

The top corners of the slide feature decorative geometric patterns. On the left, there is a dark blue shape with a network of white dots and lines. On the right, a similar pattern is visible in a lighter blue tone. The background is a solid blue color with white geometric lines forming triangles and diamonds.

**Gdevops**

# 全球敏捷运维峰会

阿里分布式数据库服务原理与实践

演讲人：沈洵



# 自我介绍



花名：沈洵

新浪微博: 淘宝沈洵\_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

01

DRDS简介

03

DRDS原理剖析

02

DRDS功能特性

04

DRDS实战



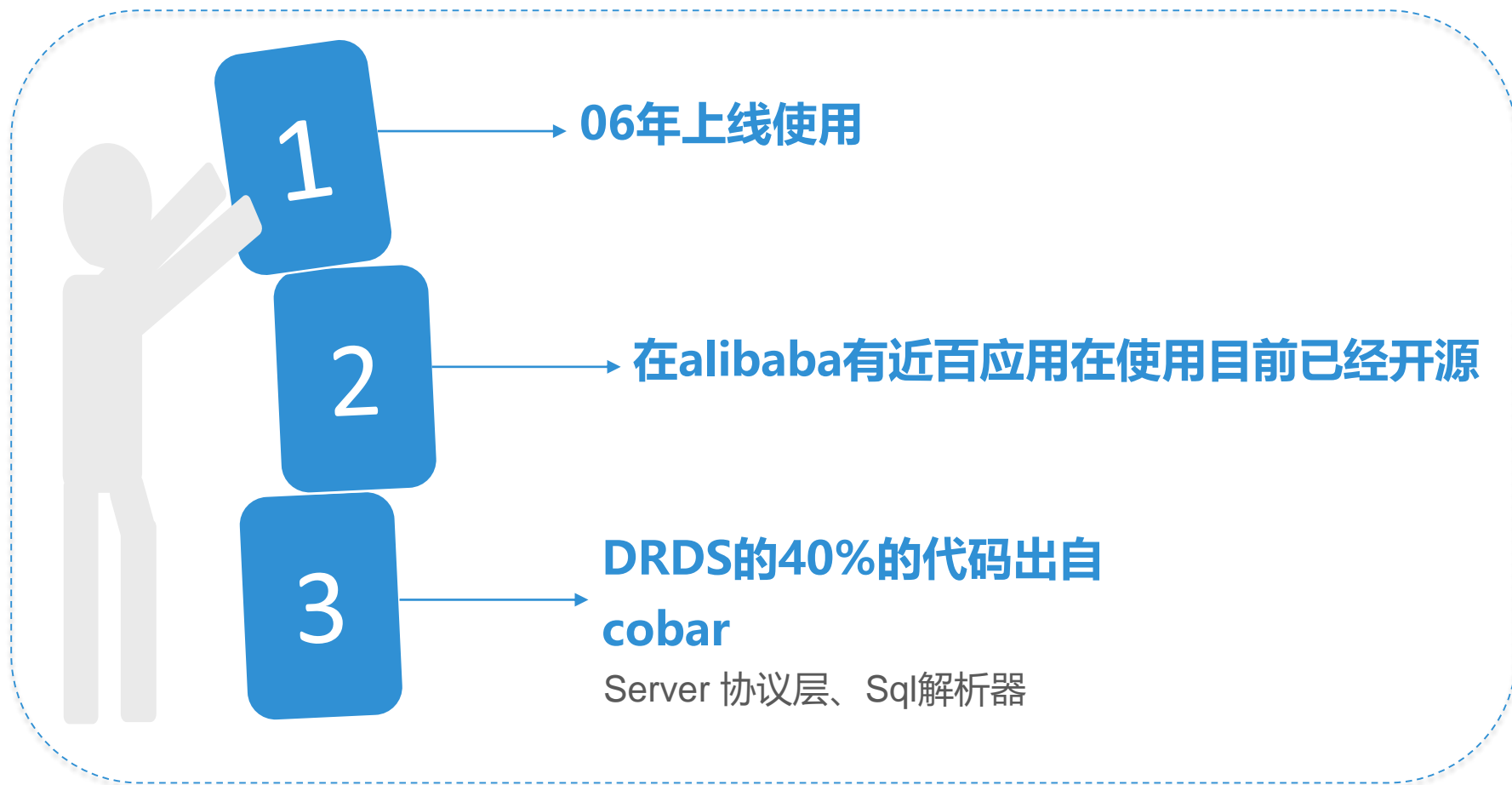
---

# DRDS起源与发展

---

