SACC

卓越 5周年 变迁

2013中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2013

大数据下的IT架构变迁

阿里云分布式RDS平台

柳彦召@阿里云

yanzhao.liu@alibaba-inc.com

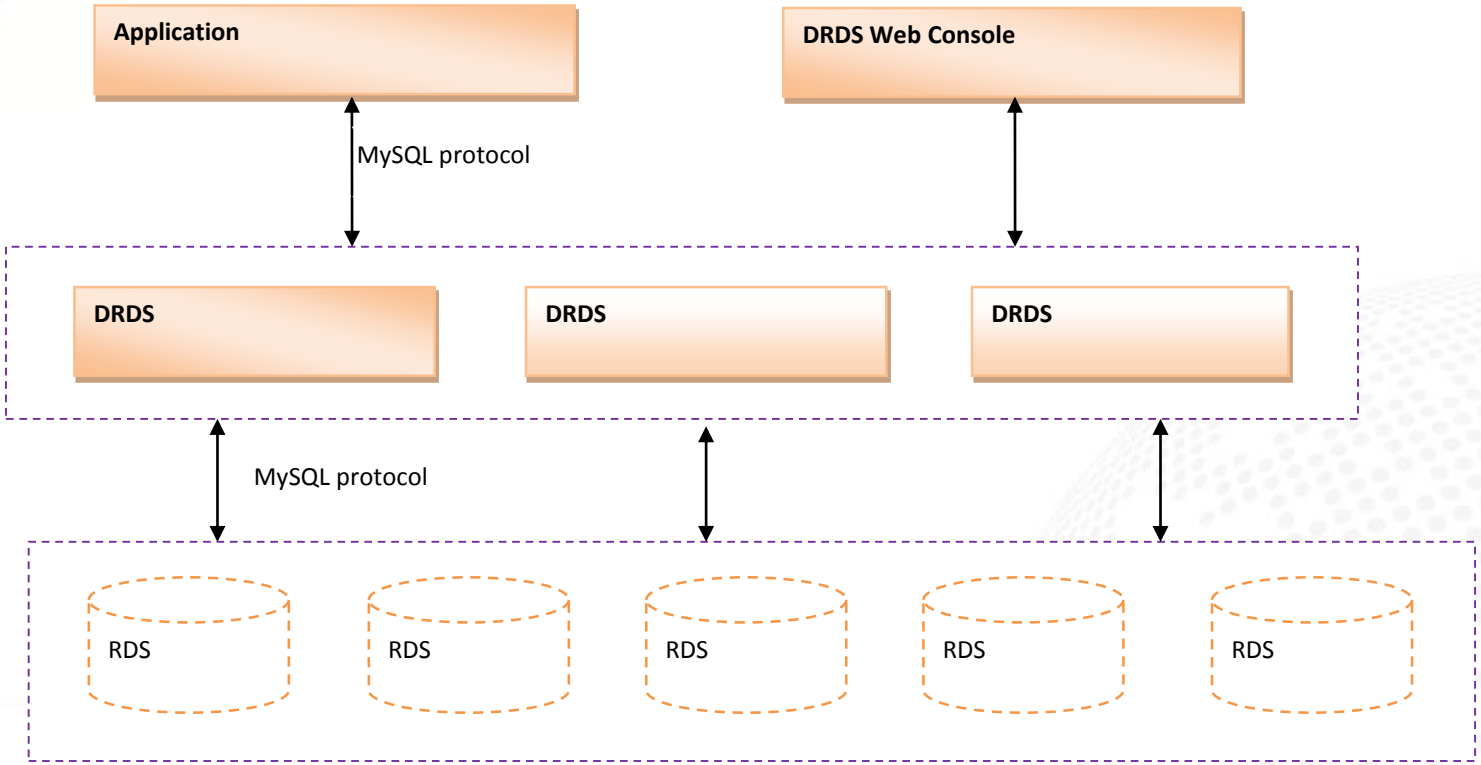
DRDS简介

- 阿里云分布式RDS平台（DRDS）是基于RDS的面向海量数据和高效访问的通用存储解决方案。
- 核心价值
  - 为用户提供透明的分库支持
  - 自动化扩容，用户根据业务需求按需添加或者删除结点，由平台负责数据的均衡
  - MySQL协议的兼容性，使基于MySQL的单库业务可以平滑的迁移到多库上
- 应用场景
  - Scale up，新一型>新二型>新三型...
  - Scale out，突破容量限制以及tps限制
  - 优先scale up，切勿过度设计

