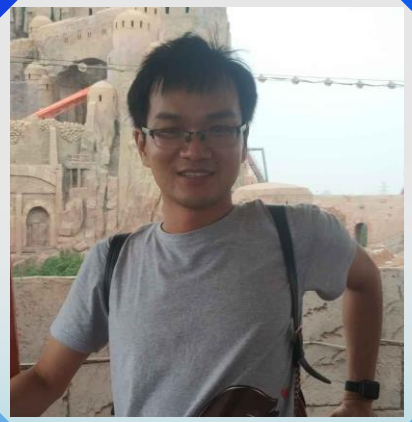
分布式数据库BaikalDB的设计与思考

李国强



李国强

百度商业平台部资深研发工程师

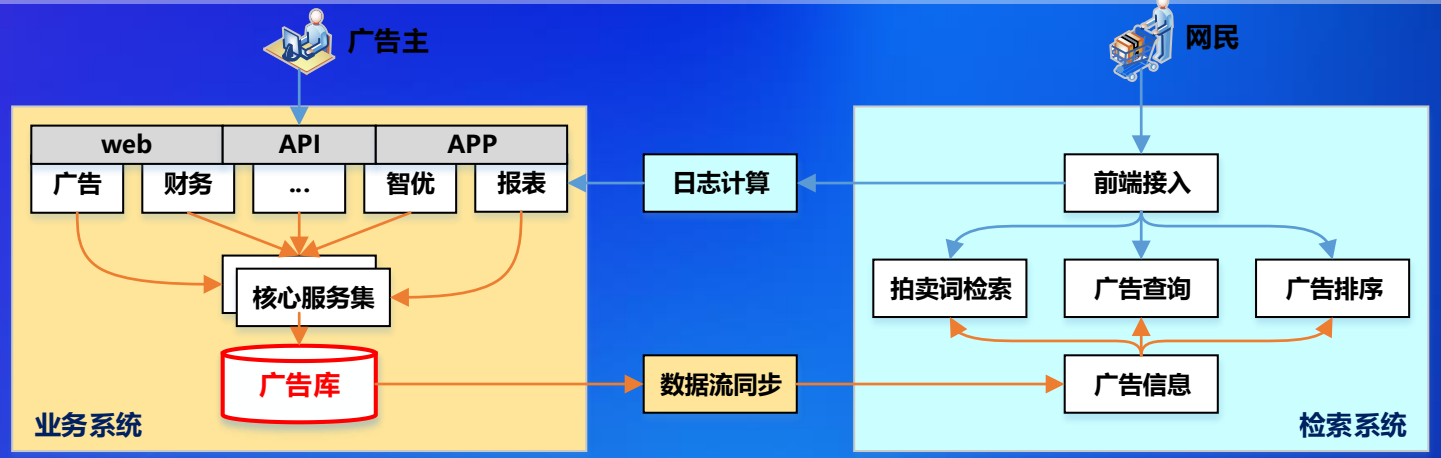
2012年加入百度，主要负责存储方向工作，包括分布式数据库系统BaikalDB，分布式SQL中间件等