# 起源

**起源：**DRDS 脱胎于 alibaba的cobar 分布式数据库引擎



**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

1

08年上线使用，去IOE的利器

2

目前正在使用的应用近千个

3

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



## 需求

单个机器早就无法满足应用需要，业务已经拆分了  
业务高速发展，小鸡模式成本太高

## 担心

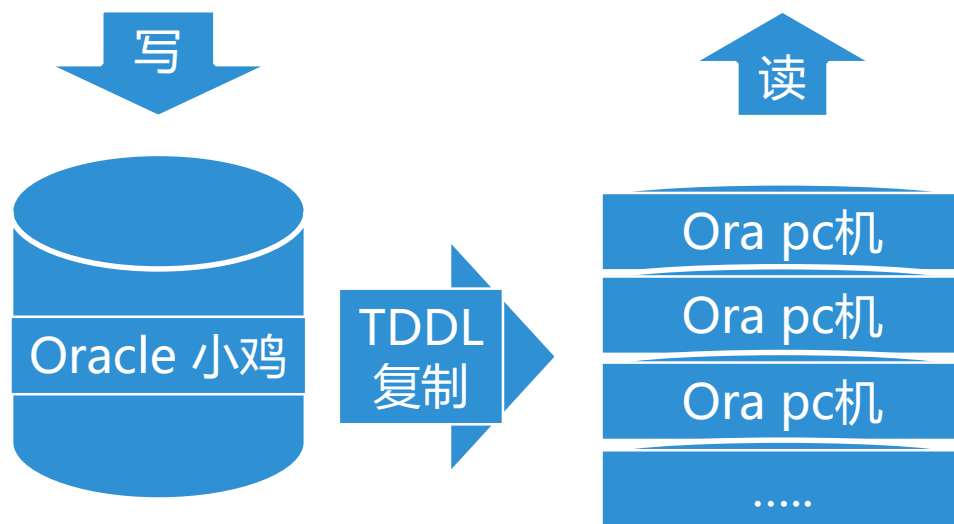
MySQL是否稳定？安全？  
这么多机器，运维成本上升，运维能否扛得住？

## 决策

重要应用采用Oracle（写）+mysql（读）架构  
—商品、用户、交易、评价、店铺  
非重要应用采取MySQL架构  
—收藏夹

## 转动历史的时刻

- ◆ 商品扩展中的失败
- ◆ Oracle(写)-> Oracle(读)
  - Oracle pc机经常挂掉
  - Oracle 11G压测出现性能问题
  - 死机后会hung住业务机
- ◆ 教训
  - Oracle不过如此
  - 架构要简单可依赖





