OceanBase 2.0的Oracle兼容模式

陈萌萌（酒满）

目录

蚂蚁金服
ANT FINANCIAL

OceanBase

01 Oracle兼容模式

从一条新闻说起...

“…. 在一个问题中，工程师被问及为什么亚马逊的仓库数据库没有遇到“上一次流量高峰期间（那时使用Oracle数据库）的同一个问题”。他们回答，“Oracle和Aurora PostgreSQL是两种不同的[数据库]技术”，处理“保存点”（savepoint）的方式不一样。…”

“市场研究公司Moor Insights & Strategy的首席分析师帕特里克•穆尔黑德（Patrick Moorhead）表示...AWS Aurora是为前瞻性应用软件设计的，而Oracle是为较传统的应用软件设计的。…”

亚马逊使用 Aurora 替换 Oracle：导致 Prime Day 促销日瘫痪

云头条 今天

- 据外媒CNBC获得的内部文件显示，亚马逊迁离Oracle的数据库使用Aurora PostgreSQL是Prime Day促销日陷入瘫痪的主要原因。
- 这次故障突显了亚马逊希望在2020年之前完全摆脱Oracle数据库的过程中可能面临的挑战。
- 近年来亚马逊和Oracle在打口水仗，声称自家数据库软件和云工具的性能更胜一筹。

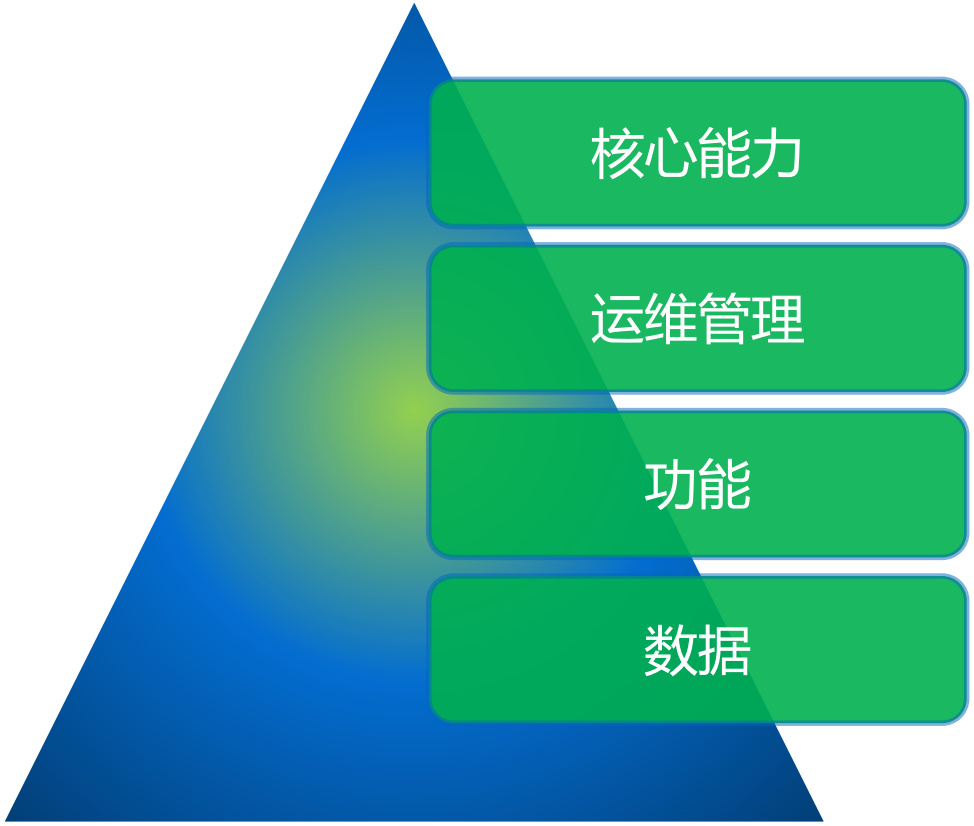
亚马逊现在算是领教了迁离Oracle数据库软件有多困难。

另一条新闻...

兼容是为了最大程度降低用户的使用成本

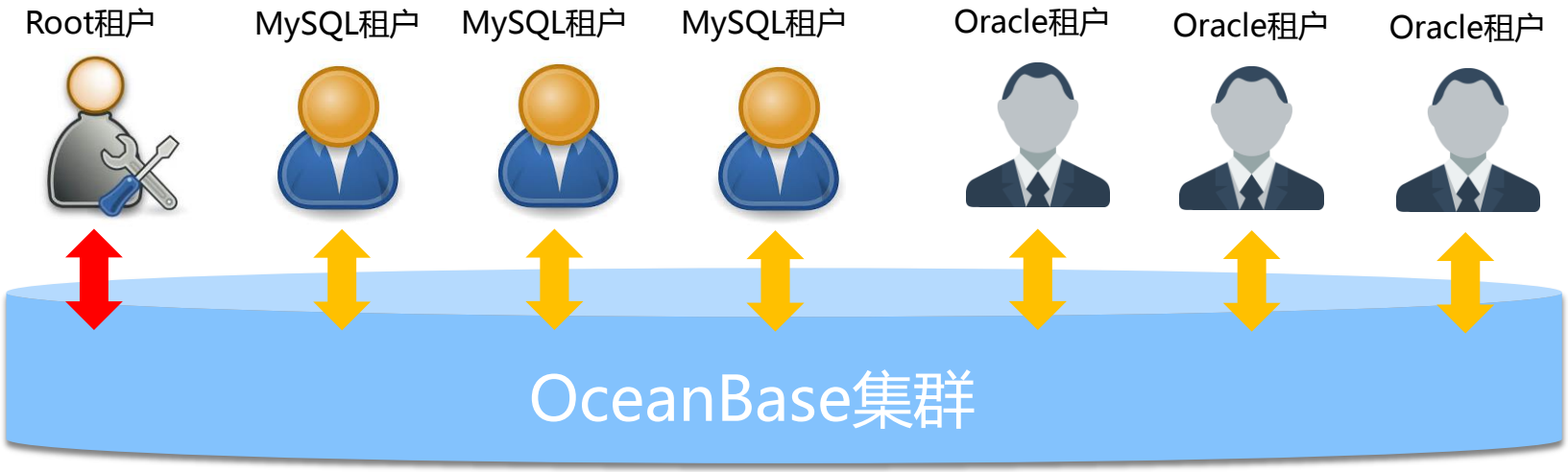
数据库兼容的几个层面

- 数据
  - 基础数据类型/表模式
- 功能
  - SQL语法、语义
  - 客户端驱动、错误码
- 运维管理
  - Oracle原生字典视图：ALL_*/DBA_*/USERS_*
  - Oracle概念——降低DBA学习成本
- 核心能力
  - 性能、稳定性