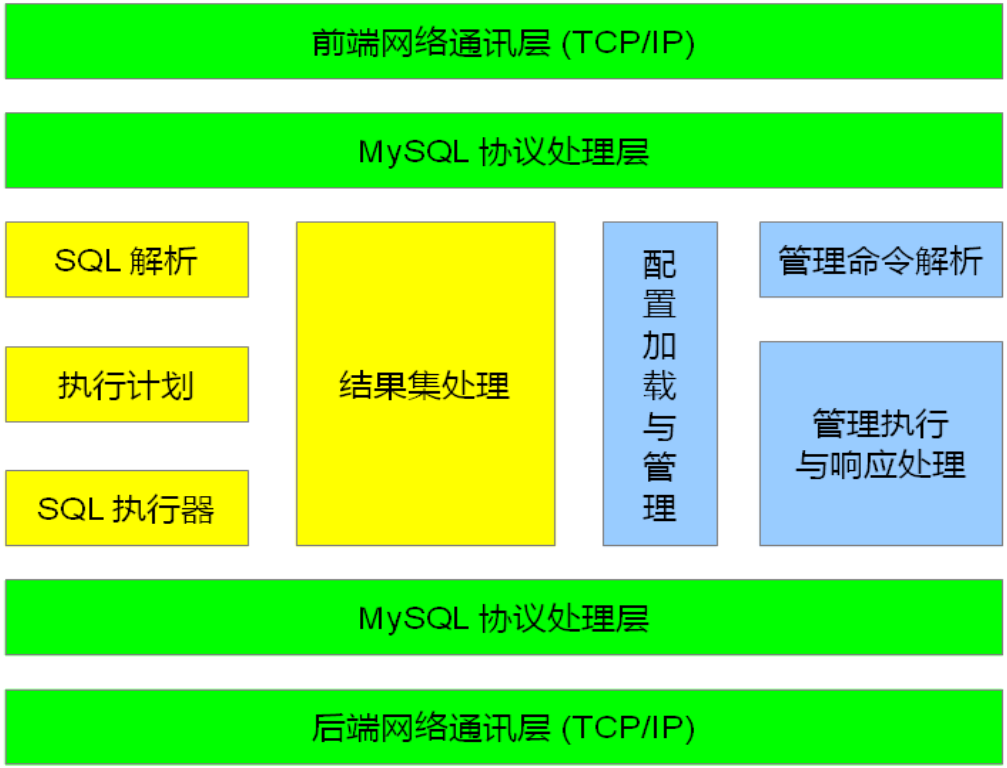
系统起源

- 脱胎于Alibaba的Cobar分布式数据库引擎
- 吸收了淘宝 TDDL分布式数据库引擎的优秀经验和部分解决方案
- 针对外部应用特色进行了部分优化开发
  - NIO异步执行器，有效隔离不同用户之间的影响，极大的提升了系统在复杂环境下的稳定性
  - 更完善的跨库查询支持
  - 全自动扩容，扩容全部通过web完成
  - 高效的管理，数据库运维操作完全通过web完成

系统架构—外部


Application
DRDS Web Console
MySQL protocol
DRDS
DRDS
DRDS
MySQL protocol
RDS
RDS
RDS
RDS
RDS