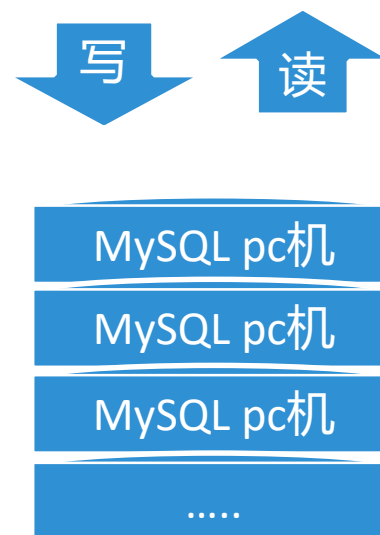
转动历史的时刻：MySQL(读写)？



新的flash cache方案

近100个系统的运维经验积累

- TDDL的稳定和可用性很高
- MySQL出现故障的可能性与Oracle一样低



# 从TDDL到DRDS

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

分销

产品化，提升SQL兼容性、Join、gui工具的兼容性  
像单机数据库一样运维分布式数据库  
自动化读写分离

DRDS专门针对外部用户  
进行了配置的重新设计

# 1

面对全中国13亿用户，以及全世界50亿的用户

# 2

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

# 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

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

## 劣势



---

# DRDS简介

---



# DRDS功能介绍

01

自动读写分离

03

弹性扩展

02

