

deeplus直播

关系型分布式数据库最佳实践

阿里云GTS-王宗瑞

分享人简介

MySQL DBA

- ✓负责百度分布式数据库方向
- ✓百信银行分布式数据库设计
- ✓主导网盘用户元信息分布式数据库，冷热数据分离设计、落地

DB-SA

- ✓负责阿里云政府、能源、新零售行业数据库商机拓展和轻交付
- ✓负责数据库架构师团队解决方案与最佳实践沉淀输出
- ✓负责分布式数据库中间件技术售前方向

DB-TM

- ✓参与保险TOP客户核心业务系统从Oracle向原生分布式数据库迁移改造
- ✓负责数据库交付标准化+工具化方向推进
- ✓负责国产数据库改造“交付平台化”方向规划与落地

目录

CONTENT

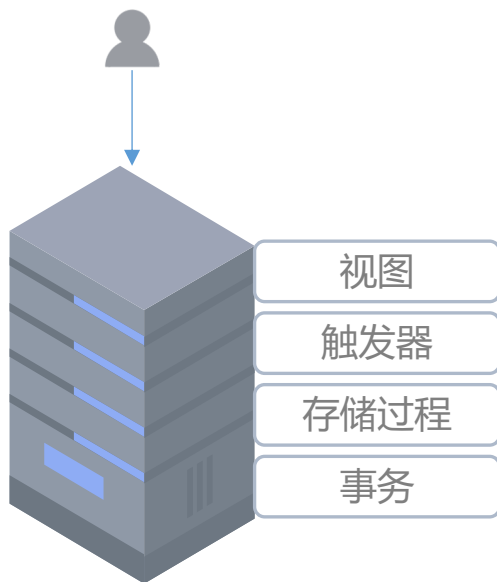
01 / 数据库从集中式到分布式的演进

02 / 分布式数据库分类

03 / 关系型分布式数据库最佳实践

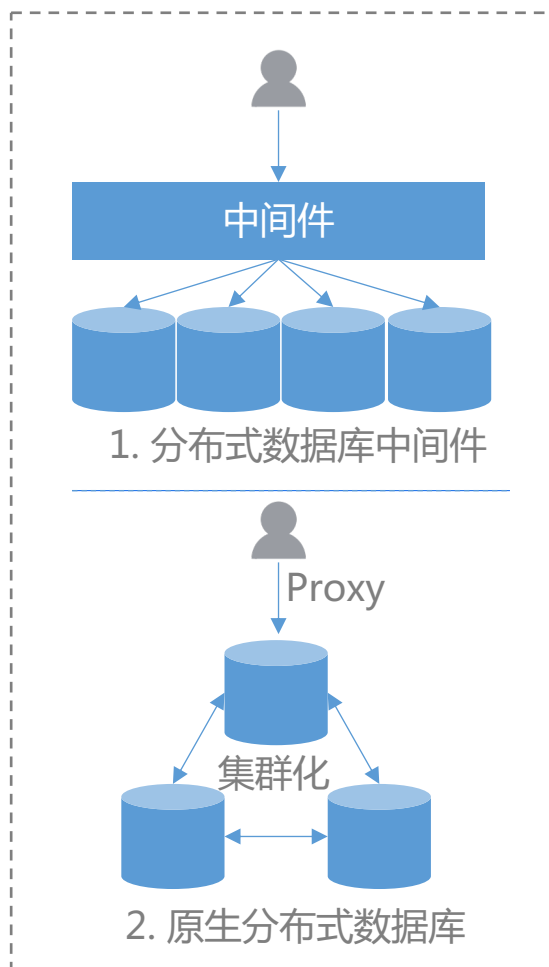
04 / 关系型分布式数据库总结与展望

数据库从集中式到分布式的演进

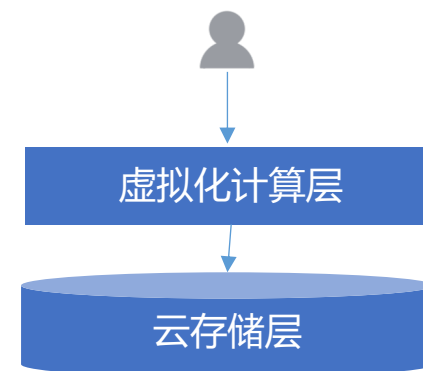


