毁的代价
 - 保证高并发下，性能稳定不衰退
- 事务系统优化
 - 锁拆分，将单把大锁拆分为多个粒度更细的锁
 - 无锁化改造，去掉readview获取时的锁保护
- TP场景整体性能提升20%以上



地理标签



- ☐ 解决多IDC数据同步的问题
- ☐ 事务提交时，每个IDC中至少一个节点确认事务
- ☐ 每个IDC中至少一个节点有最新事务
- ☐ group_replication_zone_id
 - 0 ~ 8

快速单主模式



- ☐ 不使用原来的事务认证模式，只需在本地认证
- ☐ 降低内存消耗，提升高并发时的MGR性能
- ☐ 特别适合单主模式且跨IDC部署场景
- ☐ group_replication_single_primary_fast_mode
 - 0
 - 1

仲裁节点



- ☐ 不存储用户数据
- ☐ 没有binlog，也不需要回放relay log
- ☐ 只参与MGR状态投票/仲裁
- ☐ 系统负载非常低，可以在一个服务器上部署多实例
- ☐ group_replication_arbitrator

智能选主



- ☐ 优化选主策略
- ☐ 避免可能丢失数据风险
- ☐ group_replication_primary_election_mode
 - WEIGHT_ONLY
 - **GTID_FIRST**
 - WEIGHT_FIRST

读写节点绑定VIP

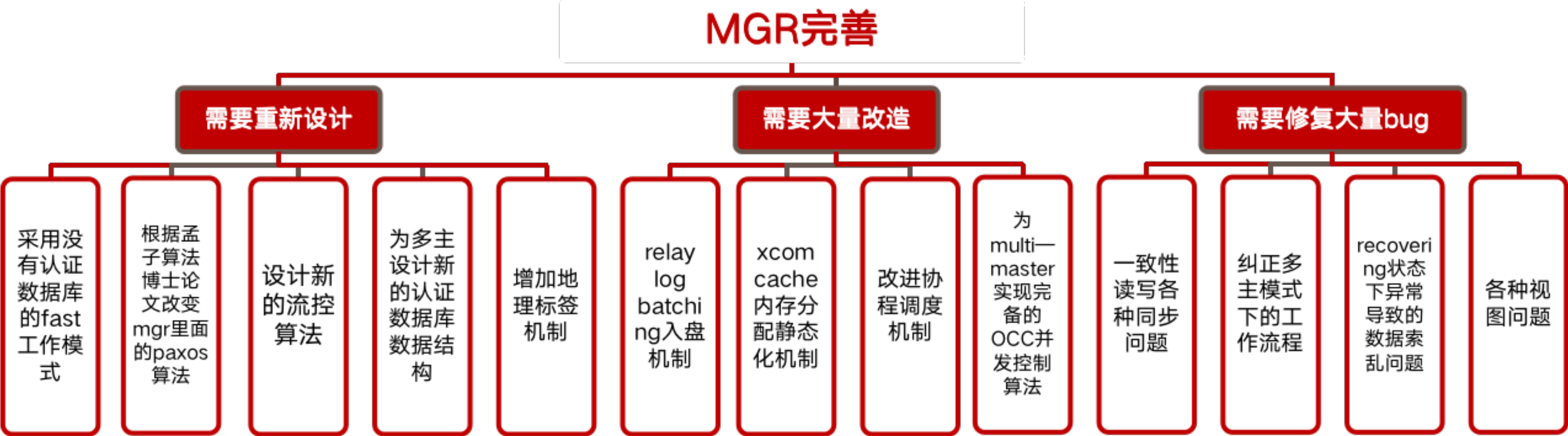


- ☐ 单主模式
- ☐ 读写节点自动绑定VIP
- ☐ 支持在线动态修改
- ☐ 高可用切换更便捷

全新流控算法



- ☐ 原生流控缺陷，导致明显抖动
- ☐ 增加主从延迟时间来计算流控阈值
- ☐ 同时考虑大事务处理因素
- ☐ group_replication_flow_control_rep_lay_lag_behind

