

ALIBABA CLOUD

# 阿里云

## 专有云敏捷版数据库场景

云数据库  
产品简介

产品版本：V1.2.0

文档版本：20240920

 阿里云

## 法律声明

阿里云提醒您在使用或阅读本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 警告	适用于可能会产生风险的场景。介绍用户在操作前就必须充分了解的信息、操作前必须要注意的事项或已具备的条件。	 <b>警告</b> 重启操作将导致业务短暂中断，建议您在业务低峰期执行重启操作，或确保已完成数据备份。如有必要，请联系阿里云技术支持提供协助。
 重要	在操作前需要用户了解的提示信息、补充信息、注意事项、限制信息等。	 <b>重要</b> 再次登录系统时，您需要修改登录账户的初始密码。
 说明	用于额外的补充说明、最佳实践、窍门等，不是用户必须了解的信息。	 <b>说明</b> 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击 <b>设置</b> > <b>网络</b> > <b>设置网络类型</b> 。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面，单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
<i>斜体</i>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.关于本手册	07
2.专有云敏捷版数据库场景DBStack整体介绍	08
2.1. 产品概述	08
2.2. 产品优势	08
2.3. 产品架构	09
2.4. 部署架构	10
2.5. 产品形态	11
2.6. 功能大图	14
3.数据库产品介绍	15
3.1. AnalyticDB PostgreSQL	15
3.1.1. 产品概述	15
3.1.2. 产品优势	15
3.1.3. 产品架构	15
3.1.4. 部署架构	17
3.1.5. 功能大图	18
3.2. RDS MySQL	19
3.2.1. 产品概述	19
3.2.2. 产品优势	19
3.2.3. 产品架构	20
3.3. PolarDB O引擎	21
3.3.1. 产品概述	21
3.3.2. 产品优势	21
3.3.3. 产品架构	22
3.3.4. 部署架构	23
3.3.5. 功能大图	23
3.4. PolarDB-X	24

---

3.4.1. 产品概述	24
3.4.2. 产品优势	24
3.4.3. 产品架构	25
3.4.4. 功能特性	26
3.5. Redis	27
3.5.1. 产品概述	27
3.5.2. 产品优势	28
3.5.3. 产品架构	28
3.5.4. 功能特性	29
3.6. Lindorm	30
3.6.1. 产品概述	30
3.6.2. 产品优势	31
3.6.3. 产品架构	32
3.6.4. 功能特性	37
3.6.5. 应用场景	38
3.6.6. 使用限制	44
3.6.7. 基本概念	45
3.7. GDB	47
3.7.1. 产品概述	47
3.7.2. 产品优势	47
3.7.3. 适用场景	47
3.7.4. 产品架构	48
3.7.5. 部署架构	49
3.7.6. 功能特性	50
3.8. ADAM	51
3.8.1. 产品概述	51
3.8.2. 产品架构	51
3.8.3. 产品优势	52

---

3.9. DTS	53
3.9.1. 产品概述	53
3.9.2. 产品优势	53
3.9.3. 部署架构	53
3.9.4. 功能特性	54
3.9.4.1. 功能大图	54
3.9.4.2. 数据迁移	54
3.9.4.3. 数据同步	57
3.9.4.4. 数据一致性	59
3.10. DMS	61
3.10.1. 产品概述	61
3.10.2. 产品优势	61
3.10.3. 产品架构	62
3.11. DAS	63
3.11.1. 产品概述	63
3.11.2. 产品优势	64
3.11.3. 产品功能	64
3.11.4. 应用场景	66

# 1.关于本手册

本手册从产品架构、部署架构、功能大图、产品形态、产品优势等角度，介绍了专有云敏捷版数据库场景DBStack以及其包含的数据库产品。

## 2. 专有云敏捷版数据库场景DBStack整体介绍

### 2.1. 产品概述

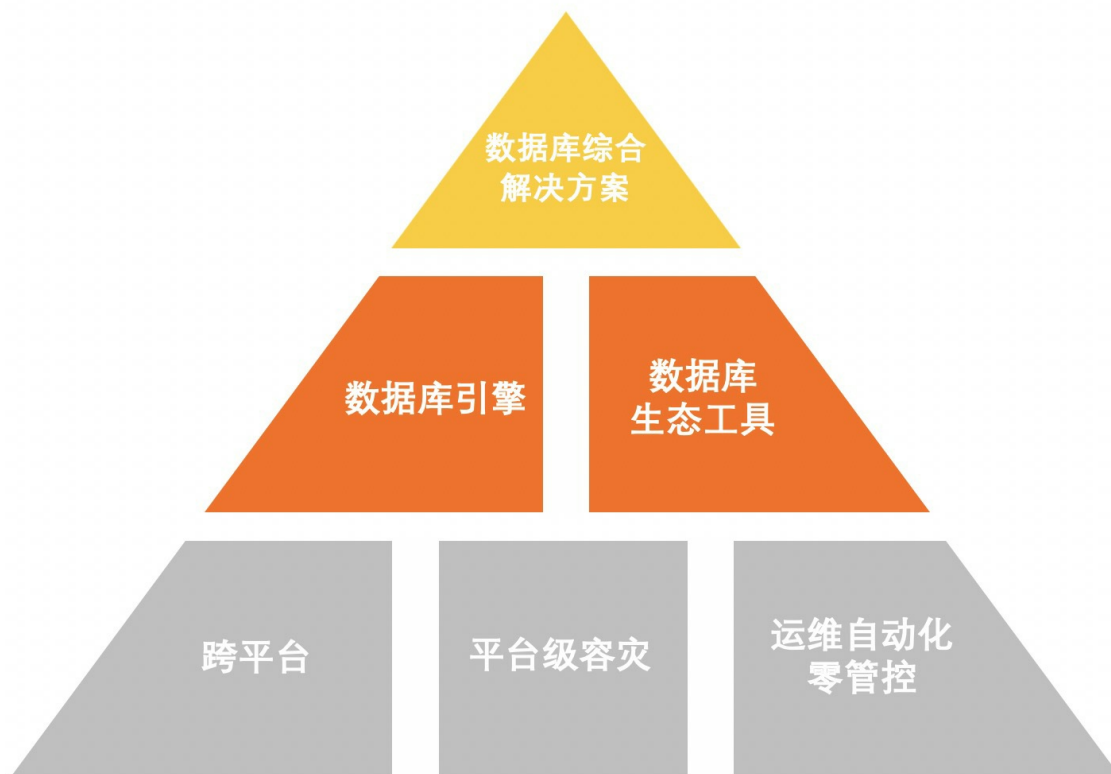
专有云敏捷版数据库场景DBStack（下文简称DBStack）是阿里云企业级交易、分析、传输、治理于一体的数据库管理平台，能够帮助企业构建稳定、安全、经济的全场景数据库解决方案。

平台提供的产品包括AnalyticDB PostgreSQL、PolarDB O引擎、PolarDB-X、Lindorm、GDB等自研数据库，与MySQL、Redis等开源数据库，以及DTS、DMS、ADAM、DAS、DBS等生态工具。

### 2.2. 产品优势

本文档主要介绍了DBStack的产品优势。

DBStack定位于一站式的数据库和数据仓库云平台。其优势包括如下三个方面：



- 综合性、一站式的数据库解决方案。
- 丰富的数据库引擎和多场景数据库生态工具。
- 跨平台、平台级容灾、运维自动化和零管控。

#### 综合性、一站式数据库解决方案

基于DBStack支持的丰富的引擎和生态工具系统，您可以方便地实现与获得一整套综合性解决方案，例如，异地多活、多平台数据灾备同步、在线离线数据同步等解决方案。

#### 丰富的数据库引擎和多场景数据库生态工具

DBStack提供了丰富的数据库引擎和数据库生态工具，可以满足几乎所有场景的数据库需求。



- 数据库引擎：数据库引擎涵盖在线分析型数据库，以及针对Oracle高度兼容的数据库，如PolarDB O引擎（高度兼容Oracle语法的阿里云自研数据库）、AnalyticDB PostgreSQL（阿里云自研的分析型数据库，100%兼容Greenplum）、基于云原生一体化架构设计的分布式数据库PolarDB-X（阿里云自研的云原生分布式数据库，可支撑千万级并发规模及百PB级海量存储）、RDS MySQL（内核采用企业级内核AliSQL，它是基于开源MySQL的独立分支）、云原生多模数据库Lindorm（支持宽表、时序、文本、对象、流、空间等多种数据的统一访问和融合处理的阿里云自研数据库）、图数据库GDB（支持Property Graph图模型、用于处理高度连接数据查询与存储的阿里云自研数据库）。
- 数据库生态工具：数据库生态工具包括数据管理DMS（企业级的数据库研发流程管理、数据管理等）、数据传输DTS（支持多种数据库间的数据传输）、Oracle兼容性分析工具ADAM、数据备份工具DBS和数据库自治服务DAS。

## 跨平台、平台级容灾、运维自动化和零管控

DBStack包含跨平台特性、平台级容灾特性以及运维自动化和零管控特性。

- 跨平台：DBStack能够运行于不同的IaaS平台，支持多云异构环境，支持多种物理服务器以及虚拟服务器。
- 平台级容灾：DBStack整体上具备容灾能力，不同服务对应不同的容灾级别，最高支持两地三中心容灾，满足金融级的行业监管要求。并且在平台中提供容灾演练和快速恢复特性，客户可根据需要量身定制容灾演练计划，以此促进整个技术体系的升级。

 说明 当前支持同城双机房级别容灾。

- 运维自动化和零管控：DBStack能够实现在管控服务器上部署业务实例，管控服务仅运行于前3台中控机上，并且资源占用消耗小于20%的服务器资源。DBStack继承了阿里云十年以上的数据库管理经验，针对数据库的运维需求提供了丰富的自动化功能，包括备份和恢复服务、监控和告警服务、安全服务、审计服务、数据库主机服务器自动化管理服务、数据库任务管理服务、数据库资源调度服务、高可用服务、数据库内核升级等。

## 2.3. 产品架构

DBStack的整体产品结构分为三大模块：用户接口模块、引擎服务模块和平台管理服务模块。



- 用户接口模块：从用户视角，DBStack提供了Web控制台、OpenAPI以及SDK三种用户接口。用户可通过任意接口管理DBStack软件。

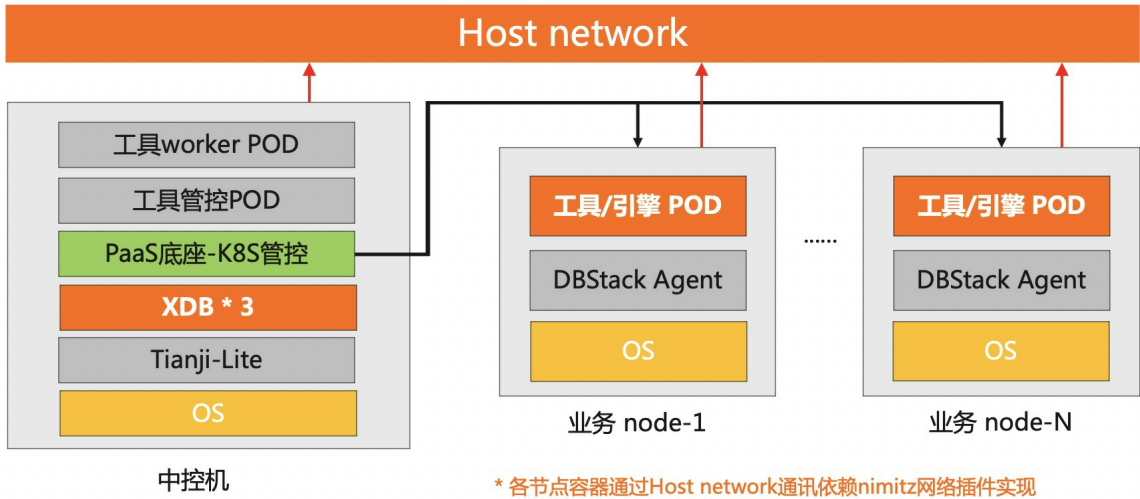
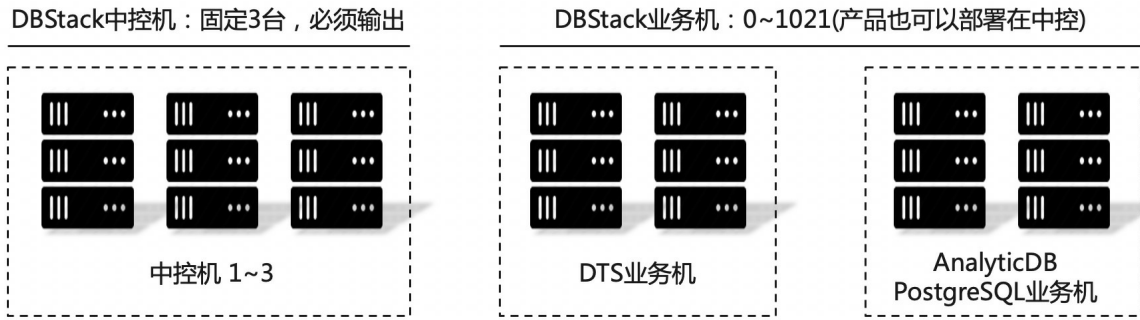
- 引擎服务模块：DBStack提供了关系型数据库服务、非关系型数据库服务、数据仓库服务以及数据库生态工具的服务。
  - 关系型数据库服务包括PolarDB O引擎（阿里云自研的高度兼容Oracle的关系型数据库）、PolarDB-X（阿里云自研的基于MySQL的分布式数据库服务）、RDS MySQL（内核采用企业级内核AliSQL，它是基于开源MySQL的独立分支）。
  - 非关系型数据库服务包括Lindorm（阿里云自研的多模数据库）、Redis（包括开源Redis内核，以及阿里云自研的企业级缓存内核Tair）、MongoDB（MongoDB官方内核版本）、GDB（阿里云自研的图数据库）。
- 说明 当前DBStack版本非关系型数据库服务仅支持Lindorm、Redis、GDB。
- 数据仓库服务包括AnalyticDB PostgreSQL（阿里云自研的云原生数据仓库，100%兼容Greenplum）、AnalyticDB MySQL（阿里云自研的云原生数据仓库，高度兼容MySQL）、DLA（云原生数据湖分析）。
- 说明 当前DBStack版本仅支持AnalyticDB PostgreSQL。
- 生态工具服务包括DTS（数据传输服务，如数据迁移、数据同步、数据订阅、异地多活等）、DMS（数据管理服务）、DBS（数据库备份服务）、DAS（数据库自治服务）、ADAM（Oracle兼容性评估服务）。
- 平台管理服务模块：包含云数据库管理的所有功能以及高级管理功能。
  - 云数据库管理功能包含数据库实例管理、备份恢复管理、安全管理、日志管理、监控高级管理，以及高可用管理；
  - 高级管理功能包括数据库主机管理、系统管理、运维任务管理、资源管理（如容量管理、实例资源迁移等能力）、容灾管理（实现平台级容灾能力），以及DBStack软件的升级管理等。

DBStack通过以上三大模块，为用户提供了平台级的数据库能力，可以支持各类虚拟化环境和硬件资源。

## 2.4. 部署架构

本章节介绍了DBStack的部署架构。

从用户整体部署视角来看，DBStack位于数据库层面，上行接入网络设备，下行接入数据离线存储设备。DBStack自身主要承担数据库平台和数据仓库平台的角色。如下图所示：



用户需要提供以下类型的设备：

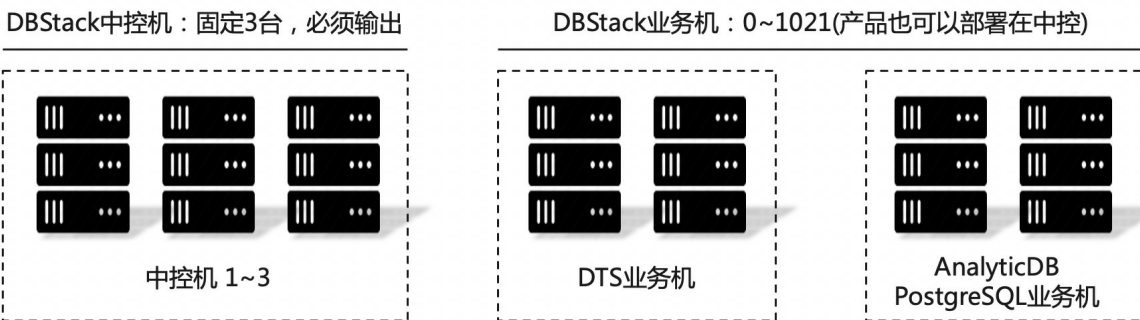
- 网络设备：其IP地址即可作为DBStack的主机IP地址，也是DBStack内部的数据库实例使用的IP地址。
- 登录设备：堡垒机。
- 安全设备：用于保存私有密钥。DBStack会从用户的密钥设备中读取密钥加解密。
- 告警设备：通过用户的告警设备发送告警信息。

此外，DBStack支持挂载外部存储设备，用于存储数据库日志文件、监控数据文件等。这些文件的保存周期由外部设备的空间大小决定。

## 2.5. 产品形态

本文介绍了DBStack的服务器构成。

整个集群中，服务器整体上分为两大角色：中控机和业务机。



- 中控机：整个集群中的管理软件都部署在中控机上。根据中控机的部署方式，可分为中控机混部和中控机

独占两种方式。

- 中控机混部：当中控机的配置高，资源充裕的时候（比如中控机64核或更高配置），管控服务只占用一小部分（如20%）的资源，此时可以将数据库引擎（如AnalyticDB PostgreSQL）或者工具（如DTS）也部署在中控机上，充分利用中控机的资源。
- 中控机独占：中控机上只部署基础服务和管控服务，不部署数据库引擎（如AnalyticDB PostgreSQL）或者工具（如DTS）。
- 业务机：每台业务机所有的资源都用于客户业务，且单台服务器只能安装一种数据库引擎或者仅部署一种数据库工具（当前仅支持DTS）。

② 说明 当前DBStack中支持的数据库引擎和工具服务产品的部署位置要求如下：

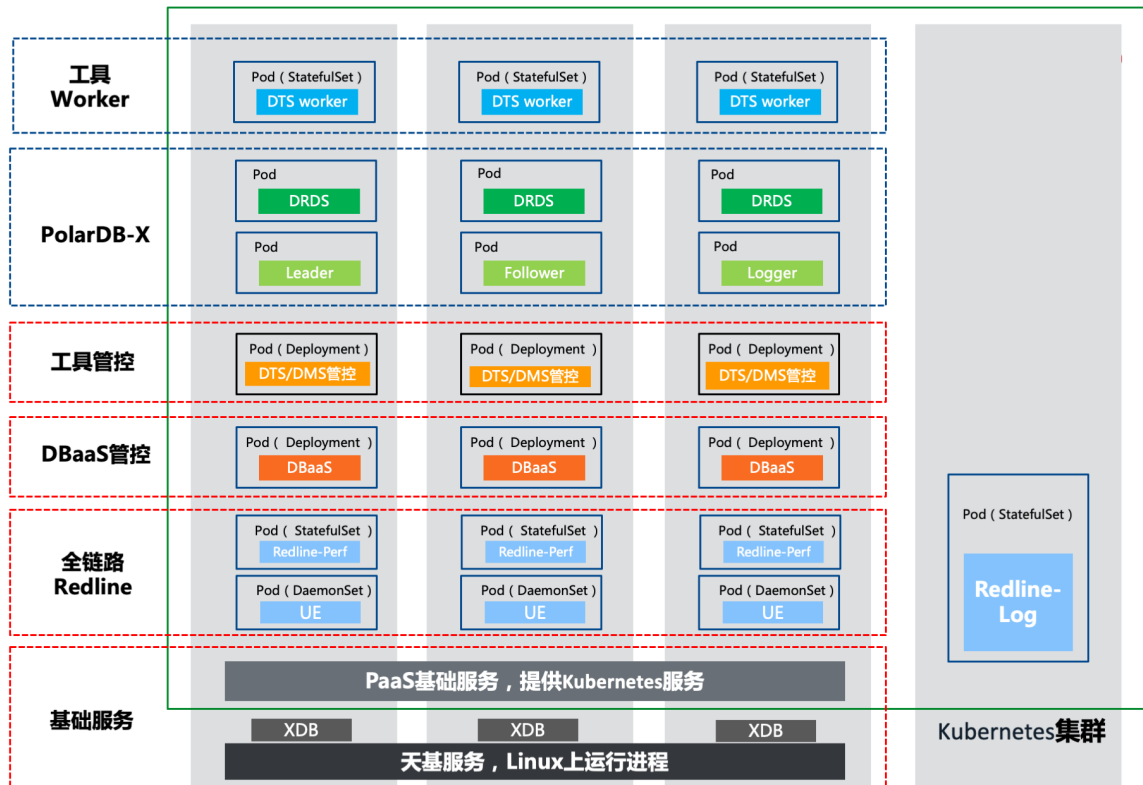
- 若需要使用AnalyticDB PostgreSQL、PolarDB-X、Redis、GDB、RDS MySQL，可将其部署在中控机或业务机上。
- 若需要使用数据库代理MaxScale，仅支持部署到业务机上。
- 若需要使用DMS和ADAM工具，工具的服务仅支持部署在中控机上。
- 若需要使用DTS工具，DTS基础服务仅支持部署在中控机上，DTS扩展服务可部署在中控机或者业务机上。
- 若需要使用Lindorm（用于备份存储），仅支持部署到业务机上。

