阿里分布式数据库服务实践

沈询
中间件

自我介绍

- 花名 沈询
- 新浪微博: 淘宝沈询_WhisperXD
- 阿里分布式数据库DRDS,TDDL负责人
- 参与过阿里集团内大部分的Oracle到MySQL的迁移工作

- 在分布式存储领域经验比较丰富

Agenda

- DRDS 简介
- DRDS 功能特性
- DRDS 原理剖析
- DRDS 实战

DRDS 简介

DRDS简介-起源

- 起源
  - DRDS 脱胎于 alibaba的cobar 分布式数据库引擎
    - 06年上线使用
    - 在alibaba有近百应用在使用，目前已经开源
    - DRDS的40%的代码出自cobar proxy
      - Server 协议层
      - Sql解析器

DRDS简介-起源

- 起源
  - DRDS吸收了Taobao TDDL分布式数据库引擎的大量优秀经验和解决方案
    - 08年上线使用
    - 目前在使用的应用近千个
    - 大量实际应用解决方案支持
      - 分布式join
      - 分布式aggregation (group sum max min)
      - 异步索引构建
      - Auto sharding ,自动扩容缩容

DRDS简介-起源

- 从TDDL到DRDS
  - DRDS专门针对外部用户进行了配置的重新设计
    - 简化了配置操作规范与流程
    - 尽可能使得应用像操作一个数据库一样的操作DRDS
    - 用户的专业化指导
  - 场景广泛
    - 互联网应用
    - 企业内大数据应用
    - 政务类应用
    - 物联网应用

DRDS简介-应用场景

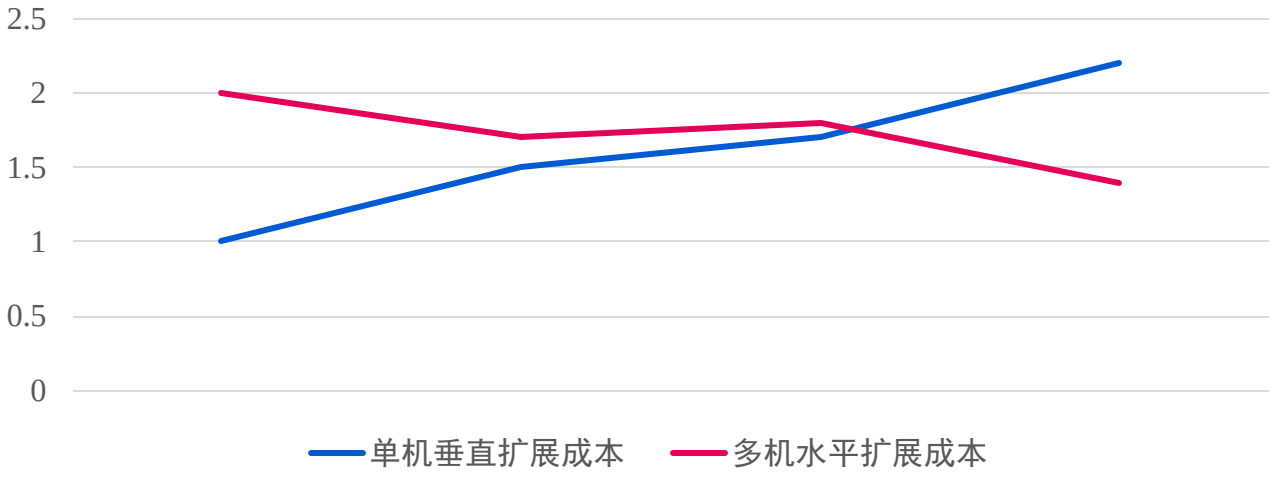
- 应用的业务需求单机已经无法满足
  - 面对全中国13亿用户，以及全世界50亿的用户
  - 单个数据库的最大实例也会出现瓶颈
    - 容量瓶颈
    - 事务数瓶颈
    - 读取瓶颈