对构建高性能、高可用分布式系统有较多实践和较深入的的理解。

目录

广告业务系统的特点

广告业务系统是以广告库为核心的大型复杂商业系统



广告库对存储设施的需求

- ✓ 稳定压倒一切
- ✓ 数据不一致的容忍度低
- ✓ 数据规模持续增长
- ✓ 业务需求多样化：OLTP，OLAP，正反KV查询，层级查询，模糊检索

业务高速发展,带动存储架构的演进

数据规模：百万级→千亿级
流量规模：千万级→十亿级
集群规模：接近百倍

应用层分库分表　定制+云化　Cloud Native+NewSQL

围绕MySQL生态的存储架构演进-主库拆分

通过主库拆分使得DB能力（存储、吞吐、低延迟等）持续扩展，消除系统瓶颈

2010年，1拆4

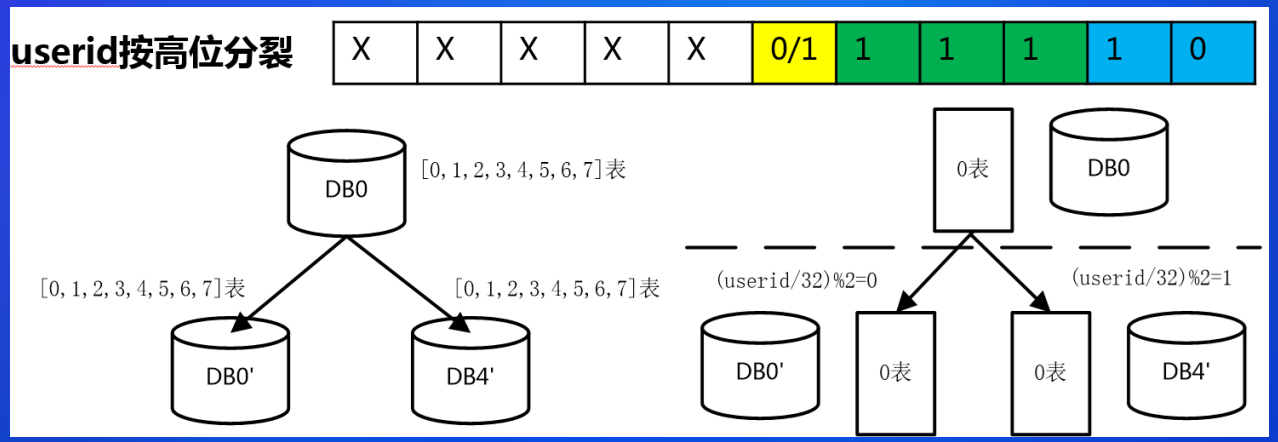
2011年，4拆8

2012年，8拆16

2013-14年，优化没拆

2015年，16拆32

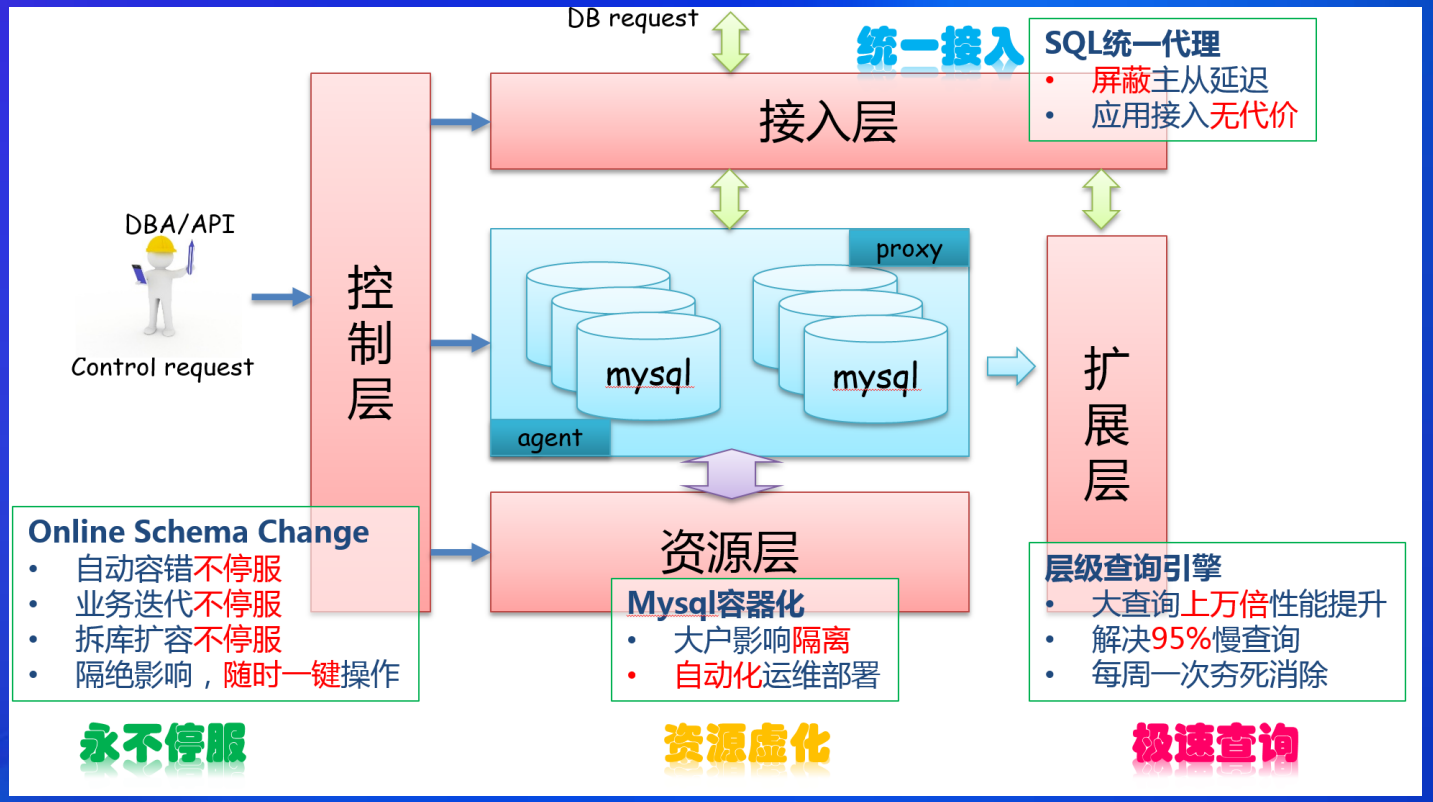
分库分表方式—裂变



- ✓ 完美解决**新老户膨胀**问题
- ✓ 奇偶分裂，负载均衡
- ✓ 支持**局部分裂**，灵活应对单库膨胀
- ✓ 衍生不跨片，先裂后删

围绕MySQL生态的存储架构演进-定制化

大规模定制化的商用MySQL集群

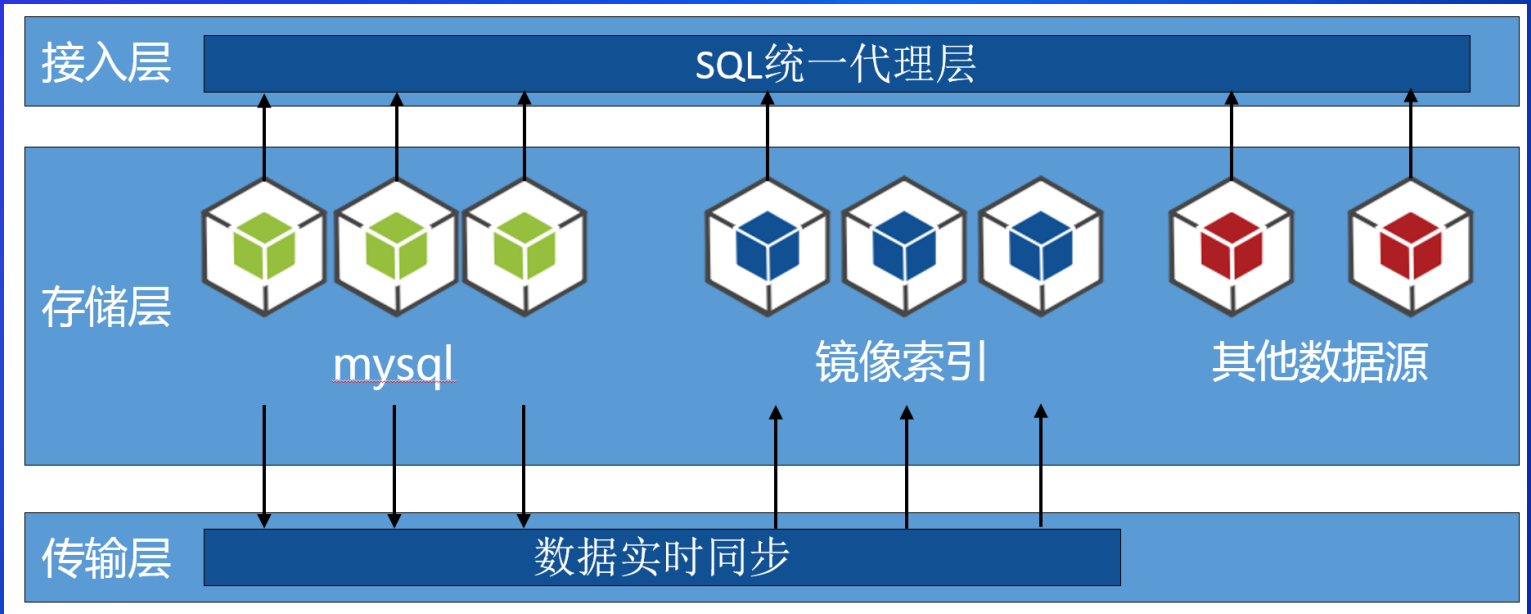


- ✓ **控制层**：指挥DDL、拓扑变化等无损调整，对外永不停服。
- ✓ **扩展层**：外挂存储引擎，实现特定SQL pattern的极速查询。
- ✓ **资源层**：基于容器实现DB实例虚拟化，自动化部署与资源隔离。
- ✓ **接入层**：屏蔽数据源，屏蔽主从延迟影响。