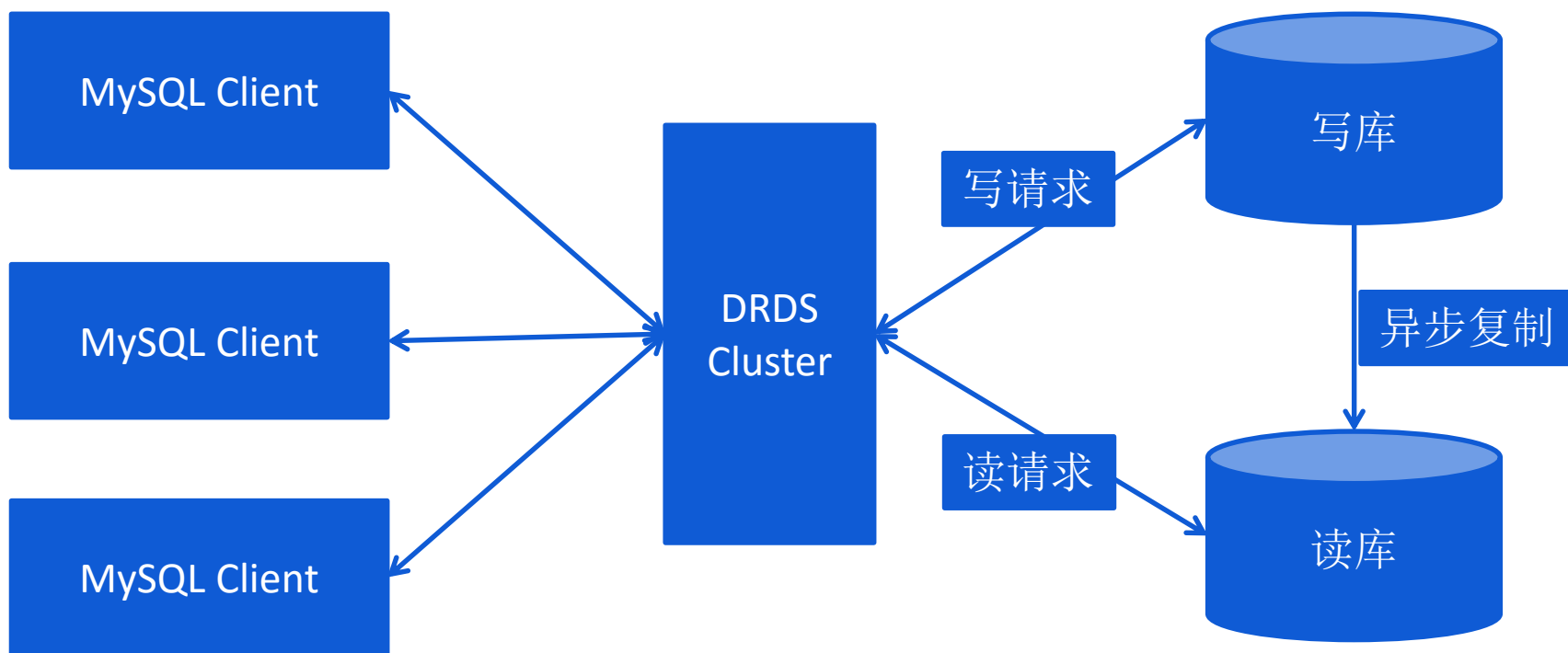
分布式MySQL执行引擎

04

小表异步广播

# 在线动态读写分离

按需“动态”读写分离



# 1

高兼容性：MySQL 5.5 的各类复杂查询

- Join
- 嵌套
- 函数

# 2

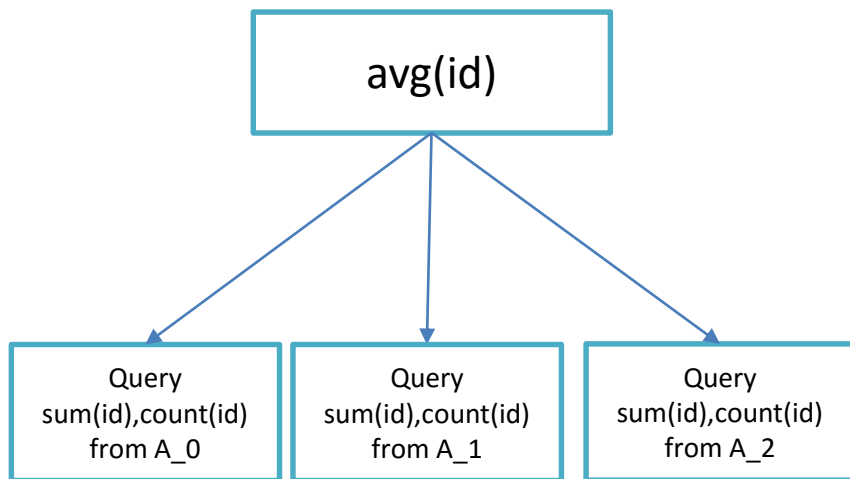
智能下推

- 减少网络传输
- 减少计算量
- 充分发挥下层存储的全部能力



- 智能下推

- 表A 分库分表3个
- `select avg(id) from A`



**Merge**

**avg ( id)**

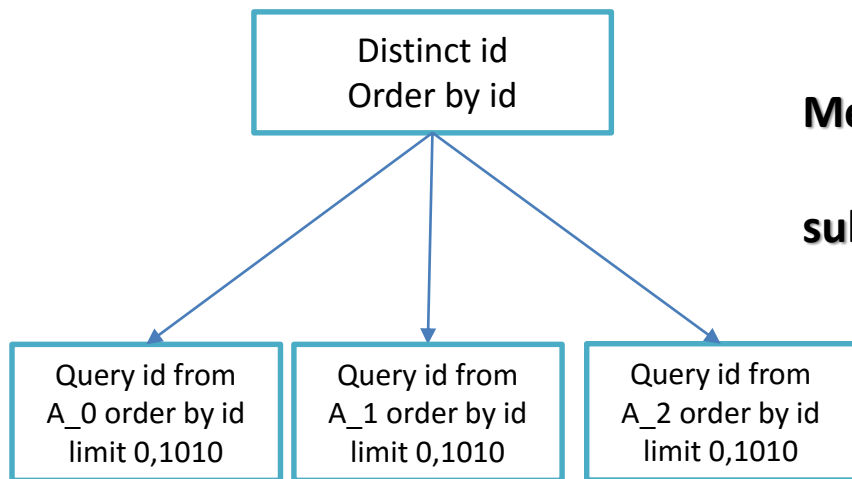
**subQuery**

Q1:select count(id),sum(id) A\_0

Q2:select count(id),sum(id) A\_1

Q3:select count(id),sum(id) A\_2

- 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**

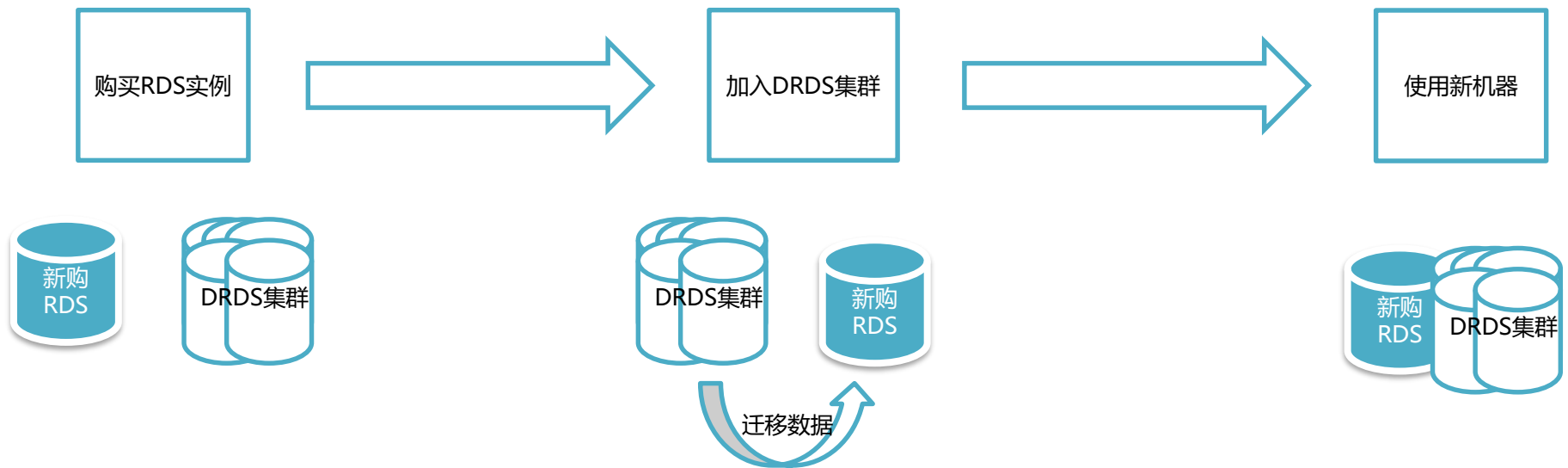
Q1:select id from A\_0 order by id limit 0,1010