DRDS简介-应用场景

- Scale out（多机水平扩展）
  - 使用廉价数据库阵列来满足用户需求--DRDS
  - 优势
    - 更轻量的使用数据库，未来更换的成本小
    - 一次重构，以后基本再无需担心系统瓶颈
  - 劣势
    - 重构迁移需要付出成本
    - 分布式环境下一些查询会被限制不允许执行
    - 完成相同功能需要比单机扩展付出更多成本

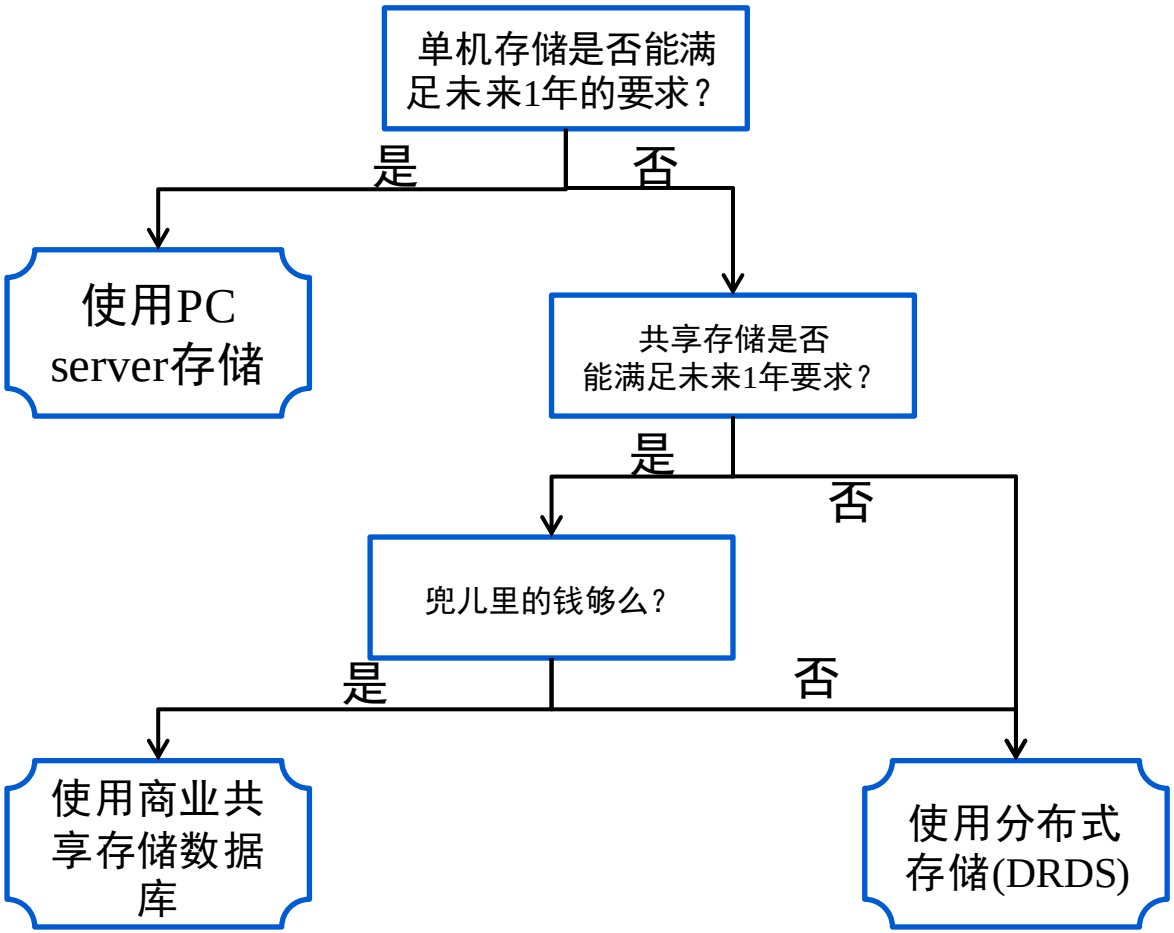
DRDS简介-应用场景

- 理想状态
  - Scale out 与scale up结合
    - 让系统架构具备scale out的能力
    - 尽可能提升单机利用率
  - 但不要过早过度设计