传统单机数据库

- 通用性强，特性全
- 可控数据规模性能优秀
- 受单机资源瓶颈制约
- 资源难以扩展



- 水平拆分场景性能优异
- 资源可线性扩展
- 需解决分布式事务、一致性
- 研发人员入门门槛高



云原生数据库

- 计算存储分离
- 资源弹性伸缩
- 单机数据库特性强兼容
- 对云平台强依赖

什么场景适合选择分布式数据库

方案选型优先级

高并发查询

高并发更新（事务）

海量在线数据存储

单表数据量大，性能退化

实时复杂分析查询超时

国产化数据库改造

数据库异地容灾

1.只读副本，读写分离 2.热点缓存（分布式缓存）

1.数据垂直拆分 2.分布式数据库

✓ 分布式数据库

1.数据归档 2.索引调优 3.分布式数据库中间件

1.流计算、预计算 2.分布式HTAP、OLAP数据库

✓ 原生分布式高兼容数据库

✓ 原生分布式数据库

分布式数据库选型

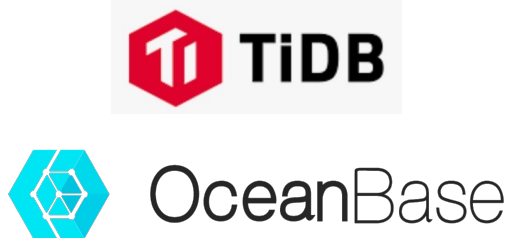
基于应用场景的分类

在线事务OLTP

分布式数据库中间件



原生分布式数据库