Q2:select id from A\_1 order by id limit 0,1010

Q2:select id from A\_2 order by id limit 0,1010

# 弹性扩展

自动扩容、缩容





# 小表异步广播

跨机JOIN

## 优势

优

一致性  
空间比较节省

网络消耗  
延迟增加

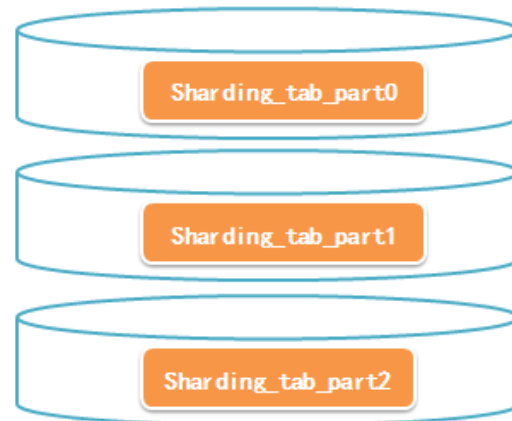
## 劣势

劣

未切分小表



切分大表



# 小表异步广播

## 小表广播JOIN

优

### 优势

性能高  
延迟低  
网络消耗小

劣

最终一致性  
小表更新量不能太巨大

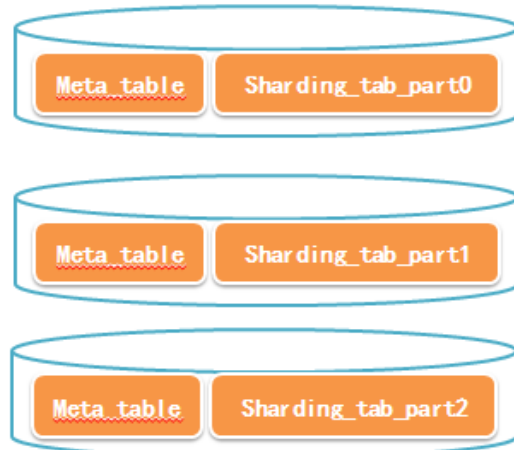
### 劣势

未切分小表



OETL (精品) 小表广播

切分大表





---

# DRDS实践

---

01

分布式查询优化

02

事务的分布式优化

03

运维平台实践

# 1

- 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

让请求可以  
水平扩展

# 2

- 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



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

# 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

# 布式查询优化

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
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