下文将对中控机和业务机进行详细介绍。

## 中控机

中控机，顾名思义就是“中央管控机器”，整套集群的所有管理系统都部署在中控机上。如果中控机服务器资源足够大，中控机上面也可以分配用户需要的数据库引擎和数据库工具（主要是DTS服务）进程。因此中控机除了分配部分资源给管理系统外，本身也是提供业务服务能力的特殊“业务机”。

下图以中控机上分配了PolarDB-X数据库引擎实例和DTS业务Worker进程，以及针对数据库引擎开启了SQL审计服务的最复杂场景为例，详细介绍了中控机内部结构。



### 管控与引擎混部 \* 3台

### 审计链路 1台

如上图所示，3台中控机内部构造包括管理进程（基础服务、全链路Redline服务、DBaaS管控服务、工具管控服务）、SQL审计服务以及数据库实例和DTS Worker。下面重点介绍管理进程：

- 基础服务：包括天基服务、PaaS基础服务、XDB数据库引擎实例。主要职能是最基础的管理职能，如包括Kubernetes的服务、鉴权、基础元数据存储等。
- 全链路Redline服务：如果要使用数据库引擎，全链路Redline服务必须部署，其职能主要是监报告警和日志存储。
  - 监报告警：主要包括主机服务器上的性能监控和数据库引擎实例的性能监控，指标丰富，同时支持多种维度和多种渠道的告警。
  - 监控日志：包括高可用切换日志、错误日志、系统日志、SQL审计日志。若要开启SQL审计服务，需要专门额外准备至少一台机器。
- DBaaS管控服务：如果要使用数据库引擎，DBaaS管控服务必须部署，且所有的数据库引擎共用一套DBaaS管控服务。DBaaS管控服务用于管理数据库的大部分管控任务，例如创建实例、升级实例等；同时提供一些基础数据库核心服务，例如高可用服务。
- 工具管控服务：如果要使用数据库工具引擎（例如：DTS引擎），则工具管控服务必须部署。工具管控服务主要提供工具的管控任务，例如创建DTS任务、监控DTS的任务状态等。

上述的管理服务进程，部分是单点服务，这类进程的异常不影响整体系统可用性。主要管理进程都是高可用服务（例如：DBaaS管控服务）。所有的数据库管理服务进程，全部不处于Kubernetes Pod中，而是通过统一架构统一管理。

## 业务机

业务机，顾名思义就是“运行某种业务的机器”，仅支持部署某种数据库引擎或者数据库工具进程，以及相关管理系统的代理，对具体的业务提供服务。

每台业务机会被某种数据库引擎，或者某种数据库工具进程（当前仅支持DTS）独占，不支持混部在一台业务机上，因此业务机都是独占模式。业务机上运行不同的数据库引擎，即成为某种引擎业务机，例如DTS业务机、AnalyticDB PostgreSQL业务机等。

相比中控机，业务机的部署逻辑比较简单，此处不与赘述。

## 服务器数量限制

中控机和业务机的服务器数量有各自的限制。

中控机只能包含3台服务器，业务机最多包含1021台服务器。整个集群最少输出3台服务器，即仅输出3台中控机；最大输出1024台服务器，即3台中控机和1021台业务机。

对于业务机，部分数据库引擎和产品有特殊要求：DTS要求业务机数量必须是3的倍数（例如3台、6台、9台等）、MaxScale要求业务机数量为2的倍数（例如2台、4台、6台）。

## 2.6. 功能大图

本文档主要介绍了DBStack支持的功能特性。

DBStack当前支持的功能如下：

- 生态工具服务，包括数据管理（DMS）、数据传输（DTS）、Oracle兼容性评估（ADAM）、数据库自治服务（DAS）、数据库备份（DBS）。
- 数据库引擎，包括阿里自研的智能化的企业级RDS MySQL、100%兼容MySQL的分布式数据库PolarDB-X、PB级在线分析型数据库引擎AnalyticDB PostgreSQL、高度兼容Oracle的自研数据库PolarDB O引擎、云原生多模数据库Lindorm、图数据库GDB和高性能缓存数据库Redis。
- 整个平台层面提供高可用服务、备份服务、恢复服务、监控告警服务、数据库审计、容灾服务和数据库主机和容量管理服务。



## 3. 数据库产品介绍

### 3.1. AnalyticDB PostgreSQL

#### 3.1.1. 产品概述

AnalyticDB PostgreSQL版是一种大规模并行处理（MPP）数据仓库服务，可提供海量数据在线分析服务。

AnalyticDB PostgreSQL版基于开源项目Greenplum构建，由阿里云深度扩展，兼容ANSI SQL 2003，兼容PostgreSQL/Oracle数据库生态，支持行存储和列存储模式。既提供高性能离线数据处理，也支持高并发在线分析查询，是各行业有竞争力的PB级实时数据仓库方案。

#### 3.1.2. 产品优势

本章节介绍了AnalyticDB PostgreSQL版的产品核心优势。

AnalyticDB PostgreSQL版是一种大规模并行处理（MPP）数据仓库服务，可提供海量数据在线分析服务。AnalyticDB PostgreSQL版兼容ANSI SQL 2003，兼容PostgreSQL/Oracle数据库生态，支持行存储和列存储模式。既提供高性能离线数据处理，也支持高并发在线分析查询，是各行业有竞争力的PB级实时数据仓库方案。

AnalyticDB PostgreSQL核心优势如下所示：

- 实时分析  
MPP水平扩展架构，PB级数据查询秒级响应；向量化计算，及列存储智能索引，性能领先传统数据库引擎；Cascade架构SQL优化器，复杂分析语句免调优。
- 稳定可靠  
支持分布式ACID数据一致性，实现跨节点事务一致，所有数据双节点强同步冗余。分布式部署，全透明化监控，切换，恢复，提高重要数据基础设施保障。
- 简单易用  
丰富的SQL语法及函数支持，支持多数Oracle函数，支持存储过程，UDF，支持分布式事务和隔离级别。业界主流BI软件和ETL工具可直接联机使用。
- 性能卓越  
支持行存储和列存储，支持多种索引机制。向量引擎分析计算性能卓越，Cascade SQL优化器实现复杂查询免调优。
- 灵活扩展  
支持扩展计算单元，CPU、内存、存储空间，从而提高OLAP性能。  
弹性扩容后，数据重分布过程中支持数据增、删、改、查。
- 资源隔离  
通过划分AnalyticDB实例实现租户间资源隔离。
- 权限管理  
支持通过控制台，实现的租户统一管理，实现租户资源的动态配置和管理，资源隔离，资源使用统计等功能。

#### 3.1.3. 产品架构

本文为您介绍云原生数据仓库AnalyticDB PostgreSQL版的产品架构。

### 集群物理架构

云原生数据仓库AnalyticDB PostgreSQL版集群物理架构如下：

图 1. 集群系统物理架构图



一个云原生数据仓库AnalyticDB PostgreSQL版的物理集群内，通过管控系统支持创建多个云原生数据仓库AnalyticDB PostgreSQL版数据库实例。每个数据库实例由两类组件组成：协调节点（Master）节点和多个计算节点（Segment）。

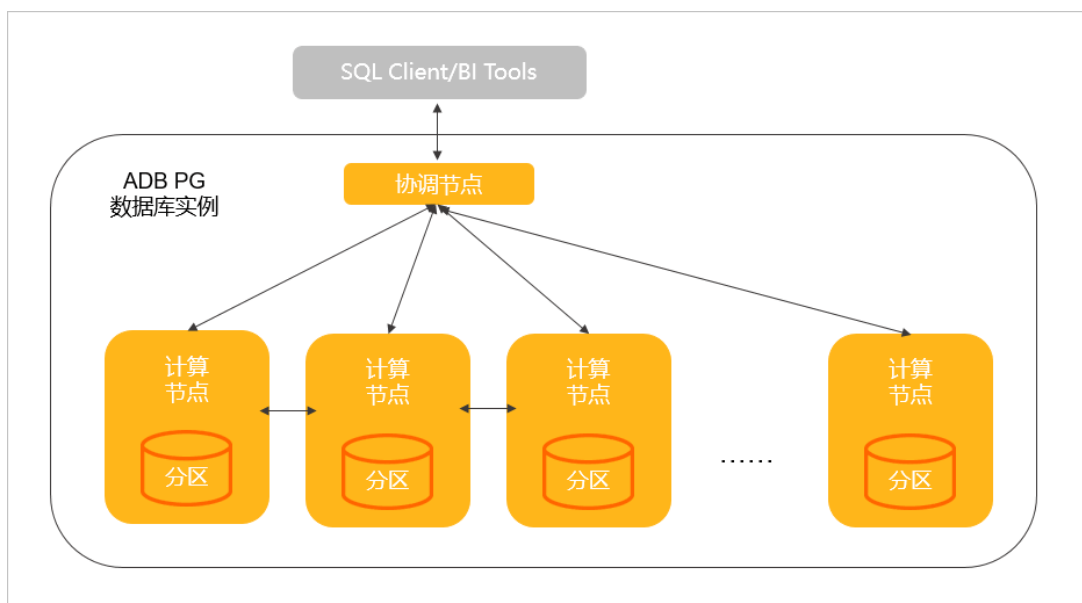
- 协调节点（Master）是云原生数据仓库AnalyticDB PostgreSQL版数据库实例的应用接入点，它负责接收客户端的连接和SQL查询请求，并将工作分配给计算节点（Segment）去并行执行。集群同时在另一台物理机上部署协调节点（Master）的备节点（Standby），采用Replication方式同步主备节点以支持故障转移。备节点（Standby）不接受外部链接。
- 每个计算节点（Segment）都是独立的云原生数据仓库AnalyticDB PostgreSQL版实例，数据按HASH或者Random方式分布存储在每个计算节点（Segment）上，同时并行地执行分析计算。每个计算节点（Segment）均采用主备（Primary/Mirror）架构，支持故障下自动切换。

### 实例逻辑架构

一个云原生数据仓库AnalyticDB PostgreSQL版集群支持创建多个云原生数据仓库AnalyticDB PostgreSQL版的数据库实例，每个数据库实例的逻辑架构如下。

图 2. 实例逻辑架构图

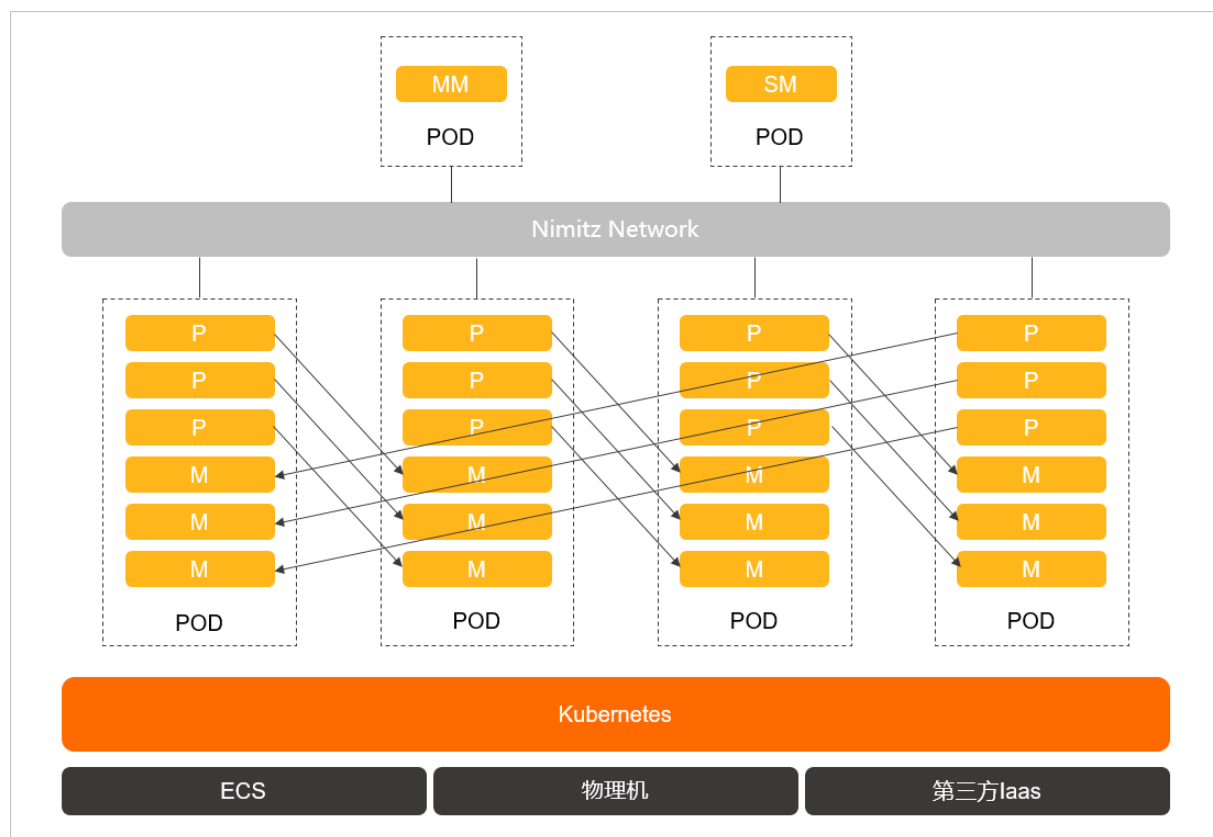




表数据可以按指定列的哈希 (HASH) 分布、随机 (RANDOMLY) 分布、复制 (REPLICATED) 分布在各个计算节点上，每个计算节点均为主备 (Primary/Minor) 架构，保证数据双副本存储，节点间支持高性能网络通信。协调节点接收应用请求后，负责SQL语句的解析优化，并生成分布式执行计划，下发给计算节点，执行计划在计算节点间全并行执行 (MassiveParallel Process)。

### 3.1.4. 部署架构

本章节介绍了AnalyticDB PostgreSQL在DBStack中的部署架构。



### 部署结构

- 统一部署于Kubernetes的POD中。
- 整体结构分为协调节点（Master）和计算组（Segment组）。
- 协调节点和计算组混合部署。

### 协调器

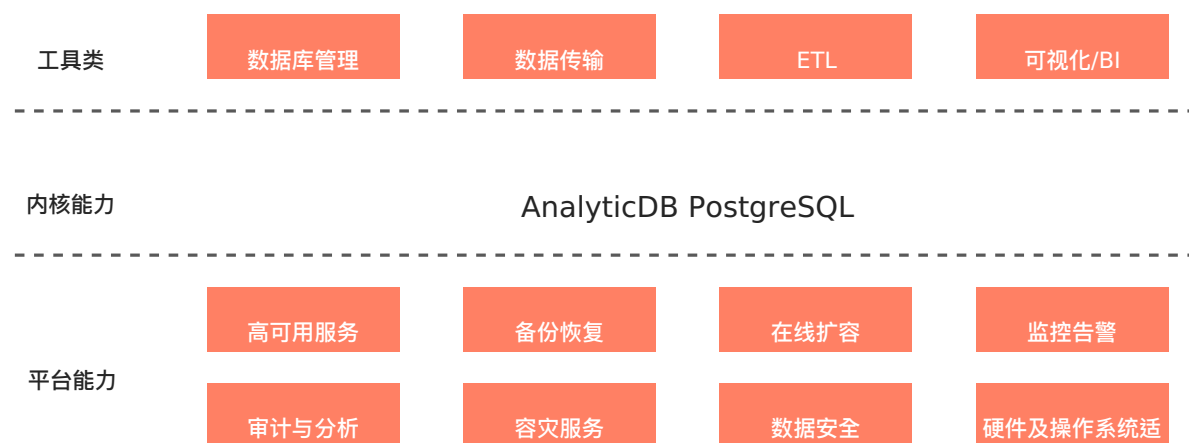
- 由主备两个Master组成（Main Master和Standby Master），提供高可用服务。
- Main Master和Standby Master部署上物理打散。

### 计算组

- 计算组对应一个Kubernetes的POD，计算组依主机打散部署。
- 一个计算组由多个Segment组成。
- Segment包含Primary和Mirror两种角色，两种角色位于不同的POD中。

## 3.1.5. 功能大图

DBStack当前版本中AnalyticDB PostgreSQL支持的功能概览如下图所示。



### 工具类

- 数据库管理：DMS企业版，中控机默认携带。
- 数据传输：支持Teradata、DB2、Oracle、PG/GP等数据源迁移。
- 第三方工具：丰富的开源或者商业第三方数据集成、迁移、ETL、BI等工具。

### 内核能力

- 离在线数据分析：离线跑批、在线实时分析（Kafka数据链路、实时物化视图）。
- 企业级特性：ACID、资源隔离、外部表、UDF、向量分析、列存储、时空引擎。

### 平台能力

- 高可用服务：秒级高可用。
- 备份&恢复服务：库表级备份服务，基于时间点快速恢复能力。
- 监控告警：全链路监控告警服务，涵盖主机、实例。
- 数据库审计与分析：自研的日志审计服务。
- 容灾服务：整个平台支持容灾服务，且支持快速恢复。
- 弹性扩容：弹性scale-out扩展。
- 硬件及操作系统适配：x86、ARM、OS认证与适配。

- 数据安全：鉴权认证。

## 3.2. RDS MySQL

### 3.2.1. 产品概述

阿里云关系型数据库RDS (Relational Database Service) 是一种稳定可靠、可弹性伸缩的在线数据库服务。基于阿里云分布式文件系统和高性能存储，提供了容灾、备份、恢复、监控、迁移等方面的全套解决方案。

RDS MySQL基于阿里巴巴的MySQL源码分支，经过双十一高并发、大数据量的考验，拥有优良的性能。RDS MySQL支持实例管理、账号管理、数据库管理、备份恢复、透明数据加密以及数据迁移等基本功能。除此之外还提供如下高级功能：

- **只读实例**：在对数据库有大量读请求和少量写请求时，单个实例可能无法承受读取压力，为了实现读取能力的弹性扩展，减少单个实例的压力，RDS MySQL的实例支持只读实例，利用只读实例满足大量的数据库读取需求，以此增加应用的吞吐量。
- **读写分离**：读写分离功能是在只读实例的基础上，额外提供了一个读写分离地址，联动主实例及其所有只读实例，创建自动的读写分离链路。应用程序只需连接读写分离地址进行数据读取及写入操作，读写分离程序会自动将写入请求发往主实例，而将读取请求按照权重发往各个只读实例。用户只需通过添加只读实例的个数，即可不断扩展系统的处理能力，应用程序上无需做任何修改。