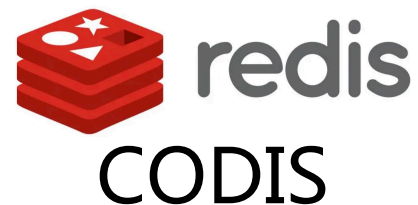


在线事务OLAP



非结构化&专有领域NOSQL

分布式缓存



文档数据库



宽列数据库



专有领域



关系型分布式数据库最佳实践

分布式数据库中间件

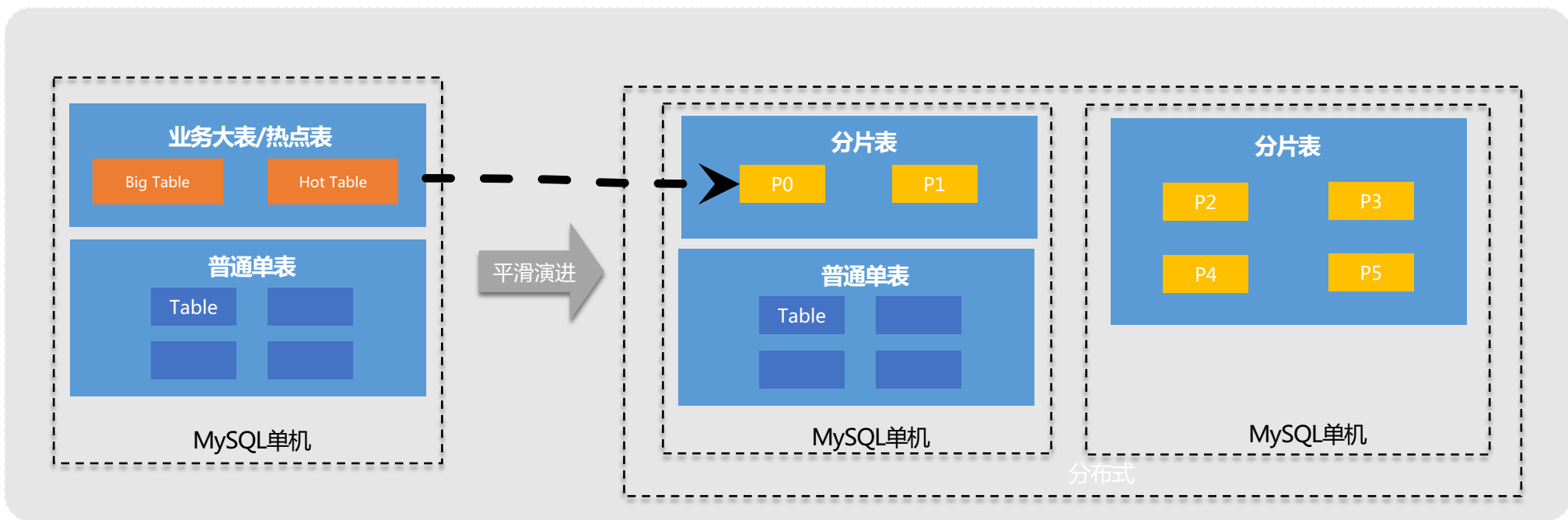
场景一 单机数据库演进分布式

痛点问题

1. 业务增长迅速，单机存储瓶颈，并发更新瓶颈
2. 大表DDL耗时长，冲击业务IO

核心技术

1. 大表拆分键选择、小表广播
2. 分布式事务保证
3. 分布式ONLINE DDL
4. 增量数据实时回流



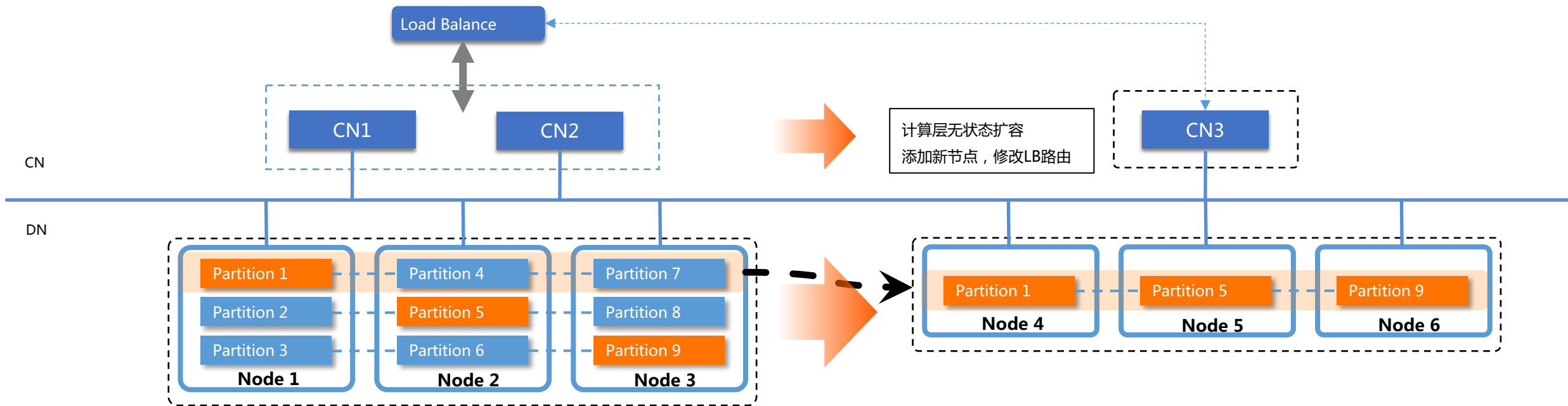
场景二 线性平滑扩容

痛点问题

1. 突破单机瓶颈，扩容对业务应无感知
2. 存在热点数据，拖累同分区数据访问
3. 分布式中间层同样存在计算瓶颈

核心技术

1. partition数据全量+增量迁移+数据校验
2. 扩容粒度层级：库级别 < 表级别 < 记录级别
3. 路由信息管理，路由性能保障、路由高可用
4. 中间层自适应升配/扩容