# 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

## 小表异步广播

Table\_bid  
buyerID % 4

type	平台名
0	商城
1	专卖店

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

最近1周内所有卖家销售的商品量？

CASE4：  
应该选择哪个列作  
为切分条件？

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

# 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**



# 事务的分布式优化

- 分布式事务的优化原则
  - 缩小锁的范围
  - 缩小锁的持有时间
  - 取消回滚



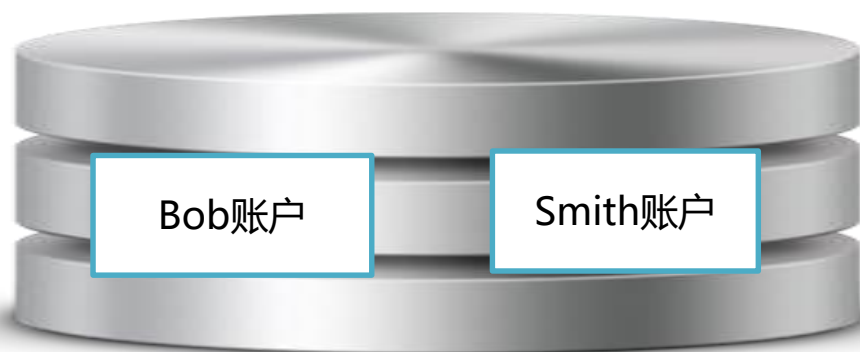


# 事务的分布式优化

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



# 事务的分布式优化

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



# 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

Bob账户

异步事务单元

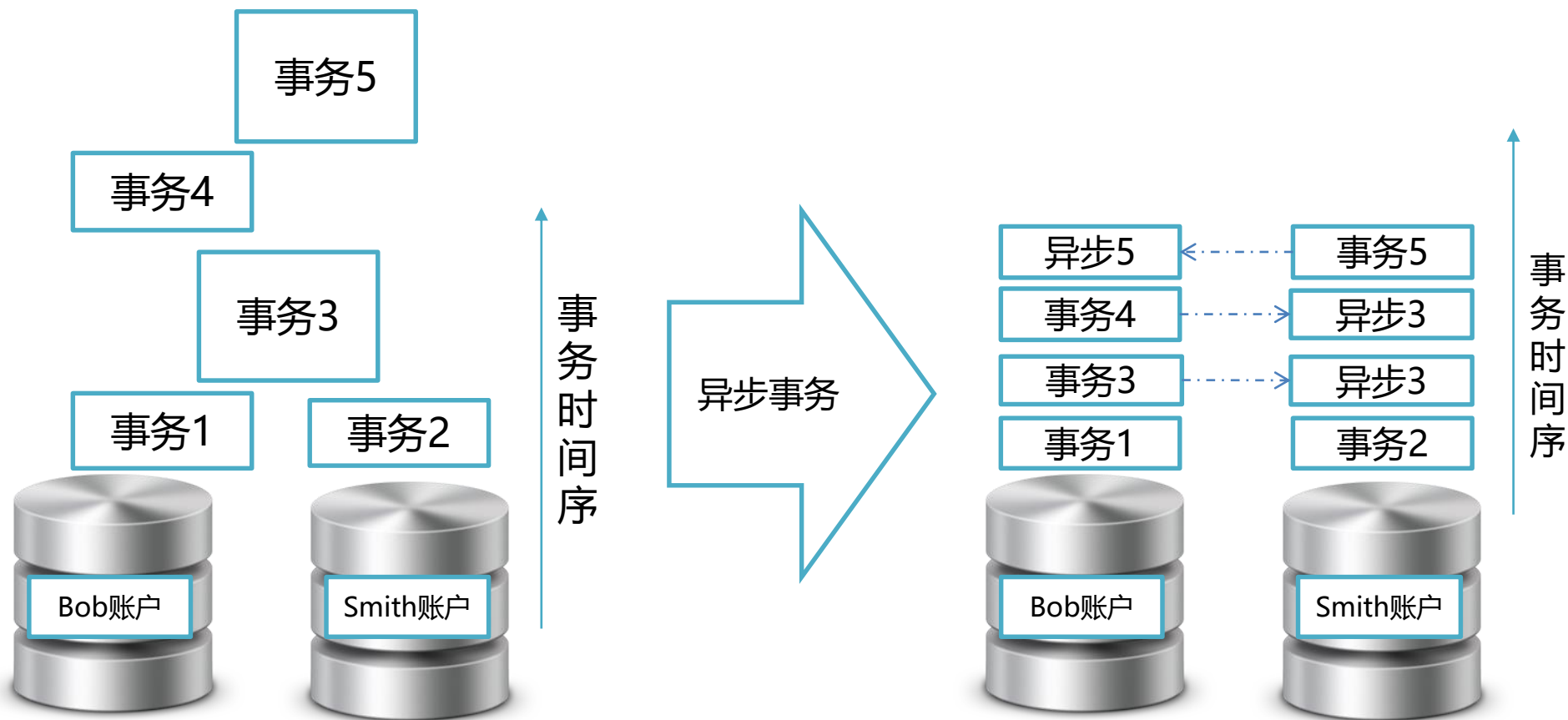
操作指令	耗时
锁定Smith账户	0.001ms
给Smith账户中增加100元	2ms
解锁Smith账户	0.001ms

Smith账户

异步并行消息

事务时间序

# 事务的分布式优化



## 利用控制台可以完成建库建表

testaaaaa [返回数据库列表](#) [如何导入数据?](#) [登录数据库](#) [数据导入](#) [删除数据库](#) [重置密码](#) [只读用户管理](#)

### 数据表信息

[新建表/分布式DD](#)

请输入表名前缀搜索 [搜索](#)

<input type="checkbox"/>	逻辑表	表属性	拆分键	高级属性 <a href="#">?</a>	操
<input type="checkbox"/>	t_item	分库	sellerid	全表扫描	<a href="#">查看表结构</a>   <a href="#">高级设置</a>   <a href="#">删</a>
<input type="checkbox"/>	<a href="#">批量设置全表扫描</a> <a href="#">批量删除</a>				