### 3.2.2. 产品优势

本章节介绍了RDS MySQL具有的产品优势。

#### 易于使用

云数据库RDS MySQL拥有按需升级、透明兼容等优点。

- **按需升级**：随着数据库压力和数据存储量的变化，您可以灵活调整实例规格，且升级期间RDS MySQL不会中断数据链路服务。
- **透明兼容**：RDS MySQL与原生数据库引擎的使用方法一致，您无需二次学习，上手即用。另外RDS MySQL兼容您现有的程序和工具。使用通用的数据导入导出工具即可将数据迁移到RDS MySQL，迁移过程中的人力开销非常低。

#### 高性能

云数据库RDS MySQL通过参数优化等来实现高性能。

- **参数优化**：所有RDS MySQL的参数都是经过多年的生产实践优化而得，在RDS MySQL实例的生命周期内，我们持续对其进行优化，确保RDS MySQL一直基于最佳实践在运行。

#### 高安全性

云数据库RDS通过访问控制策略、系统安全、SSL加密实现高安全性。

- **访问控制策略**：每个RDS MySQL账号只能查询、操作被授权的数据库，对数据库的操作权限也可按照读写（DDL+DML）、只读、仅DDL或仅DML细分。
- **系统安全**：RDS处于多层防火墙的保护之下，可以有效抵抗各种恶意攻击，保证数据的安全；RDS服务器不允许直接登录，只开放特定的数据库服务所需要的端口；RDS服务器不允许主动向外发起连接，只能接受被动访问。
- **SSL加密**：支持启用SSL (Secure Sockets Layer) 加密，提高链路安全性。SSL在传输层对网络连接进行加密，提升通信数据的安全性和完整性。

#### 高可靠性

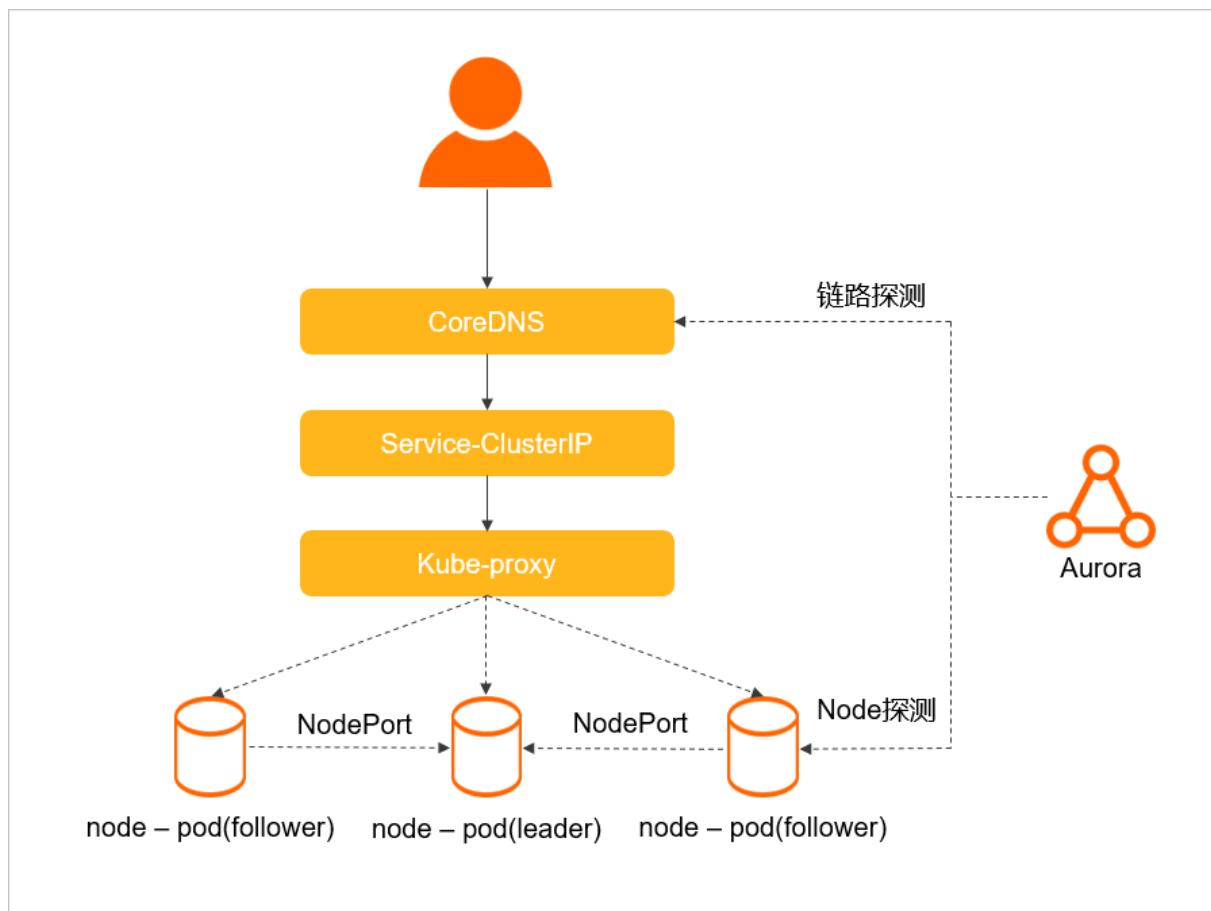
云数据库RDS通过双机热备、数据备份和数据恢复实现高可靠性。

- **双机热备**：RDS采用热备架构，服务器出现故障后服务秒级完成切换，整个切换过程对应用透明。

- 数据备份：RDS提供自动备份的机制，您可以自行选择备份周期，也可以根据自身业务特点随时发起临时备份。
- 数据恢复：支持按备份集和指定时间点来创建克隆实例恢复数据，数据验证无误后即可将数据迁回RDS主实例，从而完成数据回溯。

### 3.2.3. 产品架构

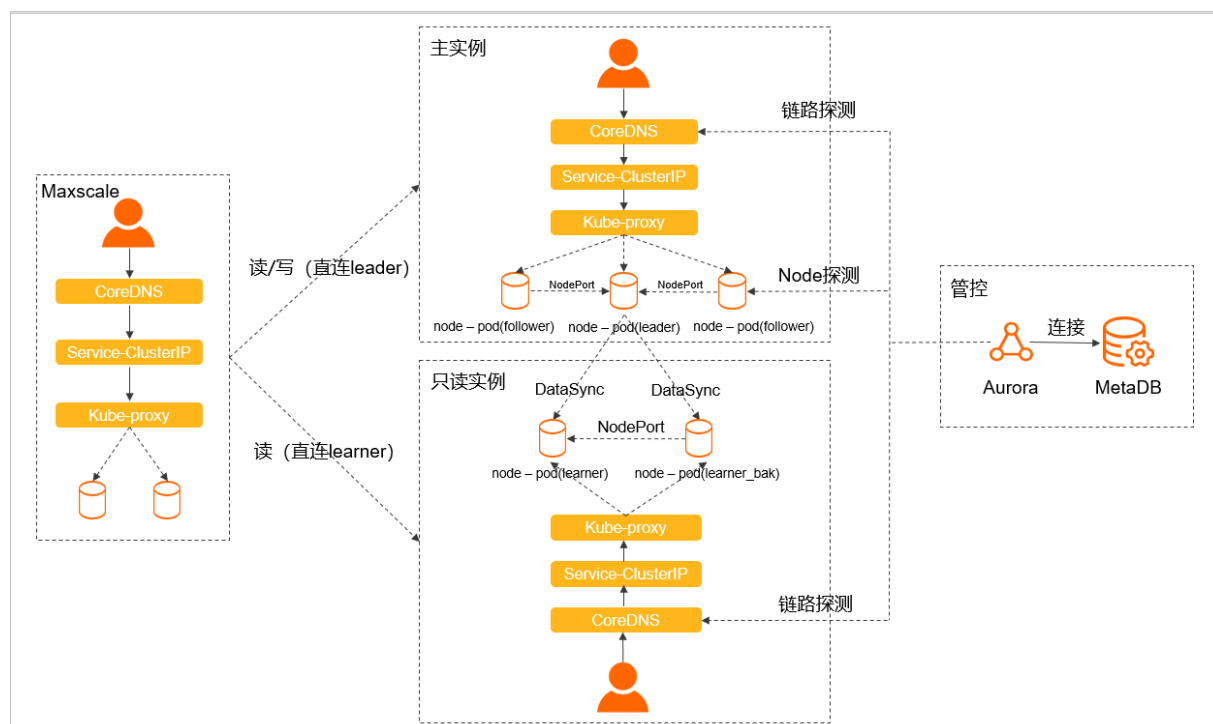
本章节介绍了RDS MySQL在DBStack中的实例架构。



在DBStack中，RDS实例数据链路为Kubernetes链路，在创建实例（ReplicaSet）时会创建一个service，该service的流量策略为cluster，实例pod（replica）会绑定在Service中。Service的clusterIP会绑定到CoreDNS中，DNS地址提供用户访问。与其他场景相同，实例级别容灾都是由Aurora组件实现（实例切换、节点摘除）。

用户通过DNS对实例进行访问，CoreDNS将地址解析为对应的ClusterIP，ClusterIP为虚IP，请求会通过kubernetes-proxy负载均衡到后端绑定的pod。由于RDS为单主实例，请求路由到follower节点后会通过nodePort转回到主实例Node中的pod，从而实现所有请求都到主实例Pod中。

除了主实例外，RDS MySQL还提供只读实例与Maxscale提供的读写分离功能。只读实例、主实例、Maxscale之间的拓扑关系如下图。



## 3.3. PolarDB O引擎

### 3.3.1. 产品概述

PolarDB O引擎是阿里云自研的云原生关系型数据库，高度兼容Oracle语法，最大限度降低业务逻辑代码的变更，保护历史投资的同时降低Oracle迁移风险、缩短迁移周期。

PolarDB O引擎基于存储计算分离架构，一写多读，资源池化高效管理，实现分钟级弹性变配，最大可扩展至10个只读节点。

### 3.3.2. 产品优势

本章节介绍了PolarDB O引擎数据库的优势和特点。

#### 自主可控

- PolarDB O引擎由阿里巴巴自主研发，安全可控，符合国家相关行业认证资质要求。
- 支持自主可控芯片和操作系统。

#### Oracle语法兼容

- PolarDB O引擎高度兼容Oracle语法，支持Oracle所有数据类型，支持PL/SQL存储过程，支持SQL常见语法，SQL兼容度大于95%。
- PolarDB O引擎提供Oracle特性兼容，支持Oracle分区表、全局索引、DBLINK、并行、AWR等常见Oracle特性。

#### 智能读写分离

- 读写分离：智能分析SQL，并根据SQL自动做读写分离。
- 负载均衡：自动在多节点做负载均衡并支持自定义业务负载。

#### RPO=0

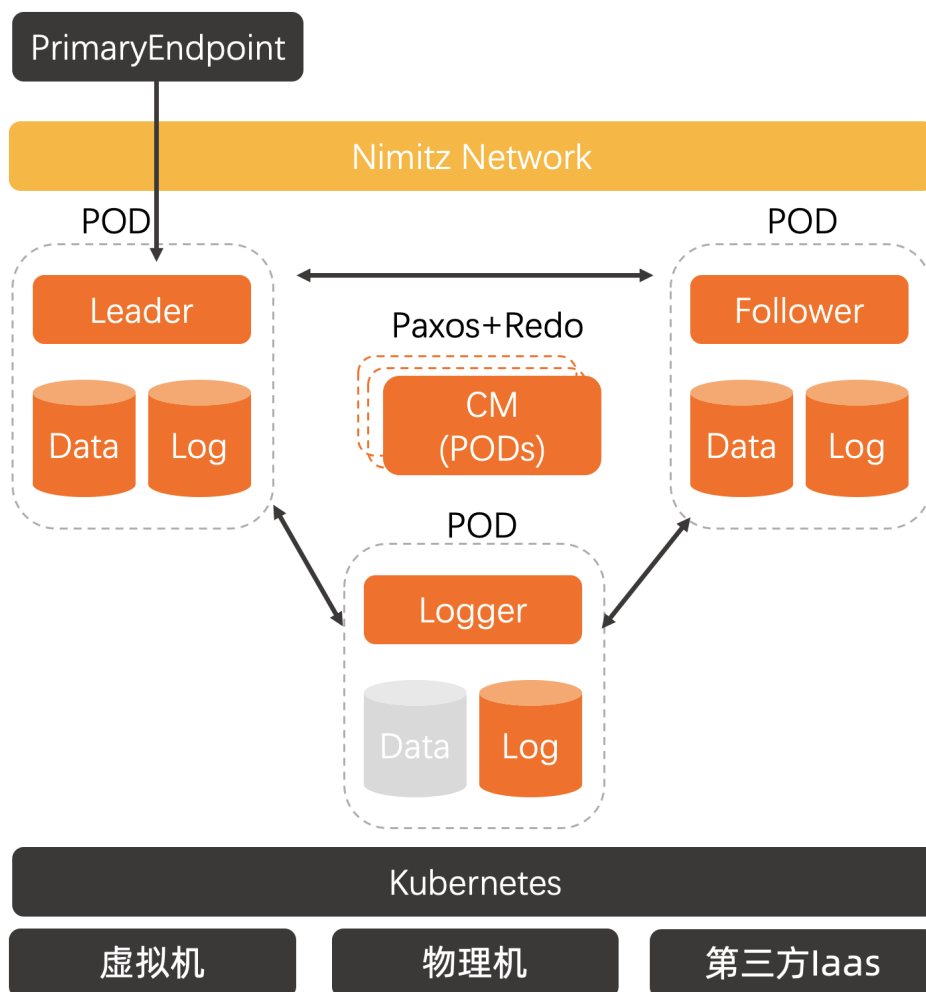
- PolarDB O引擎提供基于Paxos一致性协议的RPO=0的高可用高可靠能力。
- 最大可扩展至10个只读节点。

### 内置时空数据引擎

PolarDB O引擎内置时空数据引擎Ganos，满足时空数据和向量数据的处理需求。

## 3.3.3. 产品架构

本章节介绍了PolarDB O引擎在DBStack中的产品架构。

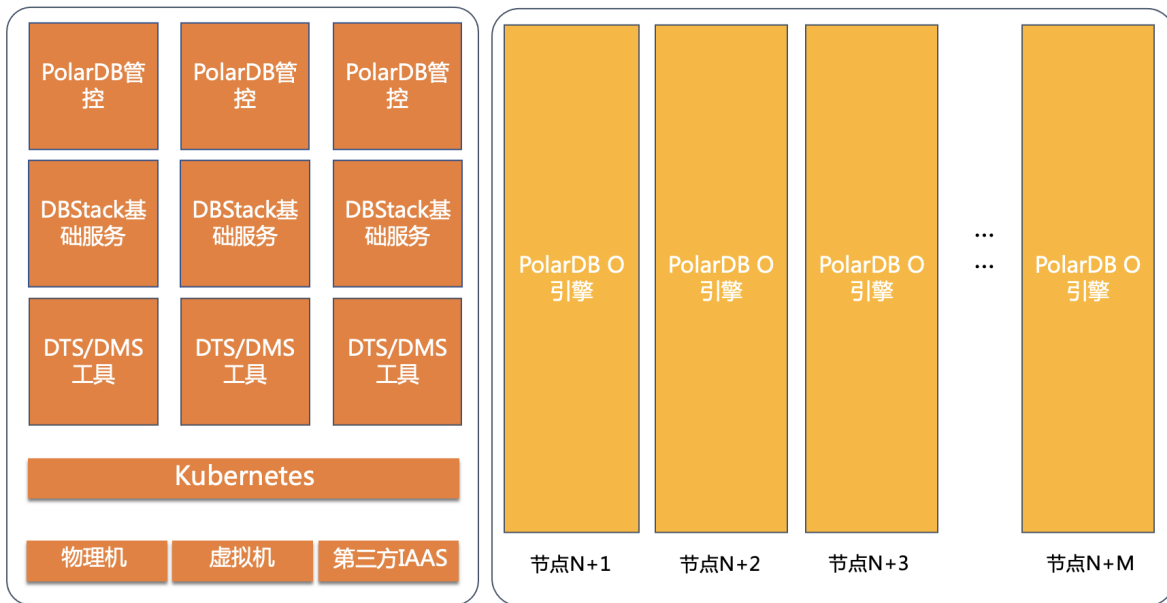


在DBStack中，PolarDB O引擎为三节点架构：

- 统一部署在Kubernetes POD中。
- 一个PolarDB O引擎集群实例包括：Leader、Follower、Logger、ClusterManager (CM)。
- Leader、Follower、Logger三者物理机维度隔离。
- Leader、Follower、Logger之间使用Paxos和Redo来维护一致性和数据同步。
- 一个PolarDB O引擎集群实例保留两份数据和三份日志。
- ClusterManager (CM) 用于实时监控集群实例的运行状态，在必要的时候实时发起运维操作。
- ClusterManager (CM) 部署采用三节点，CM内部使用raft来维护一致性和进行Leader选主。
- 应用通过PrimaryEndpoint连接到Leader节点进行读写操作。

### 3.3.4. 部署架构

本章节介绍了PolarDB O引擎在DBStack中的部署架构。

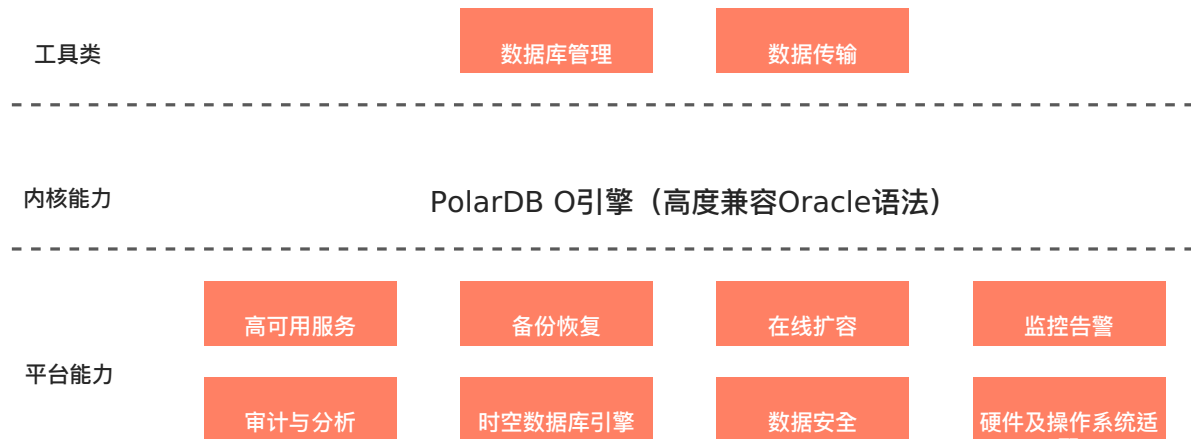


#### 部署结构

- PolarDB管控、工具和基础服务共用一个Kubernetes集群，支持物理机、虚拟机及第三方IAAS平台进行部署。
- 当前PolarDB O引擎在DBStack中为本地盘部署的形态。后续将实现共享存储形态。

### 3.3.5. 功能大图

DBStack当前版本中PolarDB O引擎支持的功能概览如下图所示。



#### 说明

上图中展示的功能与实际版本可能有所不同，请以实际版本为准。

#### 工具类

- 数据库管理：DMS企业版，中控机默认携带。

- 数据传输：支持Oracle数据全量及增量迁移，支持反向数据同步。

## 内核能力

- 云原生：存储计算分离。
- 高度Oracle语法兼容：SQL语法高度兼容、支持Oracle所有数据类型，支持分区表、分区索引，支持PL/SQL存储过程，支持Oracle原生OCI接口，支持AWR报告。
- 企业级特性：资源隔离、高可用、读写分离、时空引擎。

## 平台能力

- 高可用服务：秒级高可用。
- 备份&恢复服务：库表级备份服务，基于时间点快速恢复能力。
- 监控告警：全链路监控告警服务，涵盖主机、实例。
- 数据库审计与分析：自研的日志审计服务。
- 在线扩容：弹性scale-out扩展。
- 硬件及操作系统适配：x86、ARM、OS认证与适配，支持intel与ARM混部。
- 数据安全：鉴权认证，数据加密。