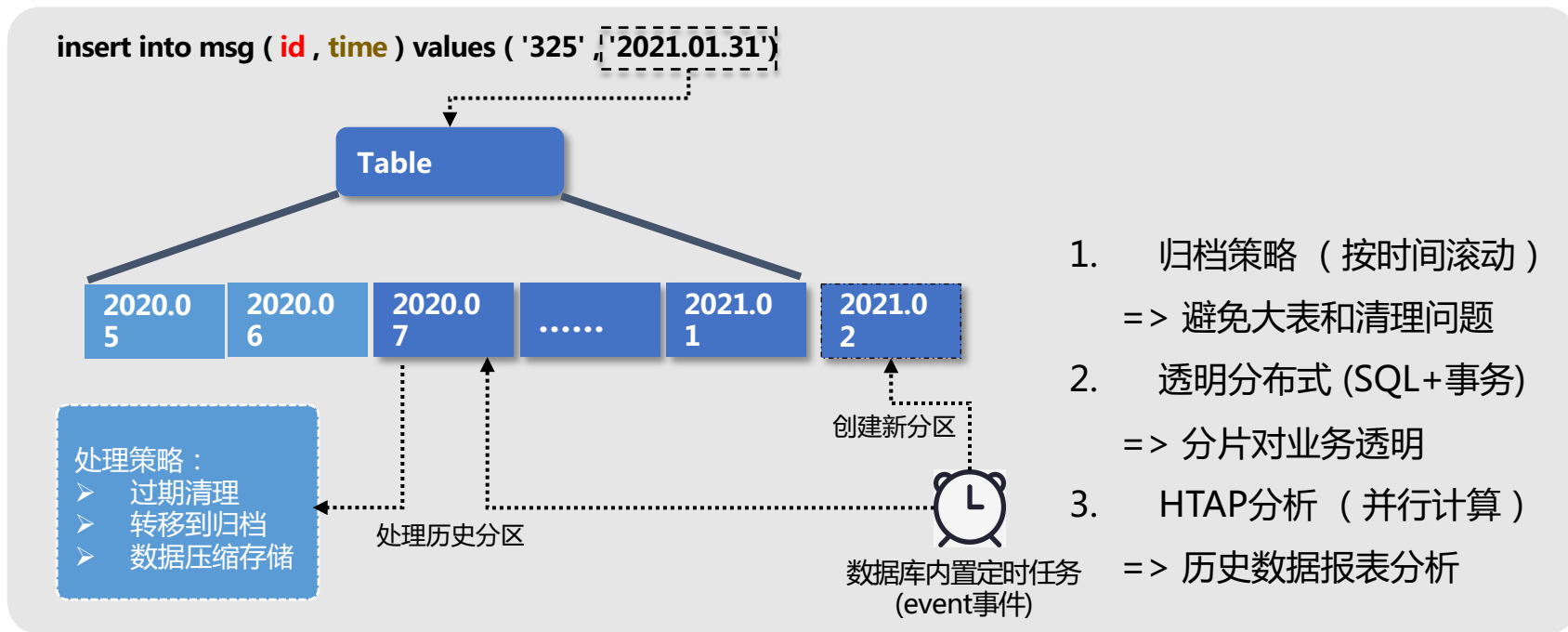
场景三 准在线数据归档

痛点问题

1. 在线业务表中数据量过大，性能恶化
2. 存储增长导致昂贵的在线数据库存储成本企高
3. 手工归档操作繁琐易出错，对IO有冲击
4. 在线服务有访问归档需求需要切换入口