围绕MySQL生态的存储架构演进-定制化

2016年的广告存储架构形态-异构复合存储

- ✓ MySQL为主存储，通过数据同步建立镜像索引，用于加速查询
- ✓ 统一SQL代理，对业务屏蔽存储细节，基于SQL pattern分发请求



劣势：

- ✓ 同步延迟时，一致性难以保障
- ✓ 资源冗余开销大

围绕MySQL生态的存储架构演进-2017年的思考

业界标杆-Google AdWords的核心存储： F1/Spanner

- ✓ 分布式强一致性事务
- ✓ 高扩展性（自动分片存储）
- ✓ 高可用（故障自愈）
- ✓ 跨数据中心多活

基于MySQL深度定制的挑战

- ✓ MySQL的单机架构无法原生的实现分布式化和虚拟化
- ✓ 扩展的镜像索引能提升查询性能，但冗余成本高，数据一致性差

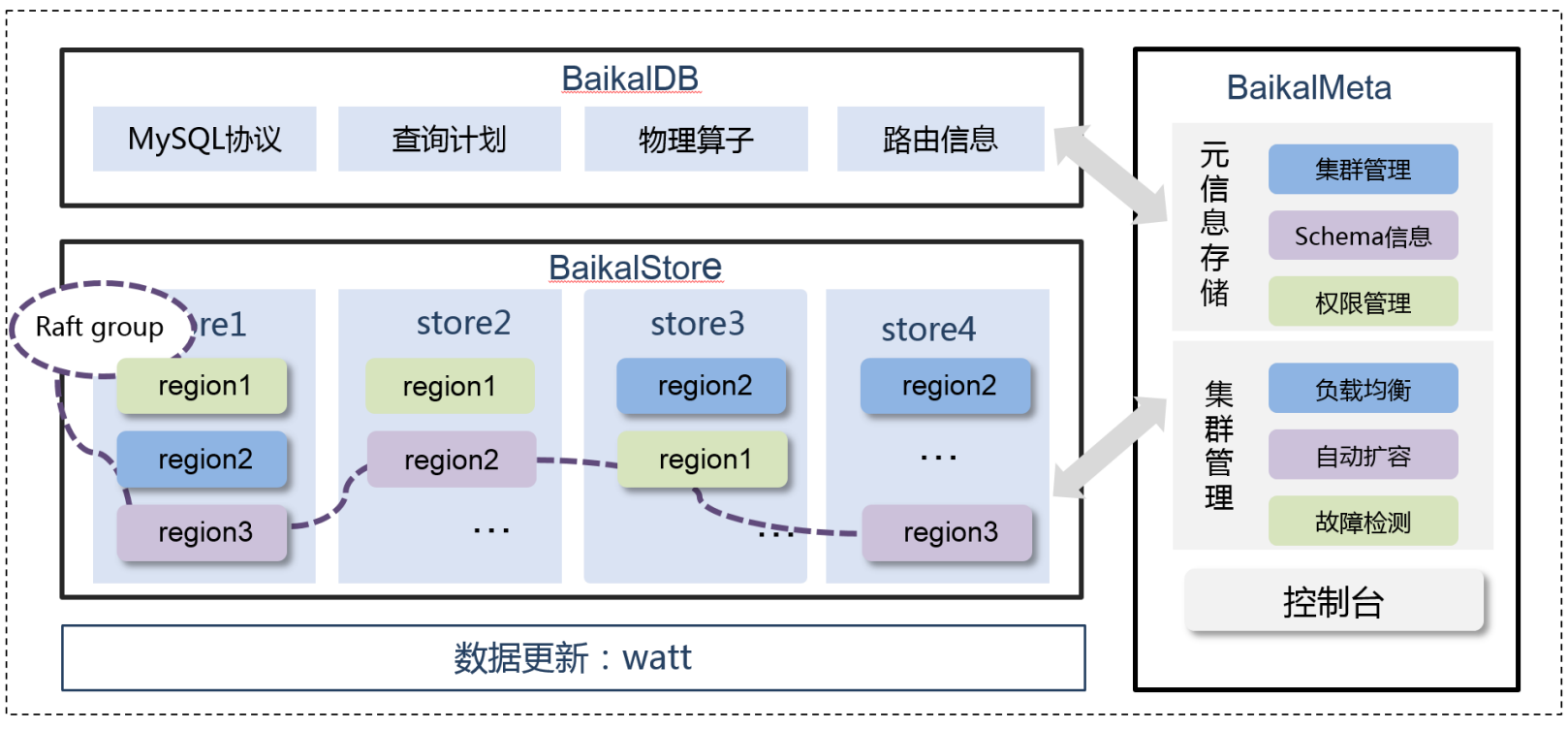
围绕MySQL生态的存储架构演进-2017年的思考

研发一个分布式+云原生+多样化索引架构+强一致 VS 寻找一个开源项目加以改造

团队特点	C++技术栈，研发过SQL代理层+定制化存储，熟悉MySQL协议，丰富的工程经验
基础设施	RPC通信框架：brpc，百度内最常使用的工业级RPC框架 一致性协议框架：braft，基于brpc的raft协议工业级实现 单机KV引擎：RocksDB, 基于LSM-Tree的高性能KV引擎
开源项目	CockroachDB（起步阶段），Impala（OLAP的SQL引擎），TiDB（2017.10发布1.0）
结　　论	基础设施很可靠，从0构建一个数据库看起来也不难，开源项目成熟度一般，那就自己干！

新一代分布式数据库-BaikalDB

BaikalDB是面向商业业务系统的新一代存储系统：baikaldb.com



- ✓ **全自主**管理，线性扩展
- ✓ **高可用**，自动故障恢复和均衡
- ✓ 兼容MySQL协议
- ✓ Online Schema Change
- ✓ 局部+全局二级索引
- ✓ 分布式事务，多表Join能力
- ✓ 支持全文检索