# 3.4. PolarDB-X

## 3.4.1. 产品概述

PolarDB-X是由阿里巴巴自主研发的云原生分布式数据库，基于云原生一体化架构设计，可支撑千万级并发规模及百PB级海量存储。

PolarDB-X专注解决海量数据存储、超高并发吞吐、大表瓶颈以及复杂计算效率等数据库瓶颈问题，历经各届天猫双十一及阿里云各行业客户业务的考验，助力企业加速完成业务数字化转型。

PolarDB-X核心能力采用标准关系型数据库技术实现，配合完善的管控运维及产品化能力，使其具备稳定可靠、高度可扩展、高度兼容MySQL、HTAP混合负载、持续可运维、类传统单机MySQL数据库体验的特点。

PolarDB-X在公共云和专有云环境沉淀打磨多年，历经各届天猫双十一核心交易业务及各行业阿里云客户业务的考验。承载大量用户核心业务，广泛应用于互联网、金融支付、教育、通信、公共事业等多行业。

## 3.4.2. 产品优势

本章节介绍了PolarDB-X所具备的产品优势。

- 高可用

经过阿里云多年双11验证的X-DB（X-Paxos共识协议能力），提供数据强一致，保证节点故障切换时RPO=0。另外支持多样化的部署和容灾能力，比如基于Paxos强同步的同城三机房、三地五中心，另外搭配binlog异步复制的两地三中心、异地灾备、异地多活等。尤其在异地长距离传输上，基于Batching & Pipelining 进行网络优化来提升性能。

- 高兼容

PolarDB-X主要兼容MySQL，包括SQL、函数类型等，技术上引入全局时间授时服务，提供全局一致性的分布式事务能力，通过TSO+2PC提供数据库完整的ACID能力，满足分布式下的Read-Committed/Repeatable-Read的隔离级别。同时在分布式事务的基础上，提供全局二级索引能力，通过事务多写保证索引和主表数据强一致的同时，引入基于代价的CBO优化器实现索引选择。除此以外，在元数据和生态对接层面，PolarDB-X基于Online DDL的技术提供了分布式下元数据的一致性。同时硬件层面，兼容主流国产操作系统和芯片认证，比如麒麟、鲲鹏、海光等。



另外在业界主流的分布式数据库里，分布式下的redolog/binlog等数据库变更日志其实一直被厂商所忽视，从关系数据库的发展历史来看，生态和标准对于市场规模化非常重要，PolarDB-X支持全局binlog能力，全面兼容和拥抱MySQL数据库生态，用户可以将PolarDB-X当做一个MySQL库，采用标准的binlog dump协议获取binlog日志。

- 高扩展

PolarDB-X基于Share-Nothing的架构支持水平扩展，同时支持数据库在线扩缩容能力，在OLTP场景下可支持千万级别的并发、以及PB级别的数据存储规模，同样在OLAP场景下，引入MPP并行查询技术，扩展机器后查询能力可线性提升，满足TPC-H等的复杂报表查询诉求。

- HTAP

随着移动互联网和IoT设备的普及，数据会产生爆炸式的增长趋势，传统的OLTP和OLAP的解决方案是基于简单的读写分离或者ETL模型，将在线库的数据T+1的方式抽取到数据仓库中进行计算，这种方案存在存储成本高、实时性差、链路和维护成本高。PolarDB-X设计中支持OLTP和OLAP的混合负载的能力，可以在一个实例里同时运行TPC-C和TPC-H的benchmark测试，保证AP的查询不影响TP流量的稳定性。核心技术层面，我们也有自己的创新性，比如我们会在计算层精确识别出TP和AP的流量，结合多副本的特性和多副本的一致性读能力，智能将TP和AP路由到不同的副本上，同时在AP链路上默认开启MPP并行查询技术，从而在满足隔离性的基础上，线性提升AP的查询能力。在存储层上，我们也在完善计算下推能力，未来也会提供高性能列存引擎，实现行列混合的HTAP能力。

- 极致弹性

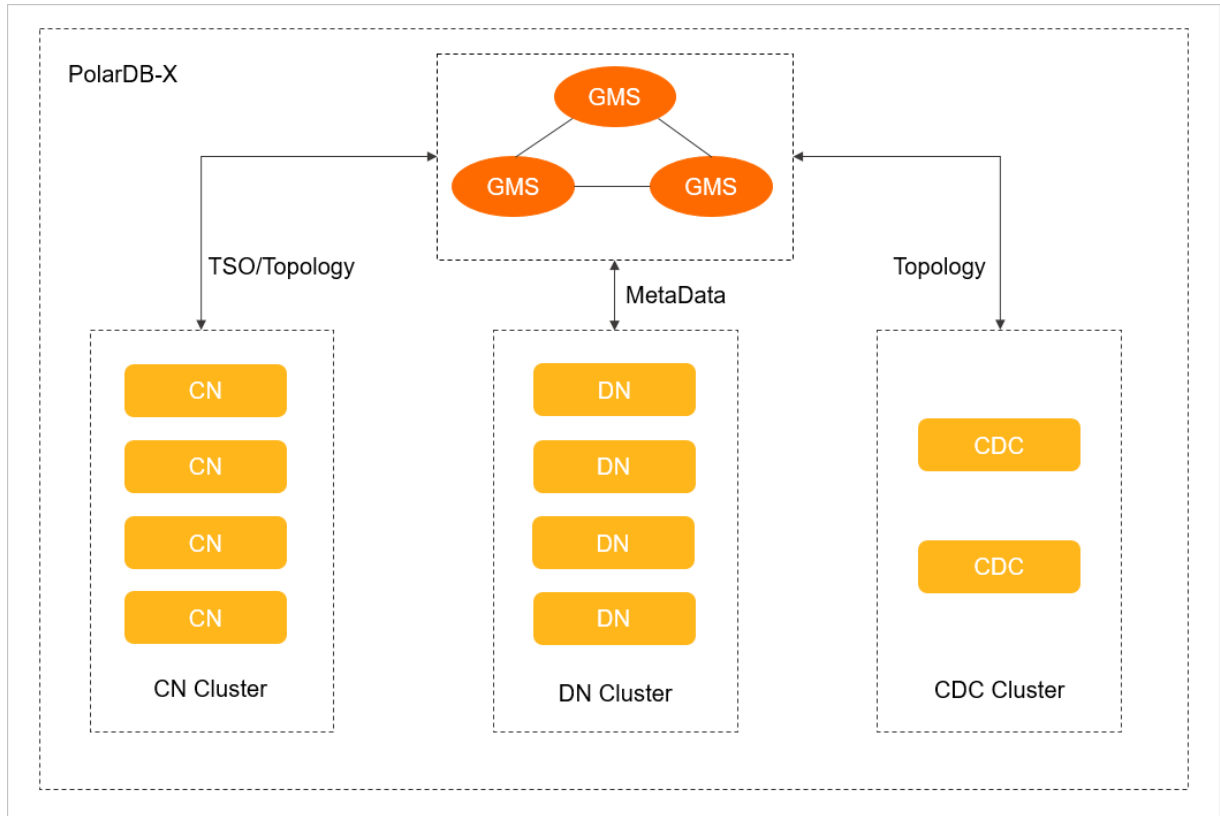
PolarDB-X结合PolarDB云原生的技术，可以基于PolarDB的共享存储+RDMA网络优化能力，提供秒级备份、极速弹性、以及存储按需扩展的能力。基于共享存储的基础上，结合分布式的多点写入能力，可以在不迁移数据的前提下提供秒级弹性的能力，给到用户完全不一样的弹性体验。

- 开放生态

PolarDB-X全面拥抱和坚定MySQL的开源生态，做到代码完全自主可控的同时满足分布式MySQL的兼容性，架构做到简单开放，只要具备一定MySQL背景的同学即可完成持续运维。除此以外，PolarDB-X和阿里云的数据库生态有完整的闭环对接，支持如DTS/DBS/DMS等，可打通阿里云的整个大生态。

### 3.4.3. 产品架构

本章节介绍了PolarDB-X在DBStack中的产品架构。



其中，对4个关键模块的介绍如下：

### 元数据服务 (Global MetaService, GMS)

- 提供全局授时服务(TSO)
- 维护Table/Schema、Statistic等Meta信息
- 维护账号、权限等安全信息

### 计算节点 (Compute Node, CN)

- 基于无状态的SQL引擎提供分布式路由和计算
- 处理分布式事务的2PC协调、全局索引维护等

### 存储节点 (Data Node, DN)

- 基于多数派Paxos共识协议的高可靠存储
- 处理分布式MVCC事务的可见性判断

### 日志节点 (Change Data Capture, CDC)

- 提供兼容MySQL生态的binlog协议和数据格式
- 提供兼容MySQL Replication主从复制的交互

## 3.4.4. 功能特性

本章节介绍了DBStack当前版本中PolarDB-X支持的功能特性。

### 水平拆分 (分库分表)

PolarDB-X具备数据水平拆分的能力，将数据库数据按某种规则分散存储到多个稳定的MySQL数据库上。这些MySQL数据库可分布于多台机器乃至跨机房，对外服务（增删改查）尽可能保证如同单MySQL数据库体验。拆分后，在MySQL上物理存在的数据库称为分库，物理的表称为分表（每个分表数据是完整数据的一部分）。PolarDB-X通过在不同MySQL实例上挪动分库，实现数据库扩容，提升PolarDB-X数据库总体访问量和存储容量。

## 计算扩展性

无论是水平拆分还是垂直拆分，PolarDB-X常常碰到需要对远超单机容量数据进行复杂计算的需求，例如需要执行多表JOIN、多层嵌套子查询、Grouping、Sorting、Aggregation等组合的SQL操作语句。

针对这类在线数据库上复杂SQL的处理，PolarDB-X额外扩展了单机并行处理器（Symmetric Multi-Processingy，简称SMP）和多机并行处理器（DAG）。前者完全集成在PolarDB-X内核中；而对于后者，PolarDB-X构建了一个计算集群，能够在运行时动态获取执行计划并进行分布式计算，通过增加节点提升计算能力。

## 平滑扩容

PolarDB-X扩容通过增加RDS/MySQL实例数，将原有的分库迁移到新的RDS/MySQL实例上，达到扩容的目标。PolarDB-X采用基于存储计算分离的Shared-Nothing架构，最大限度地发挥了云数据库的弹性扩展能力。

## 读写分离

PolarDB-X的读写分离功能是基于RDS/MySQL只读实例所做的一种相对透明读流量切换策略。当PolarDB-X存储资源MySQL主实例的读请求较多、读压力比较大时，您可以通过读写分离功能对读流量进行分流，减轻存储层的读压力。

## 分布式事务

分布式事务通常使用二阶段提交来保证事务的原子性（Atomicity）和一致性（Consistency）。

二阶段事务会将事务分为以下两个阶段：

- 准备（PREPARE）阶段：在PREPARE阶段，数据节点会准备好所有事务提交所需的资源（例如加锁、写日志等）。
- 提交（COMMIT）阶段：在COMMIT阶段，各个数据节点才会真正提交事务。

当提交一个分布式事务时，PolarDB-X 1.0服务器会作为事务管理器的角色，等待所有数据节点（MySQL服务器）PREPARE成功，之后再向各个数据节点发送COMMIT请求。

## 全局二级索引

全局二级索引（Global Secondary Index，GSI）支持按需增加拆分维度，提供全局唯一约束。每个GSI对应一张索引表，使用XA多写保证主表和索引表之间数据强一致。

## HTAP

PolarDB-X由多个节点构成计算、存储内核一体化实例，在共用一份数据的基础上避免了ETL（Extract-Transform-Load）操作，实现了在线高并发OLTP联机事务处理以及OLAP海量数据分析，即HTAP。

## 全局Binlog

PolarDB-X提供了全局Binlog文件来支持消费或订阅数据。PolarDB-X的全局Binlog（又称逻辑Binlog），是以TSO（Timestamp Oracle）为基准，将多个数据节点（Database Node，简称DN）的物理Binlog进行归并和合并，能够保证分布式事务完整性和全局有序性的Binlog文件

# 3.5. Redis

## 3.5.1. 产品概述

云数据库Redis是兼容开源Redis协议标准的数据库服务，基于双机热备架构，可满足高吞吐、低延迟及弹性变配等业务需求。支持String（字符串）、List（链表）、Set（集合）、Sorted Set（有序集合）、Hash（哈希表）、Stream（流数据）等多种数据结构，同时支持Transaction（事务）、Pub/Sub（消息订阅与发布）等高级功能。

## 3.5.2. 产品优势

本章节介绍了Redis所具备的产品优势。

### 弹性扩容

- 存储容量一键扩容：您可根据业务需求通过控制台对实例存储容量进行调整。
- 在线扩容不中断服务：可在线进行调整实例存储容量，无需停止服务，不影响您的业务。

### 资源隔离

- 针对实例级别的资源隔离，可以更好地保障单个用户服务的稳定性。
- 具备多租户隔离能力从而避免相互影响，可控制不同实例所能使用的系统资源，包括CPU、内存、IO、磁盘空间等。
- 通过多实例方式，支持云平台集群下多租户并行执行，租户任务提交到不同的实例下的队列执行。通过划分Redis实例实现租户间资源隔离。

### 数据安全

- 数据持久化存储：采用内存+硬盘的存储方式，在提供高速数据读写能力的同时满足数据持久化需求。支持从持久化数据库中读取数据加载到缓存数据库。
- 数据主从双备份：所有数据在主从节点上进行双备份，确保数据不丢失。
- 访问控制：支持密码认证方式以确保访问安全可靠。

### 高可用

- 每个实例均有主从双节点：避免单点故障引起的服务中断。
- 硬件故障自动检测与恢复：自动侦测硬件故障并在数秒内切换，尽量减少突发硬件故障对服务的影响。
- 支持对集群内服务器硬盘故障自动容错处理，支持硬盘热插拔，故障硬盘的业务恢复时间小于2分钟。

### 简单易用

- 兼容Redis命令，任何Redis客户端都可以轻松地与Redis实例建立连接进行数据操作。
- 支持批量命令。

### 权限管理

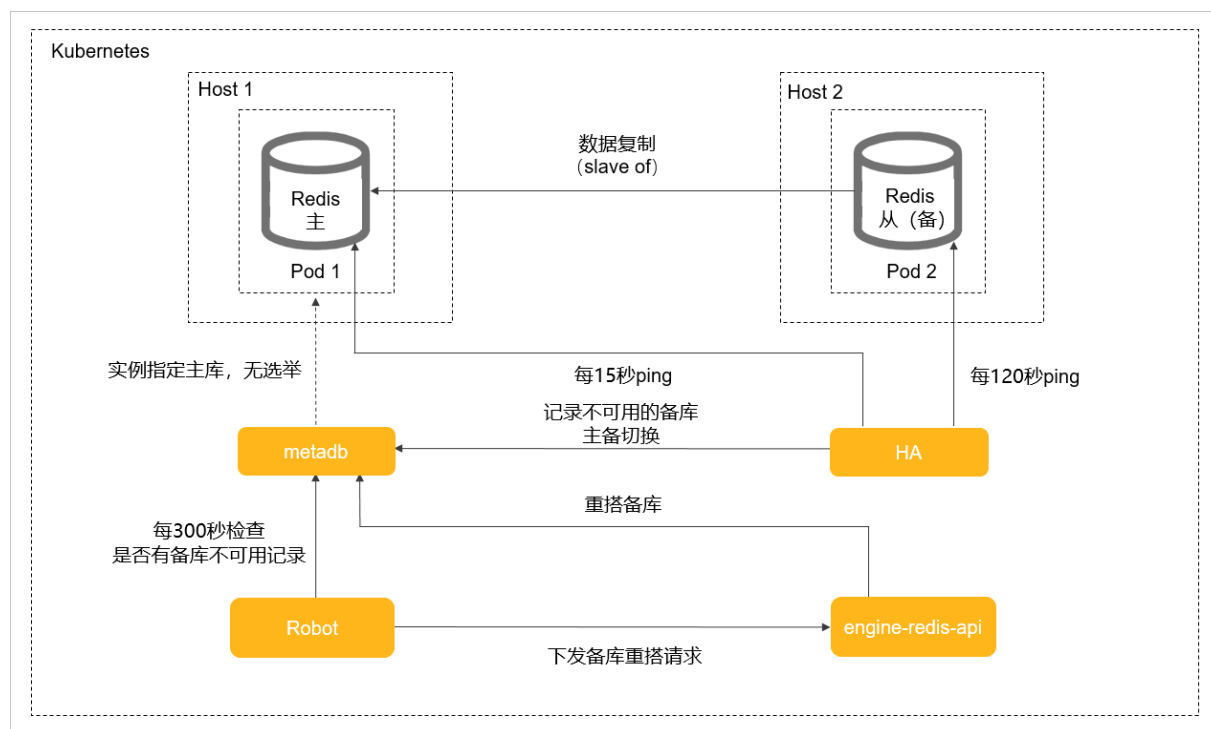
- 支持数据访问权限管理，包括登录权限、创建表权限、读写权限等。
- 提供集中统一的用户权限管理功能，将系统中各组件零散的权限管理功能集中呈现和管理，对普通用户屏蔽掉内部的权限管理细节，对管理员简化权限管理的操作方法，提升权限管理的易用性和用户体验。
- 支持通过控制台，实现多租户统一管理，以及租户资源的动态配置和管理、资源隔离、资源使用统计等功能，支持多级租户的管理功能。

### 调度

支持多集群和多资源池的多租户调度。

## 3.5.3. 产品架构

Redis支持主备双节点结构。



## Redis实例

- Redis实例只有一主一从2个保存数据的节点，没有proxy或cs之类的其他节点。
- 2个节点一定分布在2台不同机器上；只要这2台机器不同时挂掉，Redis实例可依赖高可用机制持续运行。
- 读写都集中在主库上，从（备）库做容灾，没有只读节点。
- 主从直接同步数据，异步复制。
- 实例直接指定主库，无选举，元数据保存在metadb里。

## HA

- HA（Aurora组件）直接ping Redis节点，频率是主库每15秒一次，从库每120秒一次。
- 发现备库不可用时，在metadb里插入一条数据供Robot用。
- 发现主库不可用时，操作metadb，进行主备切换。
- 主备切换会改变k8s的service，可能导致客户端闪断，需要客户端重连。

## Robot

每300秒运行一次，每次检查metadb是否有HA写入的备库不可用记录。如果有，则调用engine-redis-api（Redis的API接口服务）进行备库重搭。

## 3.5.4. 功能特性

云数据库Redis支持主备架构，数据可持久化存储，可用性高，且支持弹性扩展和智能运维。

### 主备架构

主节点的数据实时同步，主节点提供日常服务访问，备节点提供高可用HA（High Availability）能力。当主节点发生故障，系统会切换至备节点以接管业务，保证业务平稳运行。

- 对Redis协议兼容性要求较高的业务。
- 将Redis作为持久化数据存储使用的业务。
- 单个Redis性能压力可控的场景。

- Redis命令相对简单，排序和计算之类的命令较少的场景。

### 数据安全

类别	说明
数据备份与恢复	<ul style="list-style-type: none"> <li>• 备份：支持数据持久化，系统会按照默认的策略自动备份数据（基于RDB快照存储实现持久化），您可以根据业务需求修改自动备份策略，也可以手动发起临时的备份。</li> <li>• 恢复：支持从指定的备份集恢复至当前实例或新实例，恢复至新实例时，新实例中的数据将和该备份集中的数据一致，可用于数据恢复、快速部署业务或数据验证等场景。</li> <li>• 下载备份文件：备份文件会保留7天，如果需要更长时间的备份存档（例如监管或信息安全需要），您可以将备份文件下载到本地进行存储。</li> </ul>
深度内核优化	阿里云专家团队对Redis源码进行深度内核优化，有效防止内存溢出，修复安全漏洞。

### 高可用性

类别	说明
双副本	由主备节点组成双副本，主备节点的数据实时同步，主节点提供日常服务访问，备节点提供高可用HA（High Availability）能力。当主节点发生故障，系统会切换至备节点以接管业务，保证业务平稳运行。
冗余与自动检测	<ul style="list-style-type: none"> <li>• 系统各组件采用冗余设计，节点故障后持续运行，无单点失败风险。</li> <li>• 系统会自动侦测硬件故障，发生故障时能够进行故障转移，在数秒内恢复服务。</li> </ul>