MySQL/Oracle租户混合部署

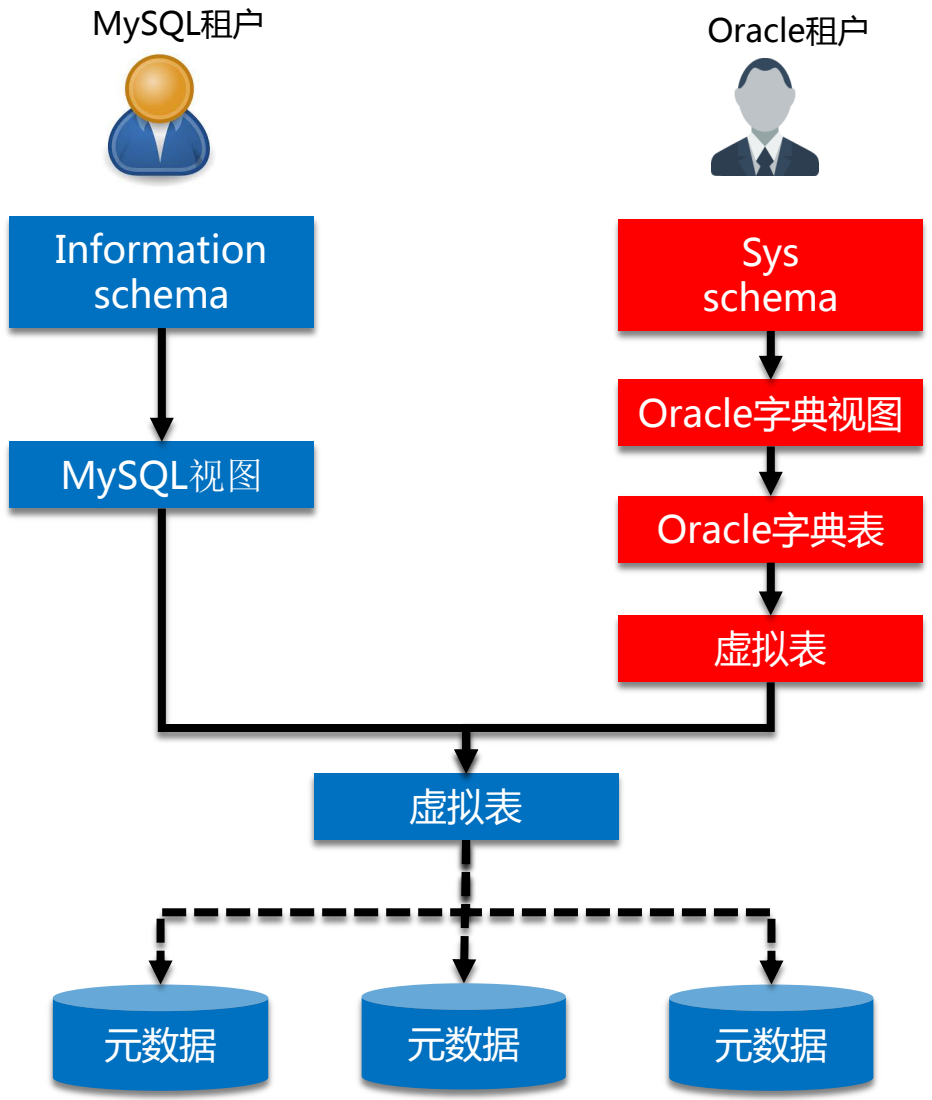
- 同时支持MySQL与Oracle两种不同类型的租户混合部署
- 支持细粒度的租户资源隔离
- 自动负载均衡



MySQL与Oracle租户混合部署

统一元数据管理

- 租户级元数据存储粒度
- 跨租户类型的统一元数据管理机制
- 兼容租户类型的展示和访问机制
  - MySQL模式支持information schema
  - Oracle模式支持sys schema下的字典视图



OceanBase 2.0 部分Oracle兼容特性

- 数据类型

支持number、 varchar2、timestamp、CLOB...

- SQL层功能

外连接(+)、窗口函数、层次查询、临时表、DB link、外键

- 事务层

隔离级别、flashback、XA

- 内部视图

部分All_/DBA_/USER_视图

- 索引

全局索引、函数索引

- 分区

hash、 range、 list分区及二级分区组合

- 伪列

rownum 、 sequence 、 virtual column等

- 存储过程

循环、复杂数据类型、cursor、异常

- 客户端

JDBC、ObProxy、C客户端

02 新特性

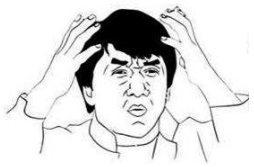
全局索引

- OceanBase 1.0仅支持分区局部索引
- 基于全局一致性，OceanBase 2.0支持全局索引的创建
  - 多维度扩展性
  - 跨分区全局唯一索引
  - 强一致
  - 支持更高效的查询方式
- 分布式全局索引的难点
  - 全局一致：分布式事务、异步化
  - 优化：更新、回表