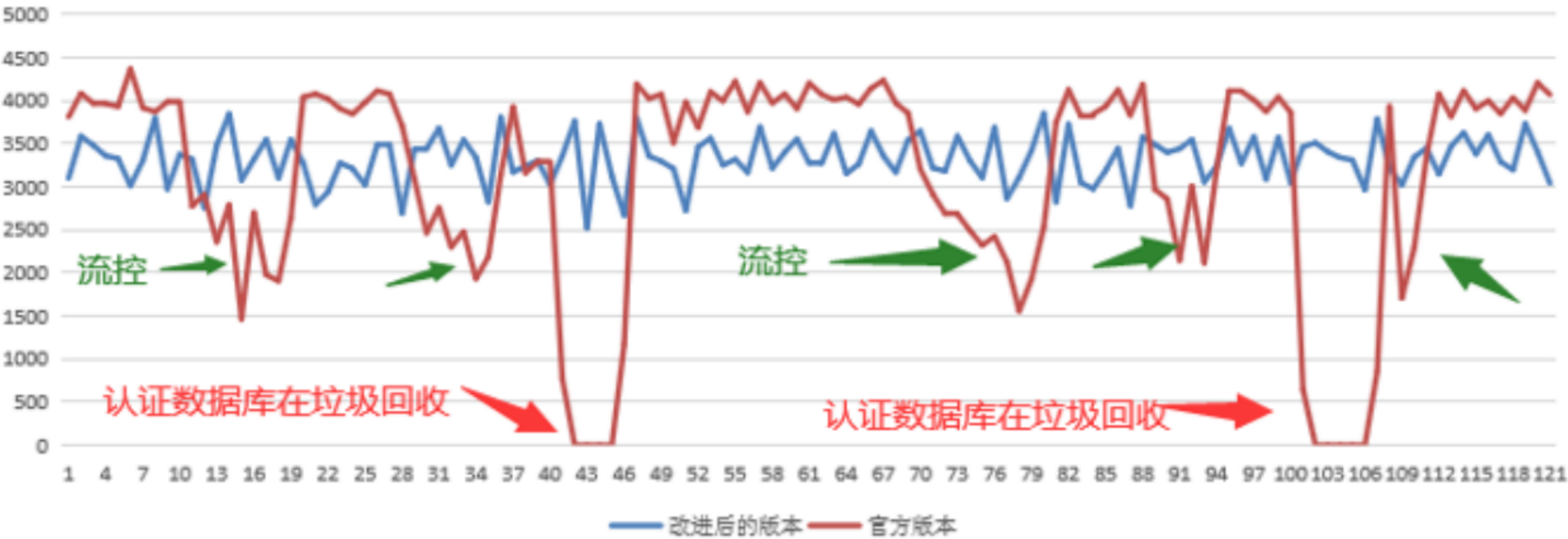


更快：快速探测异常情况；流控机制更精准；

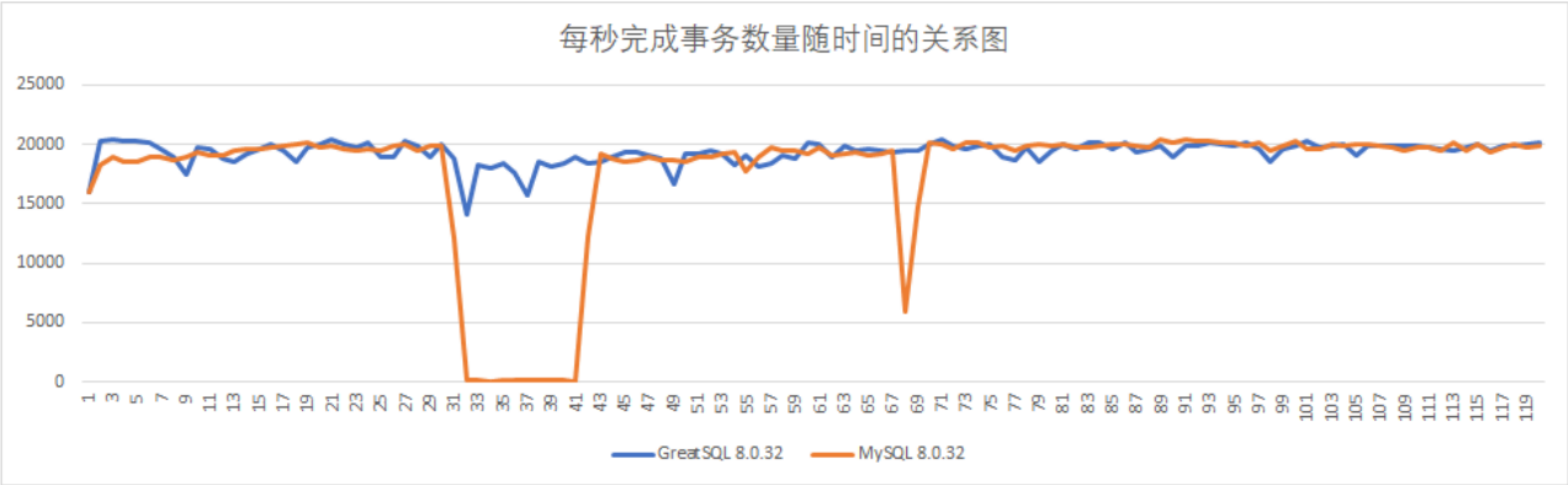
更高：高并发下，吞吐持续且稳定；

更强：更强的鲁棒性、更多的功能性。

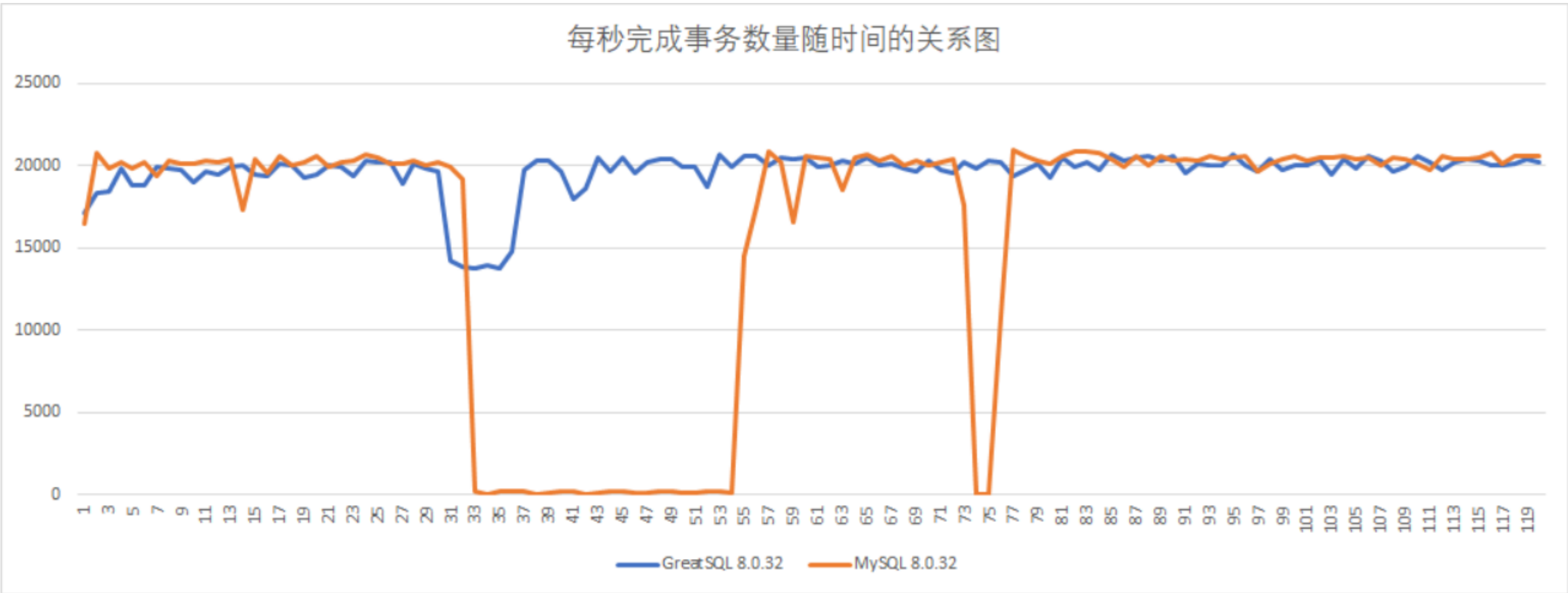
每秒订单数随时间关系图



- 新节点加入导致抖动

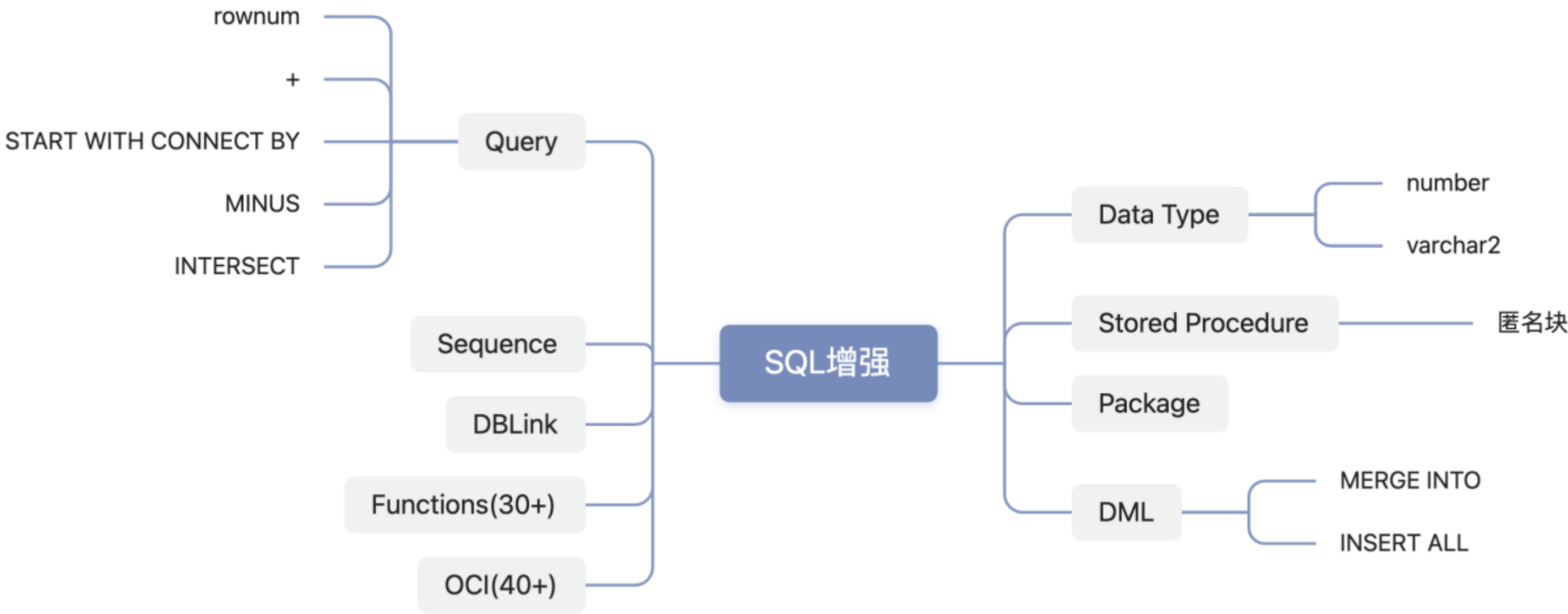


- 节点异常退出导致抖动



- 优化了加入节点时可能导致性能剧烈抖动的问题
- 解决节点异常退出集群时导致性能抖动的问题
 - 原生MGR中节点异常退出时，会造成较长时间性能抖动，最差时TPS直降为0
 - GreatSQL优化后只会产生约1~3秒的小抖动，最差时TPS可能只损失约20% ~ 30%
- 解决磁盘空间爆满时导致MGR集群阻塞的问题
- 解决了长事务造成无法选主的问题
- 完善MGR中的外键约束机制，降低或避免从节点报错退出MGR的风险
- 解决多主模式下或切主时可能导致丢数据的问题

- 优化事务认证队列清理算法，规避每60s抖动问题
- 修复了recover过程中长时间等待的问题
- 修复了传输大数据可能导致逻辑判断死循环问题
- 节点异常状态判断更完善



- InnoDB表空间加密支持国密
- SSL加密通信支持国密
- 审计日志入库
- 逻辑备份加密
- Password Validation Component增强

GreatSQL vs MySQL



1	特性	GreatSQL 8.0.32-24	MySQL 8.0.32
2	开源	✓	✓
3	ACID完整性	✓	✓