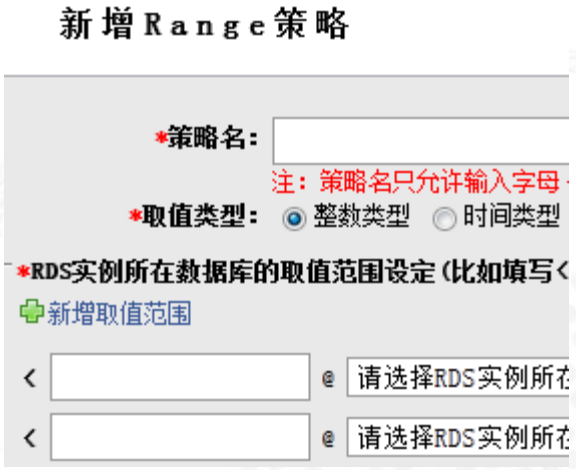
系统架构—内部



模块架构图

Sharding实现

- 拆分字段，表以那个字段的值作为分库标准
- 分布式策略
  - Hash策略，热点，无法指定结点
  - List策略，配合Hash策略使用
  - Range策略，历史数据

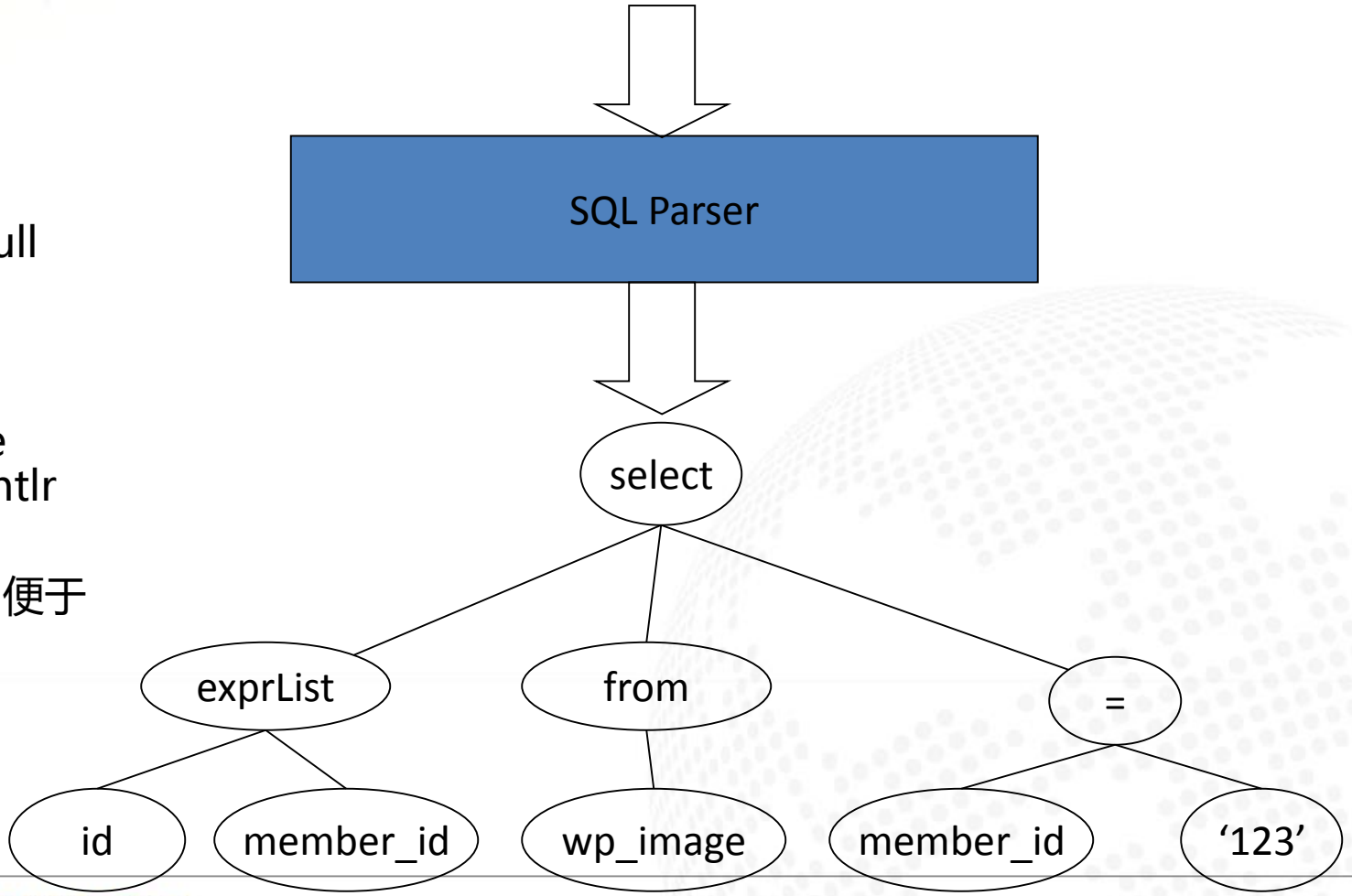


- 表与策略：多对一

查询—SQL parser

```