DRDS简介-应用场景

- 何时应该选择Sharding方案？



DRDS 简介

DRDS功能介绍

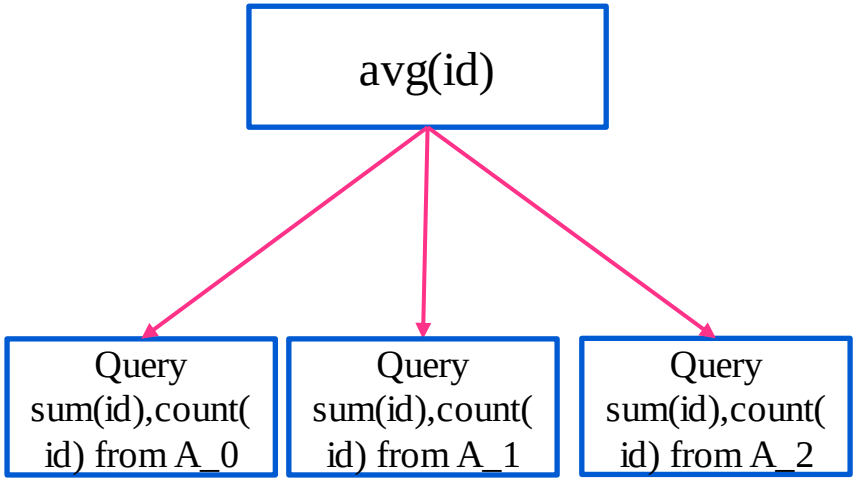
- 分布式MySQL执行引擎
- 弹性扩展
- 小表异步广播

DRDS功能介绍-执行引擎

- 高兼容性
  - MySQL 5.5 的各类复杂查询
    - Join
    - 嵌套
    - 函数
- 智能下推
  - 减少网络传输
  - 减少计算量
  - 充分发挥下层存储的全部能力

DRDS功能介绍-执行引擎

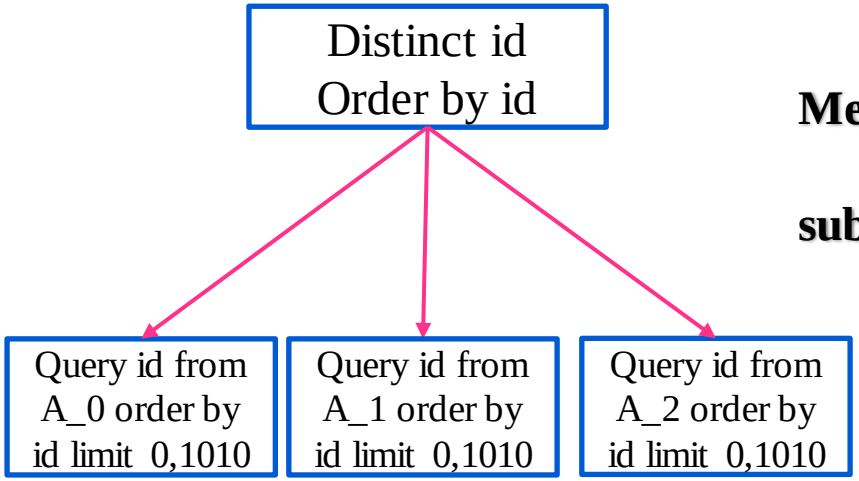
- 智能下推
  - 表A 分库分表3个
  - select avg(id) from A



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

DRDS功能介绍-执行引擎

- 智能下推
  - 全表distinct groupby的执行计划
  - Select id from A order by id limit 1000,10