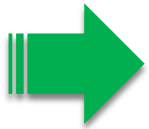
全局索引——分区维度的纠结

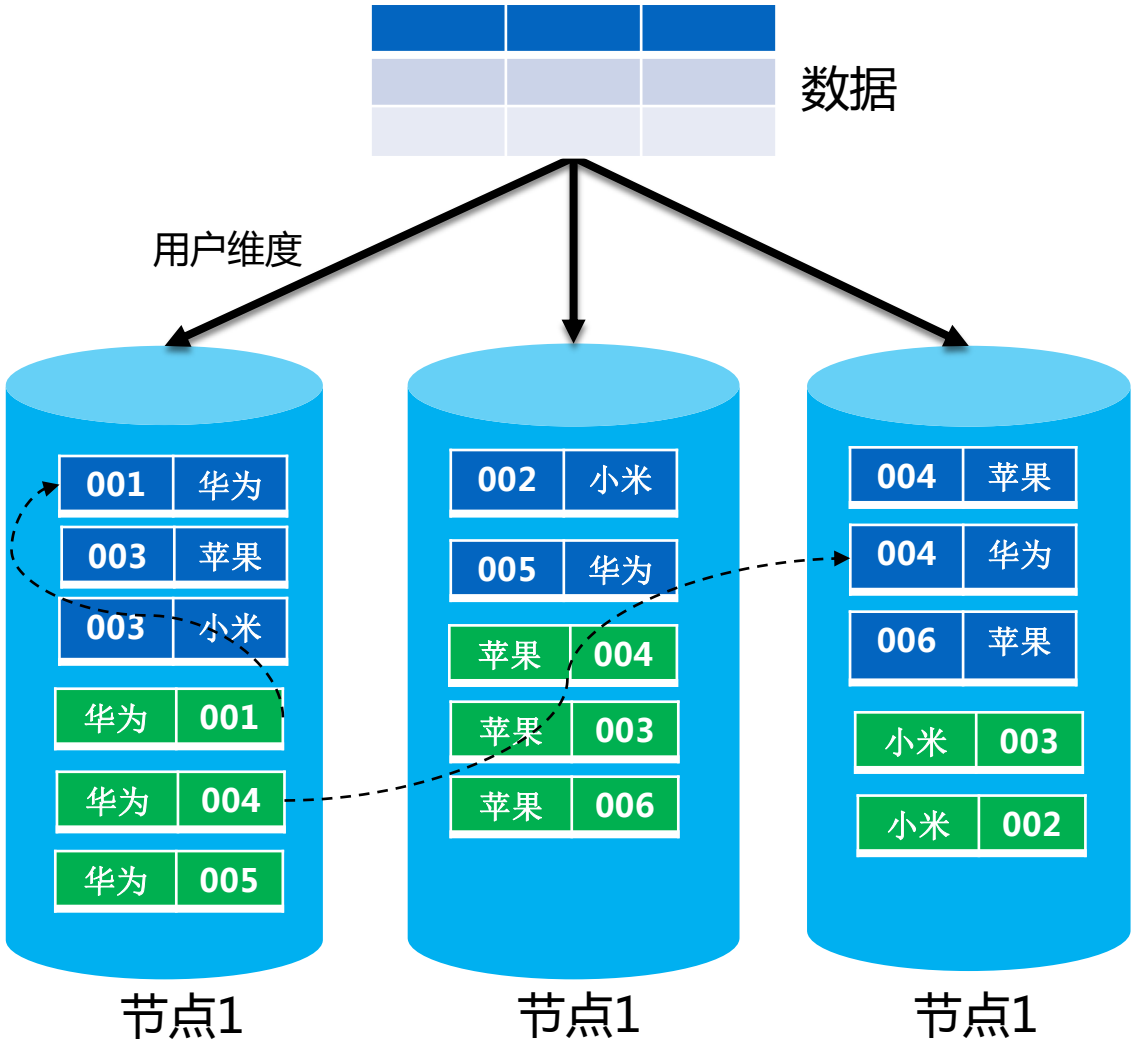
用户	商家	唯一标识
001	华为	…
002	小米	…
004	苹果	…
003	苹果	…
003	小米	…
005	华为	…
004	华为	
006	苹果	
…	…	…

分区方案一？ 分区方案二？



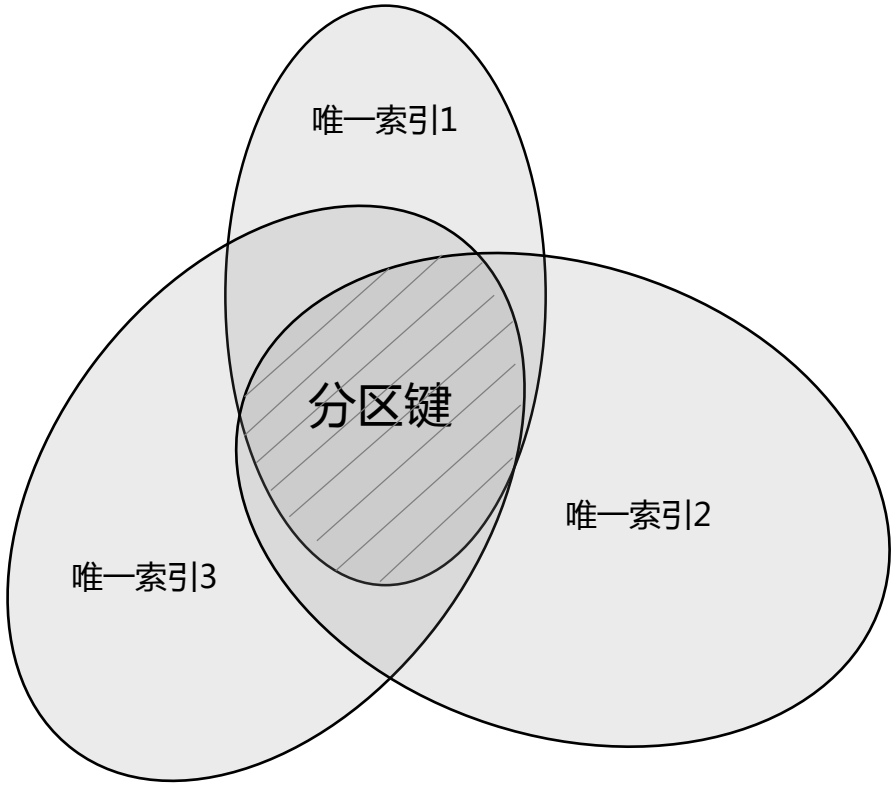
Select … from sales where 用户 = ？
还是
Select … from sales where 商户 = ？





