# 概述

## AliSQL

AliSQL是阿里云深度定制的独立MySQL分支，除了社区版的所有功能外，AliSQL提供了类似于MySQL企业版的诸多功能，如企业级备份恢复、线程池、并行查询等，并且AliSQL还提供兼容Oracle的能力，如sequence引擎等。

## RDS

RDS MySQL使用AliSQL内核，为用户提供了MySQL所有的功能，同时提供了企业级的安全、备份、恢复、监控、性能优化、只读实例等高级特性。

## TDDL

DRDS前身为淘宝TDDL，是近千核心应用首选组件。

## DRDS

DRDS诞生于2008年淘宝“去IOE”时代，过去的十多年中，DRDS经历了每年的“双11”并发挥了重要作用。5年前，DRDS正式商用，目前服务了无数企业的核心应用。

对于任何一个产品而言，它的出现以及能力的变化都与其面向的业务相关。对于DRDS而言，可以粗略地把业务场景分为3类。第一类是DRDS从一开始出现就在解决的面向最终消费者的高并发业务数据库的需求，这也是DRDS能够很好解决的场景。第二种场景就是DRDS上云提供服务之后遇到的企业级数据库需求，它希望数据库具有综合负载能力、可持续运维和7\*24小时稳定性以及并发、计算和存储的扩展性。