Select id, member_id from wp_image where member_id = '123'
```

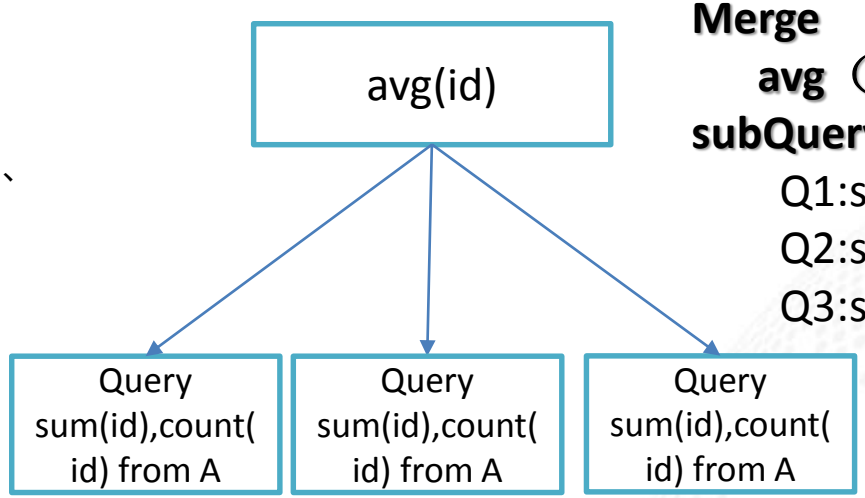
- MySQL 5.5 full support
- 10X performance javacc或者 antlr
- Visitor模式，便于AST处理



查询—Optimizer

- 支持绝大多数单库SQL
- 支持group by、order by、limit 等跨库查询
- 支持min、max、sum、avg等跨库统计
- 暂不支持跨库的join以及子查询