Merge

distinct id , group by id

subQuery

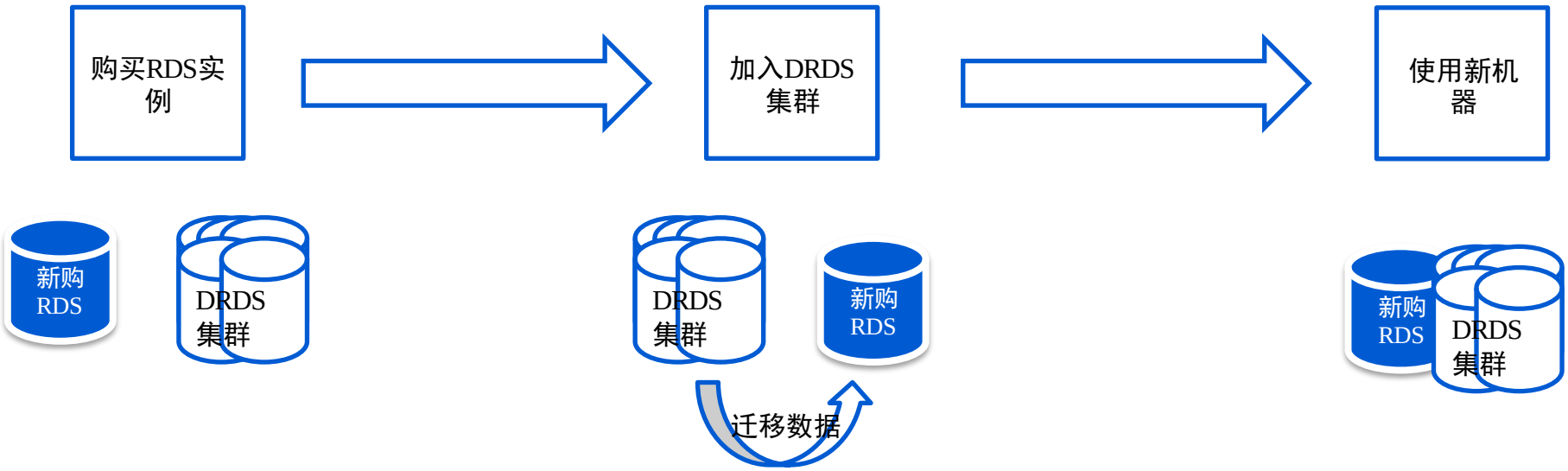
Q1:select id from A_0 order by id limit 0,1010