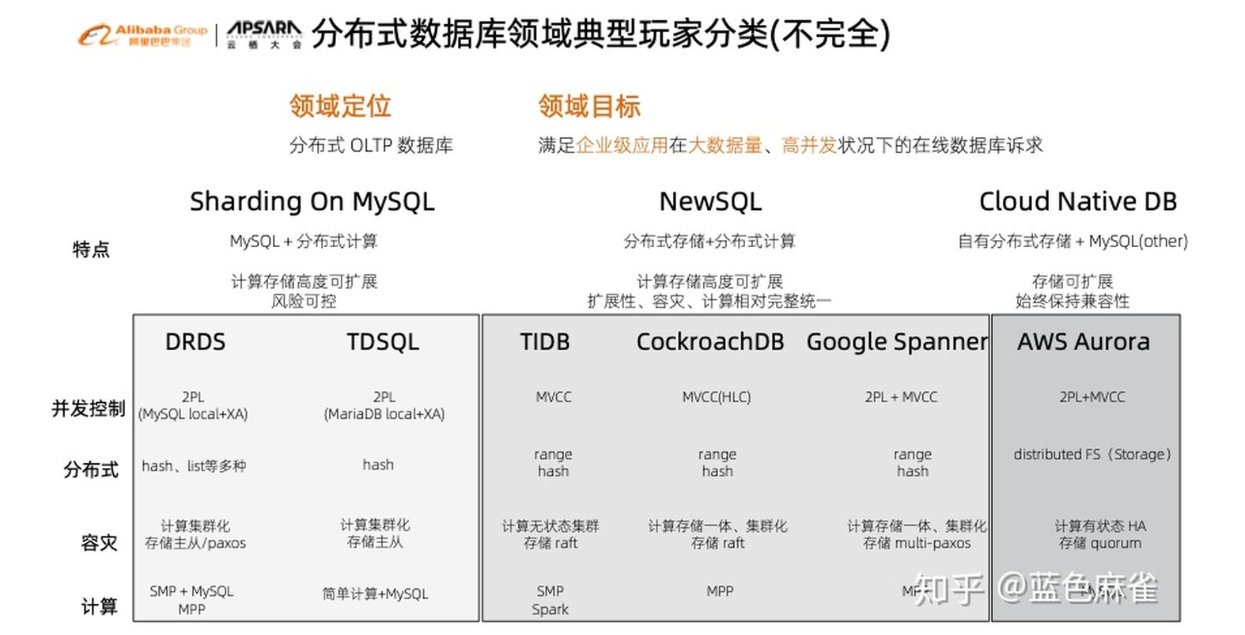
如今，解决上述某几个问题的数据库大致可分为三类：

DRDS以及Sharding On MySQL数据库，主要**基于MySQL和分布式计算能力，使得计算存储高度可扩展，风险可控**。

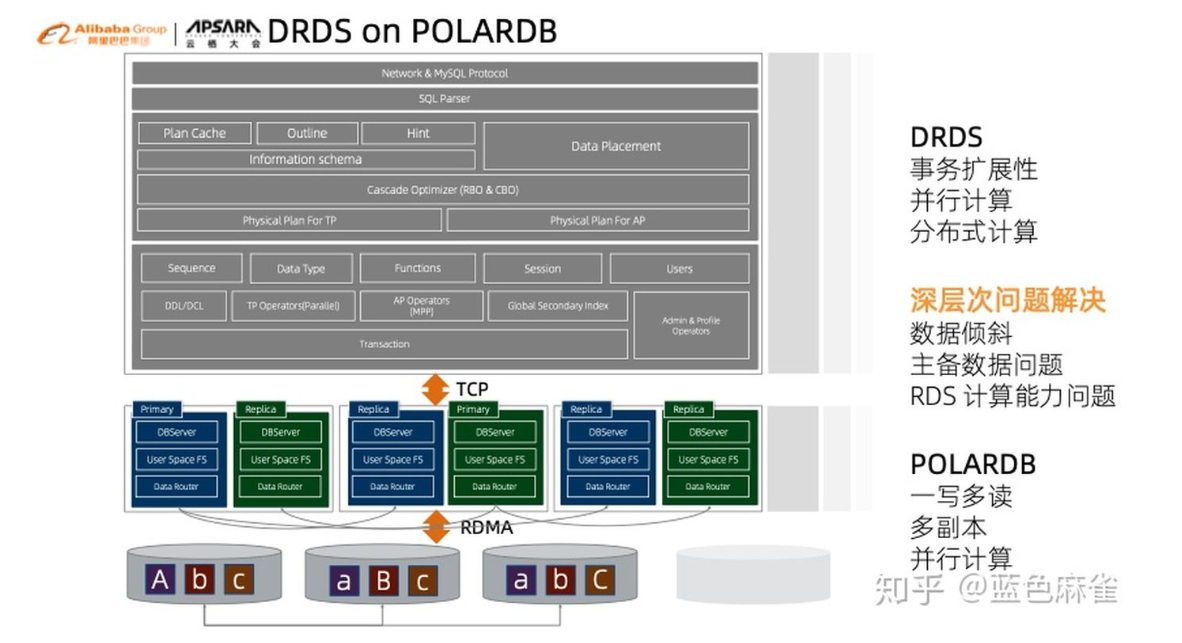
NewSQL数据库，核心特点就是存储与计算分离。

Cloud Native DB，强调存储可扩展以及全兼容的能力。

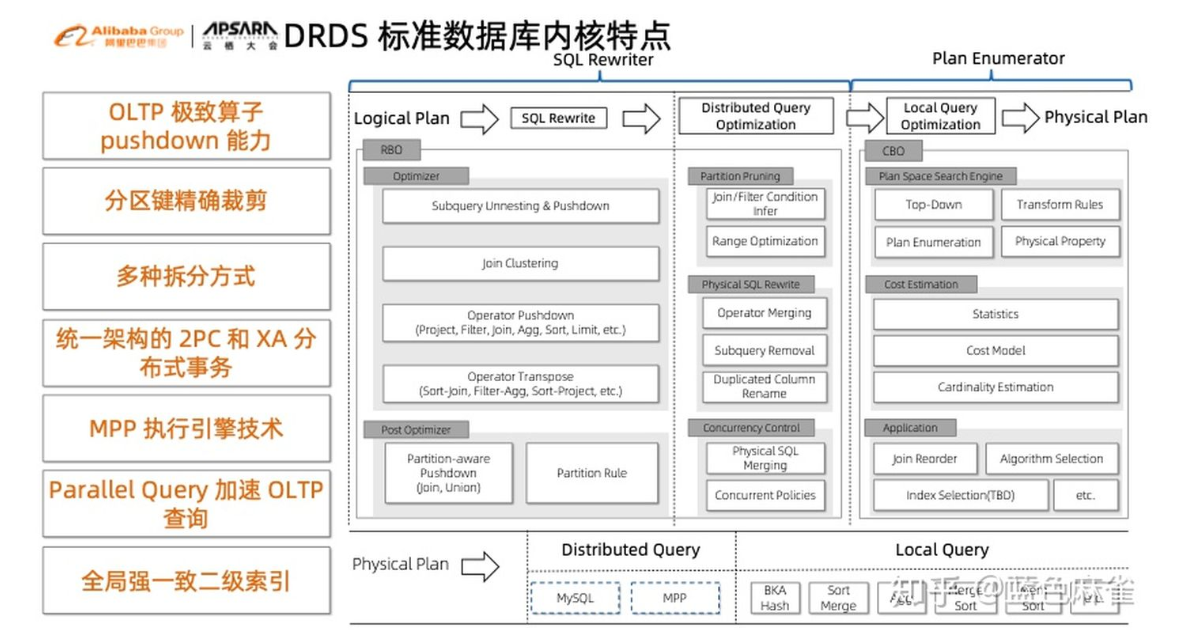
而通过并发控制、分布式、容灾以及计算这四个维度能够更加深入剖析数据库能力。



DRDS ON MySQL实现得很好，但在存储方面则让人“既爱又恨”。因此，在POLARDB上线后的第一时间，阿里云就实现了DRDS On POLARDB。DRDS和POLARDB两者的布局不同，POLARDB层面侧重一写多读的能力，DRDS层面则侧重事务扩展性。DRDS On POLARDB解决了数据倾斜、主备数据以及RDS数据能力的问题，因此相对比较稳定并且具有面向未来的一些特性。