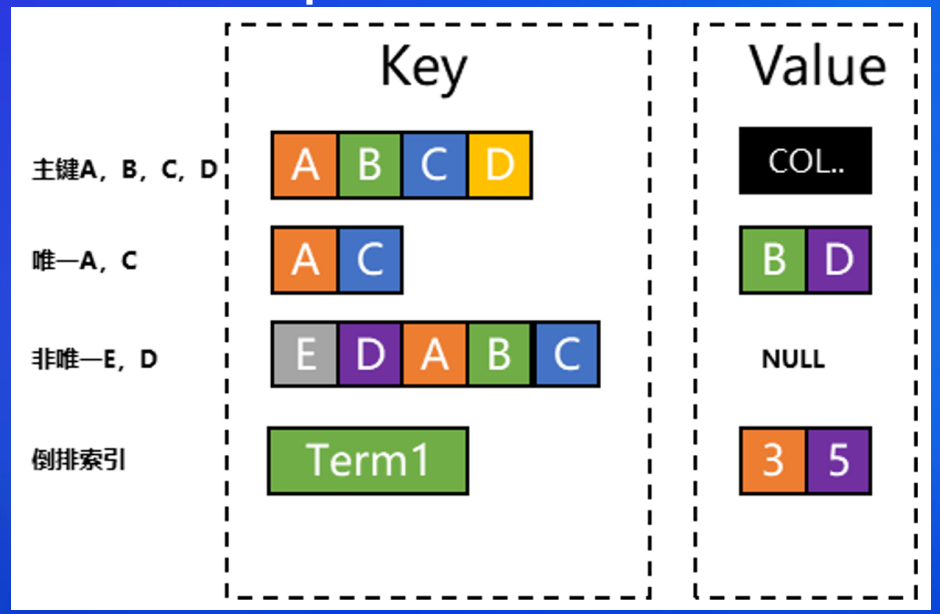
新一代分布式数据库-BaikalDB

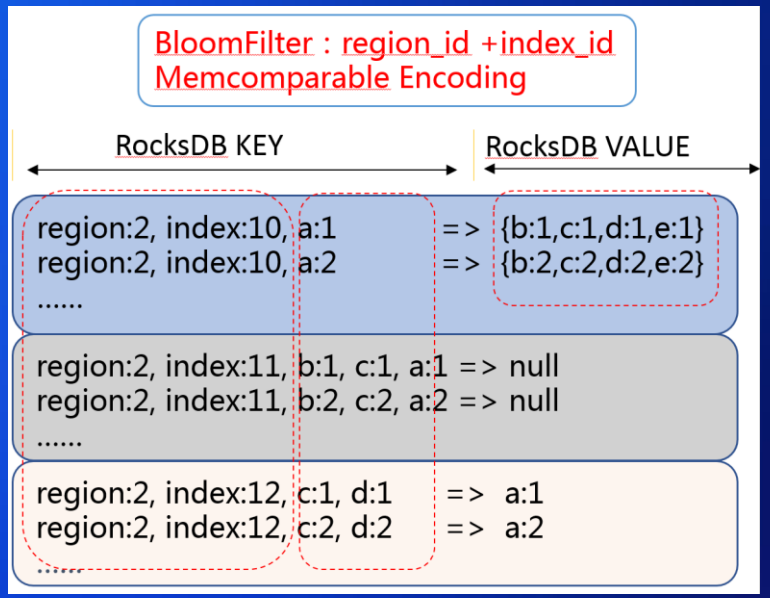
✓分布式存储系统的三大核心设计要素

- ✓ **存储**：数据结构化的存储，一致性算法
- ✓ **计算**：SQL执行过程，查询计划的优化
- ✓ **调度**：分布式集群管理，元信息管理

新一代分布式数据库-核心设计-存储

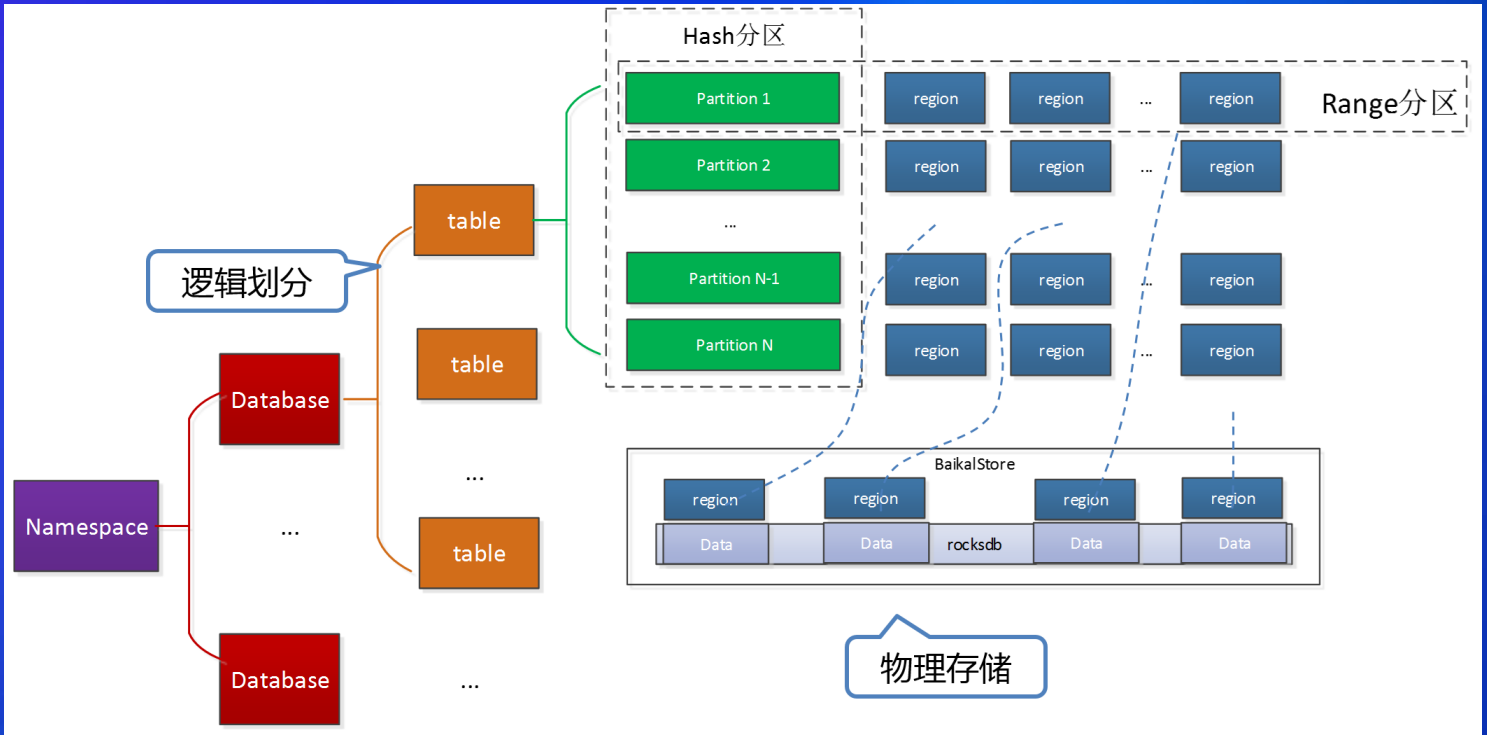
- ✓ RocksDB作为KV底层存储
- ✓ 支持索引：主键聚簇索引、联合索引、唯一索引、普通索引、倒排索引
- ✓ Value采用pb格式编码，业务可以灵活加列
- ✓ 动态解析pb格式，无需解析所有列