核心技术

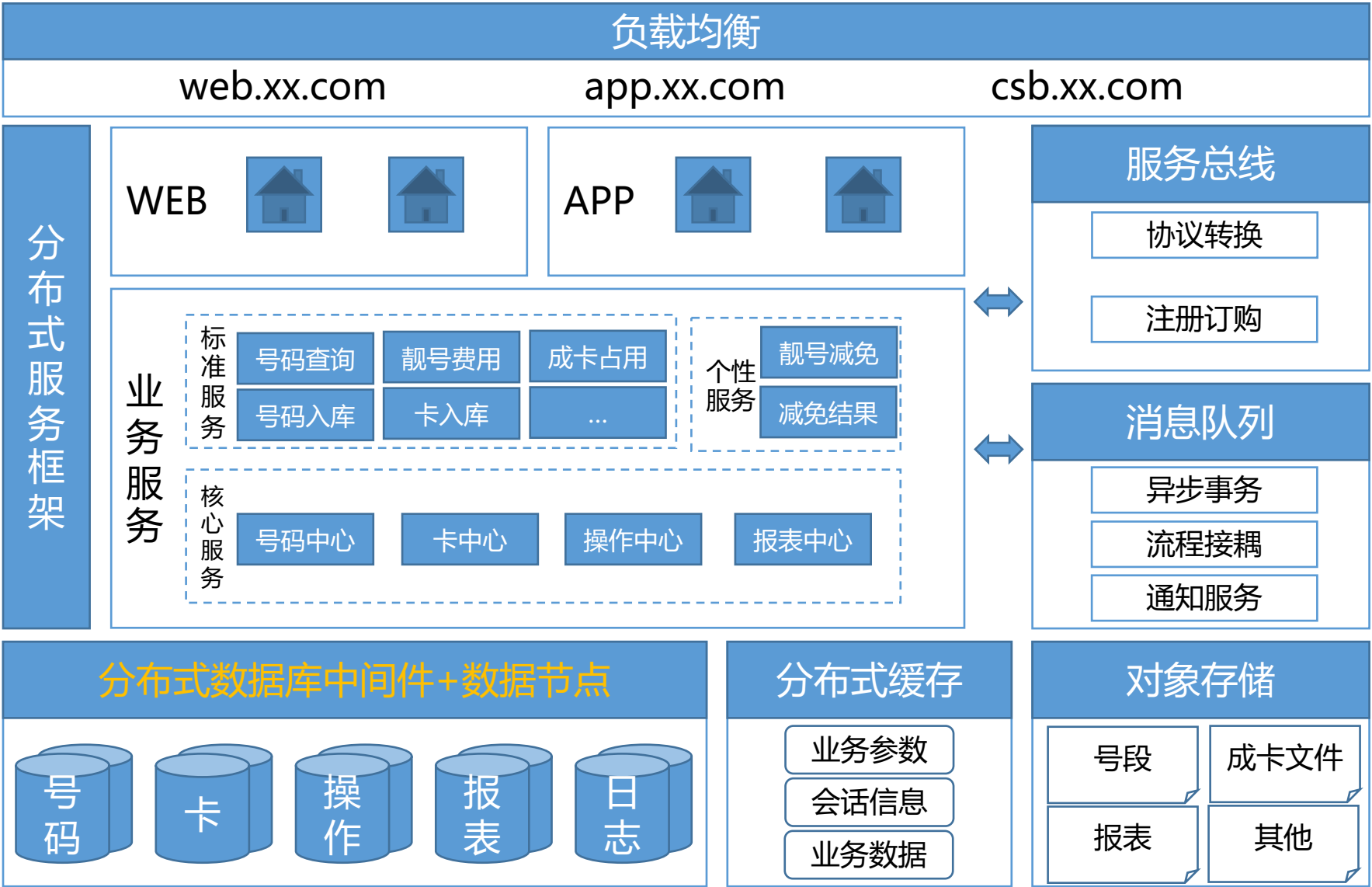
1. 归档策略可配置，自动清理过期数据，自适应限速
2. 归档数据访问路由与在线路由归一化
3. 归档库自动调度廉价存储设备
4. 并发查询提高分析效率



分布式数据库中间件设计最佳实践

1. 确保分布式表拆分键是业务最常用的等值查询条件，数据分布均匀度其次考量
2. 底层是MySQL数据节点innodb引擎条件下，单行长度不建议超过2KB，以此为条件，数据量不超过2kw，一般不会出现单表性能瓶颈，不必拆分
3. 广播表的确定，更新频率低的优先级，远大于表内数据量多寡
4. 如果支持，当然可以使用分布式事务，简化业务开发逻辑，但有代价，涉及SQL占整体SQL比例不要超过10%，最好不超过5%，且无高并发场景
5. 如果支持，也可以使用全局二级索引，但一张分布式表不建议创建超过3个，不然写入性能、事务吞吐都将受到影响（前提是有数据强一致性保证，否则不要使用）

分布式数据库中间件实战案例-运营商号卡业务



- 某运营商号码业务、资源卡业务等核心业务数据库采用分布式数据库中间件。
- 成功从传统架构转型到互联网架构。
- 通过在客户域，订单域，产品域进行分布式解耦，以及数据库拆分，使得业务域获得了水平扩展的能力。
- 对于多维查询的需求，通过增量数据传输同步到其他数据库中。
- 如同步到ES中满足实时业务的做多纬度数据查询，再如按照省的纬度同步到多个只读数据库中，为每个省提供数据库级别查询能力。