4	MVCC特性	✓	✓
5	支持行锁	✓	✓
6	Crash自动修复	✓	✓
7	表分区(Partitioning)	✓	✓
8	视图(Views)	✓	✓
9	子查询(Subqueries)	✓	✓
10	触发器(Triggers)	✓	✓
11	存储过程(Stored Procedures)	✓	✓
12	外键(Foreign Keys)	✓	✓
13	窗口函数(Window Functions)	✓	✓
14	通用表表达式CTE	✓	✓
15	地理信息(GIS)	✓	✓

1	特性	GreatSQL 8.0.32-24	MySQL 8.0.32
16	基于GTID的复制	✓	✓
17	组复制(MGR)	✓	✓
18	MyRocks引擎	✓	✗
19	SQL兼容扩展	1.数据类型扩展 2.SQL语法扩展 共超过20个扩展新特性	✗
20	MGR提升	1.地理标签 2.仲裁节点 3.读写节点绑定VIP 4.快速单主模式 5.智能选主机制 6.全新流控算法	✗
21	性能提升	1.InnoDB并行查询 2.并行load data	✗
22	安全提升	1.国密支持 2.备份加密 3.审计日志入库	✗



03 gt-checksum



- 静态数据库校验修复工具
- 支持MySQL、Oracle等主流数据库
- 采用Go语言开发

- 不支持异构数据库
- 不支持表结构校验
- 对数据库入侵性较大（需要新建校验表）
- 不支持增量数据校验
- 校验效率低（单表校验，无法进行表并发）
- 不支持无索引表校验
- 对数据库造成一定压力（在数据库端做crc32校验）
- 环境受限-强依赖主从环境
- 不支持差异数据的快速定位
- 不支持多次校验加速
- 不支持自动应用修复语句
- crc32校验存在一定的误差