新一代分布式数据库-核心设计-存储

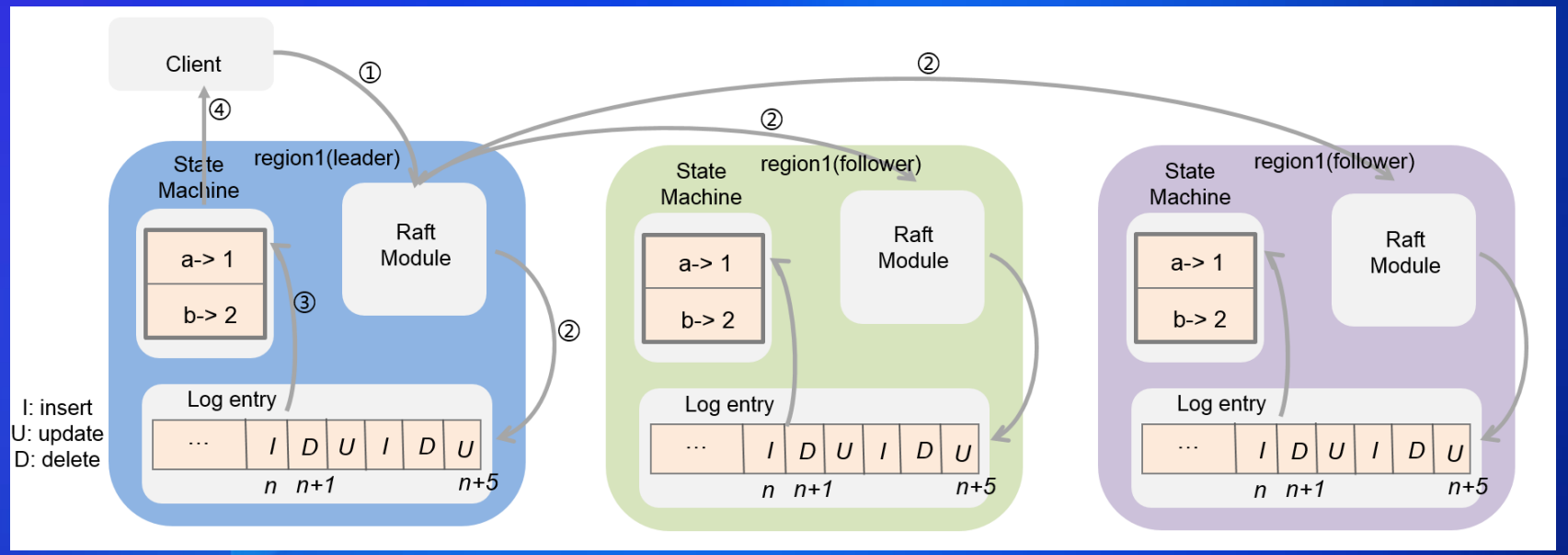
广告系统天然的层级数据模型，适合Partition+Range分区



- ✓ Range方便自动扩容，范围筛选简单
- ✓ 按层次聚簇自动聚集小户，拆分大户
- ✓ 物理部署与逻辑无关，简化调度

新一代分布式数据库-核心设计-存储

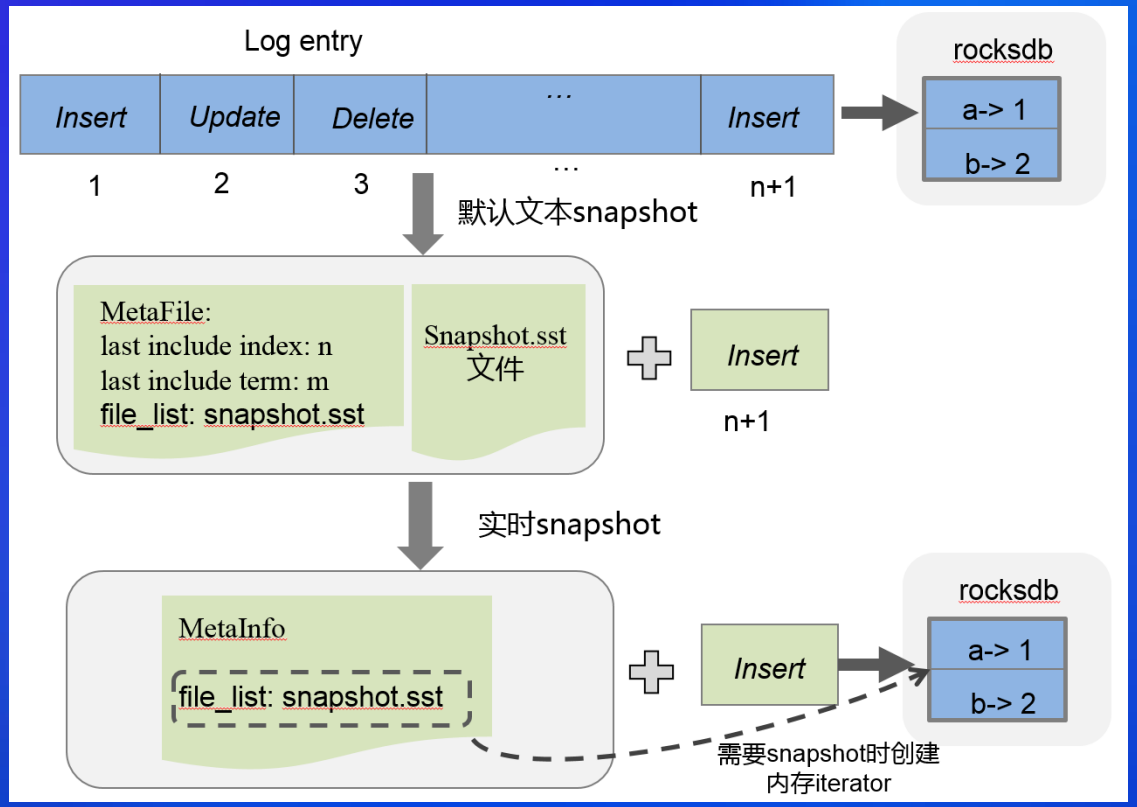
基于Raft保证多副本（Replica）的数据一致性



- ✓ 强一致性协议
- ✓ 日志复制状态机
- ✓ log_entry严格串行执行
- ✓ 业务保证执行一致

新一代分布式数据库-核心设计-存储

Raft实时snapshot优化



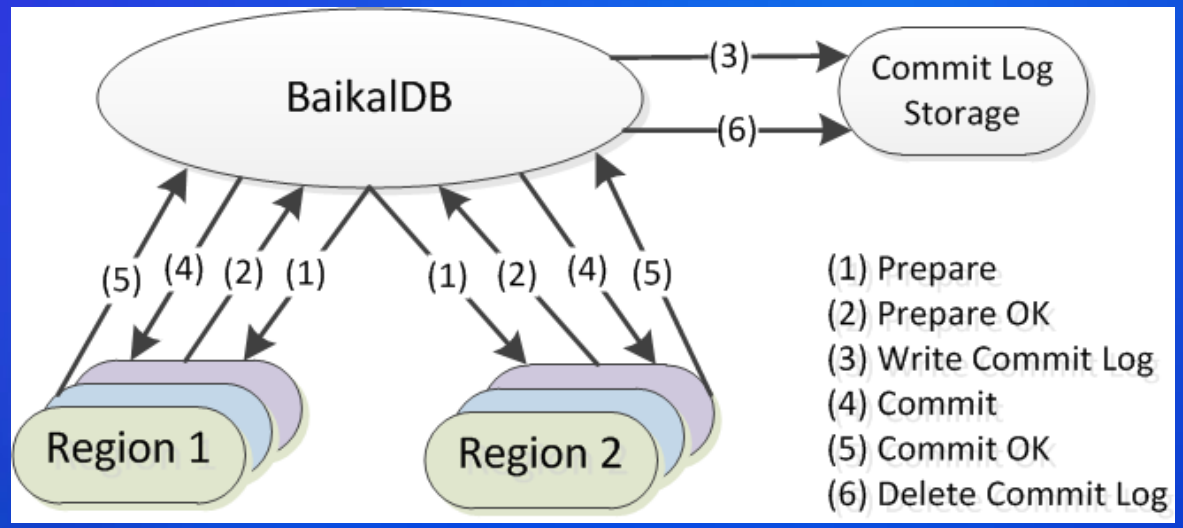
Multi-raft架构下的meta和log优化

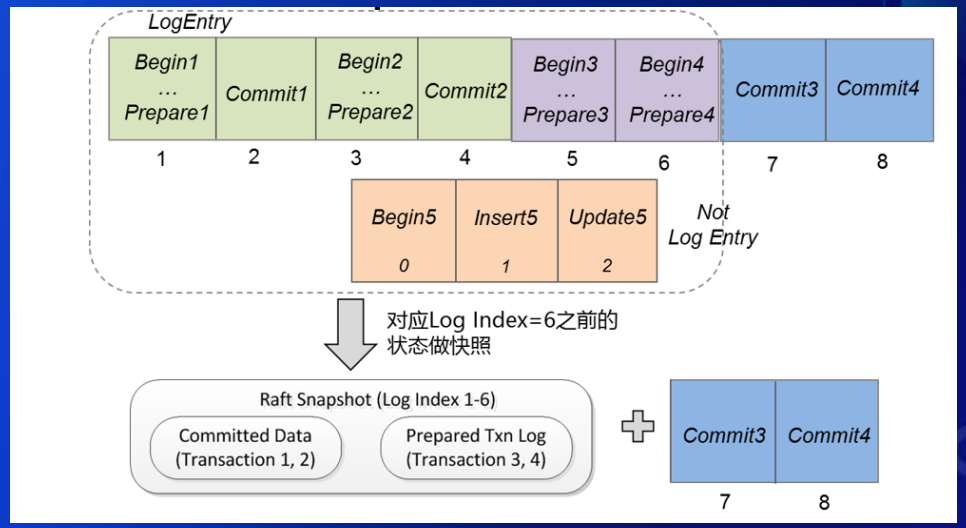
- ✓ 几千到上万个region，region共享磁盘
- ✓ 单region需要持久化两类信息：meta 和 log entry
- ✓ 改为采用rocksdb存储信息，顺序写
- ✓ 异步删除，log compaction操作转为rocksdb的compaction