全局索引——全局唯一索引

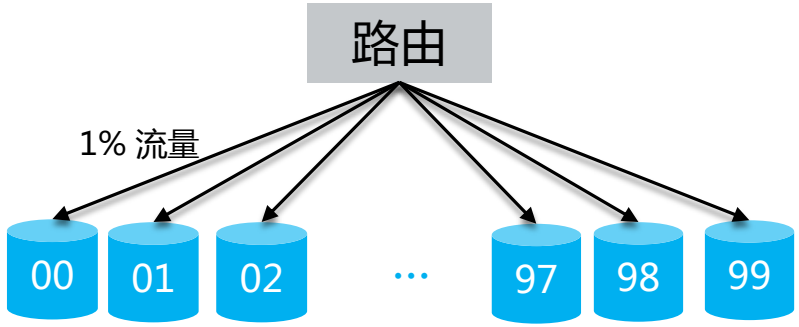
- 1.0分区键必须是唯一索引（包括主键）的集合交集的一部分
- 2.0 支持全局唯一索引
- 难点
  - 全局唯一索引的更新



OceanBase 1.0分区限制

分区

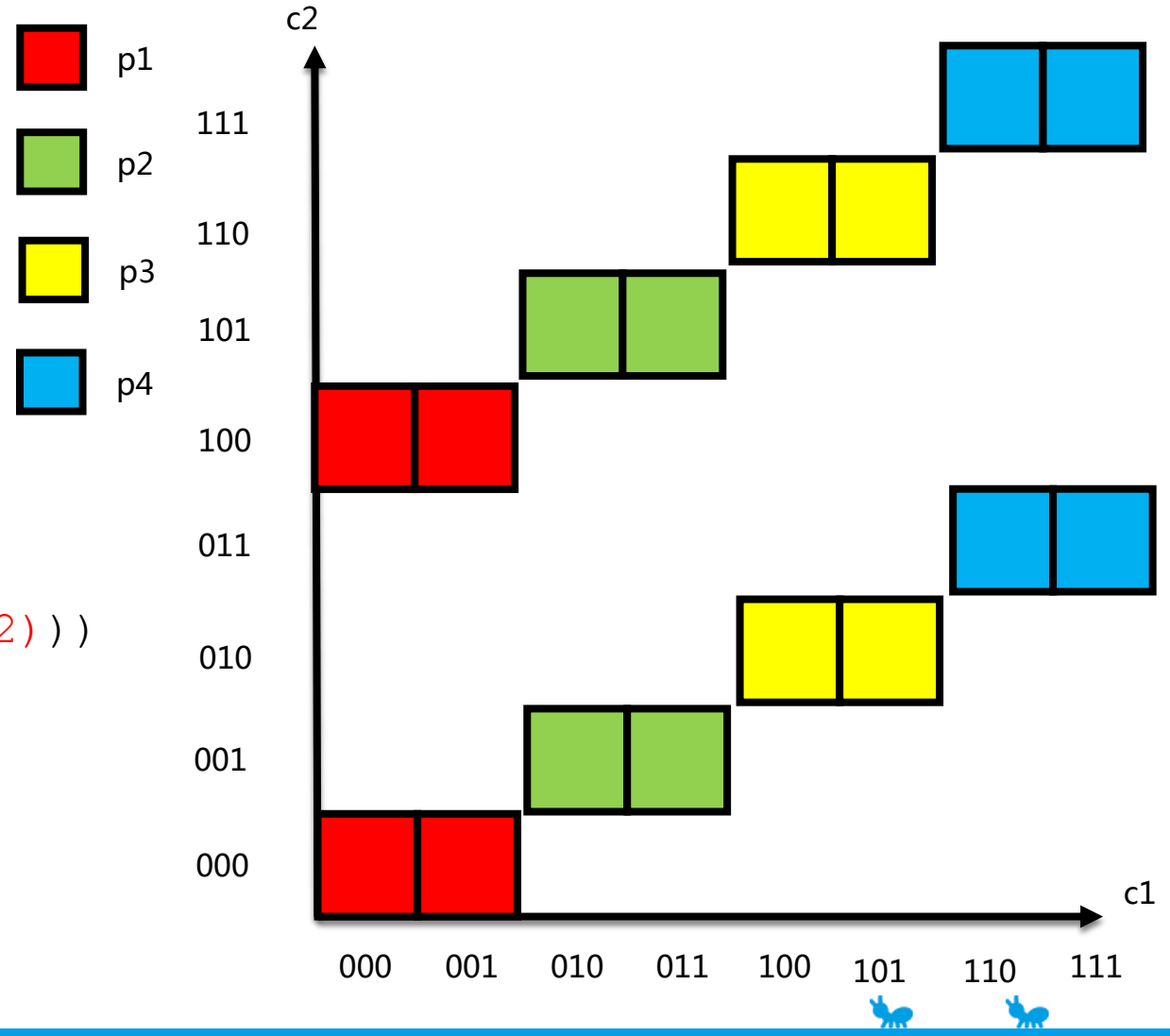
- 支持Oracle模式的hash分区与range分区
- 新增支持list分区——比hash更为精确的分区方式
- 更多管理命令
  - Add/coalesce/exchange/split/merge
- 分区裁剪能力增强
  - 基于check的分区裁剪
  - 基于虚拟列的分区及裁剪