### 弹性扩展

类别	说明
容量扩展	Redis实例创建完成后，随着业务量的变化，该实例的性能可能不足或者过剩，此时可通过变更实例的配置，调整实例的架构或内存规格，以满足不同场景下对实例的性能和容量需求。
性能扩展	

### 智能运维

类别	说明
性能监控平台	支持丰富的性能监控指标（如CPU使用率、连接数等），您可以查询过去一个月内指定时间段的监控数据，帮助您掌握Redis服务的运行状况和问题溯源。
可视化管理平台	支持基于Web的可视化管理控制台，可提供丰富的运维和管理功能（例如备份数据、参数设置等），帮助您便捷、可视化地管理实例。

## 3.6. Lindorm

### 3.6.1. 产品概述

Lindorm是一款适用于任何规模、多种模型的云原生数据库服务，支持海量数据的低成本存储处理，提供宽表、时序、搜索、文件等多种数据模型，兼容HBase、Cassandra、Phoenix、OpenTSDB、Solr、SQL等多种开源标准接口，是互联网、IoT、车联网、广告、社交、监控、游戏、风控等场景首选数据库，也是为阿里巴巴核心业务提供关键支撑的数据库之一。

Lindorm基于存储计算分离、多模共享融合的云原生架构，具备弹性、低成本、简单易用、开放、稳定等优势，适合元数据、日志、账单、标签、消息、报表、维表、结果表、Feed流、用户画像、设备数据、监控数据、传感器数据、小文件、小图片等数据的存储和分析，其核心能力包括：

- 融合多模：支持宽表、时序、搜索、文件四种模型，提供统一联合查询和独立开源接口两种方式，模型之间数据互融互通，帮助应用开发更加敏捷、灵活、高效。
- 极致性价比：支持千万级高并发吞吐、毫秒级访问延迟，并通过高密度低成本存储介质、智能冷热分离、自适应压缩，大幅减少存储成本。
- 云原生弹性：支持计算资源、存储资源独立弹性伸缩，并提供按需即时弹性的Serverless服务。
- 开放数据生态：提供简单易用的数据交换、处理、订阅等能力，支持与MySQL、Spark、Flink、Kafka等系统无缝打通。

## 多模介绍

Lindorm系统支持宽表、时序、搜索、文件四种模型，提供统一联合查询和独立开源接口两种方式，模型之间数据互融互通，帮助应用开发更加敏捷、灵活、高效。多模型的核心能力由四大数据引擎提供，包括：

- 宽表引擎

面向海量KV、表格数据，具备全局二级索引、多维检索、动态列、TTL等能力，适用于元数据、订单、账单、画像、社交、feed流、日志等场景，兼容HBase、Phoenix(SQL)、Cassandra(CQL)等开源标准接口。

支持千万级高并发吞吐，支持百PB级存储，吞吐性能是开源HBase(Apache HBase)的3-7倍，P99时延为开源HBase(Apache HBase)的1/10，平均故障恢复时间相比开源HBase(Apache HBase)提升10倍，支持冷热分离，压缩率比开源HBase(Apache HBase)提升一倍，综合存储成本为开源HBase(Apache HBase)的1/2。

- 时序引擎

面向IoT、监控等场景存储和处理量测数据、设备运行数据等时序数据。提供HTTP API接口(兼容OpenTSDB API)、支持SQL查询。针对时序数据设计的压缩算法，压缩率最高为15:1。支持海量数据的多维查询和聚合计算，支持降采样和预聚合。

- 搜索引擎

面向海量文本、文档数据，具备全文检索、聚合计算、复杂多维查询等能力，同时可无缝作为宽表、时序引擎的索引存储，加速检索查询，适用于日志、账单、画像等场景，兼容开源Solr标准接口。

- 文件引擎

提供宽表、时序、搜索引擎底层共享存储的服务化访问能力，从而加速多模引擎底层数据文件的导入导出及计算分析效率，兼容开源HDFS标准接口。

对于目前使用类HBase+ElasticSearch或HBase+OpenTSDB+ES的应用场景，比如监控、社交、广告等，利用Lindorm的原生多模能力，将很好地解决架构复杂、查询痛苦、一致性弱、成本高、功能不对齐等痛点，让业务创新更高效。

## 3.6.2. 产品优势

Lindorm是一款适用于任何规模、多种模型的云原生数据库服务，其基于存储计算分离、多模共享融合的云原生架构设计，具备弹性、低成本、稳定可靠、简单易用、开放、生态友好等优势。

### 云原生弹性

- 基于存储计算分离的全分布式架构，支持计算资源和存储资源的独立弹性伸缩。
- 存储资源支持秒级在线扩缩，计算资源（宽表引擎、时序引擎、搜索引擎）支持分钟级在线伸缩。
- 提供按需即时弹性的Serverless服务，自适应弹性伸缩，无需人工容量管理。

### 低成本

- 提供性能型、标准型、容量型多种存储规格，可满足不同场景的性价比选择。
- 多种引擎共享统一的存储池，减少存储碎片，降低使用成本。

- 容量型存储单价为业界最低标准，大幅低于基于ECS本地盘自建。
- 内置深度优化的压缩算法，数据压缩率高达10:1以上，相比snappy提高50%以上。
- 内置面向数据类型的自适应编码，数据无需解码，即可快速查找。
- 支持智能冷热分离，多种存储规格混合使用，大幅降低数据存储综合成本。

### 高性能

- 宽表引擎：支持千万级并发吞吐，支持百PB级存储，吞吐性能是开源HBase的3~7倍，P99时延为开源HBase的1/10。
- 时序引擎：写入性能和查询性能是InfluxDB的1.3倍，是OpenTSDB的5~10倍。
- 搜索引擎：基于Lucene引擎深度优化，综合性能比开源Solr/ES提升30%。

### 多模互通融合

- 多模型之间支持数据互通，搜索引擎可无缝作为宽表引擎、时序引擎的索引存储，加速复杂条件查询。
- 支持统一的SQL访问，以及跨多模引擎关联查询。

### 易用

- 兼容多种开源标准接口，包括HBase/Cassandra/Phoenix、OpenTSDB、Solr，业务可以无缝迁移。
- 云托管服务，免运维。
- 图形化系统管理和数据访问，操作简单。

### 高可用

- 系统采用分布式多副本架构，集群自动容灾恢复。
- 支持跨可用区、强一致的容灾能力，具备金融级可用性标准。
- 支持全球多活部署。

### 高可靠

- 底层多副本存储，实现了数据的高可靠性。
- 提供企业级备份能力。
- 在阿里巴巴部署上万台，支持过10年天猫双十一，久经验证。

### 开放生态

- 支持与MySQL、HBase、Cassandra等系统的平滑在线数据搬迁。
- 可轻松与Spark、Flink、DLA、MaxCompute等计算引擎无缝对接。
- 支持无缝订阅Kafka、SLS等日志通道的数据，并具备快速处理能力。
- 通过Lindorm Stream，可以实时订阅Lindorm的增量变更数据，自定义消费。

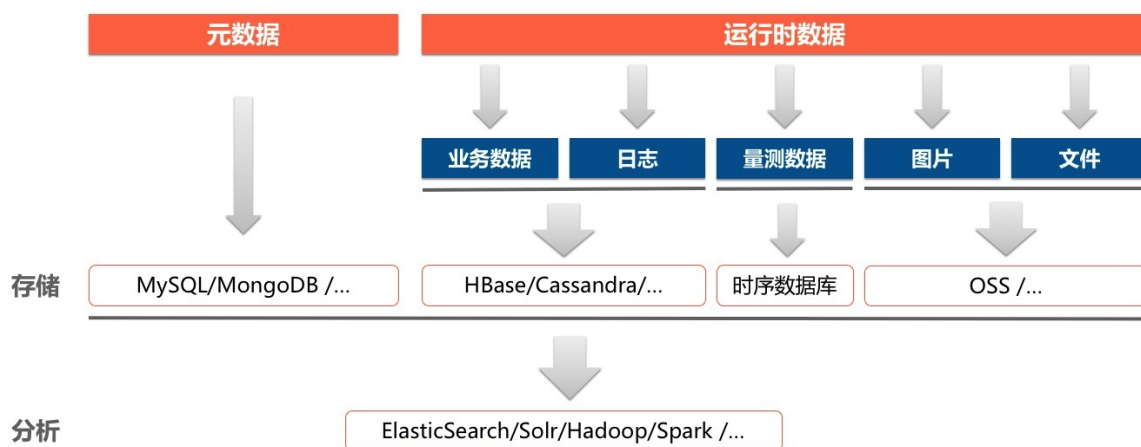
## 3.6.3. 产品架构

本文介绍云原生多模数据库Lindorm的产品架构。

### 业务背景

伴随着信息技术的飞速发展，各行各业在业务生产中产生的数据种类越来越多，有结构化的业务元数据、业务运行数据、设备或者系统的量测数据，也有半结构化的业务运行数据、日志、图片或者文件等。按照传统方案，为了满足多种类型数据的存储、查询和分析需求，在设计IT架构时，需要针对不同种类的数据，采用不同的存储分析技术，如下图：

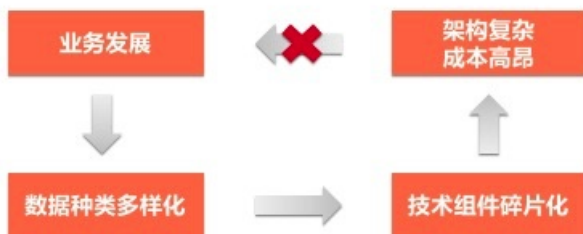




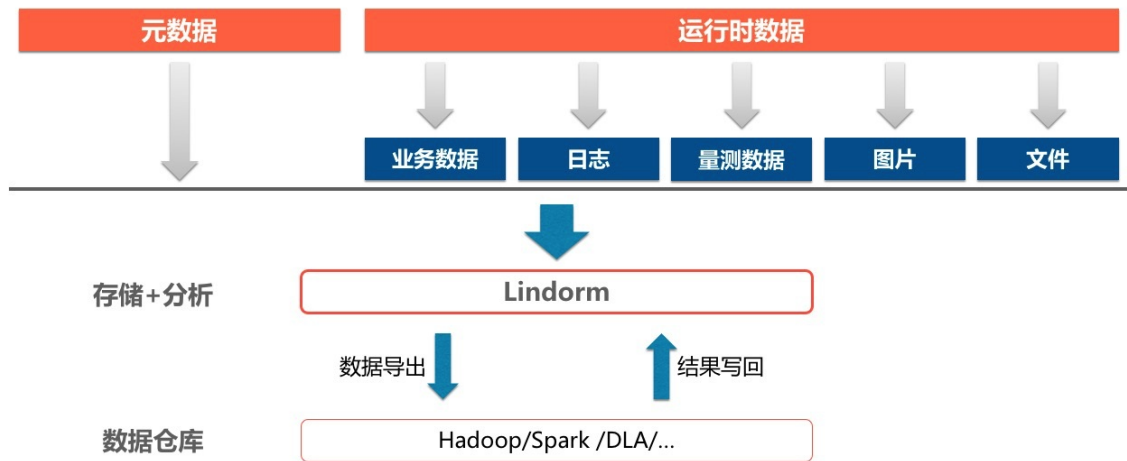
这种技术方案，是一种典型的技术碎片化的处理方案。针对不同的数据，使用不同的数据库来处理。有如下几个弊端：

- 涉及的技术组件多且杂
- 技术选型复杂
- 数据存取、数据同步的链路长

这些弊端会对信息系统建设带来巨大的问题，对技术人员要求高、业务上线周期长、故障率高、维护成本高。更进一步，技术碎片化导致技术架构割裂，不利于技术架构的长期演进和发展，最终导致技术架构无法跟进业务前进的步伐。举个简单的例子，当业务发展，需要支持跨可用区高可用、全球同步或者降低存储成本时，各技术组件都需要独立演进和发展，耗时耗人耗力，是一件非常痛苦的事情。并且随着业务的发展，数据的类型会越来越多，对不同种类数据的差异化处理需求会日渐增加，会导致数据存储碎片化更加严重。

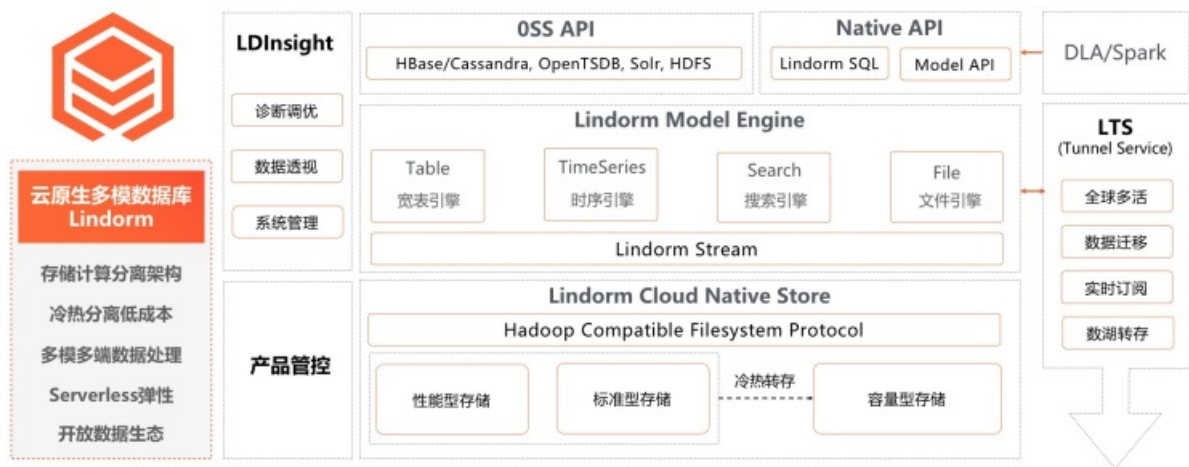


当前信息化技术发展面临的一个主要矛盾是“日益多样的业务需求带来的多种类型数据与数据存储技术架构日趋复杂成本快速上升之间的矛盾”。伴随5G、IoT、智能网联车等新一代信息技术的逐步普及应用，这个矛盾会越来越突出。为了解决这个问题，阿里云自研了业界独家云原生多模数据库Lindorm，满足多模型数据的统一存储、查询和分析需求。如下图所示，与传统方案相比，Lindorm系统将极大地简化数据存储技术架构设计，大幅度提升系统稳定性，降低建设成本投入。



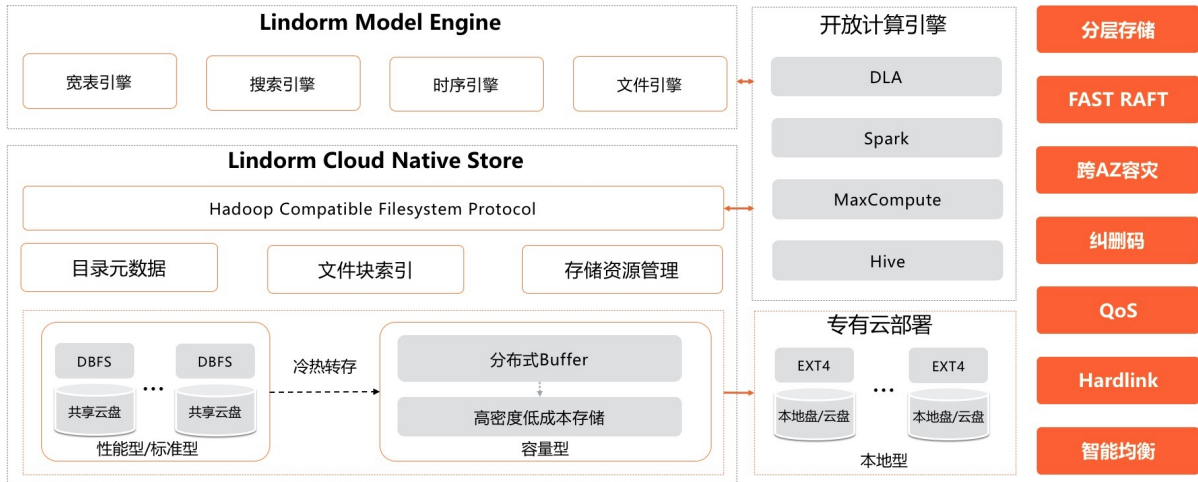
### 总体架构

Lindorm创新性地使用存储计算分离、多模共享融合的云原生架构，以适应云计算时代资源解耦和弹性伸缩的诉求。其中云原生存储引擎LindormStore为统一的存储底座，向上构建各个垂直专用的多模引擎，包括宽表引擎、时序引擎、搜索引擎、文件引擎。在多模引擎之上，Lindorm既提供统一的SQL访问，支持跨模型的联合查询，又提供多个开源标准接口（HBase/Phoenix/Cassandra、OpenTSDB、Solr、HDFS），满足存量业务无缝迁移的需求。最后，统一的数据Stream总线负责引擎之间的数据流转和数据变更的实时捕获，以实现数据迁移、实时订阅、数湖转存、数仓回流、单元化多活、备份恢复等能力。



### 存储引擎

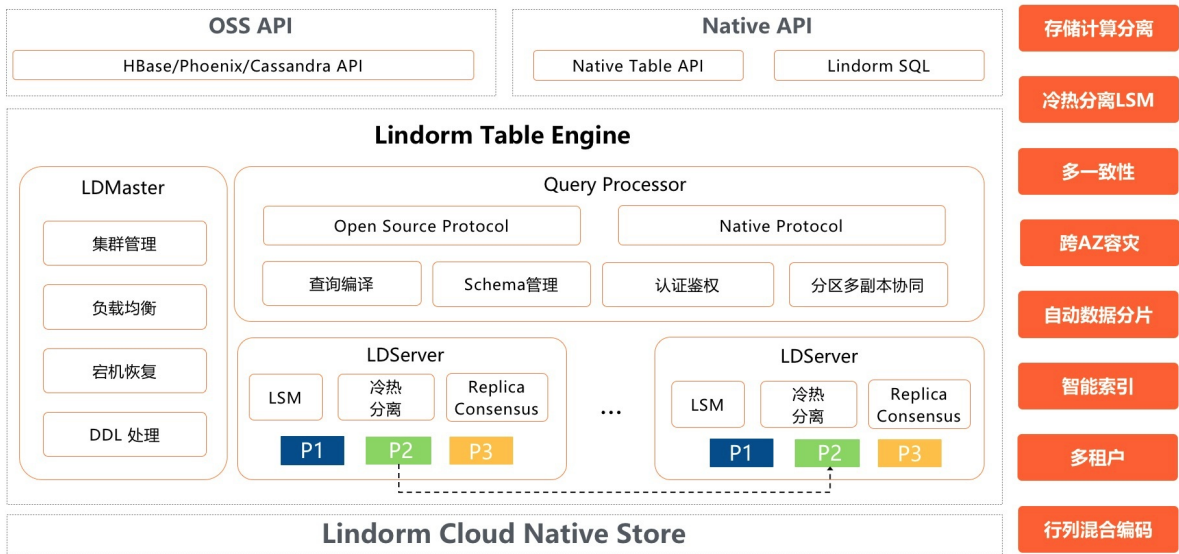
LindormStore是面向云基础存储设施设计、兼容HDFS协议的分布式存储系统，并同时支持运行在本地盘环境，以满足部分专有云、专属大客户的需求，向多模引擎和外部计算系统提供统一的、与环境无关的标准接口，整体架构如下：



LindormStore提供性能型、标准型、容量型等多种规格，并且面对真实场景的数据冷热特点，支持性能型/标准型、容量型多种存储混合使用的形态，以适应真实场景的数据冷热特点，可结合多模引擎的冷热分离能力，实现冷热存储空间的自由配比，让用户的海量数据进一步享受云计算的低成本红利。