新一代分布式数据库-核心设计-存储

基于2PC保证多分片（Region）的数据一致性

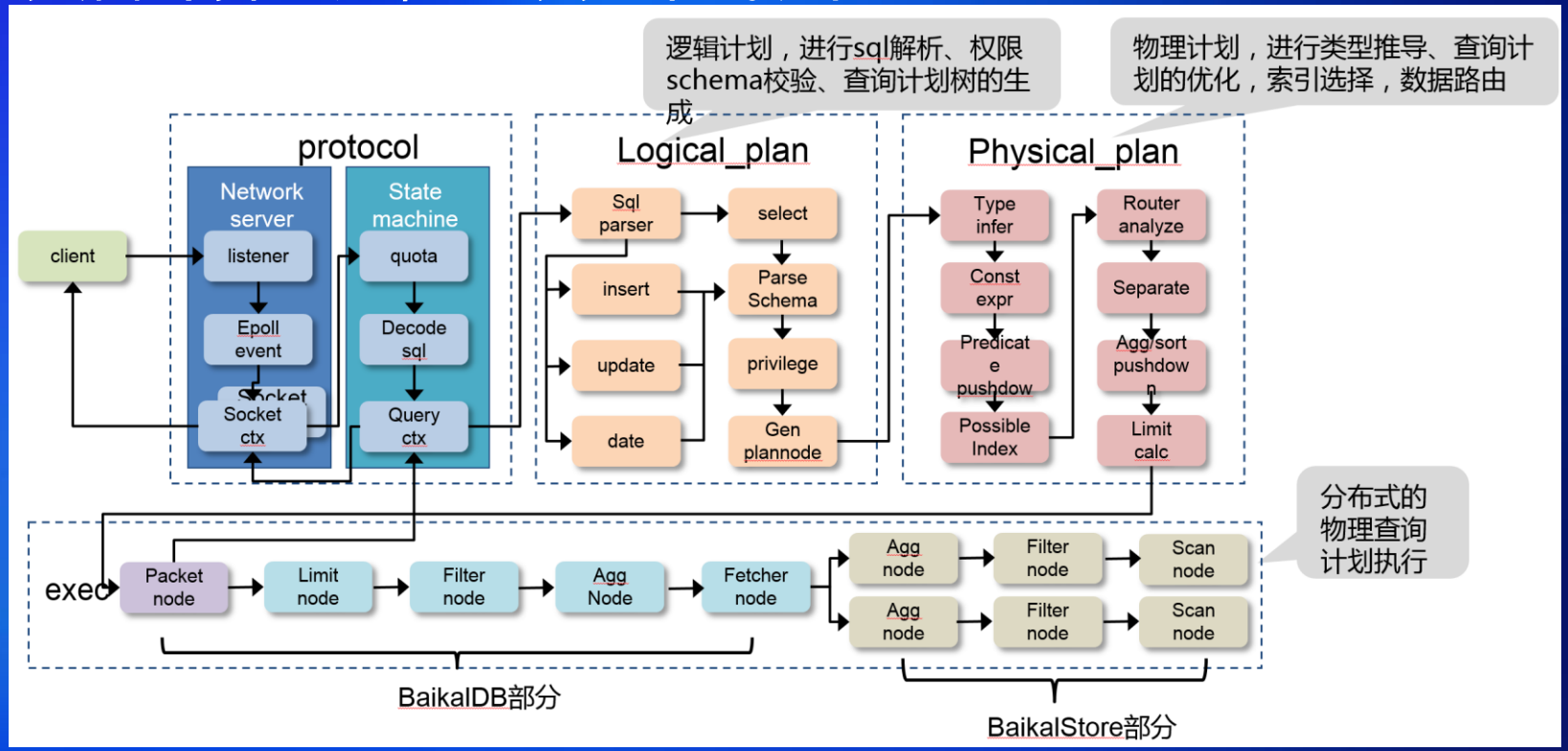
- ✓ 结合RocksDB单机事务的两阶段提交
- ✓ BaikalDB模块兼任协调者
- ✓BaikalStore的实时Snapshot存储已commit数据和Prepare命令
- ✓重启后重放Snapshot恢复