分区——基于check约束的分区裁剪

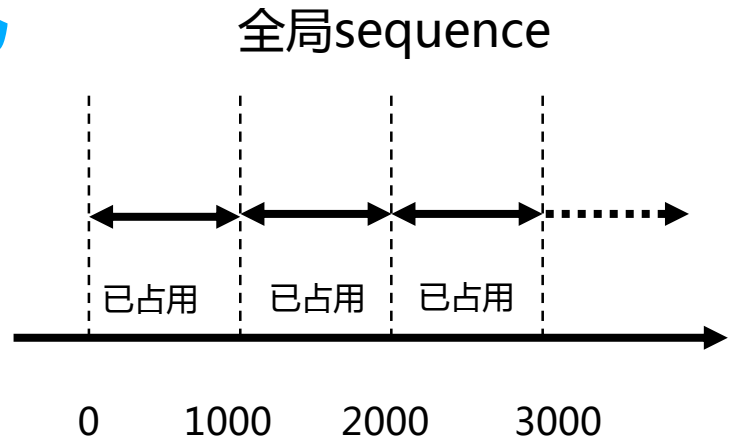
- 基于check的分区裁剪可以允许用户使用多个维度进行高效查询

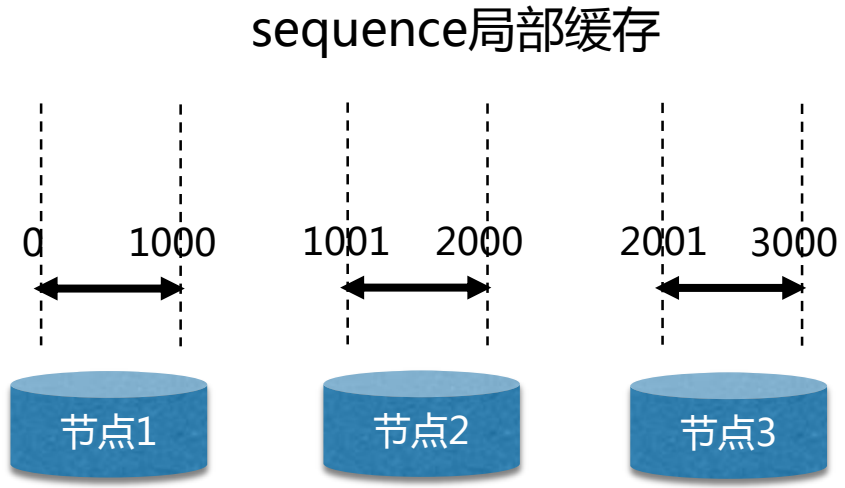
```
create table m(
  c1 varchar(10),
  c2 varchar(3),
  c3 as (substr(c1, 0, 2)),
  constraint check_c2 check(
    substr(c2, 1, 2) = substr(c1, 0, 2)))
partition by list(c3)
(partition p1 values('00'),
 partition p2 values('01'),
 partition p3 values('10'),
 partition p4 values('11'));
```



Sequence——分布式id生成

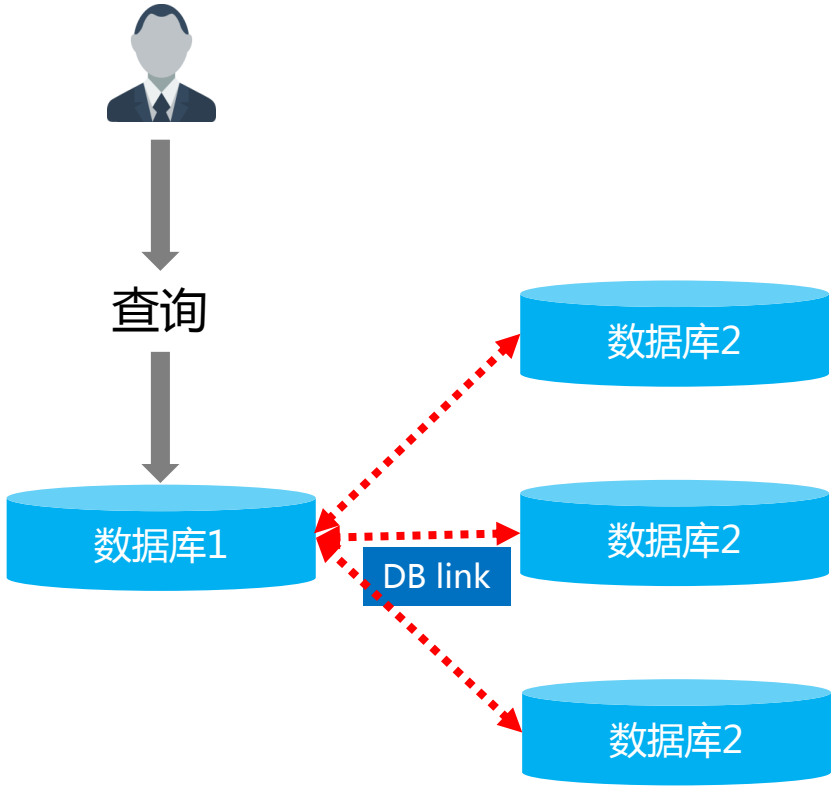
- 各节点缓存部分sequence值域
- 与MySQL自增列的机制类似，但更宽松、高效
  - 无递增要求——避免空洞问题
- 提前预取
  - 避免耗尽时同步可能带来的瞬时不可用





DB link——跨库查询

- 跨库的“分布式”查询——降低业务代码的复杂度
- 支持跨库的分布式事务
- 暂时支持OB-OB的DB link，后续计划支持异构库的访问



基于DB link的跨库访问

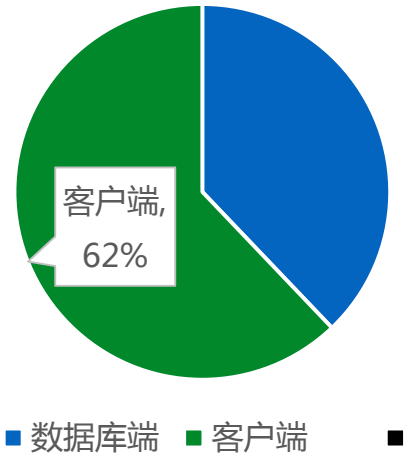
存储过程——Oracle用户迁移的拦路虎

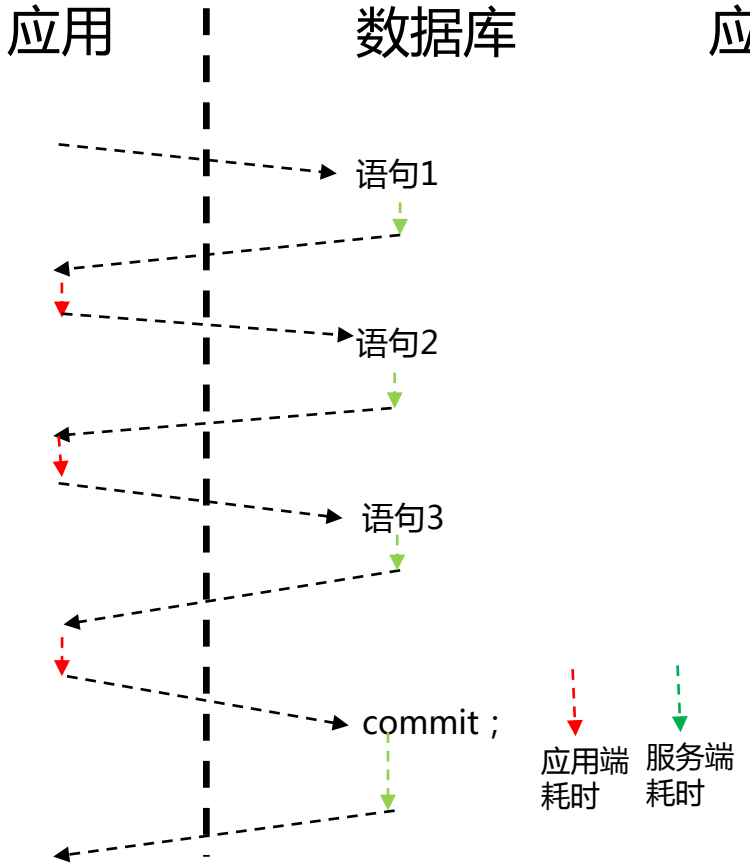
- 大量历史遗留代码无法重构
- 性能关键路径
  - 客户端交互占处理延迟 > 50%
  - 热点行处理
- 存储过程的高效实现
  - 基于llvm的编译执行框架

“多一行PL/SQL的代码我们就多一行拴住客户的锁链...”