select avg(id) from A

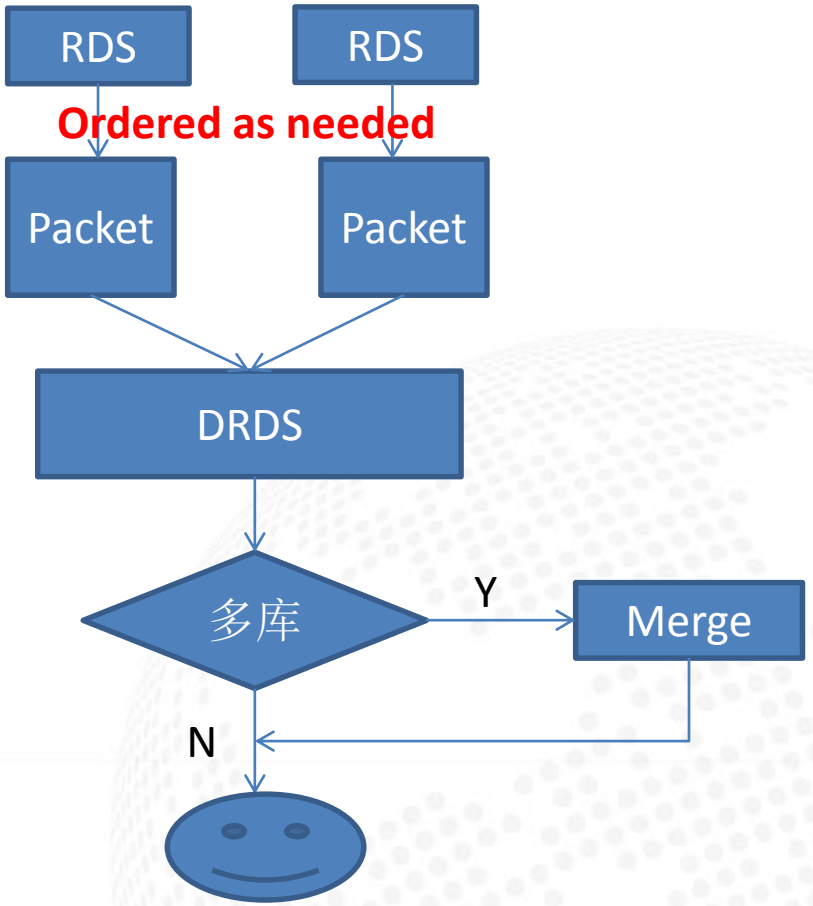


Merge
avg（id)
subQuery
Q1:select count(id),sum(id) A_0
Q2:select count(id),sum(id) A_1
Q3:select count(id),sum(id) A_2

查询—执行器.流式

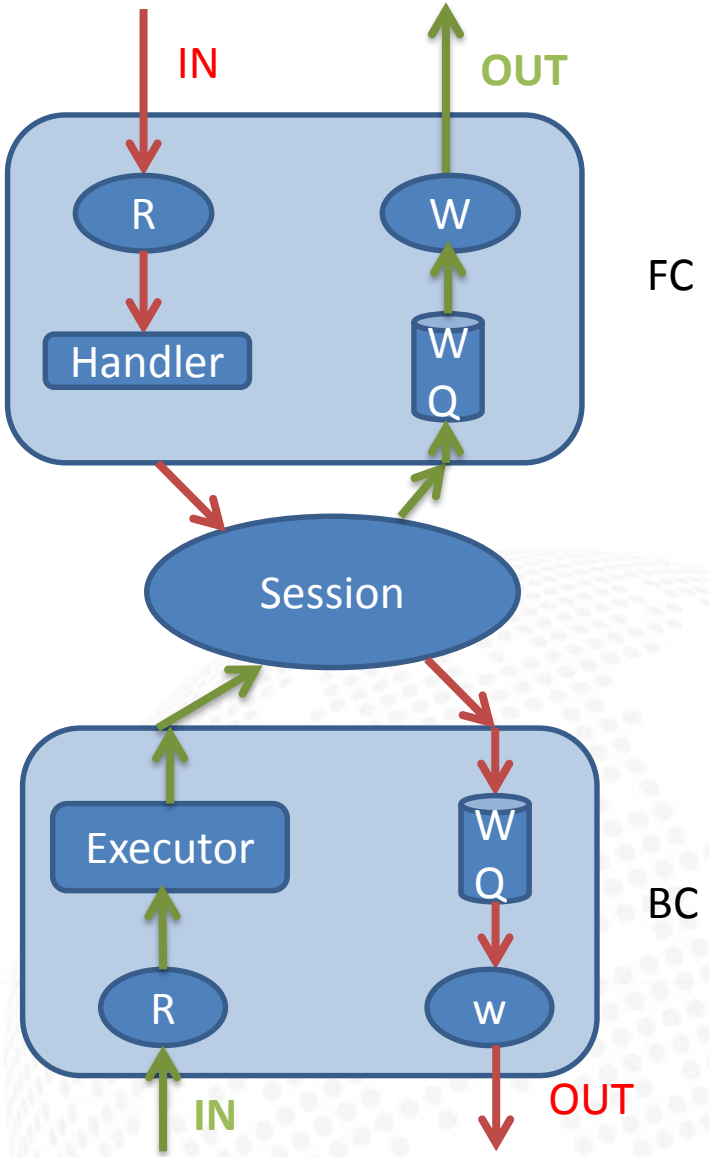
Select * from t limit 10 offset 10000 order by id