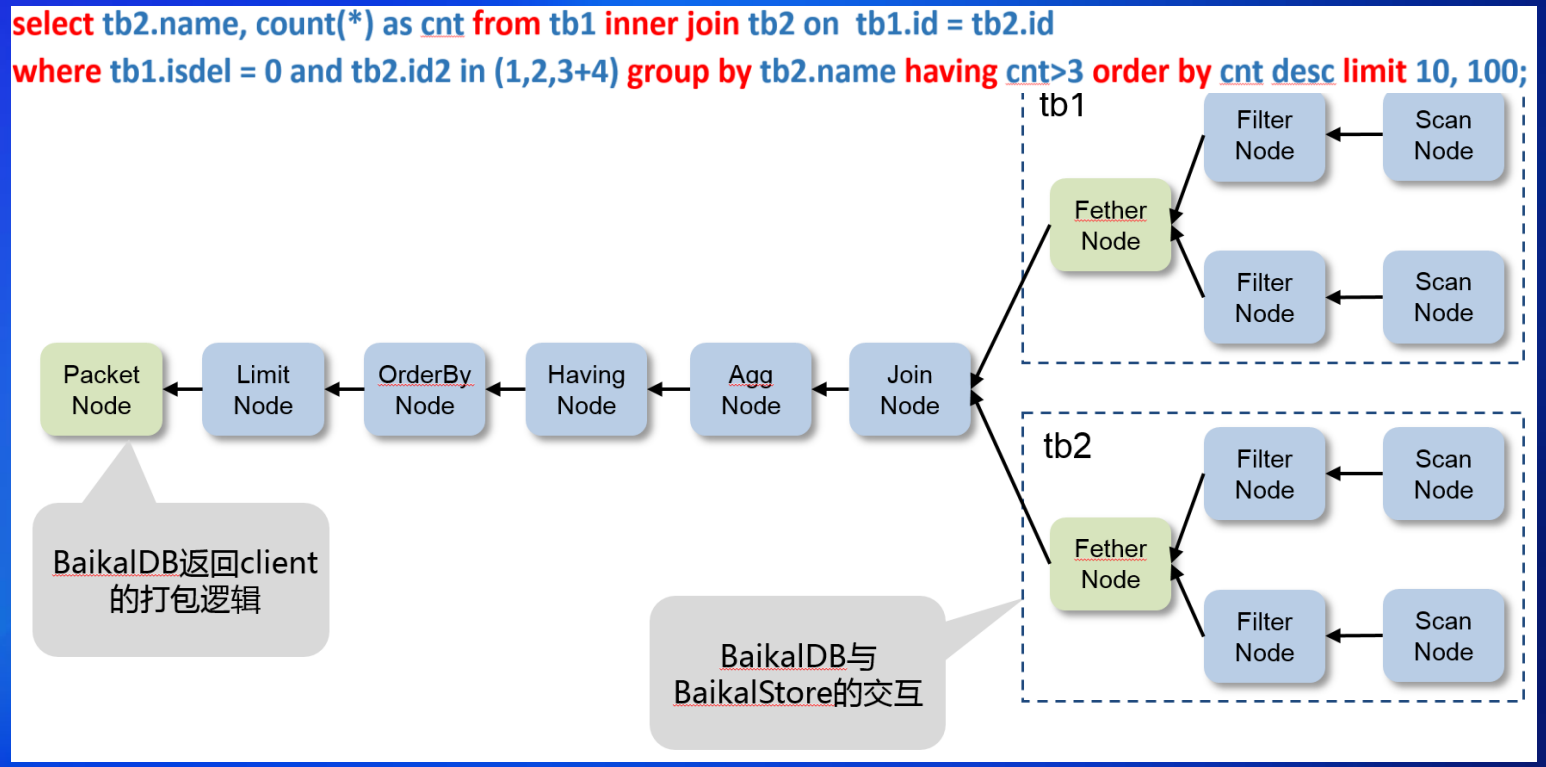
新一代分布式数据库-核心设计-计算

- ✓分层架构降低SQL处理复杂性
- ✓存储节点具备计算能力，可以分布式执行



新一代分布式数据库-核心设计-计算

- ✓ 一切皆算子，灵活拼接，高可扩展
- ✓ 经典的火山模型, 每个算子提供open、next、close操作

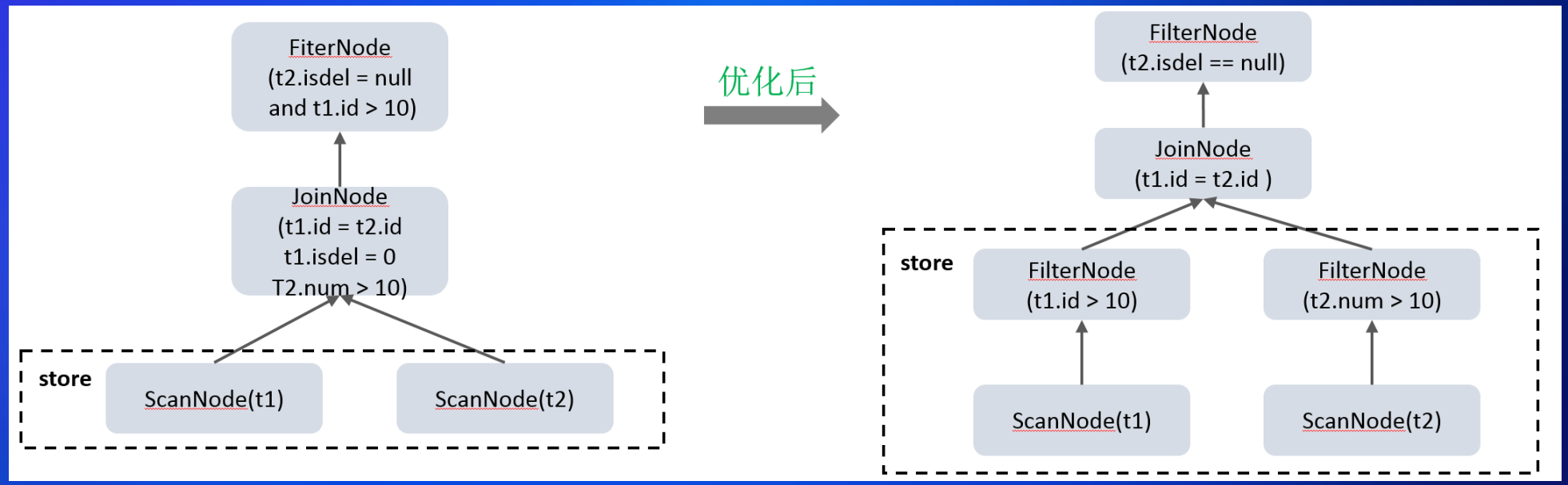


新一代分布式数据库-核心设计-计算

分布式查询框架关键点：查询优化

索引覆盖　　谓词下推　　投影下推

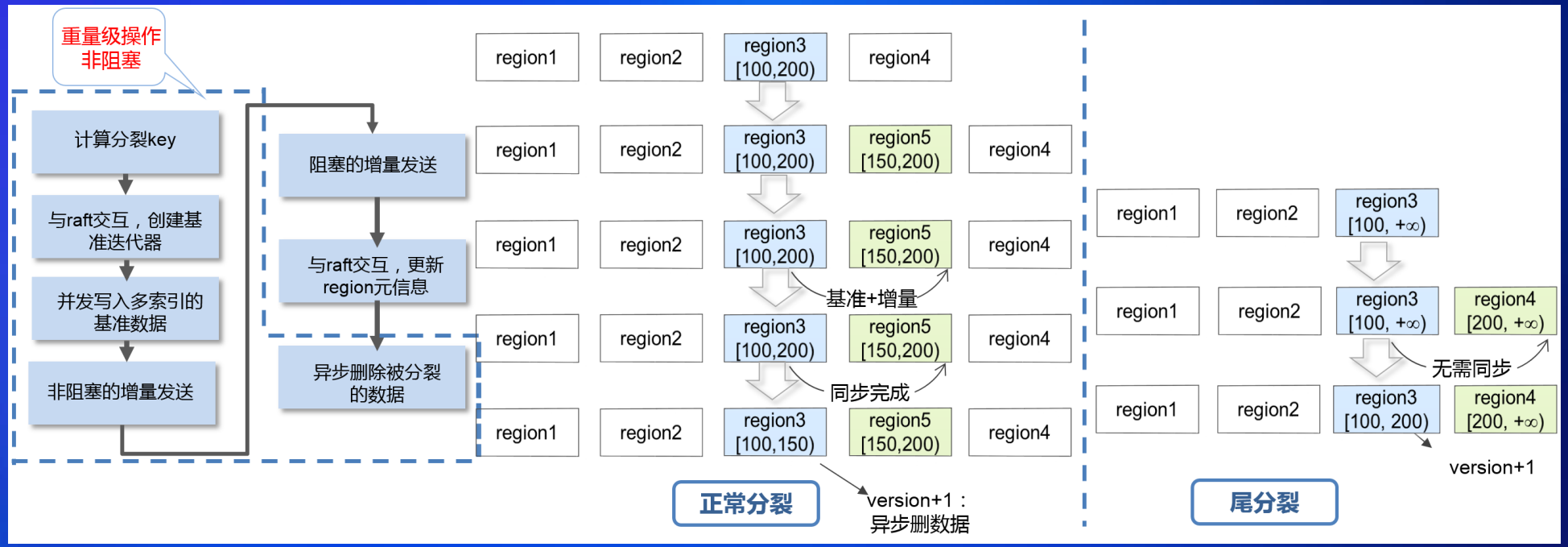
limit下推　　Join Reorder　　常量表达式预计算



新一代分布式数据库-核心设计-调度

对业务透明的自动分裂机制

单region过大→性能下降，重启恢复慢　　核心思想：基准 + 增量



新一代分布式数据库-核心设计-调度

自动故障检测与自动迁移保证系统高可用

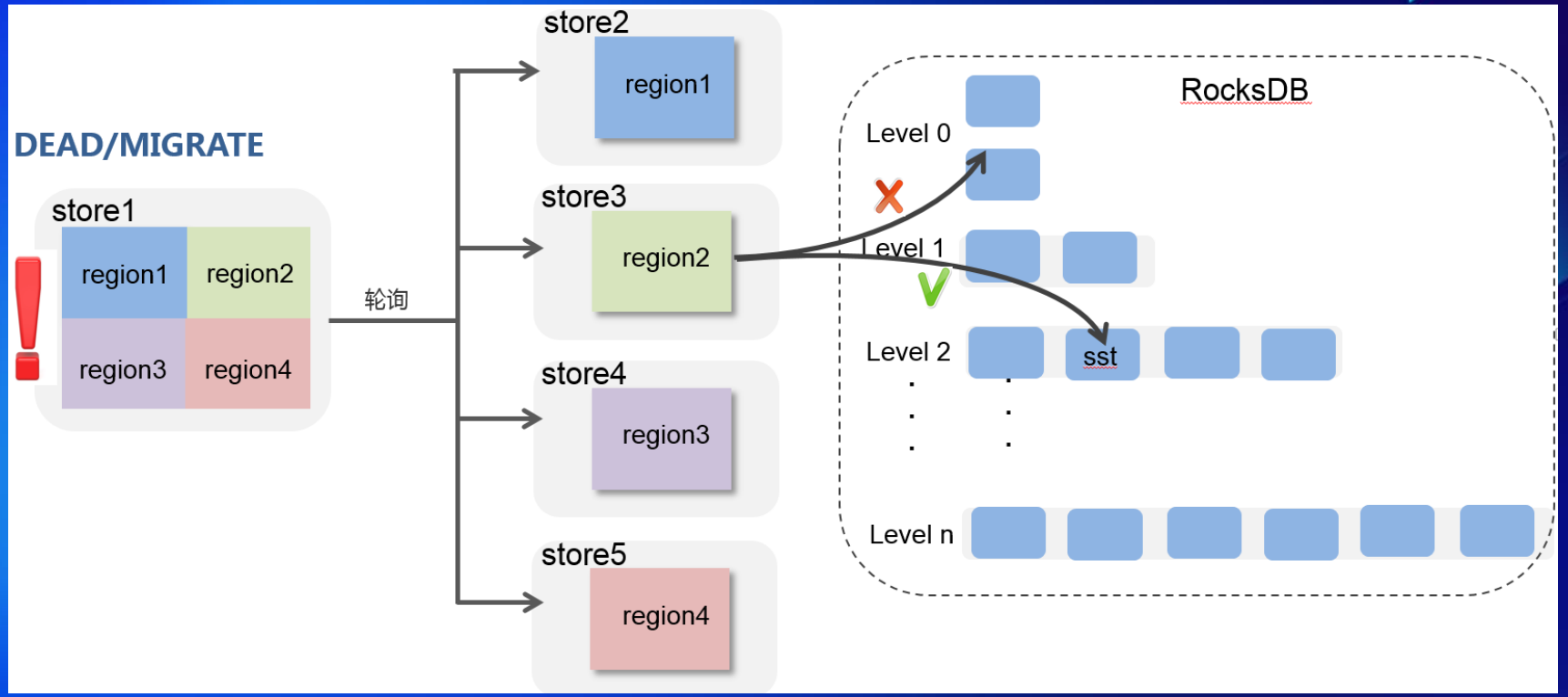
故障检测：

FAULTY/DEAD

数据迁移：

轮询分散压力

INGEST到合适层

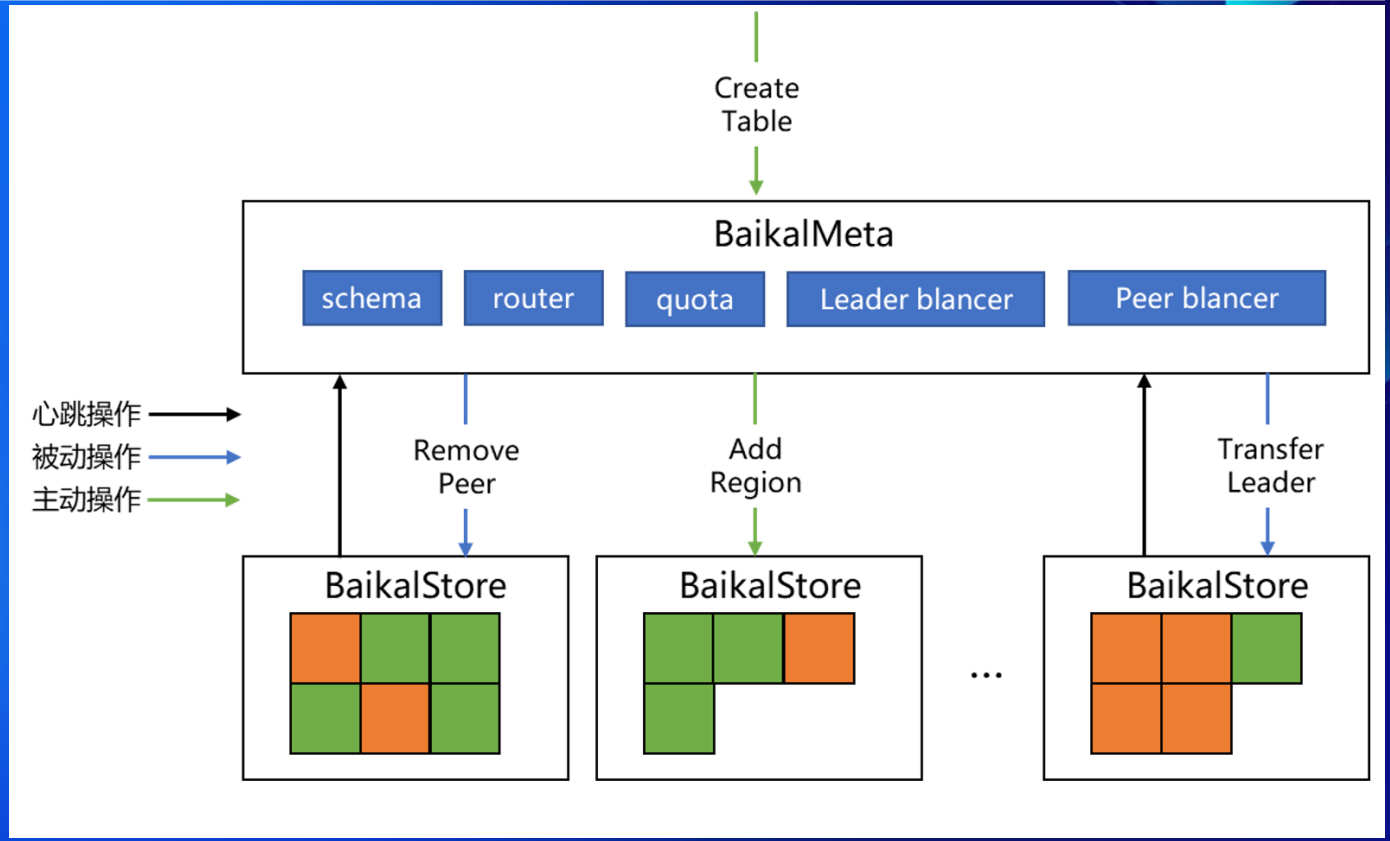


新一代分布式数据库-核心设计-调度

完善的负载均衡机制对整体系统的性能至关重要

两种负载均衡策略：

- ✓ Leader均衡：均衡计算
- ✓ Peer均衡：均衡存储



自主运维的思考

存储规模不断扩大，运维难度不断增加

业务方无法正确建立索引、查询需求增多导致旧索引无法支撑