### 宽表引擎

LindormTable是面向海量半结构化、结构化数据设计的分布式NoSQL系统，适用于元数据、订单、账单、画像、社交、feed流、日志等场景，兼容HBase、Phoenix(SQL)、Cassandra等开源标准接口。其基于数据自动分区+分区多副本+LSM的架构思想，具备全局二级索引、多维检索、动态列、TTL等查询处理能力，支持单表百万亿行规模、千万级并发、毫秒级响应、跨机房强一致容灾，高效满足业务大规模数据的在线存储与查询需求。面向海量半结构化、结构化数据设计的分布式NoSQL系统，能够兼容HBase、Phoenix(SQL)、Cassandra等开源标准接口。整体架构如下：



LindormTable的数据持久化存储在LStore中，表的数据通过自动Sharding分散到集群中的多台服务器上，并且每一个Partition可以拥有1至N个副本，这N个副本拥有主、从两种角色，主从副本可以加载在不同的Zone，从而保障集群的高可用和强一致。针对不同的一致性模式，主从副本之间的数据同步和读写模式如下：

- 强一致模式：只有主副本提供读写，数据会异步回放到从副本,主副本所在节点故障，从副本晋升为主(晋升之前会保障数据同步完成，从副本拥有所有最新数据，整体过程由Master协调负责。
- 最终一致模式：主从副本都提供读写，数据会相互同步，保证副本之间的数据最终一致。

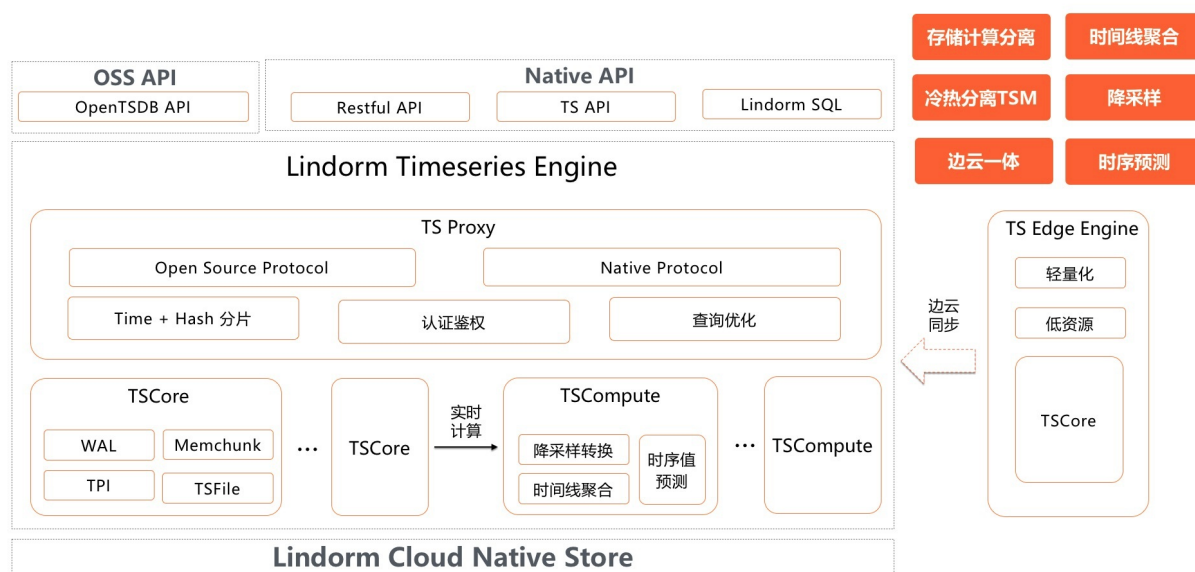
LindormTable的多副本架构基于PACELC理念设计，每一个数据表都支持单独支持设置一致性模式，从而拥有不同的可用性和性能。在最终一致模式下，服务端会对每一个读写请求在一定条件下触发多副本并发访问，从而大幅提升请求的成功率和减少响应毛刺。该并发机制建立在内部的异步访问框架上，相比于启动多线程，额外资源消耗可以忽略不计。对于并发访问的触发条件，主要包括两个类型：其一是限时触发，对于每一个请求，都可以单独设置一个GlitchTimeout，当请求运行时间超过该值未得到响应后，则并发一个请求到其他N-1个副本，最终取最快的那个响应；其二是黑名单规避，服务端内部会基于超时、抛错、检测等机制，主动拉黑存在慢、停止响应等问题的副本，使得请求能够主动避开受软硬件缺陷的节点，让服务最大可能保持平滑。对于像掉电终止这样的不想应场景，在节点不可服务至失去网络心跳通常会存在一两分钟的延迟，利用LTable的这种多副本协同设计，可以将影响控制在10秒以内，大幅提升服务的可用性。

LindormTable的LSM结构面向冷热分离设计，支持用户的数据表在引擎内自动进行冷热分层，并保持透明查询，其底层结合LStore的冷热存储混合管理能力，大幅降低海量数据的总体存储成本。的LSM结构面向冷热分离设计，支持用户的数据表在引擎内自动进行冷热分层，并保持透明查询，其底层结合LStore的冷热存储混合管理能力，大幅降低海量数据的总体存储成本。

LindormTable提供的数据库模型是一种支持类型的松散表结构。相比于传统关系模型，LTable除了支持预定义字段类型外，还可以随时动态添加列，而无需提前发起DDL变更，以适应大数据灵活多变的特点。同时，LTable支持全局二级索引、倒排索引，系统会自动根据查询条件选择最合适的索引，加速条件组合查询，特别适合如画像、账单场景海量数据的查询需求。

### 时序引擎

LindormTS是面向海量时序数据设计的分布式时序引擎，兼容开源OpenTSDB标准接口，其基于时序数据特点和查询方式，采用Timerange+hash结合的分区算法，时序专向优化的LSM架构和文件结构，支持海量时序数据的低成本存储、预降采样、聚合计算、高可用容灾等，高效满足IoT/监控等场景的测量数据、设备运行数据的存储处理需求，整体架构如下：

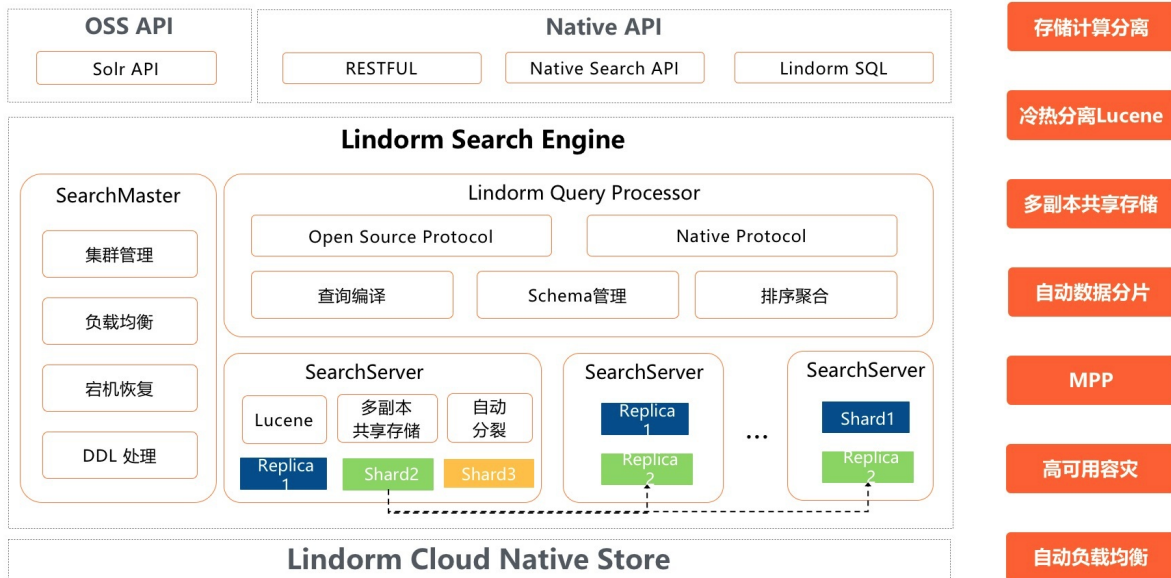


TSCore是时序引擎中负责数据组织的核心部分，其整体思想与LSM结构相似，数据先写入Memchunk，然后Flush到磁盘，但由于时序数据天然顺序写入特征，定向专用的时序文件TSFile的结构设计为以时间窗口进行切片，数据在物理和逻辑上均按时间分层，从而大幅减少Compaction的IO放大，并使得数据的TTL、冷热分离等实现非常高效。

TSCoordinate是负责时序数据实时计算的组件，重点解决监控领域常见的降采样转换、时间线聚合、时序值预测等需求，其通过Lindorm Stream进行数据订阅，并完全基于内存计算，所以，整体非常的轻量、高效，适合系统已预置的计算功能。针对部分灵活复杂的分析需求，用户仍可以通过对接Spark、Flink等系统实现，从而支撑更多场景和适应业务变化。

### 搜索引擎

LindormSearch是面向海量数据设计的分布式搜索引擎，兼容开源Solr标准接口，同时可无缝作为宽表、时序引擎的索引存储，加速检索查询。其整体架构与宽表引擎一致，基于数据自动分区+分区多副本+Lucene的结构设计，具备全文检索、聚合计算、复杂多维查询等能力，支持水平扩展、一写多读、跨机房容灾、TTL等，满足海量数据下的高效检索需求，具体如下：



LindormSearch的数据持久化存储在LindormStore中，通过自动Sharding的方式分散到多台SearchServer中，每一个分片拥有多个副本，支持一写多读，提升查询聚合的效率，同时这些副本之间共享存储，有效消除副本之间的存储冗余。

在Lindorm系统中，LindormSearch既可以作为一种独立的模型，提供半结构化、非结构化数据的松散文档视图，适用于日志数据分析、内容全文检索；也可以作为宽表引擎、时序引擎的索引存储，对用户保持透明，即宽表或时序中的部分字段通过内部的数据链路自动同步搜索引擎，而数据的模型及读写访问对用户保持统一，用户无需关心搜索引擎的存在，跨引擎之间的数据关联、一致性、查询聚合、生命周期等工作全部由系统内部协同处理，用简单透明的方式发挥多模融合的价值。

### 文件引擎

LindormFile是基于LindormStore轻量封装的分布式文件模型服务，其核心是提供安全认证、限流保护、多网络访问等服务化能力，从而使得外部系统可以直接访问多模引擎的底层文件，大幅提升备份归档、计算分析等场景的效率。同时，用户可以在离线计算系统直接生成底层数据格式的物理文件，通过LindormFile入口，将其快速导入到LindormStore中，以减少对在线服务的影响。对于部分存储计算的混合场景，用户可以将计算前的源文件存在LindormFile，计算后的数据结果存在LindormTable，结合Spark/DLA等大规模计算引擎，实现简单高效的数据湖分析能力。

## 3.6.4. 功能特性

云原生多模数据库Lindorm支持云原生弹性、低成本存储、高性能读写等功能，为您的业务稳定提供保障。

### 云原生弹性

- 基于存储计算分离的全分布式架构，支持计算资源和存储资源的独立弹性伸缩。
- 存储资源支持秒级在线扩缩，计算资源（宽表引擎、时序引擎、搜索引擎）支持分钟级在线伸缩。

### 低成本存储

- 提供多种存储规格，可满足不同场景的性价比选择。
- 多种引擎共享统一的存储池，减少存储碎片，降低使用成本。
- 内置深度优化的压缩算法，数据压缩率高达10:1以上，相比snappy提高50%以上。
- 内置面向数据类型的自适应编码，数据无需解码，即可快速查找。

- 支持智能冷热分离，多种存储规格混合使用，大幅降低数据存储综合成本。

### 高性能读写

- 宽表引擎：支持千万级并发吞吐，支持百PB级存储，吞吐性能是开源HBase的3~7倍，P99时延为开源HBase的1/10。
- 时序引擎：写入性能和查询性能是InfluxDB的1.3倍，是OpenTSDB的5~10倍。
- 搜索引擎：基于Lucene引擎深度优化，综合性能比开源Solr/ES提升30%。

### 开发生态

- 支持与MySQL、HBase、Cassandra等系统的平滑在线数据搬迁。
- 可轻松与Spark、Flink、DLA、MaxCompute等计算引擎无缝对接。
- 支持无缝订阅Kafka、SLS等日志通道的数据，并具备快速处理能力。
- 通过Lindorm Stream，可以实时订阅Lindorm的增量变更数据，自定义消费。

## 3.6.5. 应用场景

Lindorm是阿里云自研的云原生多模型数据库，面向海量多模型数据的低成本存储分析，构建万物互联时代的数据底座。Lindorm支持宽表模型、时序模型，提供自研的宽表引擎、时序引擎和搜索引擎，兼容HBase、Phoenix、OpenTSDB、Solr等多种开源标准接口，提供SQL查询、时序处理、检索分析等能力，满足结构化、半结构化的存储和分析需求，同时支持在线业务和离线业务。

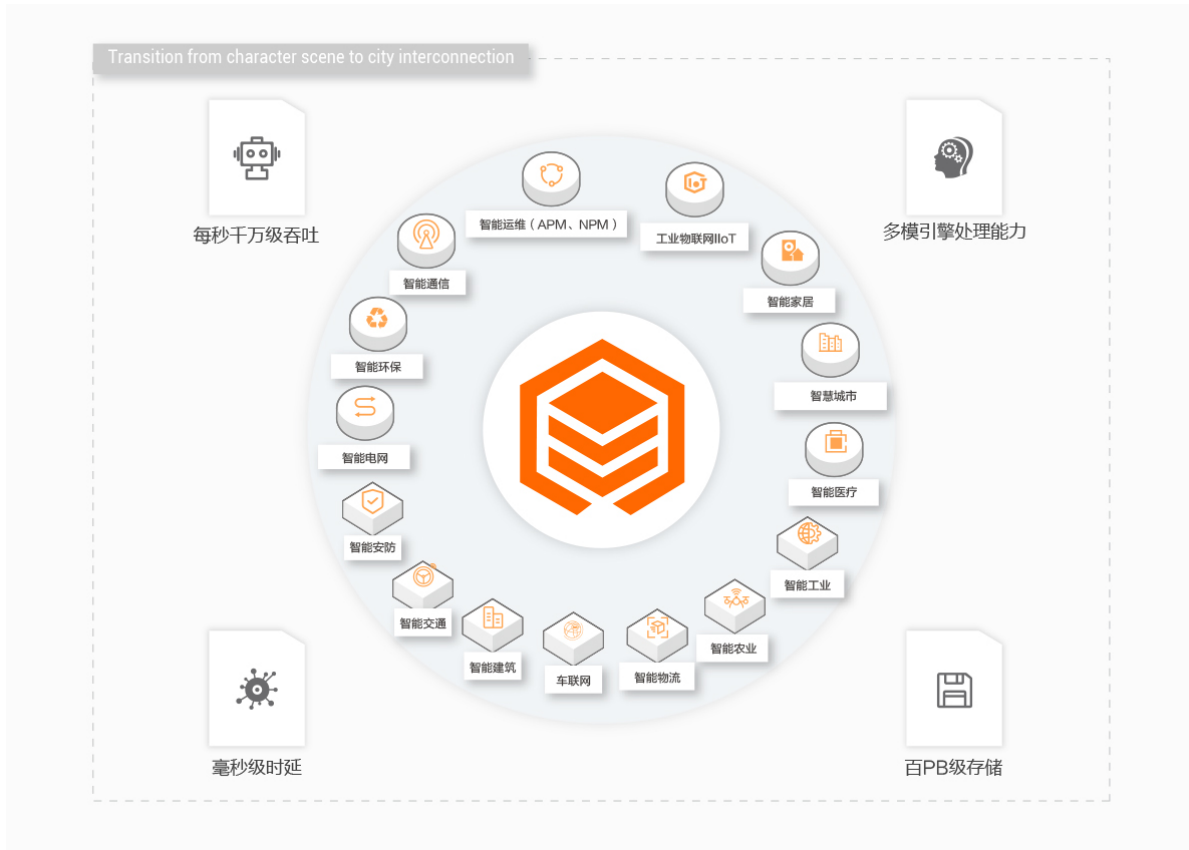
Lindorm可以提供单个毫秒响应的性能，支持水平扩展到PB级存储和千万级QPS，在IoT、淘宝、支付宝、菜鸟等众多阿里巴巴核心服务中起到了关键支撑的作用。

### 阿里巴巴集团内部最佳实践

- Lindorm在阿里巴巴集团内部成熟业务中得到广泛使用。

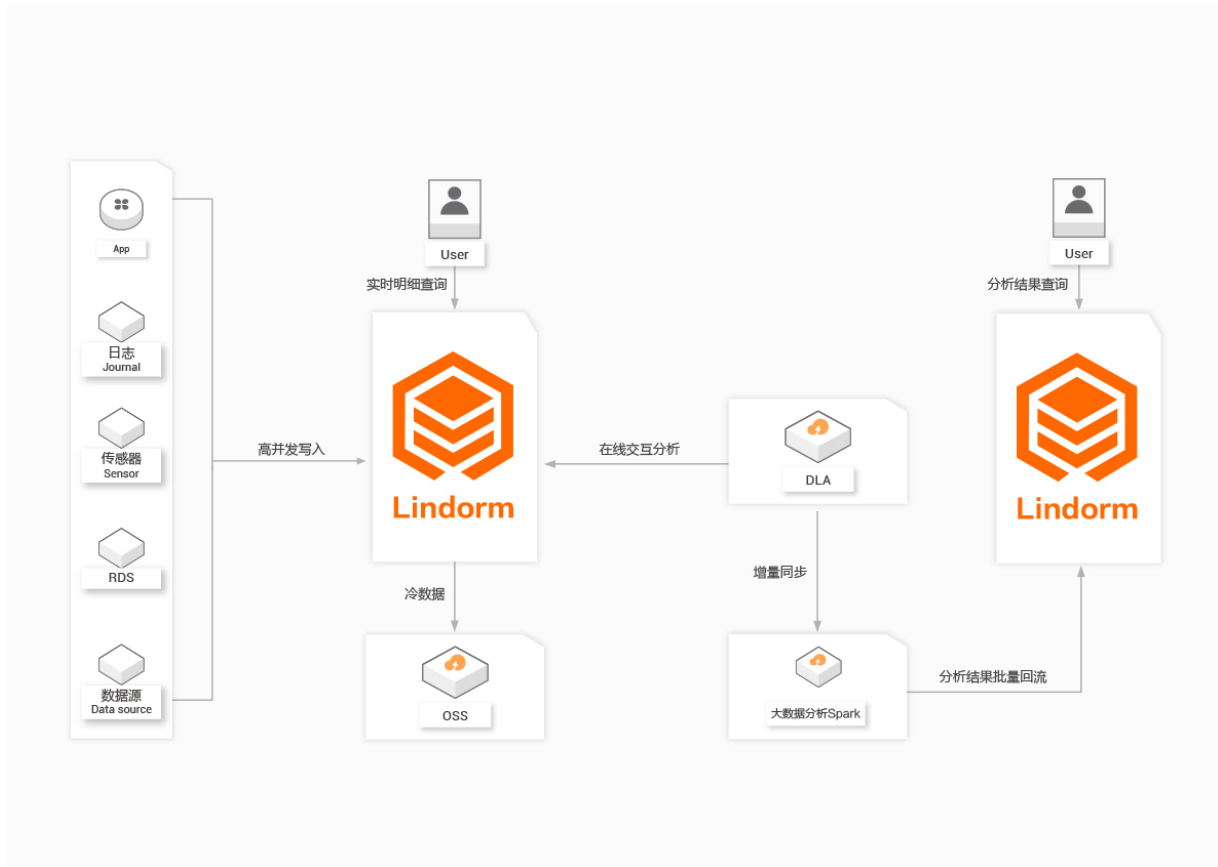


- Lindorm基于自研的云原生多模架构支撑IoT业务飞速发展。



### 大数据场景：海量数据存储与分析

Lindorm支持海量数据的低成本存储、快速批量导入和实时访问，具备高效的增量及全量数据通道，可轻松与Spark、MaxCompute等大数据平台集成，完成数据的大规模离线分析。