关系型分布式数据库最佳实践

原生分布式数据库

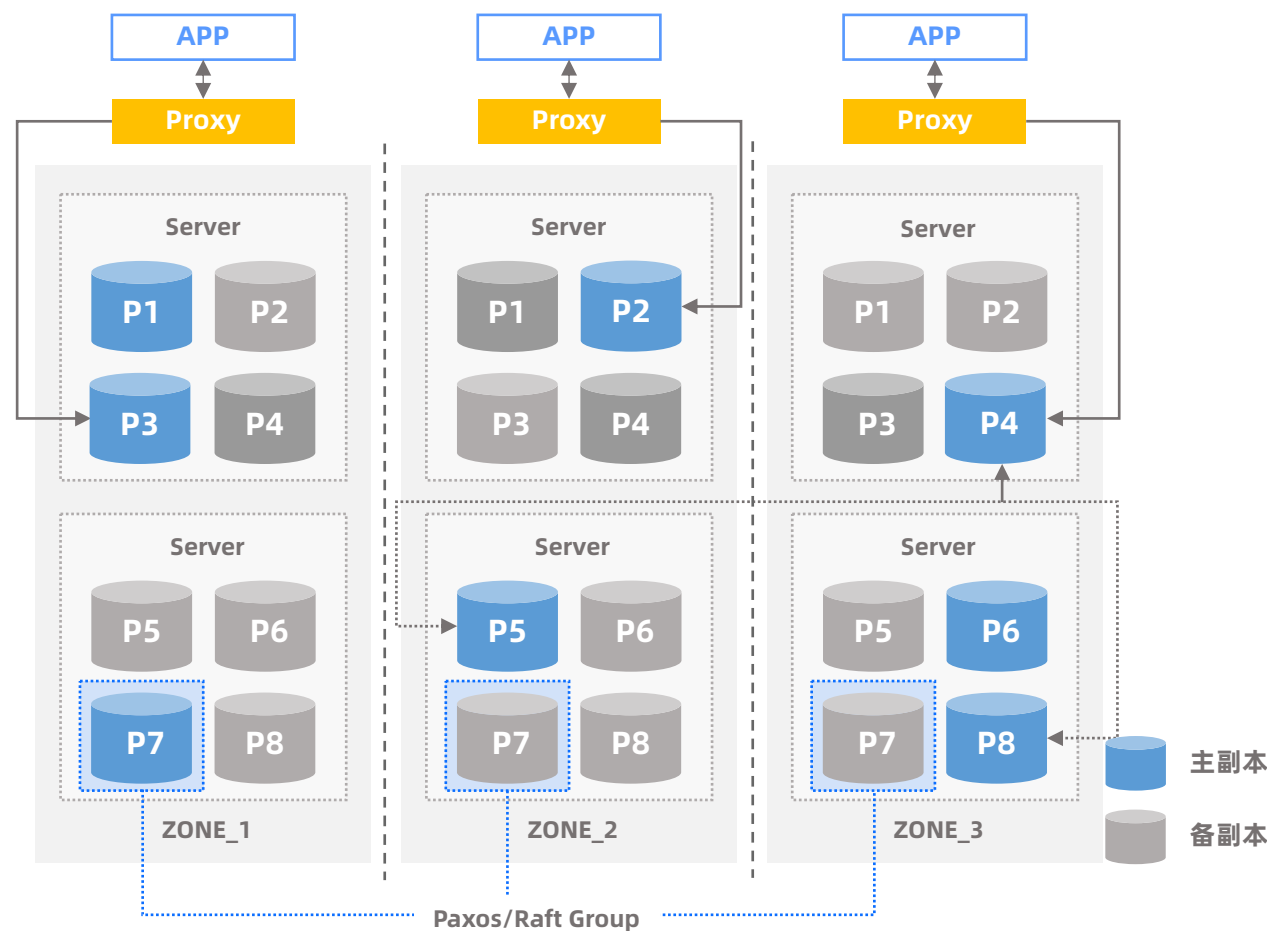
场景一 高并发交易业务

痛点问题

1. 事务更新并发高，单机难以承载或scale up成本高昂
2. 业务有大促、抢购场景，资源需要快速伸缩
3. 数据一致性要求非常高
4. 分布式改造希望平滑，应用改造少