- 挑战
  - Server的压力，内存？
  - 如何应对大数据量查询？
  - 跨库的排序放在什么地方
- 应对
  - offloading
  - 流程化处理，防止内存爆满



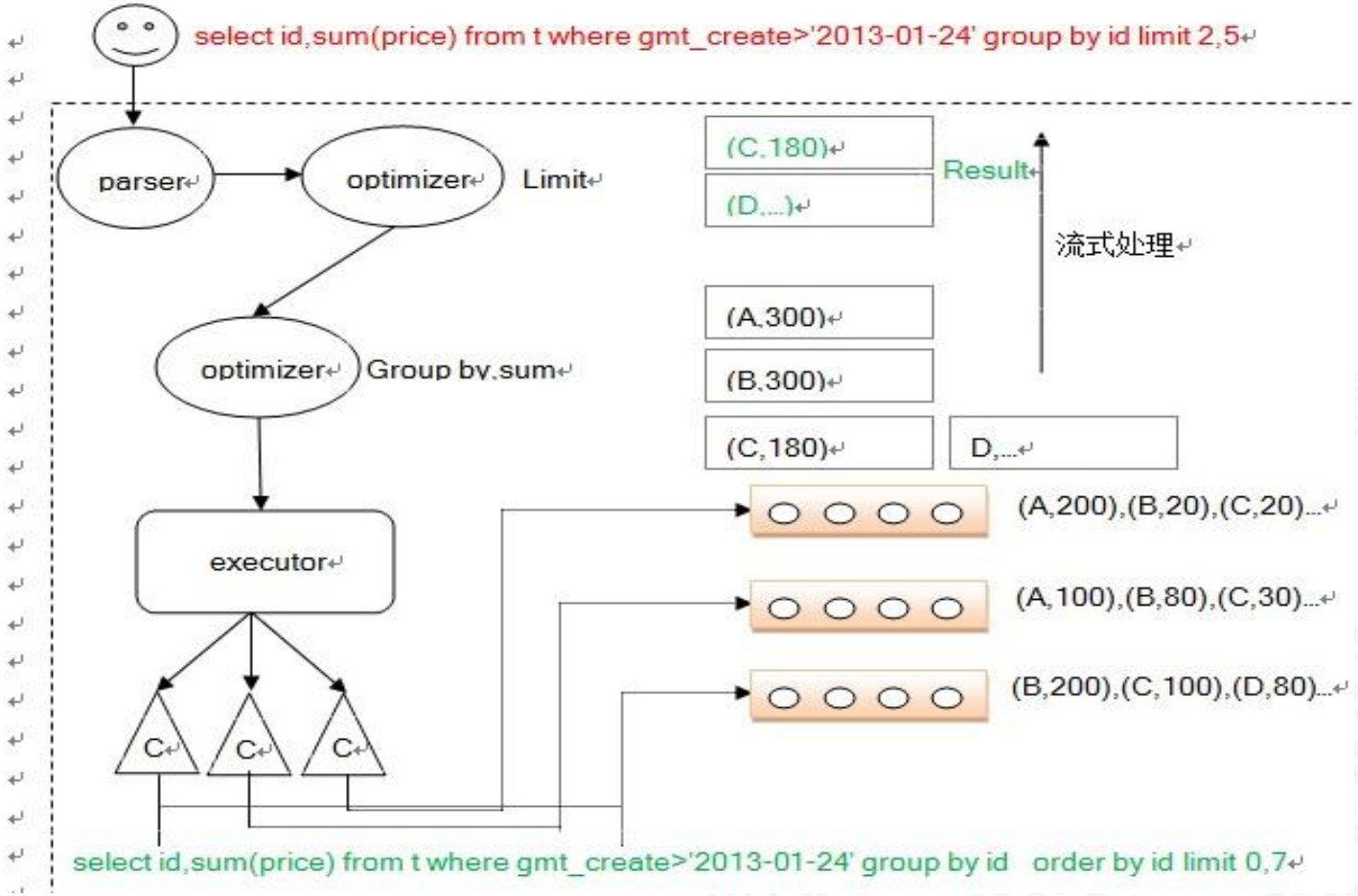
查询—执行器.异步

- 挑战
  - 如何防止因底层RDS原因导致的雪崩？
  - 如何应对耗时较长的SQL？
  - 如何应对短连接
- 应对
  - 异步化处理，连接与线程剥离，防止Hang
  - 底层RDS连接采用连接池的方式