DRDS标准数据库内核的发展经历了从超高并发用户侧服务逐步转向了企业级应用场景的转变。发生这样转变的驱动力也有三个，即业务场景、经典数据库理论以及Benchmark。DRDS标准数据库内核非常注重分布式的能力，比如OLTP极致算子Pushdown能力、分区键精确裁剪、多种拆分方式、统一架构的2PC和XA分布式事务、全局强一致二级索引、MPP执行引擎技术、OLTP查询加速等。



## X-Engine

之前AliSQL使用InnoDB引擎，而InnoDB存在扩展性瓶颈。X-Engine引擎则采用了LSM-Tree架构，并进行了创新。在架构最上层提供了高度并发的事务处理流水线，中间实现了无锁内存表Memtable。此外，为了解决读写冲突，X-Engine将每个Meta信息作为一个独立版本。X-Engine对于磁盘存储层也进行了整体重构，并且还引入了FPGA作为硬件加速器。

## X-DB

16年，阿里内部开始了X-DB的研发。X-DB首先就是要克服副本数据一致性的问题。 X-DB的团队选择自研Paxos协议来替代MySQL内置的主备同步逻辑。

## PolarDB

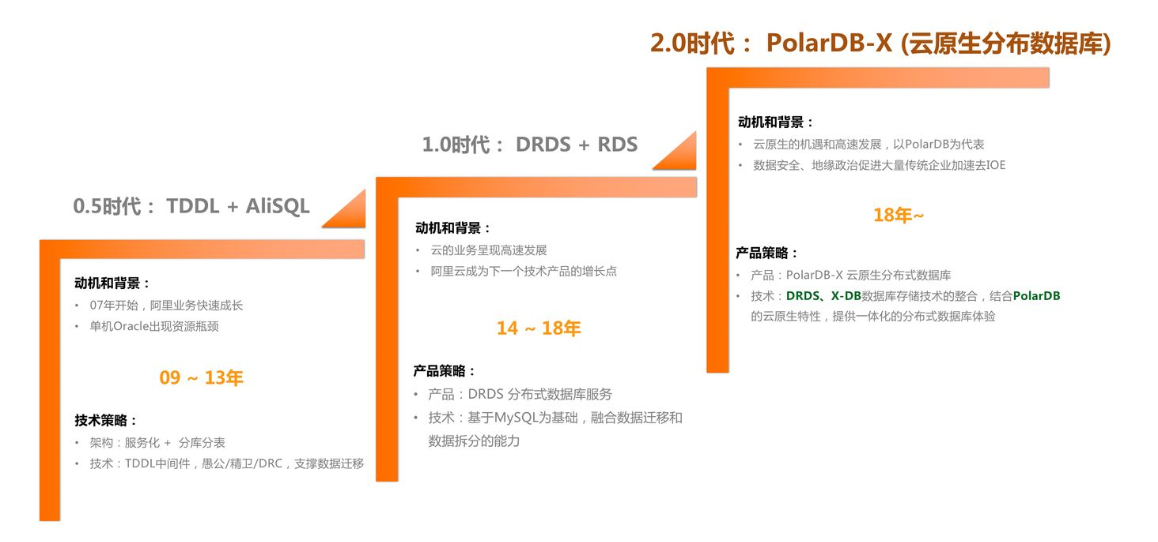
PolarDB（这里指PolarDB For MySQL）是一个基于共享存储技术的云原生数据库。PolarDB在存储空间上可以做到很强的弹性能力，但一般使用情况下，其计算能力、写入能力依然存在单机的上限。

PolarDB-X 2.0这种share-nothing的架构，使得包括计算、写入、读取、存储等在内的所有资源，都具备了可水平扩展的能力，因此不会存在单机的瓶颈上限。

但是，share-nothing的架构在单纯的数据容量的弹性上，是不如PolarDB的共享存储架构的。

## PolarDB-X

2003 年淘宝网成立之初采用的是经典的LAMP架构，随着用户量迅速增长，单机MySQL数据库很快便无法满足数据存储需求，之后淘宝网进行了架构升级，数据库改用Oracle。随着用户量的继续快速增长，Oracle数据库也开始成批成批的增加，即使这样，仍然没有满足业务对数据库扩展性的诉求，所以阿里巴巴内部在2009年时发起了著名的去IOE运动，PolarDB-X也开启了自己的演进之路。