共有1条, 每页显示: 20条 [«](#) [«](#) [1](#) [»](#) [»](#)



## 利用控制台可以完成建库建表

The screenshot displays the DRDS console interface. On the left is a sidebar with navigation links: 信息, 管理, 表管理, 管理, 单设置, and 信息. The main content area is titled '数据表信息' (Data Table Information) and shows details for a database named 'testaaaaa'. At the top right of the main area are buttons for '如何导入数据?' (How to import data?), '登陆数据库' (Login to database), '数据导入' (Data import), '删除数据库' (Delete database), '重置密码' (Reset password), and '只读用户管理' (Read-only user management). Below the title bar, there is a search input field labeled '请输入表名前缀搜索' (Please enter table name prefix search) and a '搜索' (Search) button. The main content area contains a table with columns: 逻辑表 (Logical table), 表属性 (Table attribute), 拆分键 (Split key), and 高级属性 (Advanced attribute). The table lists one entry: 't\_item' with attributes '分库' (Shard), 'sellerid', and '全表扫描' (Full table scan). To the right of the table entry are links for '查看表结构' (View table structure), '高级设置' (Advanced settings), and '删除' (Delete). At the bottom right of the table, there is a pagination bar showing '共有1条, 每页显示: 20条' (Total 1 item, 20 items per page) and a page number '1'.

逻辑表	表属性	拆分键	高级属性
<input type="checkbox"/> t_item	分库	sellerid	全表扫描



- 支持标准的DDL扩展

```
CREATE TABLE multi_db_multi_tbl
(id int auto_increment, id2 int, name
varchar(30), primary key(id))
dbpartition by hash(id) tbpartition by
hash(id2) tbpartitions 3;
```



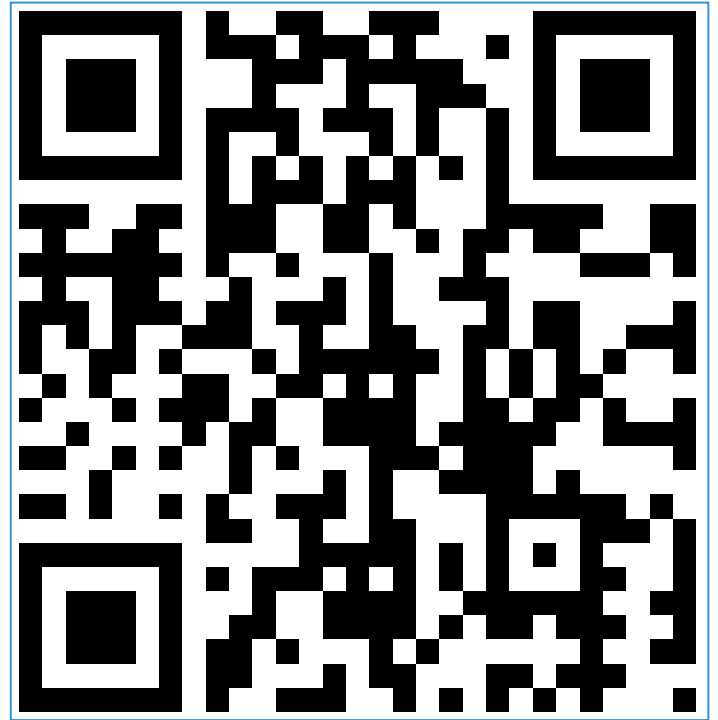
- GUI工具兼容
  - Navicat
  - MySQLWorkbench
  - Etc...



欢迎选择

<http://www.aliyun.com/product/drds>

期待更多的行业案例



The top corners of the slide feature decorative geometric shapes. On the left, there is a dark blue sphere with a network of white lines and dots. On the right, there is a similar structure, a dark blue sphere with a network of white lines and dots. The background is a solid blue color with white geometric lines forming a large 'V' shape in the center and several diagonal lines extending from the corners towards the center.

**Gdevops**

# 全球敏捷运维峰会

The bottom corners of the slide feature decorative geometric shapes. On the left, there is a dark blue sphere with a network of white lines and dots. On the right, there is a similar structure, a dark blue sphere with a network of white lines and dots. The background is a solid blue color with white geometric lines forming a large 'V' shape in the center and several diagonal lines extending from the corners towards the center.

THANK YOU !