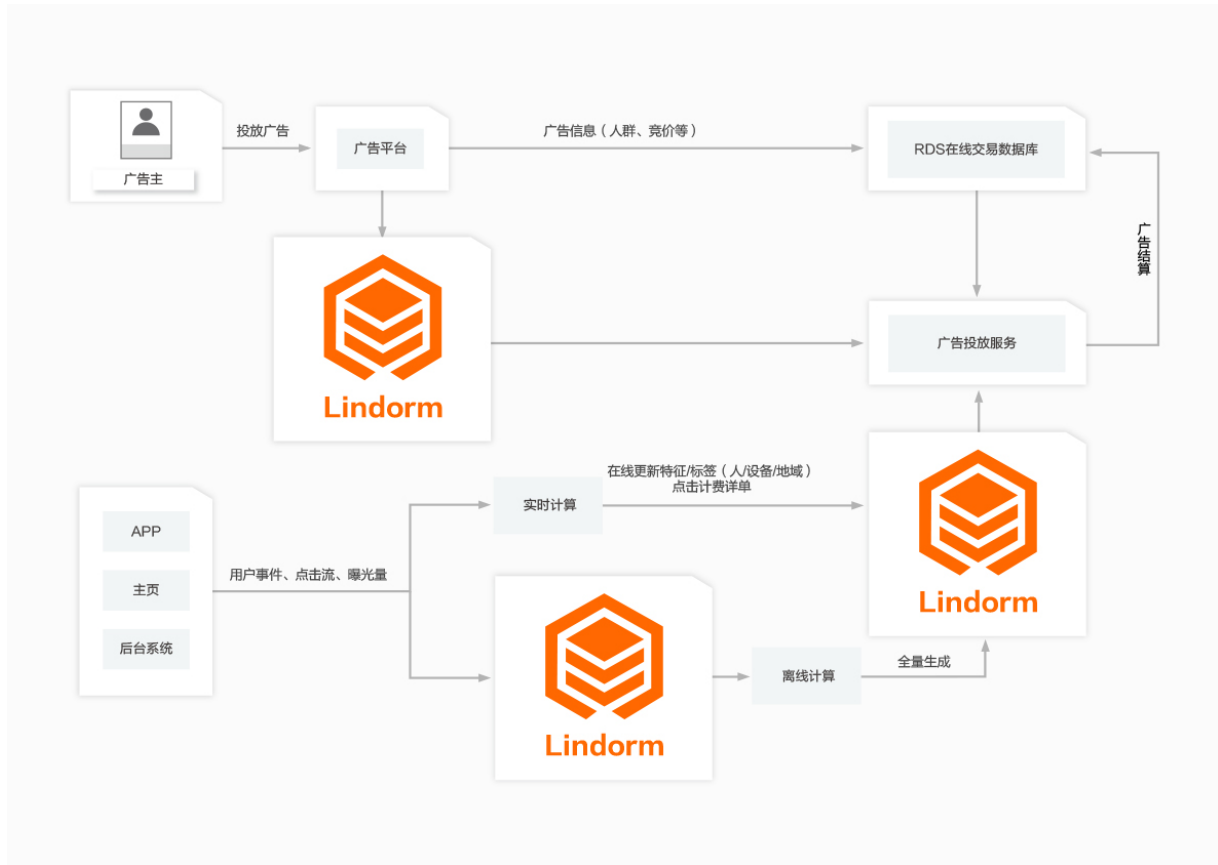
### 优势

- 低成本：高压缩比，数据冷热分离，支持HDD/OSS存储。
- 数据通道：通过BDS构建云Lindorm与异构计算系统的高效、易用的数据链路。
- 快速导入：通过BulkLoad将海量数据快速导入Lindorm，效率比传统方式提升一个数量级。
- 高并发：水平扩展至千万级QPS。
- 弹性：存储计算分离架构，支持独立伸缩，自动化扩容。

### 广告场景：海量广告营销数据的实时存储

使用Lindorm存储广告营销中的画像特征、用户事件、点击流、广告物料等重要数据，提供高并发、低延迟、灵活可靠的能力，帮助您快速构建领先的实时竞价、广告定位投放等系统服务。



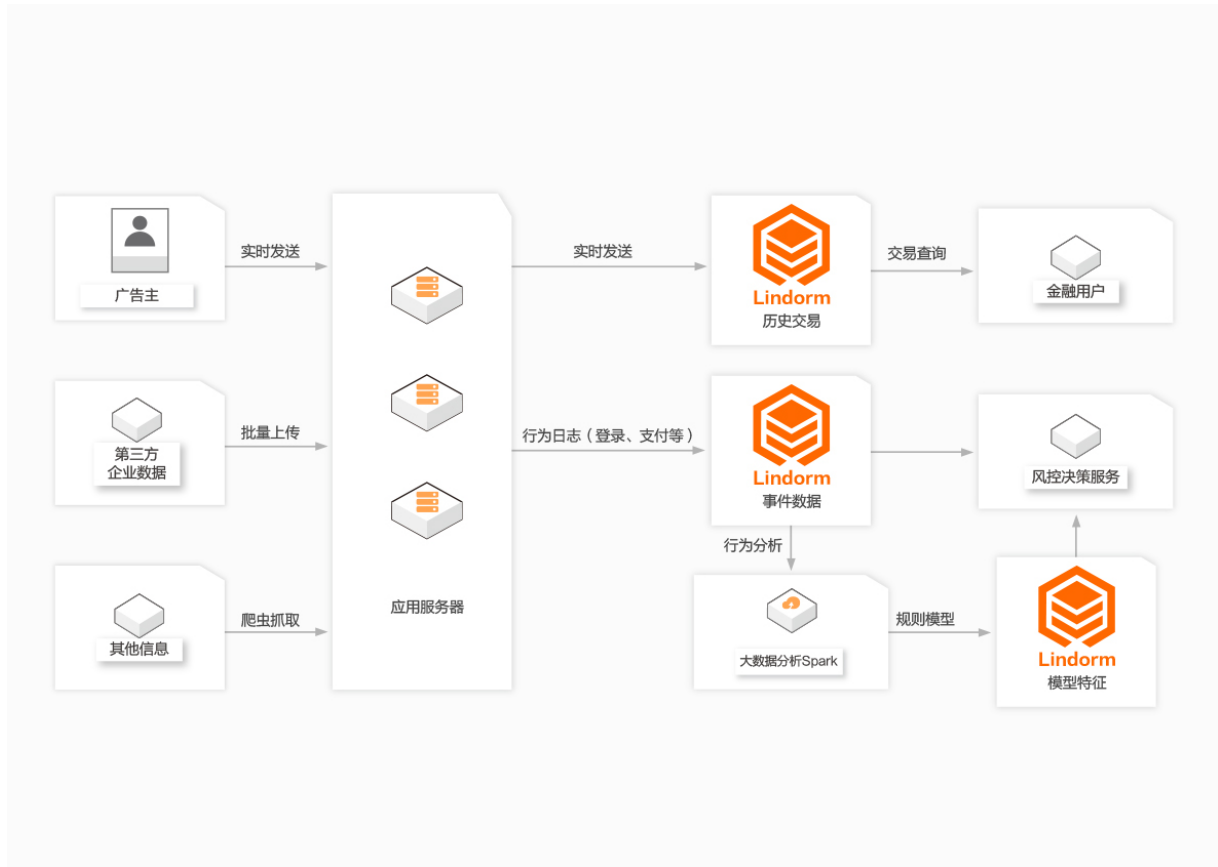


### 优势

- 低延迟：单个毫秒响应，支持双集群请求并发加速。
- 高并发：水平扩展至千万级QPS。
- 使用灵活：动态列，自由增减特征/标签属性；TTL，数据自动过期。
- 低成本：高压缩比，数据冷热分离，支持HDD/OSS存储。
- 数据通道：通过BDS构建云Lindorm与异构计算系统的高效、易用的数据链路。
- 高可用：主备双活容灾，请求自动容错。

### 金融&零售：海量订单记录与风控数据的实时存储

使用Lindorm存储金融交易中的海量订单记录，金融风控中的用户事件、画像特征、规则模型、设备指纹等重要数据，提供低成本、高并发、灵活可靠的能力，帮助您构建领先的金融交易与风控服务。

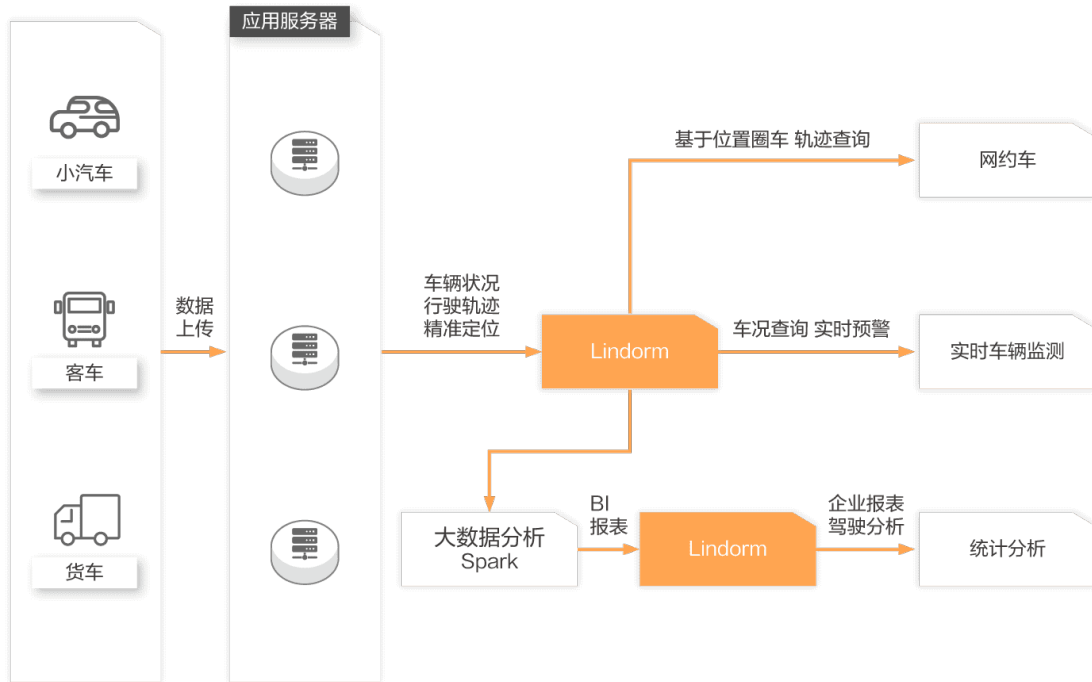


**优势**

- 低成本：高压缩比，数据冷热分离，支持HDD/OSS存储。
- 高并发：水平扩展至千万级QPS。
- 使用灵活：动态列，自由增减特征/标签属性；TTL，数据自动过期；多版本。
- 低延迟：单个毫秒响应，支持双集群请求并发加速。
- 数据通道：通过BDS构建Lindorm与异构计算系统的高效、易用的数据链路。
- 高可用：主备双活容灾，请求自动容错。

**车联网：车辆轨迹与状况数据的高效存储处理**

使用Lindorm存储车联网中的行使轨迹、车辆状况、精准定位等重要数据，提供低成本、弹性、灵活可靠的能力，帮助您构建领先的网约车、物流运输、新能源车检测等场景服务。

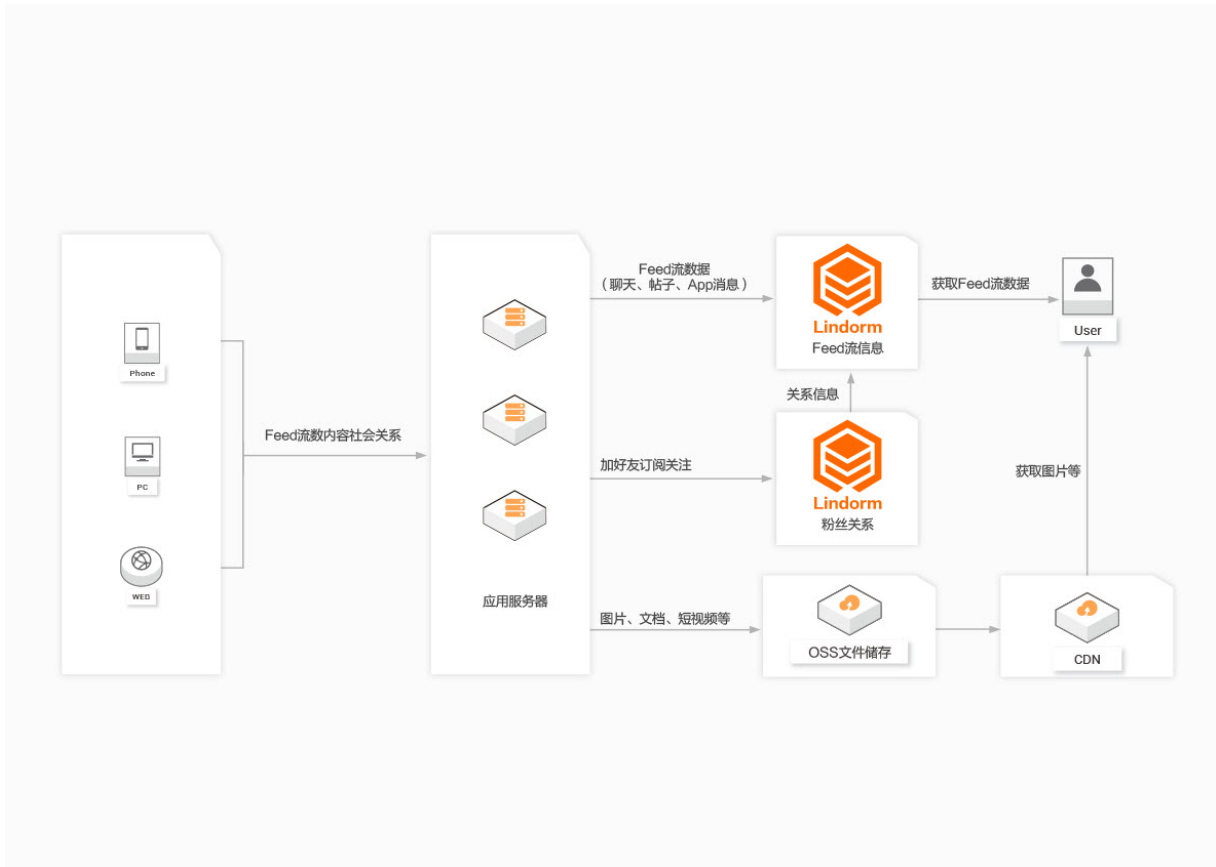


**优势**

- 低成本：高压缩比，数据冷热分离，支持HDD/OSS存储。
- 弹性伸缩：存储计算分离架构，支持独立伸缩，自动化扩容。
- 使用灵活：动态列，自由增减特征/标签属性；TTL，数据自动过期；多版本。
- 低延迟：单个毫秒响应，支持双集群请求并发加速。
- 数据通道：通过BDS构建Lindorm与异构计算系统的高效、易用的数据链路。
- 高可用：主备双活容灾，请求自动容错。

**互联网社交：高效、稳定的社交Feed流信息存储**

使用Lindorm存储社交场景中的聊天、评论、帖子、点赞等重要数据，提供易开发、高可用、延迟的能力，帮助您快速构建稳定可靠的现代社交Feed流系统。



**优势**

- 易开发：提供社交IM场景专属的Feed流功能，开发效率和运行性能提升一个数量级。
- 高可用：主备双活容灾，请求自动容错。
- 低延迟：单个毫秒响应，支持双集群请求并发加速。
- 低成本：高压缩比，数据冷热分离，支持HDD/OSS存储。
- 弹性：存储计算分离架构，支持独立伸缩，自动化扩容链路。

### 3.6.6. 使用限制

本章节介绍了Lindorm数据库的使用限制。

#### 宽表引擎

限制项	限制范围	限制说明
单集群规模	单集群规模最大100，单次最大5台	超过限制联系运维人员处理

#### 时序引擎

时序引擎 (TSDB) 对某些具体指标进行了约束，您在使用TSDB时注意不要超过相应的限制值，以免系统出现异常。

单个时序引擎节点 (30核96 GB)，具体的限制项和限制值请参见下表。

限制项	限制值
每秒写入数据点数	48万

每秒查询并发数	400
单次查询返回数据点数量	9000万
单次查询中子查询数量	200
单次子查询的查询时间	90s

## 搜索引擎

限制项	限制值
单集群规模	建议最大50台
单集群索引表数量	建议最大1000个

## 3.6.7. 基本概念

本章节介绍了Lindorm数据库中的一些基本概念。

### Lindorm

- HBase

HBase是一个开源的非关系型分布式数据库（NoSQL），基于谷歌的BigTable建模，是一个高可靠性、高性能、高伸缩的分布式存储系统，使用HBase技术可在廉价PC Server上搭建起大规模结构化存储集群。

- Phoenix

Phoenix是一个开源的HBase SQL层。它不仅可以使用标准的JDBC API替代HBase client API创建表，插入和查询HBase，也支持二级索引、事务以及多种SQL层优化。

- QPS

每秒查询率QPS是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准。

- 二级索引

HBase的一级索引就是rowkey，rowkey之外的索引能力称为二级索引。

- Hadoop

Hadoop是一个由Apache基金会所开发的分布式系统基础架构。

### 时序引擎

- 时间序列数据库

英文全称为Time Series Database，提供高效存取时序数据和统计分析功能的数据管理系统。

- 时序数据

基于稳定频率持续产生的一系列指标监测数据。例如，监测某城市的空气质量时，每秒采集一个二氧化硫浓度的值而产生的一系列数据。

- 度量

监测数据的指标，例如风力和温度。

- 标签

度量（Metric）虽然指明了要监测的指标项，但没有指明要针对什么对象的该指标项进行监测。标签（Tag）就是用于表明指标项监测针对的具体对象，属于指定度量下的数据子类别。

一个标签 (Tag) 由一个标签键 (TagKey) 和一个对应的标签值 (TagValue) 组成，例如“城市 (TagKey) = 杭州 (TagValue)”就是一个标签 (Tag)。更多标签示例：机房 = A、IP = 172.220.110.1。

**说明** 当标签键和标签值都相同才算同一个标签；标签键相同，标签值不同，则不是同一个标签。

在监测数据的时候，指定度量是“气温”，标签是“城市 = 杭州”，则监测的就是杭州市的气温。

- **标签键**  
为指标项监测指定的对象类型（会有对应的标签值来定位该对象类型下的具体对象），例如国家、省份、城市、机房、IP 等。
- **标签值**  
标签键 (TagKey) 对应的值。例如，当标签键 (TagKey) 是“国家”时，可指定标签值 (TagValue) 为“中国”。
- **值**  
度量对应的值，例如15 级（风力）和20 °C（温度）。
- **时间戳**  
数据（度量值）产生的时间点。
- **时间点**  
针对监测对象的某项指标（由度量和标签定义）按特定时间间隔（连续的时间戳）采集的每个度量值就是一个数据点。“一个度量 + N 个标签 (N >= 1) + 一个时间戳 + 一个值”定义一个数据点。
- **时间序列**  
也称作时间线。针对监测对象的某项指标（由度量和标签定义）按特定时间间隔（连续的时间戳）采集的一系列数据点就是一个时间序列。“一个度量 + N 个标签 (N >= 1) + N 个时间戳 (N >= 1) + N 个值 (N >= 1)”定义一个时间序列。示意图如下：

Metric: Temperature	
Tags: Floor = 33 Room = 3302 Device ID = 7649501	
Timestamp	Value
1492158910	26
1492158920	25.8
1492158930	26.1
1492158940	26.3
1492158950	26.5

5 个数据点 (Data Point)

1 个时间序列 (Time Series)

- **时间线**  
等同于时间序列的概念。
- **时间精度**  
时间线数据的写入时间精度——秒、分钟、小时或者其他稳定时间频度。例如，每秒一个温度数据的采集频度，每 5 分钟一个负载数据的采集频度。
- **数据组**

如果需要对比不同监测对象（由标签定义）的同一指标（由度量定义）的数据，可以按标签这些数据分成不同的数据组。例如，将温度指标数据按照不同城市进行分组查询，操作类似于该SQL语句：`select temperature from xxx group by city where city in (shanghai, hangzhou)`。