- 功能丰富： 满足多种使用场景
- 性能更好： 通过go的协程并发及阻塞队列来实现快速校验
- 简便易用： 跨平台编译可执行文件，支持极简配置运行，无额外依赖
- 国内开源项目

- 支持多种业务需求场景
 - 主从复制
 - MGR组复制
 - 上云下云业务场景
 - 异构迁移场景

- 支持多种校验对象

- 数据

- 表结构

- 索引

- 分区

- 外键

- 存储过程

- 支持多种校验模型
 - count, 只校验源、目标表的数据量
 - rows, 逐行校验源、目标数据
 - sample, 只进行抽样数据校验, 配合参数ratio设置采样率

- 支持数据库名、表名设置支持多种正则表达式
- 支持校验没有索引的表
- 支持多线程并行校验数据提高校验效率
- 支持设定每次校验分片大小，避免一次性读取太大数据集
- 支持指定修复模式为在线修复或生成SQL文件
- 支持设定每次修复数据事务大小
- 支持设定数据修复方向，可选以源或目标为准

- 极简

- 编辑配置文件 gc.conf-simple, 设定校验源及目标, 校验对象, 而后直接运行

```
$ ./gt-checksum
```

- 标准

- 编辑配置文件后指定

```
$ ./gt-checksum -f ./gc.conf
```

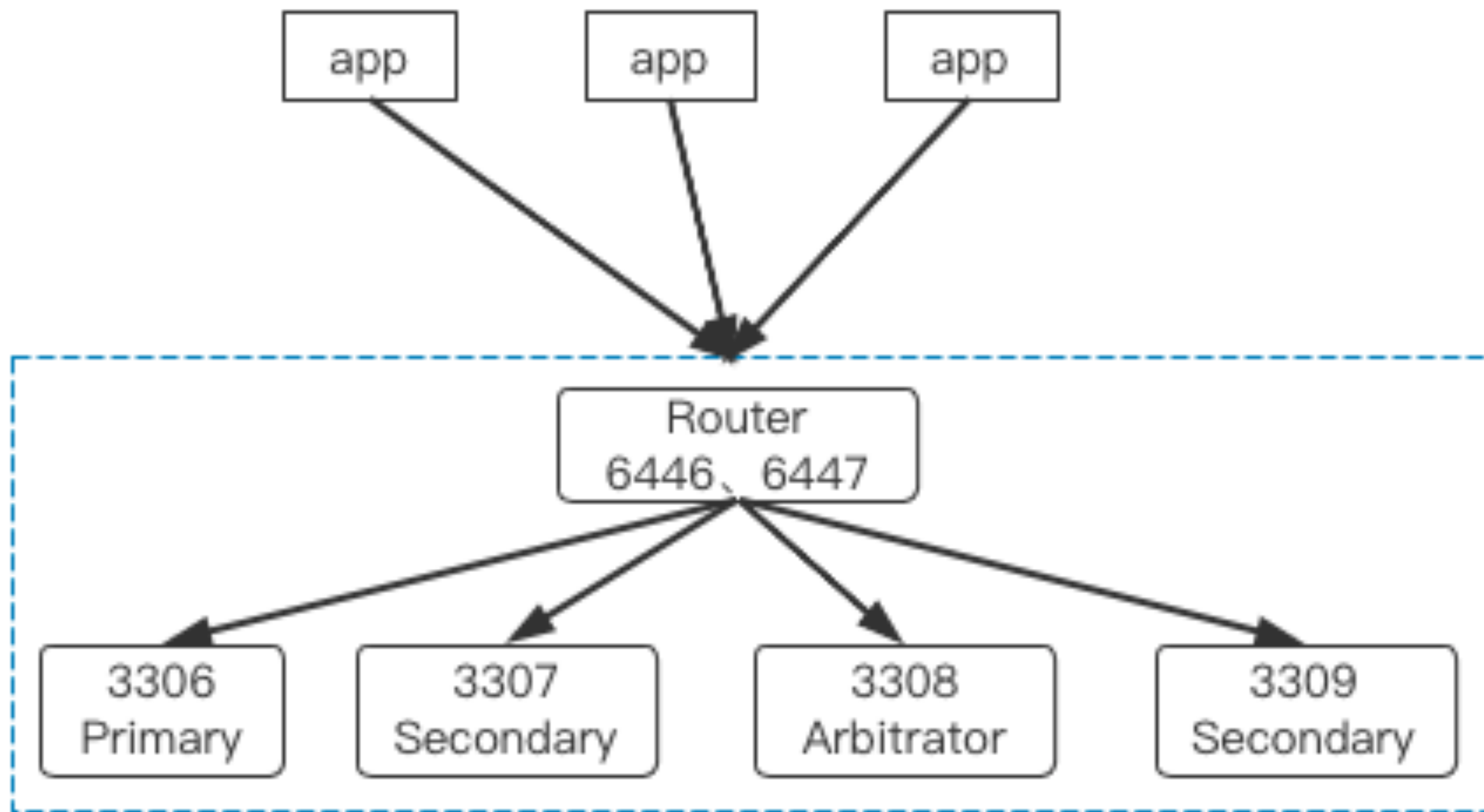
- 也可直接在命令行中传递参数

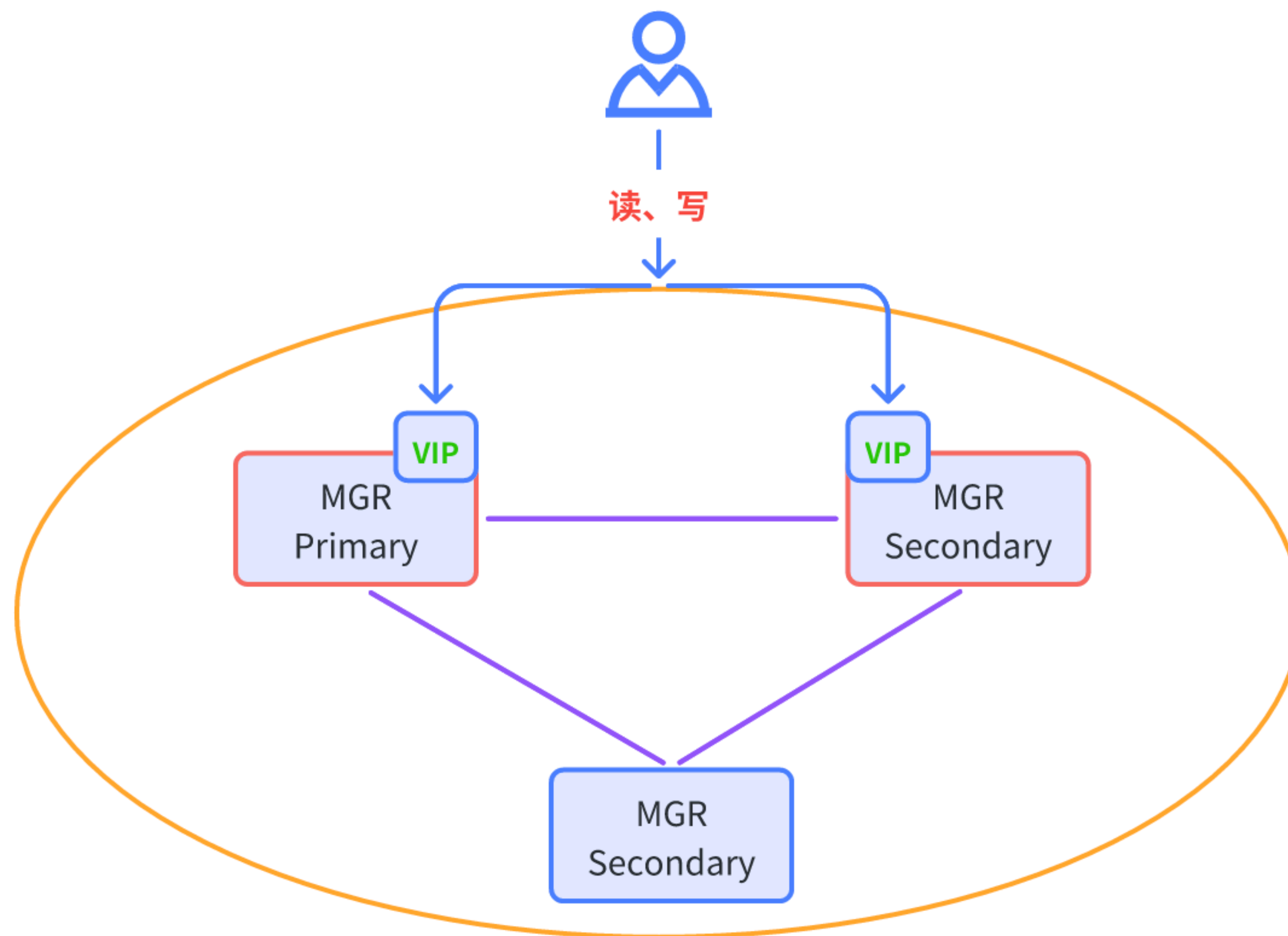
```
$ ./gt-checksum -S type=mysql,user=root,passwd=abc123,host=172.16.0.1,port=3306,charset=utf8 -D  
type=mysql,user=root,passwd=abc123,host=172.16.0.2,port=3306,charset=utf8 -t db1.* -it test.* -nit yes
```