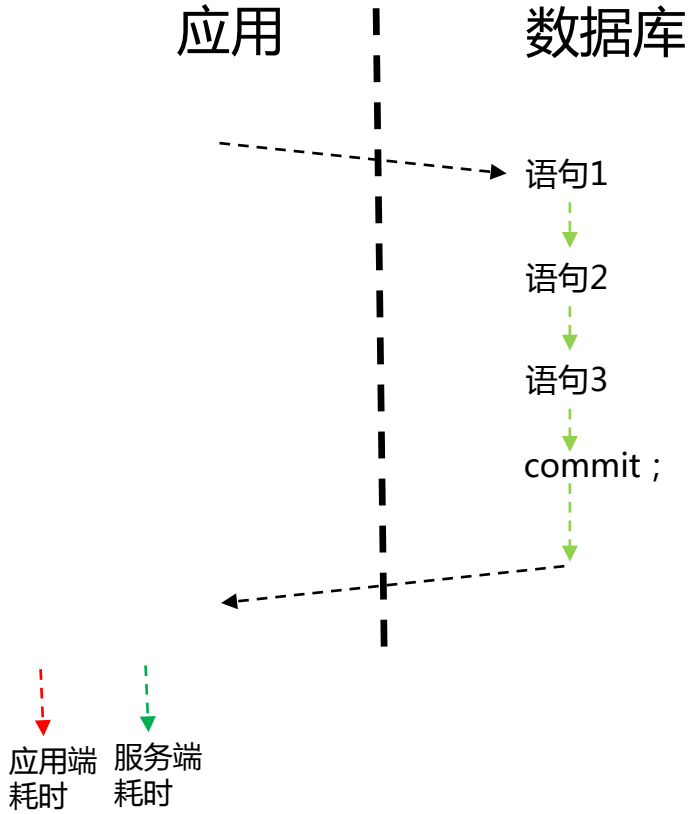
——Oracle PL/SQL组资深工程师

存储过程——优化事务延迟





应用端处理



服务端处理（存储过程）

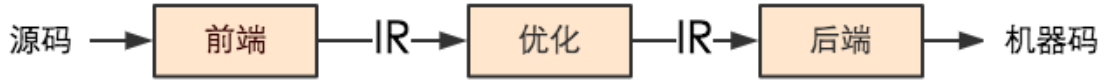
如果第一条语句是

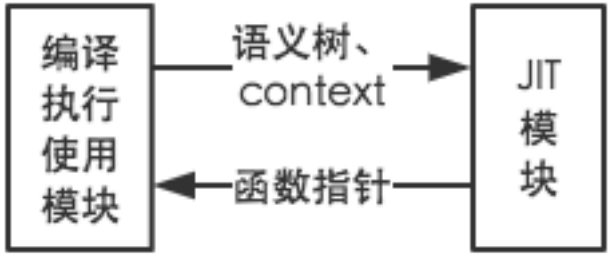
Select price from item where id = 3000123 for update