Q2:select id from A_1 order by id limit 0,1010

Q2:select id from A_2 order by id limit 0,1010

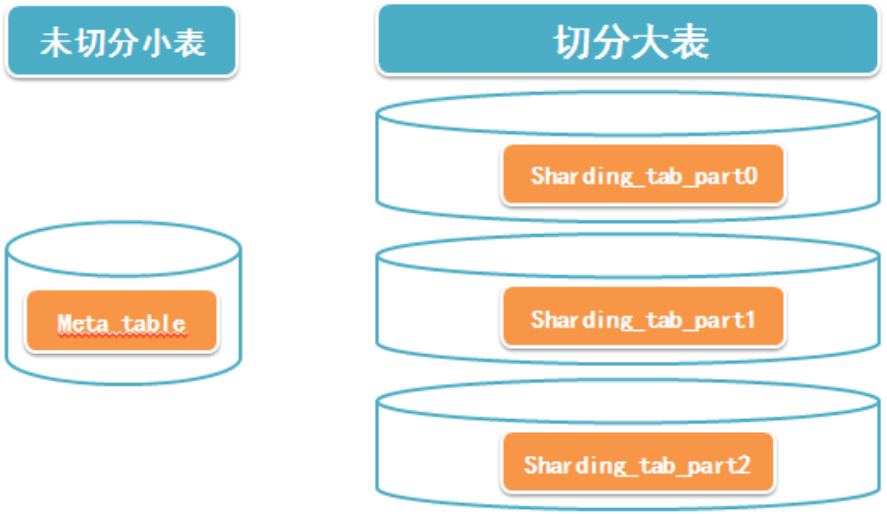
DRDS功能介绍-弹性扩展

- 自动扩容、缩容



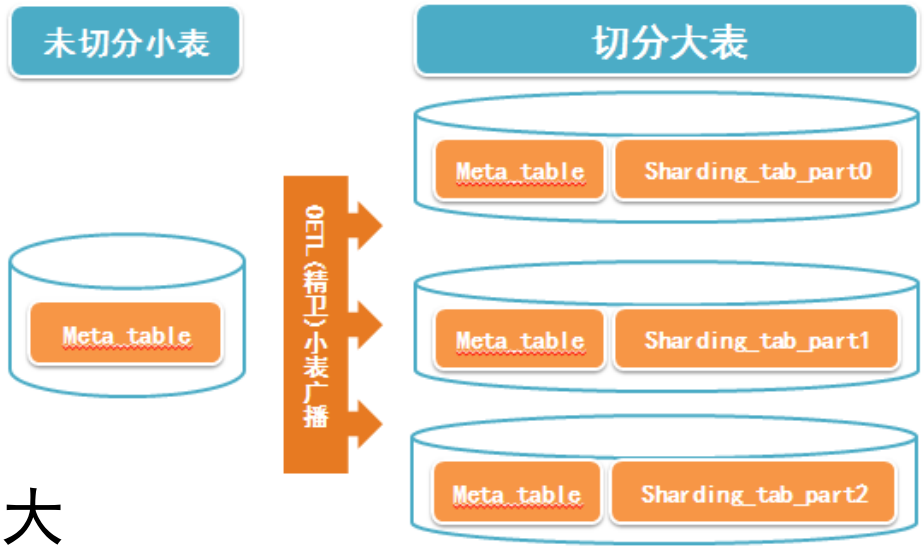
DRDS功能介绍-小表异步广播

- 跨机JOIN
  - 优势：
    - 一致性
    - 空间比较节省
  - 劣势
    - 网络消耗
    - 延迟增加



DRDS功能介绍-小表异步广播

- 小表广播JOIN
  - 优势
    - 性能高
    - 延迟低
    - 网络消耗小
  - 劣势
    - 最终一致性
    - 小表更新量不能太巨大



DRDS 实践

DRDS 实践

- 分布式查询优化
- 事务的分布式优化
- 从单机存储到DRDS迁移流程

DRDS 实践-分布式查询优化

- 让请求可以水平扩展
  - 原则1：选择的shardingKey要能够让所有存储节点均衡的负载读写请求
    - 系统可以简单加机器来扩展
    - 没有系统瓶颈
  - 原则2：查询尽可能带上shardingKey
    - 将跨网络请求尽可能减少
    - 减少并行查询时的机器消耗，从而节省成本

DRDS 实践-分布式查询优化

- CASE1:
  - 应该选择哪个列作为切分条件？
    - 按照买家ID的查询（买家查看自己买了哪些商品）

bizOrderID	buyerID	sellerID	content
0	0	1	床上用品
1	0	2	路上用品
2	0	3	销售路由器
3	0	4	中文书籍
4	0	5	电脑
5	1	0	ipad
6	2	0	笔记本
7	3	0	铅笔
8	4	0	桌面

DRDS 实践-分布式查询优化

- CASE2:
  - 应该选择哪个列作为切分条件？
    - 按照买家ID的查询（买家查看自己买了哪些商品）
    - 按照卖家ID的查询（卖家查看自己卖了哪些商品）