核心技术

1. 分区表技术，全局+本地索引
2. 多点写入，实现分布式数据写入负载均衡
3. 资源智能调配，一键扩缩容
4. 单机数据特性、语法兼容度100%
5. 数据由分布式一致性算法保障



场景二 数据库多地多中心多活

痛点问题

1. 业务对数据容灾有高要求
2. 不同基础设施下的数据库高可用需求
3. 海量数据条件下，备份恢复难题
4. 小型数据故障需快速修复+数据安全需求



核心技术

1. 分布式一致性算法，RPO=0（少数派故障）
2. 自恢复机制，确保RTO分钟级
3. 快照+增量日志的快速备份恢复技术
4. 数据闪回、全链路数据审计



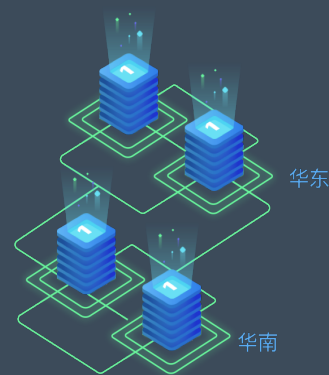
单机房架构



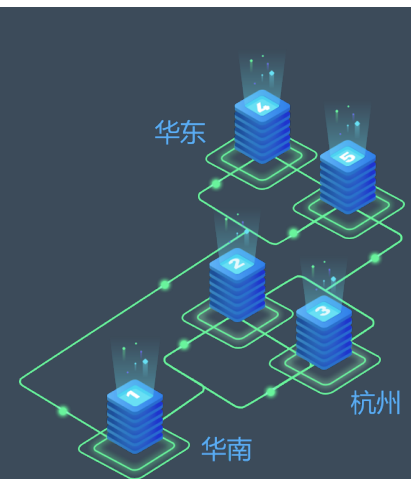
同城双机房架构



两地三中心架构



异地多活架构



三地五中心多活架构

场景三 国产化数据库改造

痛点问题

1. 老系统在Oracle、DB2，业务复杂改不动
2. 正式割接难以确定新系统表现，存在未知风险
3. 需要平滑可靠的改造回滚方案
4. 改造分布时候，性能退化严重



核心技术

1. 高兼容传统数据库的SQL、特性
2. 完善的评估报告、采集回放方案
3. 可靠的异构数据双向回流技术
4. 根据需要选择分布式，不要刻舟求剑

可行性和
风险评估

风险保障

迁移规划
和准备

切割上线

迁移改造和验证

原生分布式数据库设计最佳实践

- 没有高并发更新的数据表，不需要刻意分区，付出不必要的性能代价
- 关联查询条件固定的表，要放在相同表组，杜绝跨节点关联
- 数据的自均衡，同样有性能代价，要根据业务访问压力权衡是否打开
- 关注分布式执行计划，通过hint指定最优的join方式，弥补优化器自身的失准
- 如果可能，绑定执行计划是很好的调优手段
- 可能的话，复杂分析类查询，走OLAP入口
- 特定场景下，可配合数据库中间件食用，效果更佳

原生分布式数据库实战案例-XX银行/保险

水平扩展、高并发处理、低采用成本等分布式优势，为银行业务的快速发展提供了强有力的保障。

业务挑战：

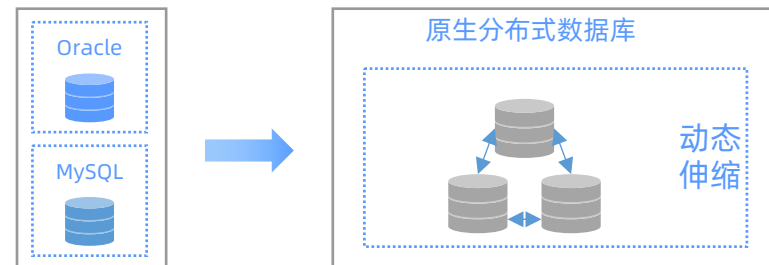
- 银行业传统的IOE架构无法支撑高并发量和数据库的高弹性需求，限制业务的快速发展。
- 互联网金融业务在“科技引领”战略目标指引下，发展需求巨大。