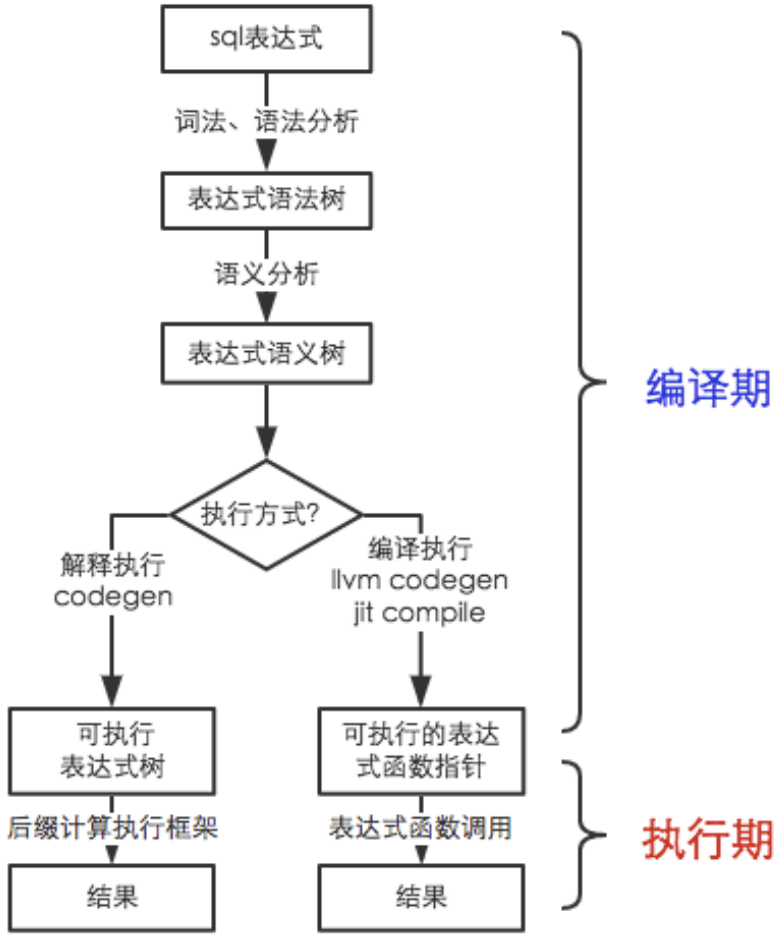
会怎么样？

存储过程——编译执行优化

- 存储过程采用全编译的方式生成可执行代码

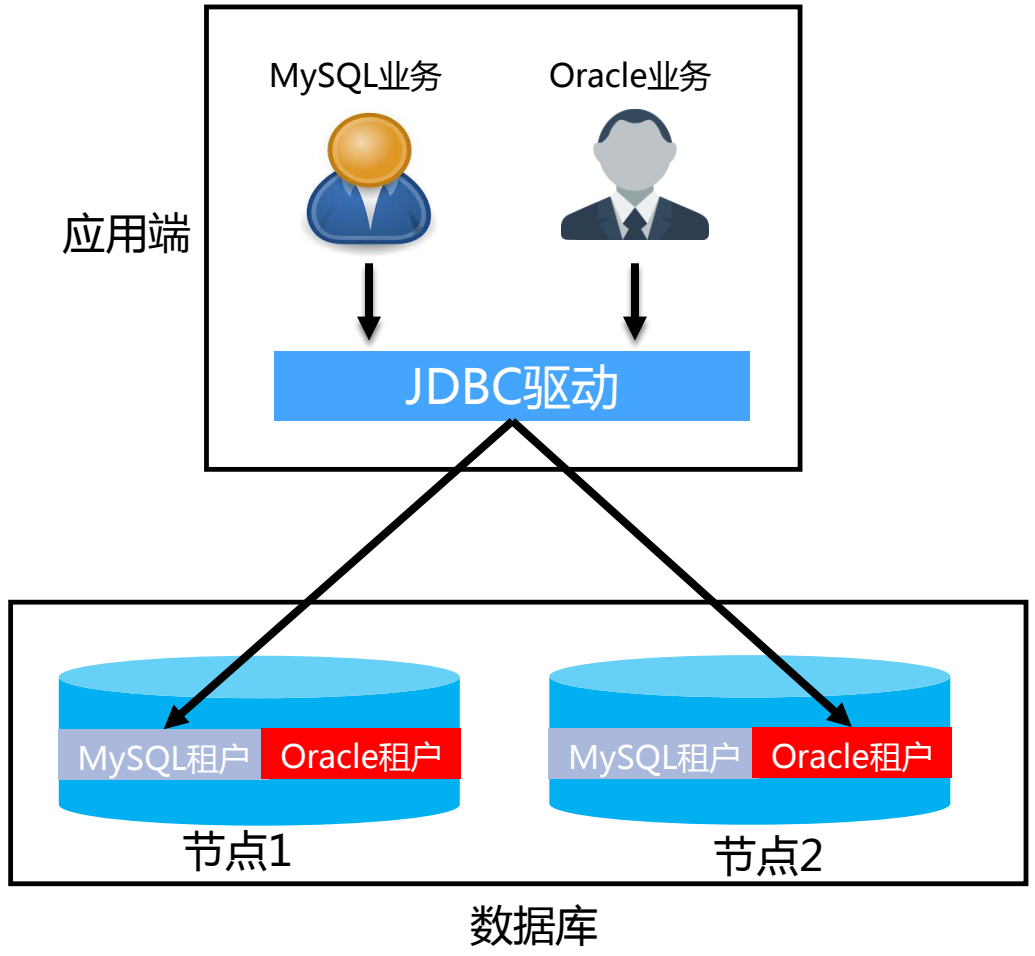






客户端驱动——JDBC driver

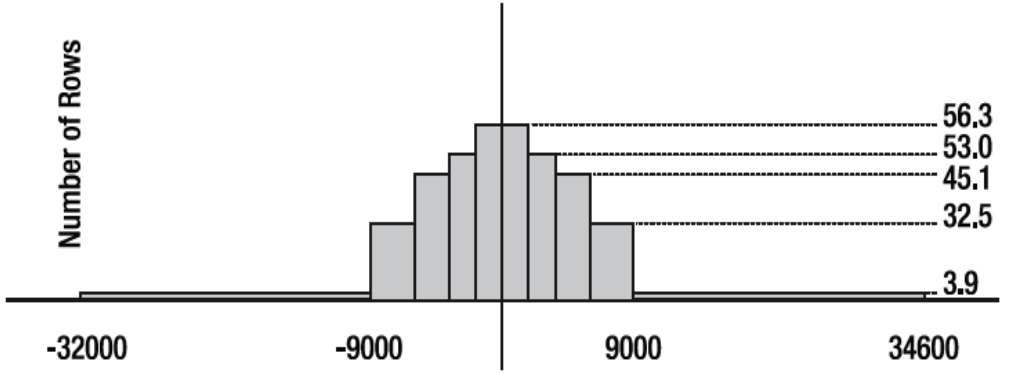
- 支持Oracle基础数据类型
  - Number/varchar2/timestamp w. timezone/CLOB
- 支持Array、Struct等复杂数据类型
- 基本分区裁剪功能
- 同时兼容MySQL/Oracle两种连接方式



03 内核升级

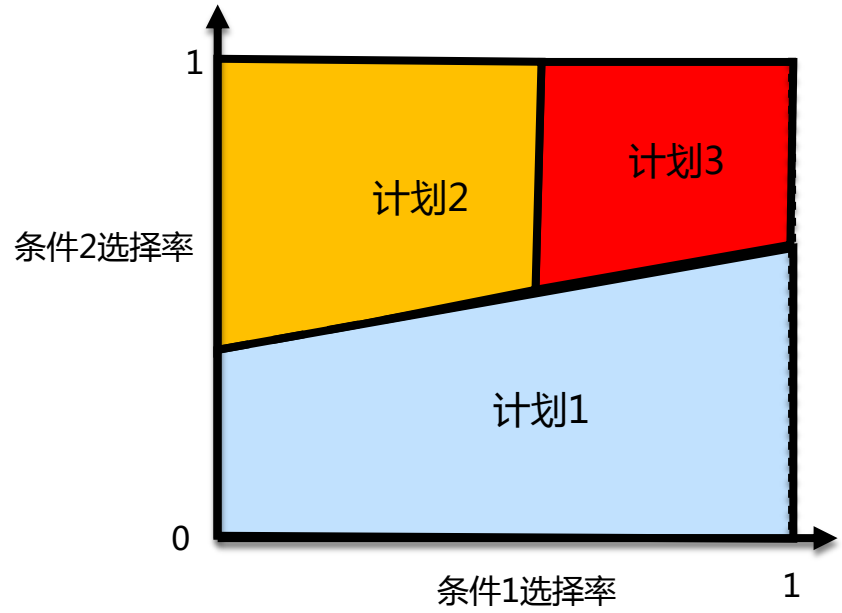
更强大的优化器——直方图

- 频率直方图
  - 频率直方图为列中的每个不同值维护一个行数信息
  - 适用于NDV比较小的场景
- 等高直方图
  - 等高直方图收集的是一个数值分布的信息
  - 适用于NDV比较大的场景
- 直方图建议
  - 自动探测需要收集直方图的列
  - 自动选择采样率
  - 全表扫描小表；采样大表，综合考虑系统资源与准确率并计算出一个合理的采样比例。