Table_bid
buyerID % 4

bizOrderID	buyerID	sellerID	content
0	0	1	床上用品
1	0	2	路上用品
2	0	3	销售路由器
3	0	4	中文书籍
4	0	5	电脑
8	4	0	桌面

bizOrderID	buyerID	sellerID	content
5	1	0	ipad

bizOrderID	buyerID	sellerID	content
6	2	0	笔记本

bizOrderID	buyerID	sellerID	content
7	3	0	铅笔

DRDS 实践-分布式查询优化

- 异构复制

Table_bid

buyerID % 4

bizOrderID	buyerID	sellerID	content
0	0	1	床上用品
1	0	2	路上用品
2	0	3	销售路由器
3	0	4	中文书籍
4	0	5	电脑
8	4	0	桌面

bizOrderID	buyerID	sellerID	content
5	1	0	ipad

bizOrderID	buyerID	sellerID	content
6	2	0	笔记本

bizOrderID	buyerID	sellerID	content
7	3	0	铅笔

异构复制

Table_sid

sellerID % 4

bizOrderID	buyerID	sellerID	content
5	1	0	ipad
6	2	0	笔记本
7	3	0	铅笔
8	4	0	桌面
3	0	4	中文书籍

bizOrderID	buyerID	sellerID	content
0	0	1	床上用品
4	0	5	电脑

bizOrderID	buyerID	sellerID	content
1	0	2	路上用品

bizOrderID	buyerID	sellerID	content
2	0	3	销售路由器

DRDS 实践-分布式查询优化

- CASE3:
  - 卖家在商城销售的所有商品

type	平台名
0	商城
1	专卖店

Table_bid
buyerID % 4

bizOrderID	buyerID	sellerID	type	content
0	0	1	0	床上用品
1	0	2	1	路上用品
2	0	3	0	销售路由器
3	0	4	1	中文书籍
4	0	5	0	电脑
8	4	0	0	桌面

bizOrderID	buyerID	sellerID	type	content
5	1	0	1	ipad

bizOrderID	buyerID	sellerID	type	content
6	2	0	0	笔记本

bizOrderID	buyerID	sellerID	type	content
7	3	0	1	铅笔

DRDS 实践-分布式查询优化

- 小表异步广播

type	平台名
0	商城
1	专卖店

Table_bid
buyerID % 4

bizOrderID	buyerID	sellerID	type	content
0	0	1	0	床上用品
1	0	2	1	路上用品
2	0	3	0	销售路由器
3	0	4	1	中文书籍
4	0	5	0	电脑
8	4	0	0	桌面

type	平台名
0	商城
1	专卖店

bizOrderID	buyerID	sellerID	type	content
5	1	0	1	ipad

type	平台名
0	商城
1	专卖店

bizOrderID	buyerID	sellerID	type	content
6	2	0	0	笔记本

type	平台名
0	商城
1	专卖店

bizOrderID	buyerID	sellerID	type	content
7	3	0	1	铅笔

DRDS 实践-分布式查询优化

- CASE4:
  - 应该选择哪个列作为切分条件？
    - 最近1周内所有卖家销售的商品量？

bizOrderID	buyerID	sellerID	content	GMT_MODIFIED
0	0	1	床上用品	2014-09-01
1	0	2	路上用品	2014-09-01
2	0	3	销售路由器	2014-09-01
3	0	4	中文书籍	2014-09-01
4	0	5	电脑	2014-09-02
5	1	0	ipad	2014-09-02
6	2	0	笔记本	2014-09-04
7	3	0	铅笔	2014-09-03
8	4	0	桌面	2014-09-05

DRDS 实践-分布式查询优化

- 让请求可以水平扩展
  - 原则1：选择的shardingKey要能够让所有存储节点均衡的负载读写请求
    - 系统可以简单加机器来扩展
    - 没有系统瓶颈
  - 原则2：查询尽可能都带上shardingKey
    - 将跨网络请求尽可能减少
    - 减少并行查询时的机器消耗，从而节省成本