解决方案：

- 原生分布式数据库DBPaaS，提供核心基础数据服务，遵循以高性能和稳定性为主、高兼容性为辅的设计原则，通过集群部署，能够自由增加数据库节点，实现弹性扩展，不需要停止业务系统进行割接操作，满足了客户对性能及高可用的严苛要求。
- 本地化的弹性扩展能力，解决了传统关系型数据的实时结构化数据存储容量瓶颈问题；数据安全能力，采取一主多备，当主节点异常时，系统自动切换至备节点，保障系统操作连续服务不中断，从而保证了SLA。

客户收益：

- 在线水平扩展能力：只需增配节点即可，且能够在不中断业务的情况下，快速扩展硬件能力
- 高并发处理能力：能够轻松应对远高于原架构的突发高并发流量
- 软硬件和运维成本：采用普通的PC服务器，其成本低于小型机+SAN存储，使用分布式数据库后的硬件成本和硬件维保费用也有所降低



分布式数据库总结与展望

关系型分布式数据库总结

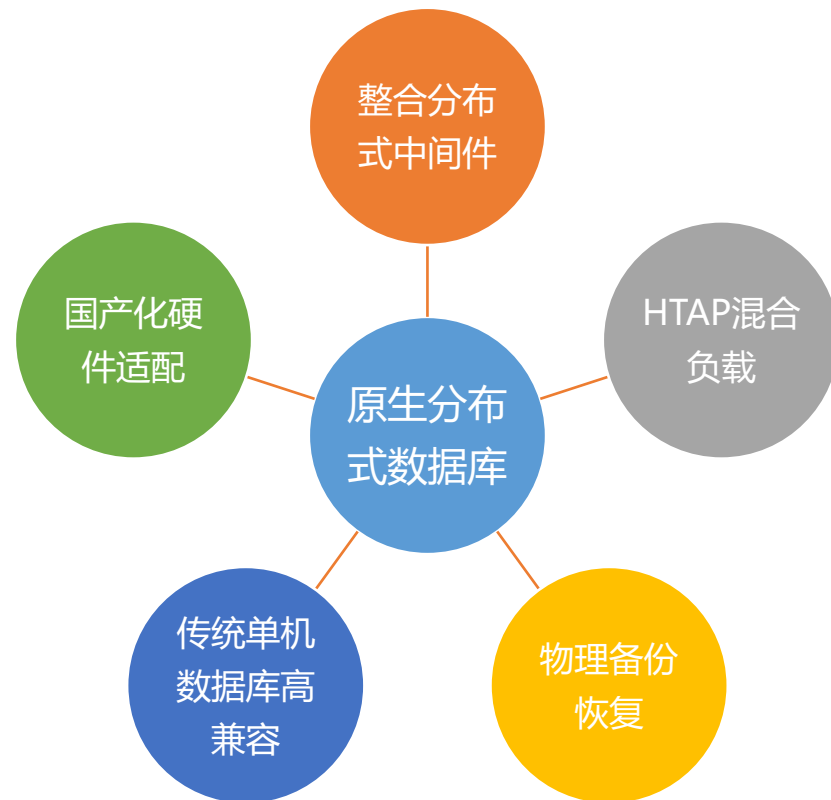
分布式数据库中间件

- 优势
 - 架构简单，易学习理解
 - 灵活，可挂载不同数据节点
 - ShareNothing架构，数据天然分片
场景几乎具有无限线性扩展型
- 劣势
 - 对于分区键的取舍是技术活
 - 分布式事务是天然难题
 - 与存储节点解耦，过于依赖存储节点实现，如数据一致性

原生分布式数据库

- 优势
 - 计算存储高耦合，技术深入，如CBO、与MySQL、Oracle协议高度兼容等
 - 天然分布式一致性保证，异地多活
 - 使用体验与单机数据库几乎一致
- 劣势
 - 往往架构复杂，单机数据库维护调优经验难复用
 - 对于OLAP场景表现欠佳
 - 扩展性依然受限于固有硬件条件

关系型分布式数据库展望



二者虽然在各自起点选了不同的科技树，但现在大有融合趋势；开源是趋势所向

deeplus直播

Thank You!