基于规则的优化器无法正确选择索引

业务大规模混布，集群变慢无法快速定位根源

自主运维的思考

高效的TRACE降低运维难度

- ✓ 聚合sql trace，找出拖慢集群的sql
- ✓ 单sql trace，找出sql性能瓶颈
  - 统计扫描行数，过滤行数，索引情况，每个算子时间

pv↓	平响 ⇕	SQL ⇕
2,909,649	40,725	DELETE FROM FC_Content.ideacontent_new WHERE ((((account_id = ?) &&
1,404,890	5,953	REPLACE INTO cluster_status.region(region_id, table_id, parent, table_name sed_size, num_table_lines, main_table_id, raw_start_key) VALUES (?)
261,585	3,881	UPDATE FC_Content.word_expect_rank SET modtime = ?, top_expect_rank

经验总结

- ✓基于优秀的开源组件和参考优秀系统的设计
- ✓基础设计不能妥协：存储格式，索引格式，支持类型等
- ✓有效的方案：基准+增量
  - 分裂，DDL，meta更新，raft复制
- ✓高效Profile：火焰图，DOT图

ctrip tech Ctrip 携程

github.com/baidu/baikaldb

欢迎STAR和提出宝贵意见~

Thanks For Watching

本PPT来自2019携程技术峰会

更多信息请关注“携程技术中心”微信公众号~