DRDS 实践-事务的分布式优化

- 目标：
  - 完整的事务支持
    - 像传统单机事务一样的操作方式
    - 可按需无限扩展

- 快醒醒~~别做梦了
- 容易理解的模型往往性能都不好，性能好的模型往往不容易理解

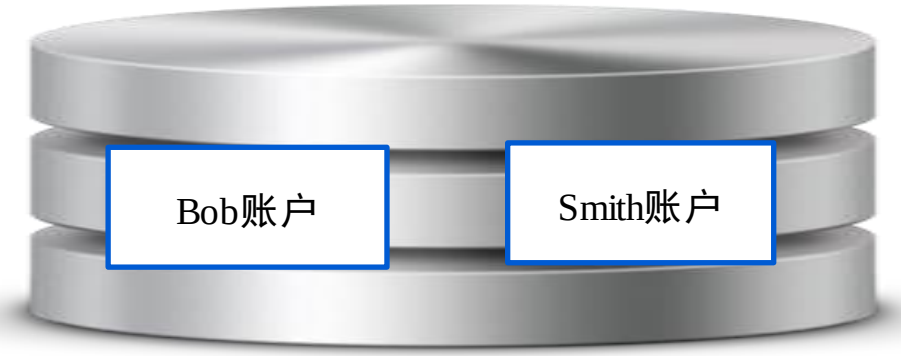
这就是生活

DRDS实践 – 事务的分布式优化

事务单元

操作指令	耗时	总耗时
锁定Bob账户	0.001ms	5.004ms
锁定Smith账户	0.001ms	
查看Bob是否有100元	1ms	
从Bob账号中减少100元	2ms	
给Smith账户中增加100元	2ms	
解锁Bob账户	0.001ms	
解锁Smith账户	0.001ms	

事务时间序

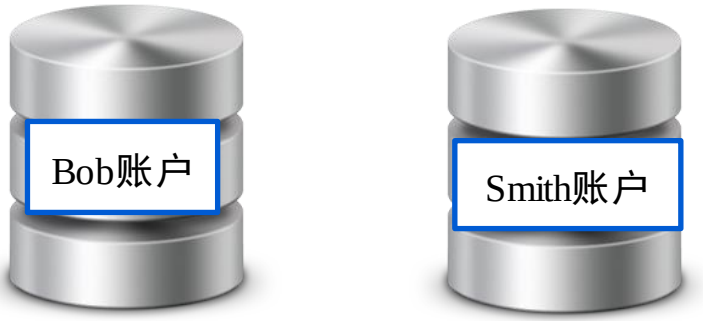


DRDS实践 – 事务的分布式优化

延迟增加
用户体验
下降

操作指令	耗时	总耗时
锁定Bob账户	0.001ms	11.004ms
通过网络锁定Smith账户	2ms+0.001ms	
查看Bob是否有100元	1ms	
从Bob账号中减少100元	2ms	
通过网络给Smith账户中增加100元	2ms+2ms	
解锁Bob账户	0.001ms	
通过网络解锁Smith账户	2ms+0.001ms	