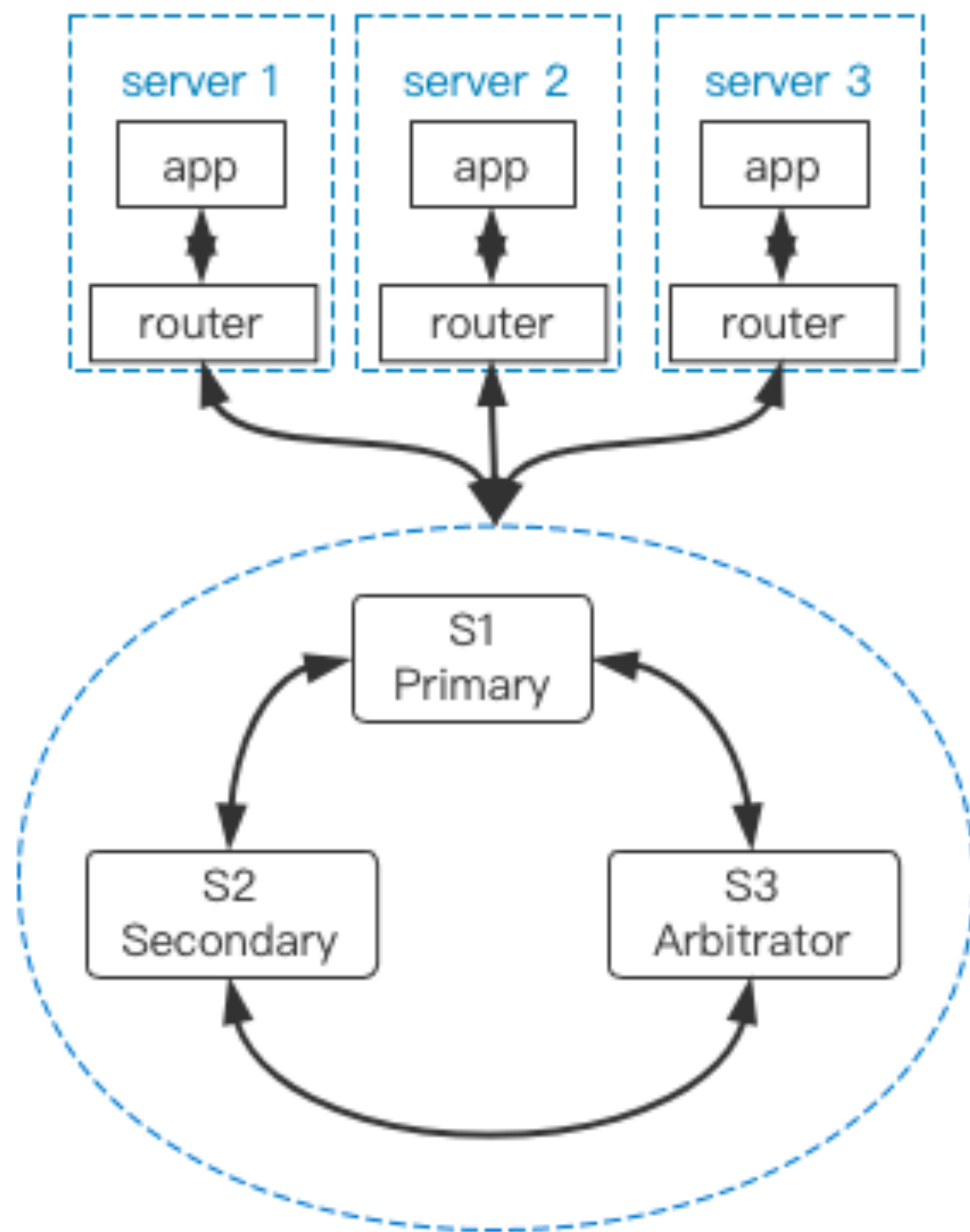


04 高可用架构

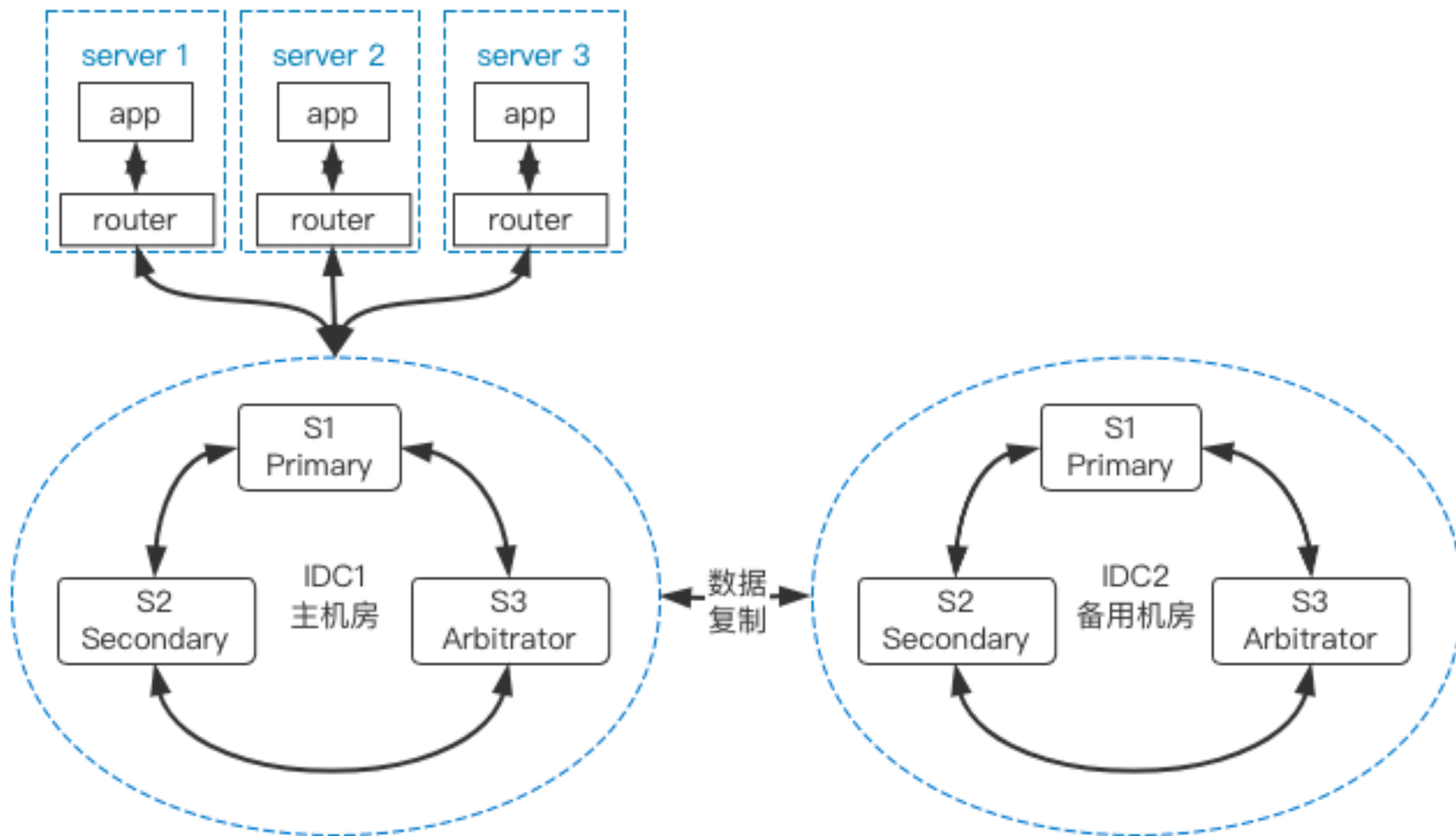




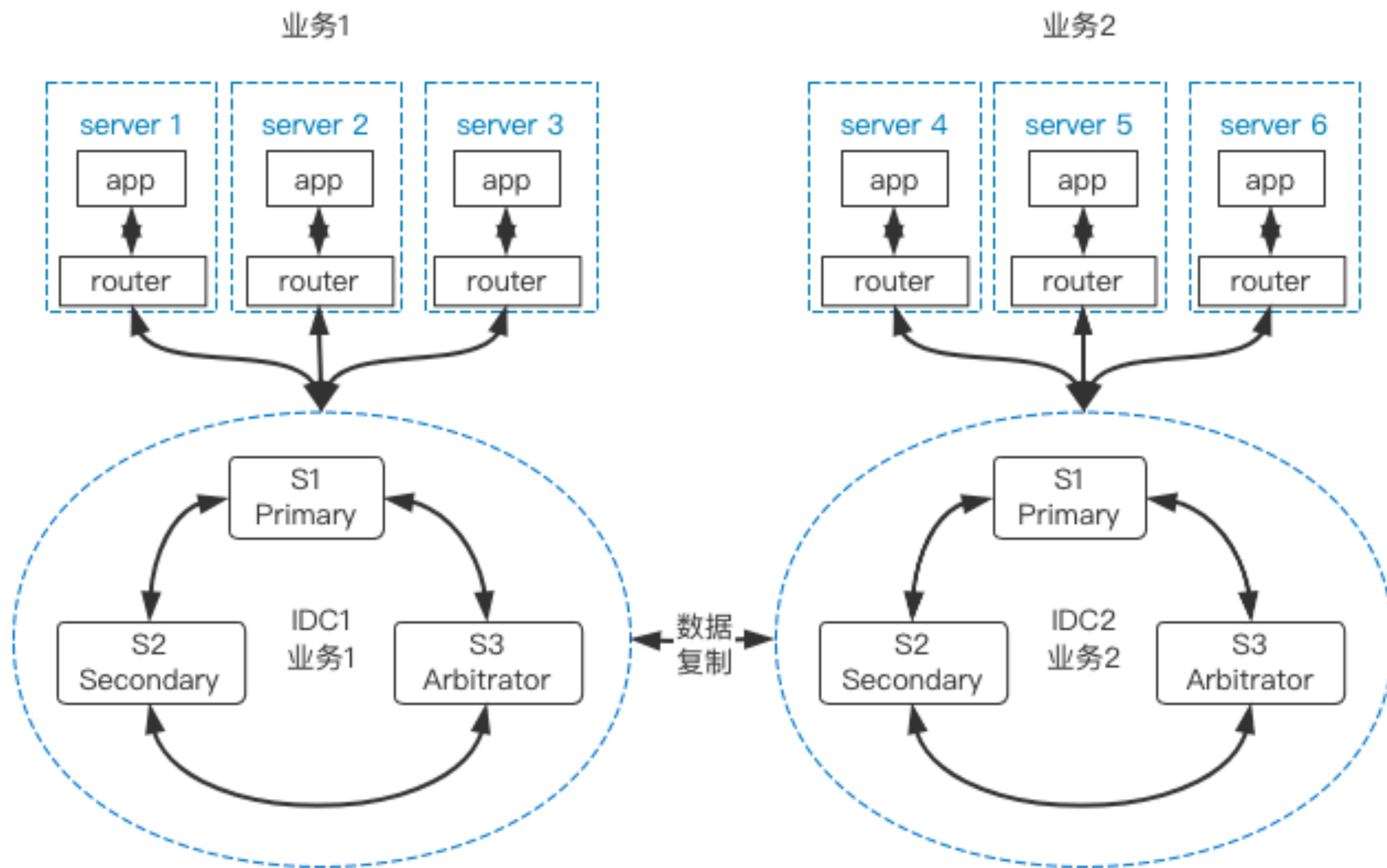


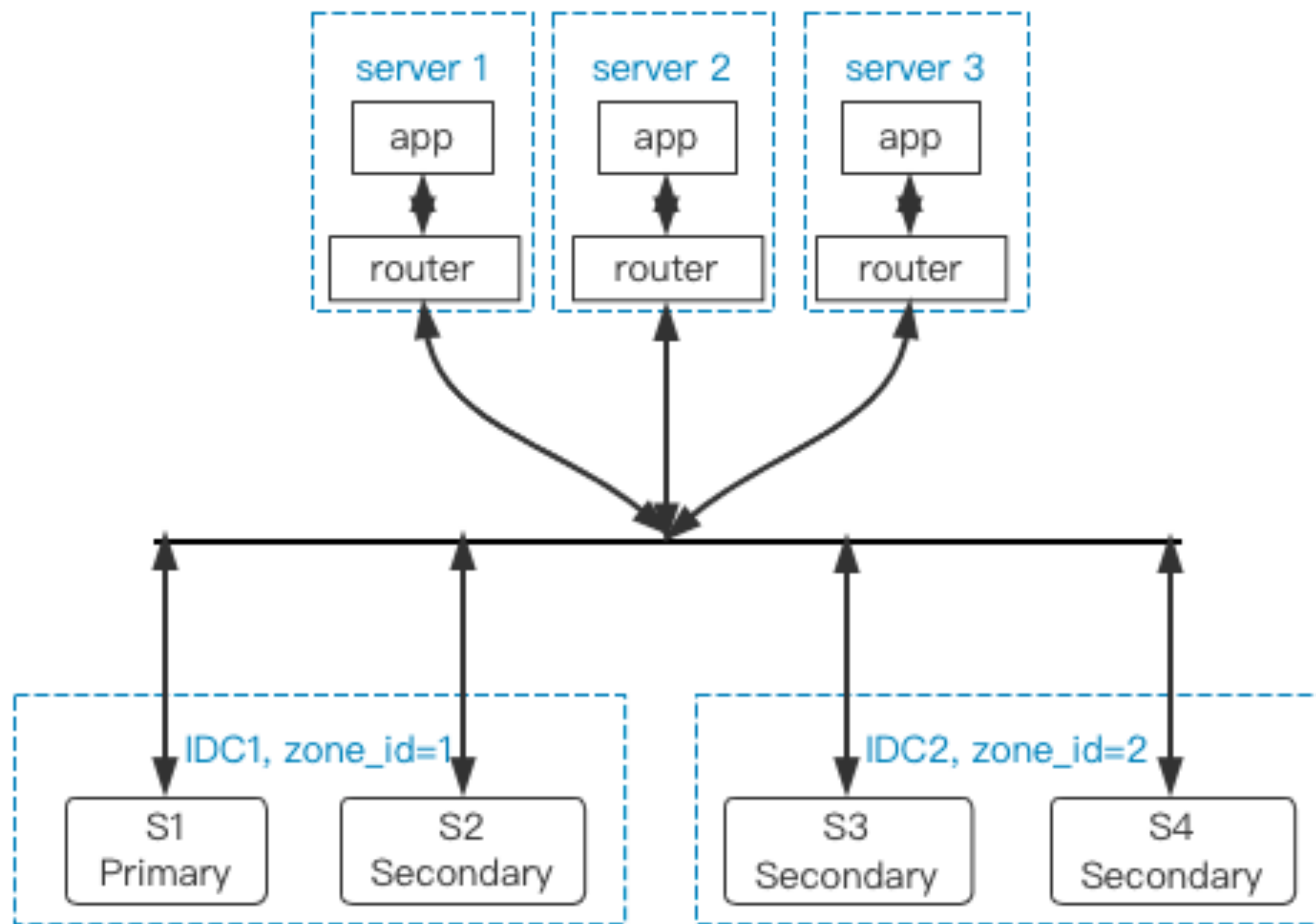


同城跨IDC高可用 - 主备架构

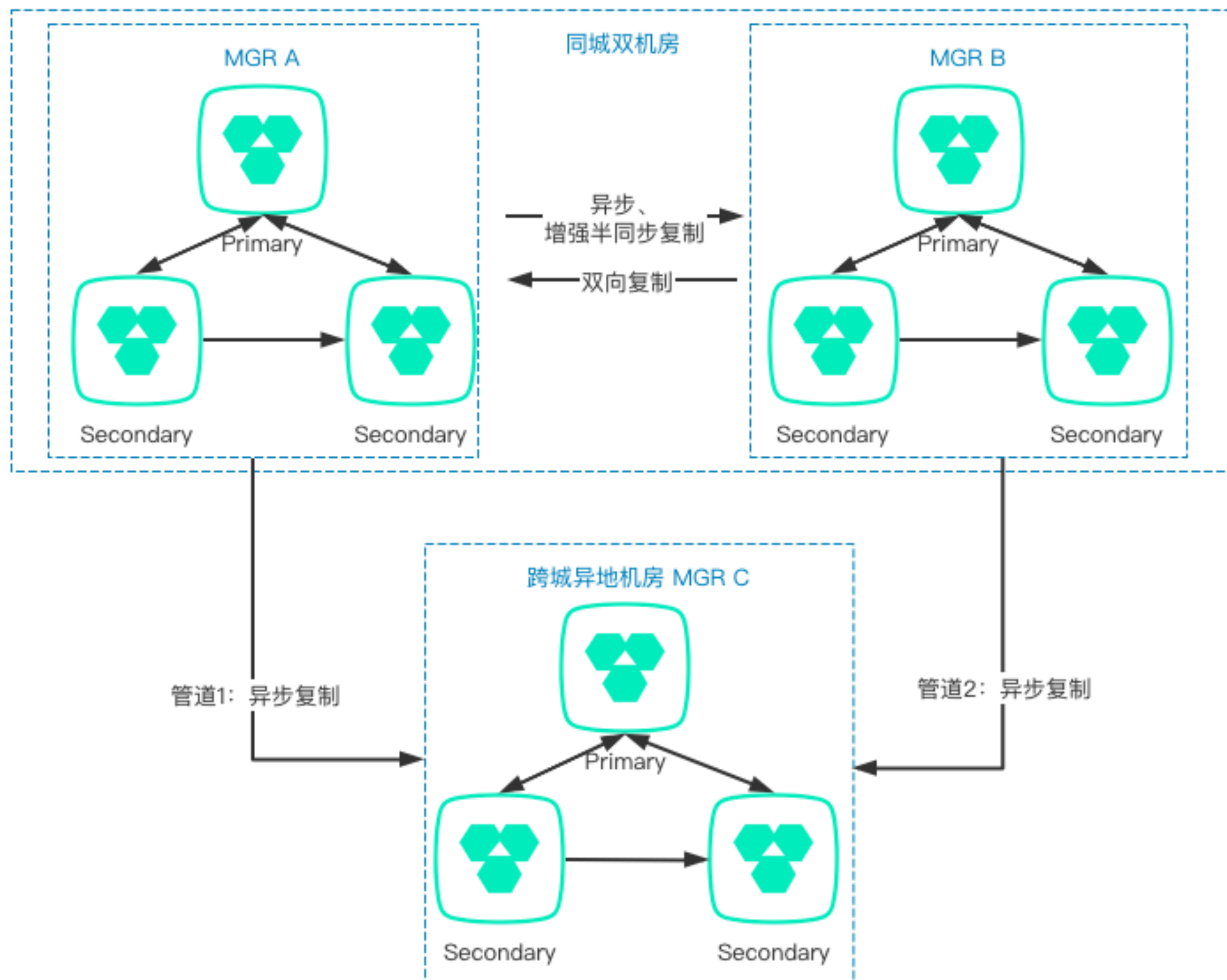


同城跨IDC高可用 - 垂直区分业务架构





跨城多IDC高可用





05 SQL优化



- Q1：数据库慢语句调优可以实现自动化吗？有什么方式
 - 没有全自动化的
 - 可以利用pt-query-digest汇总分析，再手动优化
 - 或者利用公有云的RDS产品