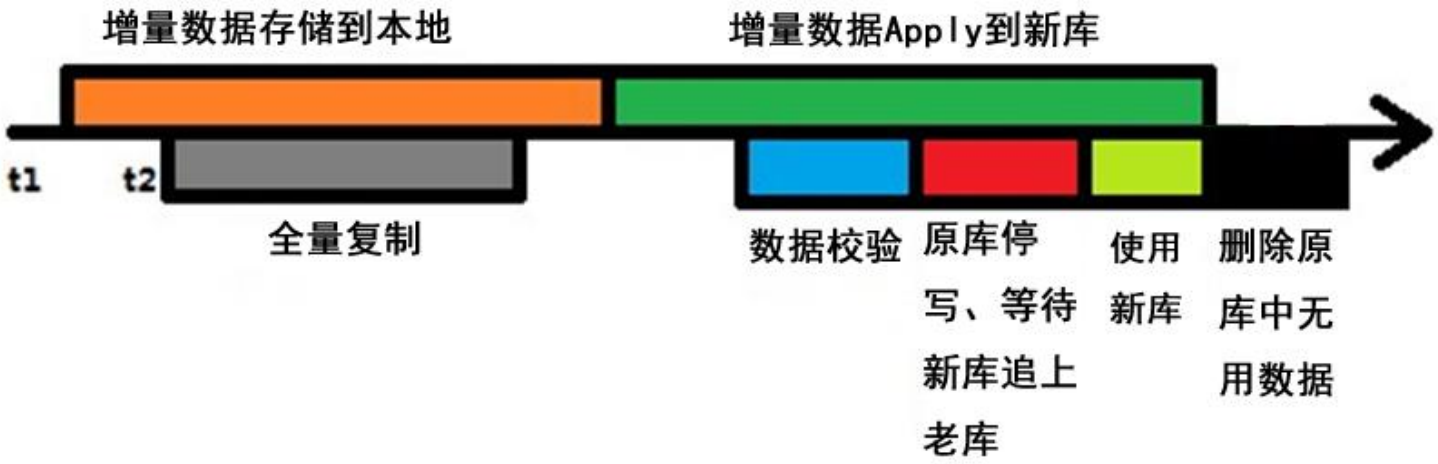
查询—流程总结

- 单库查询
- 跨库查询



自动迁移

- 流程


增量数据存储到本地
增量数据Apply到新库
t1
t2
全量复制
数据校验
原库停写、等待新库追上老库
使用新库
删除原库中无用数据

迁移主要处理过程

- 迁移准备，结点、空间、表等
- 制定好迁移后的策略，开启任务
- 程序会进行全量复制+增量追赶
- 提示“catch up”状态时，可以认为数据的搬迁已经完毕，并且针对原库的所有写操作也会被持续的重放到目标库
- 进行必要验证
- 停原库写几秒钟，让备库与原库一致
- 进行切换