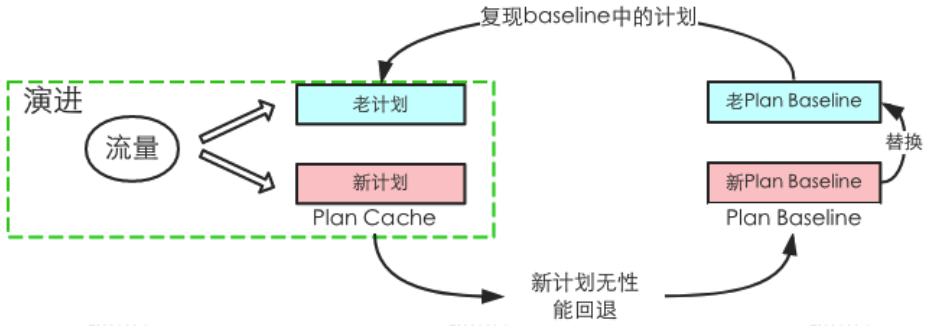
更强大的优化器——自适应计划匹配

- 相同SQL不同参数的计划匹配（大小账号问题）
  - 精确匹配（optimize always）
  - 模糊匹配（optimize once)
- 基于参数的计划匹配
  - Optimize as necessary
  - 参数值域空间映射为选择率空间
  - 利用直方图或者存储层采样估算参数选择率



更强大的优化器——计划演进

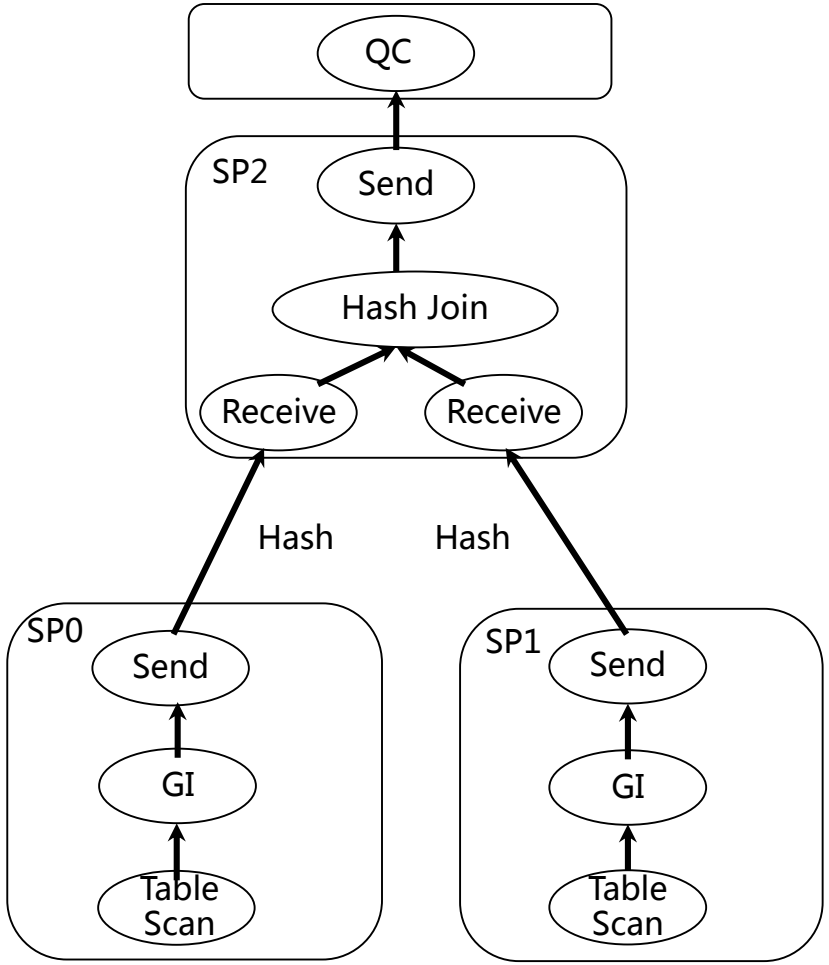
- 触发计划演进的场景
  - 统计信息更新
  - 版本升级
  - Schema变更
- 基于执行反馈的计划演进
  - 自动/手动计划基线捕捉
  - 计划变更时通过灰度引流的方式验证新老计划
  - 兼容计划绑定
  - 确定性能提升后升级新计划
  - 与ACS功能应用场景互斥




```
select cpu_time/evo_executions as avg_cpu_time, evolution, evo_executions, outline_data
from oceanbase.gv$plan_cache_plan_stat
where SQL_ID = '7B2EEF7D0BBF189782665D92CF8A1C8F';
avg_cpu_time evolution evo_executions  outline_data
3836         1         10             /*+ BEGIN_OUTLINE_DATA INDEX(@"SEL$1" "spm_db.t1"@"SEL$1" "idx_u") END_OU
2376         1         90             /*+ BEGIN_OUTLINE_DATA FULL(@"SEL$1" "spm_db.t1"@"SEL$1") END_OUTLINE_DAT
```

更强大的优化器——并行查询优化

- 更多优化维度
  - 数据重分布代价
  - 局部性
  - 最佳并行度
- 更多的执行路径
  - 动态filter
  - 动态分区裁剪
  - 局部排序
  - 中间数据落盘



OceanBase 2.0并行查询新框架

一切刚开始。。。

- 魔鬼在细节中
- 优化与性能
- 稳定性

谢谢

THANK YOU

微信公众号
OceanBase

蚂蚁金服
ANT FINANCIAL

OceanBase