### PolarDB-X 0.5时代

去IOE的关键一环是实现对Oracle的替换，当时淘宝的业务体量已很难用成熟的技术产品支撑，为了避免以后出现卡脖子情况，技术的自力更生和自主可控成为一个核心诉求。一方面，随着x86技术日趋成熟，稳定性与小型机的差距不断缩小，另一方面，MySQL采用轻量化线程模型并具备高并发的支持能力，其生态逐步完善，因此新方案采用了基于Sharding技术+开源MySQL的分布式架构（TDDL + AliSQL），我们称为PolarDB-X 0.5版本时代。这代产品的特征是以解决扩展性为目标、面向系统架构使用，尚不具备产品化能力。

### PolarDB-X 1.0时代

随着这套架构的逐渐成熟，14年开始，我们基于阿里云走上了云数据库发展之路。作为分库分表技术的开创者，我们推出了DRDS + RDS的分布式云数据库服务，我们称为PolarDB-X 1.0时代。这代产品的特征是采用Share-Nothing架构、以解决存储扩展性为出发点、提供面向用户的产品化交付能力。1.0版本是国内第一家落地分布式技术的云服务，成为云市场上分布式数据库技术方向的开创者和引领者。

针对用户使用中的痛点，我们不断进行产品能力迭代，陆续支持了分布式事务、全局二级索引、异步DDL等内核特性，持续改进SQL兼容性，实现子查询展开、Join下推等复杂优化，并开发了平滑扩容、一致性备份恢复、SQL闪回、SQL审计等运维能力。这段时间我们不断扩展所谓分库分表中间件技术的能力边界，试图找到它的能力上限。这个探索的过程，一方面使我们的计算层能力更加稳定、丰富和标准化，另一方面也促使着DRDS从中间件到分布式数据库的蜕变。

### PolarDB-X 2.0时代

18年开始，我们逐渐触碰到了计算层的能力边界，比如无法提供RR隔离级别的事务能力、计算下推受限于SQL表达能力、数据查询的传输效率底下、多副本的线性一致性不可控等，这些问题像一个无法穿透的屏障，我们能看到屏障的对面是什么，能看到所有障碍都指向了同一个方向：计算层需要与存储层深度融合。

值得高兴的是，我们的AliSQL分支从诞生起就没有停止前进的步伐，通过集团业务多年的技术锤炼，基于AliSQL演化而来的X-DB数据库（包括X-Paxos 协议库、X-Engine存储引擎等），在全球三副本、低成本存储等技术有了非常好的沉淀。

与此同时，基于云原生架构理念的PolarDB，通过引入RDMA网络优化存储计算分离架构，实现一写多读的能力，并提供资源池化降低用户成本，优化并提供秒级备份恢复、秒级弹性等能力，成为公有云增速最快的数据库产品。

这些技术探索和沉淀，使PolarDB-X团队有底气开始思考基于云架构的分布式数据库应该是什么样的形态，从宏观角度来看，会有云原生、国产化、分布式、HTAP等诉求，从用户角度来看，需要满足用户使用云的一些期望，比如用户的数据库数据永远不会丢，即使主机异常宕机，这里需要有数据强一致以及高可用容灾等能力。再比如随着移动互联网和IoT的普及，数据层面会有爆炸式的增长，以及今年疫情之后有更多的企业会关注IT成本，因此高性能、低成本和可扩展的计算和存储能力也成为普适性诉求，另外类似Snowflake的按查询付费的弹性能力，也是市场的另一个诉求。因此，简单总结一下，下一代的分布式数据库需要具备：金融级高可用和容灾、水平扩展、低成本存储、按需弹性、透明分布式、HTAP混合负载、融合新硬件等。

所以，2019年PolarDB-X团队完成DRDS SQL引擎和X-DB数据库存储技术的融合，并结合PolarDB的云原生特性，承上启下推出了新一代的云原生分布式数据库。专注解决单机解决不好的分布式扩展性问题，满足分布式数据一致性要求，并支持从单机到分布式的平滑演进，利用云原生技术的优势提供低成本和弹性能力，在交付上具备线上公有云、线下专有云、轻量化等全形态输出。

# 架构

参考：

<https://zhuanlan.zhihu.com/p/88899448>

PolarDB-X承担着OLTP在线核心数据库的职责与定位，可与数据集成、数据传输，缓存、大数据生态配合使用。

DRDS/TDDL是典型的**水平扩展**分布式数据库模型，区别于传统单机数据库share anything架构，DRDS/TDDL采用**share nothing**架构，share nothing架构核心思路利用普通的服务器，**将单机数据拆分到底层的多个数据库实例上，通过统一的Proxy集群进行SQL解析优化、路由和结果聚合，对外暴露简单唯一的数据库链接**。整体架构如图所示，包含DRDS服务模块、DRDS管控模块、配置中心、监控运维、数据库服务集群、域名服务模块。

# 事务

# 数据分布

# 复制/一致性

# 备份恢复

# 容错/故障切换

# 兼容性

# 扩展性

# 高并发

# 高可用

# 数据安全

# 数据压缩

# 数据迁移

# 性能优化

# 未来发展