迁移特点

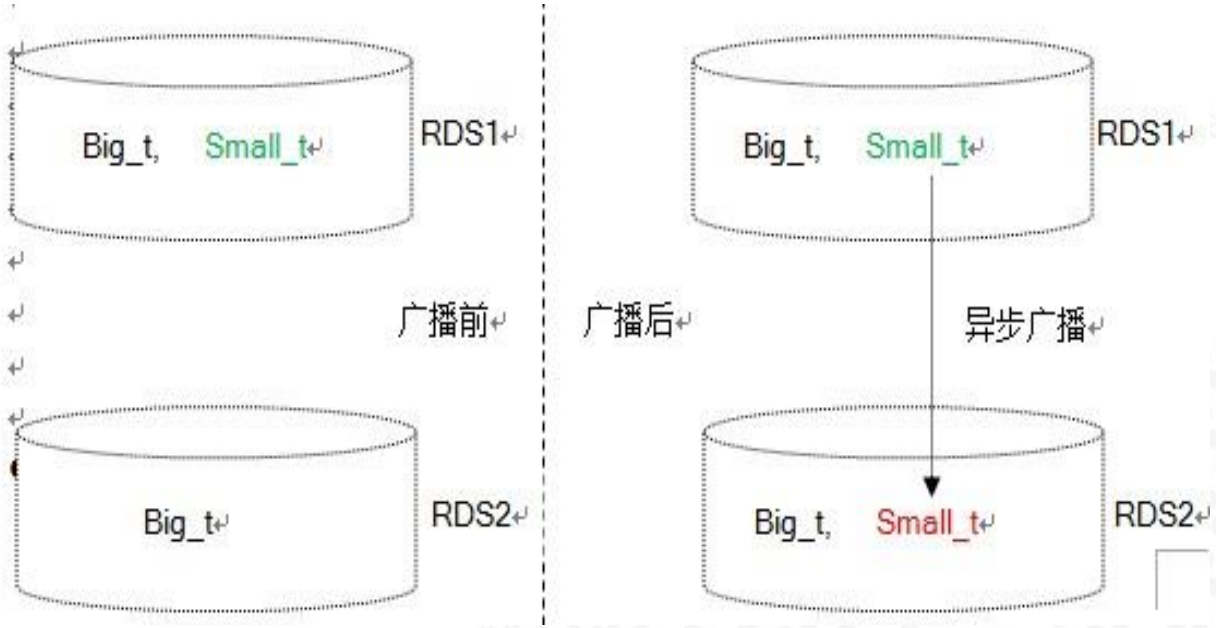
- 对程序透明
- 全量复制+增量追赶，时间取决于需要迁移的数据量，非即时
- 切换时应用有短暂的不可用，该不可用时间用户可以指定

小表广播

- 小表异步复制到多个节点
- 跨库查询转换为本地查询

小表

异步



全局ID

- 问题：单结点的RDS无法保证全局唯一
- 挑战：单点故障？性能瓶颈？
- 方案一，系统自动产生
  - 使用以及获取方式与MySQL auto_increment完全一致
  - 时间戳+SeverID，高可用，但不连续
- 方案二，sequence
  - 使用与Oracle Sequence类似
  - 可指定起始值，步长等
  - 底层使用多个RDS结点，高可用

系统管理

rds_perf > 所有表

新增表 修改表结构 增删索引

表名	应用策略	分区字段	主键选项	表状态	操作
kpi_value_host	rds_perf_policy	host_id	系统生成	运行中	修改表名 清空
kpi_value_ins	rds_perf_policy	custins_id	系统生成	运行中	修改表名 清空
kpi_value_db	rds_perf_policy	db_id	系统生成	运行中	修改表名 清空
myclass	rds_perf_policy	id	系统生成	运行中	修改表名 清空
tunning_sql_stat	rds_perf_policy	custins_id	系统生成	运行中	修改表名 清空
merger_value_db_per_day	rds_perf_policy	db_id	系统生成	运行中	修改表名 清空
merger_value_db_per_hour	rds_perf_policy	db_id	系统生成	运行中	修改表名 清空
merger_value_host_per_day	rds_perf_policy	host_id	系统生成	运行中	修改表名 清空
merger_value_host_per_hour	rds_perf_policy	host_id	系统生成	运行中	修改表名 清空
merger_value_ins_per_day	rds_perf_policy	custins_id	系统生成	运行中	修改表名 清空

最佳实践

- 拆分后，application good design，99%单机查询和事务
- 单机查询
- 单机事务
- 合理的冗余，减少走网络的次数

Thanks!


SequeMedia
盛拓传媒
IT168.com
www.it168.com
ChinaUnix.net
ITPUB