- Q2: 怎样对如oracle同步到mysql的数据稽核, 且判断数据不一致是否为同步延迟导致
 - 可以用gt-checksum工具做数据校验和修复

- Q3: 有没有什么提高数据库配置的方法。如通过获取主机参数等信息, 生成最优数据库配置。或者是交流会上根据一些场景、环境推荐如何配置
 - my.cnf生成工具 <https://imysql.com/my-cnf-wizard.html>
 - MySQL运行状态分析工具 <https://imysql.com/my-state-diag.html>

- Q4: 多表的时候索引怎么建比较高效, 有没有什么坑可以规避?
 - 多次DDL合并成一次
 - 不支持Online DDL的分开执行
 - Online DDL Operations, <https://dev.mysql.com/doc/refman/8.0/en/innodb-online-ddl-operations.html>
 - 通常, 不涉及到PK、FULLTEXT、SPATIAL索引, 基本上都支持Online DDL
 - 其他不支持Online DDL的场景: 修改列类型、转换字符集、表空间加密、大部分表分区相关操作

- Q5：本次GreatSQL最新版本的特性
 - InnoDB并行查询、并行load data、事务&锁优化
 - MGR增强，仲裁节点、地理标签、动态VIP、智能选主、快速单主、流控优化
 - SQL兼容
 - 国密支持，备份加密

- Q6: 关于数据库设计优化的一些思路
 - 窄表好 - 数据类型够用就好, 保持高效
 - 小表好 - 数据冷热分离, 只留最新热数据
 - 请求快 - 令每个事务/请求效率足够高, 不阻塞
 - 请求少 - 从产品需求端削减不必要的请求, 动态请求静态化或降频

GreatSQL

更流畅，更安心



成为中国广受欢迎的
开源数据库