事务时间序



DRDS实践 – 事务的分布式优化

事务单元

操作指令	耗时
锁定Bob账户	0.001ms
查看Bob是否有100元	1ms
从Bob账号中减少100元	2ms
解锁Bob账户	0.001ms

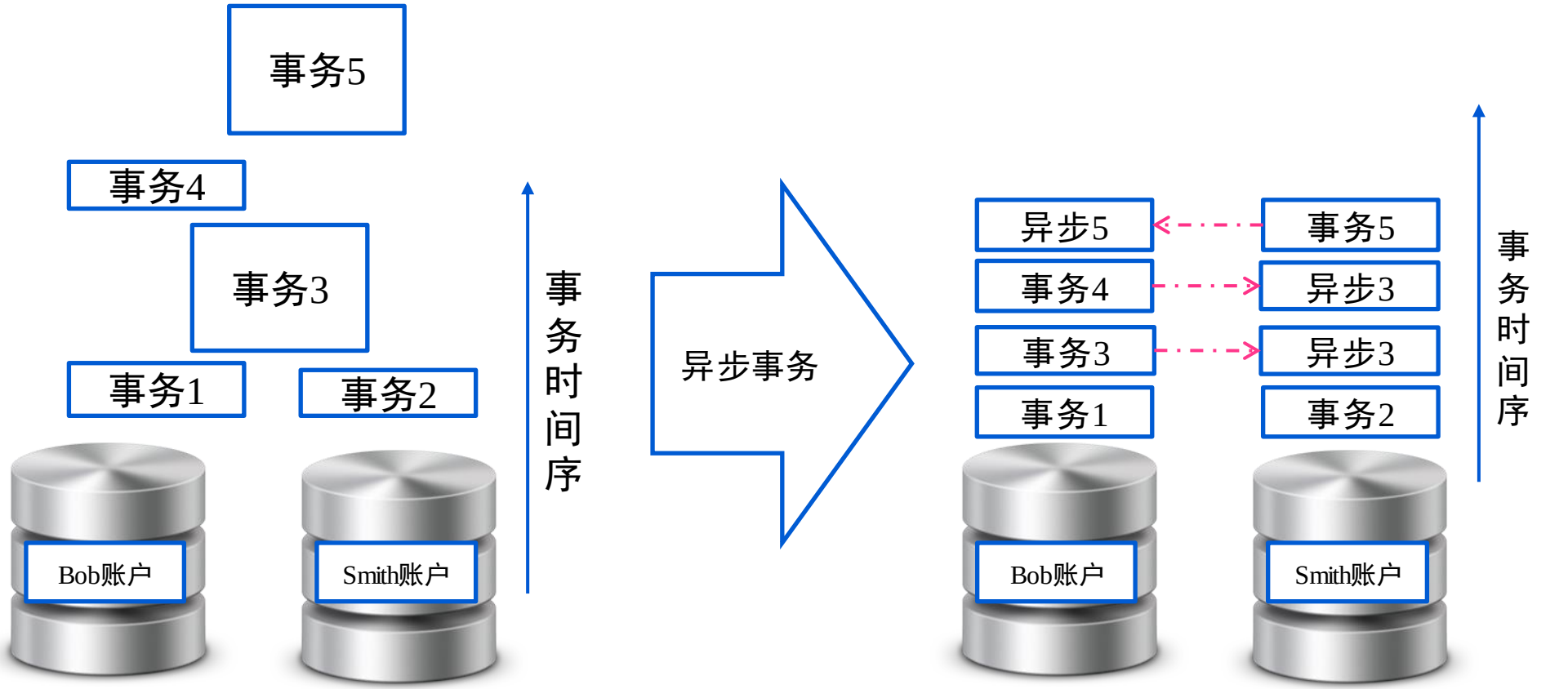
异步并行消息

异步事务单元

操作指令	耗时
锁定Smith账户	0.001ms
给Smith账户中增加100元	2ms
解锁Smith账户	0.001ms

事务时间序

DRDS实践 – 事务的分布式优化



DRDS 实践-DRDS数据迁移

- 目标：
  - 保证业务线上正常运转
  - 平滑过渡
  - 减少运维

DRDS 实践-DRDS迁移流程

- SETP1：
  - 读写在原来的单机数据库
  - 数据通过“愚公数据迁移平台”写入云上DRDS
- SETP2：
  - 验证云上数据是否正确
  - 验证云上DRDS是否能够很好的应对读流量压力
- SETP3：
  - 夜间，停写几分钟
  - 读写切换到DRDS
  - 数据通过“愚公数据迁移平台”写回到云下单机数据库

小结

小结

- DRDS/TDDL系统靠谱
- DRDS/TDDL 服务靠谱
  - 双11的核心应用全部跑在DRDS/TDDL体系
    - 2000多个各类型应用
  - 在云上已经有5家客户正式上线
    - 网聚宝
    - 某GIS大数据应用
    - 某历史数据管理类业务
    - 某保险业物

小结

- 正式公测
  - http://www.aliyun.com/product/drds
  - RDS用户即可免费申请
  - 期待更多的行业案例
  - DRDS QQ群 326140964