- 空间聚合

当同一个度量（Metric）的查询有多条时间线产生（多个指标采集设备），那么为了将空间的多维数据展现为成同一条时间线，需要进行合并计算，例如，当选定了某个城市某个城区的污染指数时，通常将各个环境监测点的指标数据平均值作为最终区域的指标数据，这个计算过程就是空间聚合。

- 降采样

当查询的时间区间跨度较长而原始数据时间精度较细时，为了满足业务需求的场景、提升查询效率，就会降低数据的查询展现精度，这就叫做降采样，比如按秒采集一年的数据，按照天级别查询展现。

- 数据时效

数据时效是设置的数据的实际有效期，超过有效期的数据会被自动释放。

## 3.7. GDB

### 3.7.1. 产品概述

图数据库（Graph Database，简称GDB）是一种支持Property Graph图模型、用于处理高度连接数据查询与存储的实时、可靠的在线数据库服务。它支持Apache TinkerPop Gremlin和OpenCypher两种查询语言，可以帮您快速构建基于高度连接的数据集的应用程序。

图数据库GDB非常适合社交网络、欺诈检测、推荐引擎、实时图谱、网络/IT运营这类高度互连数据集的场景。例如，在一个典型的社交网络中，常常会存在“谁认识谁，上过什么学校，常住什么地方，喜欢什么餐馆”之类的查询，传统关系型数据库对于超过3张表关联的查询十分低效，但图数据库可轻松应对社交网络的各种复杂存储和查询场景。

### 3.7.2. 产品优势

本文介绍图数据库GDB的产品优势，帮助您更好地了解GDB。

- 图数据库GDB支持属性图，高度兼容Gremlin图查询语言和OpenCypher图查询语言。
- 图数据库GDB由阿里云自主研发，实现了高度优化的图引擎计算层和存储层，通过云盘多副本保障数据超高可靠，并支持ACID事务。
- 图数据库GDB支持高可用版实例，当出现故障时，系统将会基于主备故障的各种异常情况迅速将故障转移，保障业务连续性。
- 图数据库GDB提供了丰富的数据库运维管理能力，包括备份恢复、自动升级、监报告警、故障切换等，能够帮助您大幅降低运维成本。

### 3.7.3. 适用场景

图数据库GDB针对高度互联数据的存储和查询场景进行设计，并在内核层面进行了大量优化，非常适合金融风控、商品推荐、社交推荐、异常检测、异常指标监控和违规团伙挖掘等场景。

#### 金融风控

传统的金融风控模型，能够汇集各个数据源的属性特征信息，但是比较难挖掘数据源之间的深度关联关系。要深度并且快速的挖掘海量数据的关联特征，使用传统的方法则会面临非常大的技术挑战。通过图表示学习技术，提取知识图谱中的拓扑信息特征，作为风控模型的输入条件，参与模型训练，可以帮助金融机构构建更高精度的风控模型。

#### 商品推荐

基于图的推荐算法是当前推荐系统中的一种重要的技术方向，在兼顾了推荐精度的同时，还能让模型具备较好的可解释性。通过图的共性关系发现和分析方法，通过计算共同邻居数进行相似节点推荐。适用于电商、保险的商品推荐场景。

## 社交推荐

图数据库GDB可以轻松应对海量高度互连社交数据的实时存储和高效查询，帮助您快速构建复杂的社交网络系统。例如，在一个典型的社交网络中，常常会存在“谁认识谁，谁上过什么学校，谁常住什么地方，谁喜欢什么餐馆”等查询，传统关系型数据库对于超过3张表关联的查询往往会很低效甚至无法支持，但图数据库从基因层面提供了解决方案，轻松应对社交网络的各种复杂存储和查询。

## 异常检测

金融场景中，债权人所持有的债券是否超过检测抵押物多次担保的价值，对于风险控制尤为重要。以图计算中的环检测算法为核心，找到图中闭环连接，帮助金融机构发现抵押物和债权人之间的深层关联关系，从而实现降低风控成本的目的。

## 异常指标监控

交易如电子支付、移动支付发展的同时，也带来了更为严峻的交易安全问题。为防止违法金融行为带来的经济损失，通过使用异常检测算法在网络金融交易领域的应用，准确识别网络金融交易中存在的异常交易情况，及时阻止违法金融行为。通过图数据库，识别支付用户的设备信息、支付环境信息、转账信息、社交信息，检测可能出现的异常风险，提高支付的安全性。

## 违规团伙挖掘

很多平台上都有团伙作案的场景出现，如团伙性信用卡套现、羊毛党，甚至于团伙性的违法、犯罪性活动。团伙性作案的核心特征是，其成员之间具备某种关联关系，或行为之间具备某些相似特征。通过图结合机器学习的方式，提取团伙关系图谱之中的拓扑关系信息，以已知团伙信息作为样本，训练机器学习模型，针对目标节点进行分类，或其关系进行预测。

## 3.7.4. 产品架构

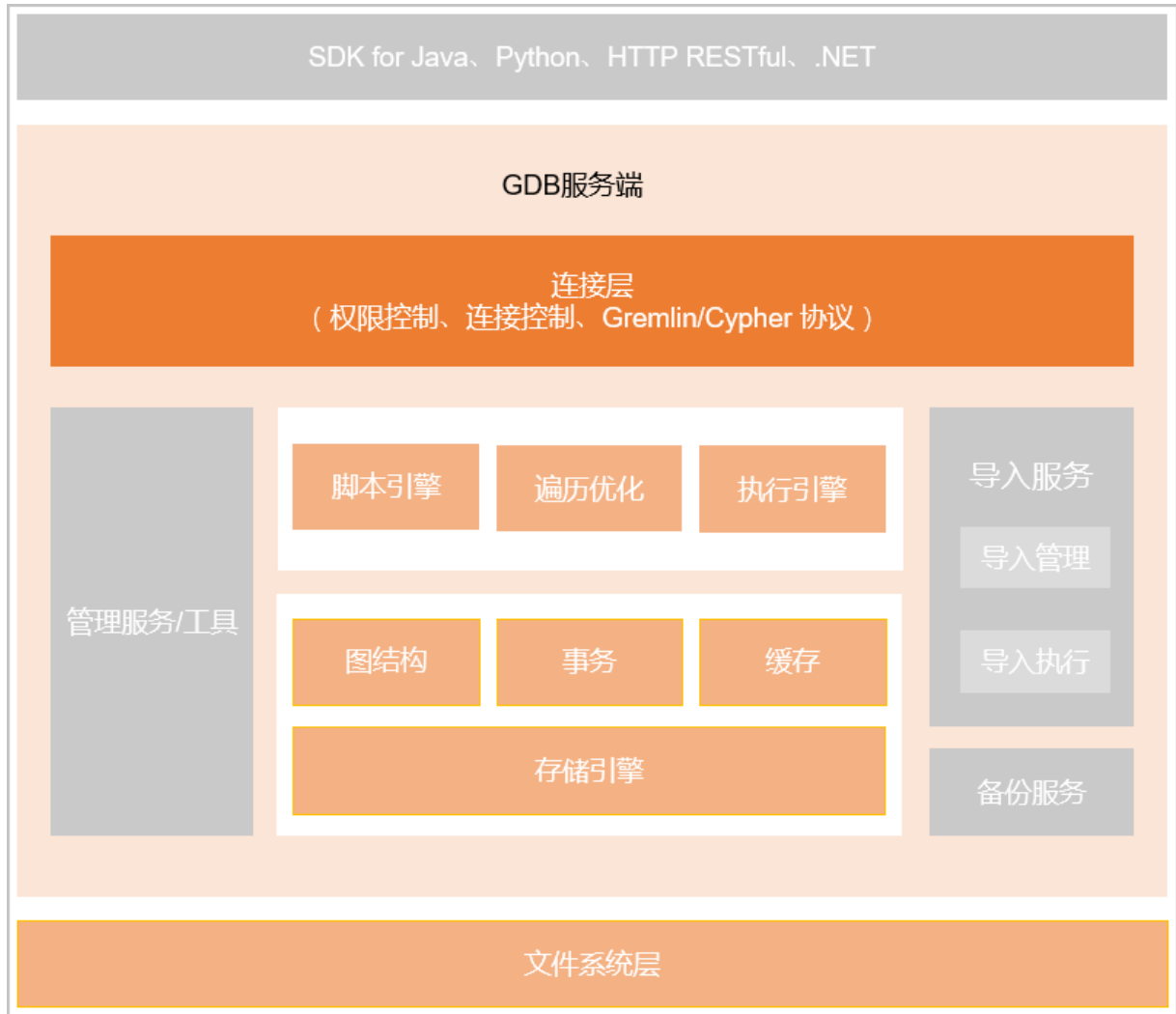
本文档主要介绍了图数据库GDB的产品架构。

图数据库GDB的产品架构从下到上可以分为四层：

- 存储层：实现图结构数据存储；
- 计算层：实现图核心逻辑，包括图层次优化、索引维护、查询优化、Object Cache、执行计划等核心功能；
- 连接层：实现权限认证、连接控制、Gremlin和Cypher连接协议；
- 应用层：提供应用层操作图数据库的SDK，包括Java、Python、Http Restful、.NET等方式。

图数据库GDB的产品架构如下图所示：

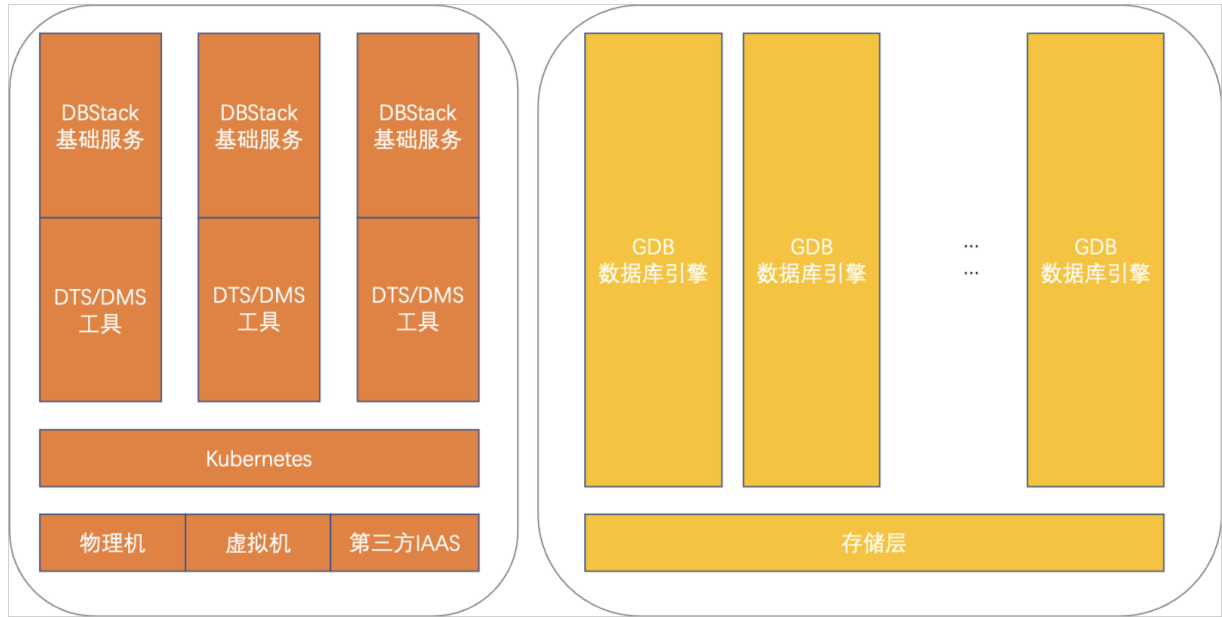




### 3.7.5. 部署架构

本文档主要介绍了图数据库GDB在DBStack中的部署架构。

图数据库GDB在DBStack中的部署架构如下图所示：



- 工具和基础服务共用一个Kubernetes集群，支持物理机、虚拟机及第三方IaaS平台进行部署。
- PolarDB O引擎在另一个Kubernetes集群，仅支持物理机部署，需要额外的存储设备。
- 两个Kubernetes集群采用统一的用户帐号体系，通过hosting网络进行打通。

### 3.7.6. 功能特性

本文档介绍了GDB支持的主要功能特性。

#### 图查询

- 支持对接DMS可视化控制台  
通过对接DMS可视化控制台，可以实现可视化查询分析。
- 优化超级顶点查询  
超级顶点是一些紧密连接的节点簇，处理不当将拖慢图数据库性能。
- 全文索引  
与ElasticSearch深度融合，提供分词检索功能。在图数据库GDB的统一入口中，输入相应的DSL语句，即可使用分词查询、模糊匹配、正则查询等全文检索功能，简易快捷。
- 支持Gremlin、Cypher  
能够兼容市面90%的图查询代码，方便客户迁移现有图数据库产品至阿里云，降低代码编写成本。
- 支持Schema free  
满足更灵活多变的架构调整需求。
- 支持事务  
支持ACID事务，满足关键业务应用需求。
- 权限管理  
支持针对点、边进行权限控制。
- 自动索引  
自动建立索引，优化查询效率的同时更易维护。

#### 图计算

图计算算子，集成中心度、路径查找、社群划分三大类图计算算法，满足常用的图计算需求，同时支持客户通过DSL编写自定义图算子。

### 接口

支持Java、Go、Python、.Net、NodeJS等开发SDK，保证开发对接的可行性。

### 系统维护

- 备份恢复
  - 当备库发生异常时，您可以执行备份恢复操作，保障主备高可用架构的持续性。
- 支持升降配
  - 可满足客户业务潮汐扩缩规格需求。

## 3.8. ADAM

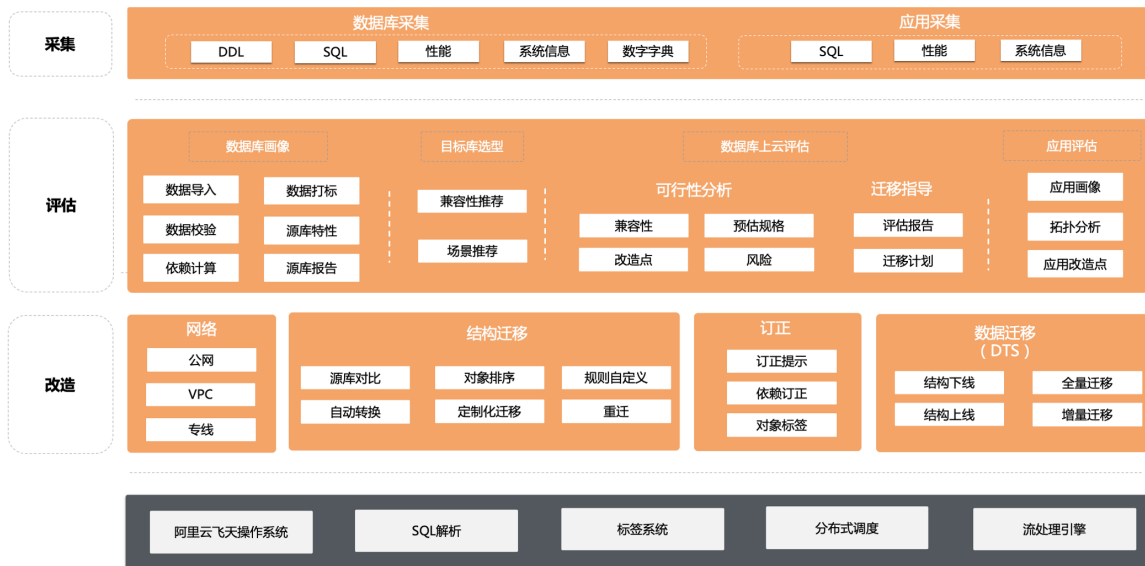
### 3.8.1. 产品概述

数据库和应用迁移（Advanced Database and Application Migration，简称ADAM）是阿里巴巴基于多年的Oracle数据库和应用架构梳理、架构选型、系统改造经验，研发的一套帮助企业客户进行传统数据库和应用架构改造、云化的数字化迁移解决方案产品。

ADAM为企业提供了从数据和应用架构梳理、架构选型、业务改造、数据迁移、系统割接及运维优化等一整套产品化服务能力，并结合丰富的合作伙伴生态，为企业用户提供专业数据和应用改造支持服务。

### 3.8.2. 产品架构

本章节介绍了ADAM的产品结构和功能特性。



### 采集器

采集器包括数据采集器和应用采集器两种：

- 数据采集器负责收集并汇总Oracle数据库信息，包含环境、对象、SQL、空间、性能和事务六方面信息，

全面覆盖数据库实际运行状况。同时，针对数据冗余、信息安全问题，对采集结果中SQL数据进行脱敏、去重、一致性校验等处理，保证采集结果的准确性。

- 应用采集器负责分析并收集应用框架与运行时信息，包含应用机器性能信息、应用代码中SQL信息、运行时SQL调用堆栈、应用与数据库间调用关系、应用与应用间调用关系五方面信息，全面覆盖应用实际运行状况。同时，也支持对结果数据进行脱敏、去重以及一致校验处理。

## 智能分析平台

智能分析平台的目标是根据从Oracle采集到的源库数据、从应用采集到的运行时数据，给出最适合的目标库方案。智能分析平台包括数据库评估、数据库改造迁移、应用评估改造三部分。

- 数据库评估
  - **源库画像分析：**源库画像通过对数据库采集器采集到的源库数据进行分析，给出多个维度的评估分析，包括对源库性能、容量、Oracle特性、外部连接(DBLink)和对象全景查询。其中对象全景分析对Oracle对象提供关联关系、特征标识等信息。
  - **目标库选型建议：**计算出每一个表组对应的云数据库规格，提供 CPU 核数，内存大小，磁盘大小等指标，满足用户的性能需求。根据评估分析结果给出目标方案的云资源成本，方便客户进行迁移到目标库的成本分析。
  - **目标库兼容评估服务：**针对不同的目标数据库及版本，ADAM提供对Oracle数据库对象和目标数据库之间的兼容性分析和改造的建议。兼容性级别分为完全兼容、不兼容两种。其中修改后兼容的方案由ADAM智能分析引擎自动化提供并写入迁移计划。
- 数据库改造迁移
  - **生成迁移计划：**根据评估分析的结果生成从Oracle库到目标数据库的迁移计划，使用迁移工具可以利用迁移计划将源库的Schema快速迁移到目标数据库，其中迁移计划包含的ADAM智能转换结果将保证最大兼容度。
  - **迁移计划校验：**校验迁移计划与源库中结构的一致性，并针对不一致的数据提示合并，这样保证了目标库与源库的数据结构最大一致性。
  - **结构迁移/订正：**根据迁移计划中的转换建议，生成目标库的DDL，连接到目标库进行数据库结构创建，并为转换流程提供实时更正功能，保证结构迁移中最大成功率。
  - **数据迁移服务：**自动连接DTS数据迁移服务。
- 应用评估改造
  - **应用画像：**应用画像通过对采集到的应用数据或应用的WAR包进行分析，提供多维度的评估分析，包括对应用的软件栈、系统信息、对象详情、SQL和调用栈等的评估分析。应用画像包括静态分析和动态分析，其中动态分析依赖对应用数据的采集，而静态分析可以直接对Java应用的WAR包进行分析，定位应用对数据库对象的访问点。
  - **应用评估：**应用评估主要针对数据库与应用迁移的改造过程，帮助用户快速梳理数据库异构迁移过程中的应用修改内容。

## 3.8.3. 产品优势

本章节介绍了ADAM具有产品优势点。

### 多年迁移经验

ADAM积累并提炼了阿里巴巴内部多年的成功迁移经验，尤其是从传统 IT 架构向互联网和云架构改造升级方面的成功实践。从迁移前的可行性分析、场景分析、目标数据库选型、兼容性评估、工作量评估、应用改造建议，以及最后新老系统割接、自动化测试等有着全流程的工具和实践经验。

### 智能分析

在迁移前，ADAM会采集您系统运行环境里的 Oracle 信息，然后对采集到的数据进行评估分析，最后得出分析报告集（包含专家意见）。报告集的内容包括目标数据库方案、源库不同类型对象的兼容度和不兼容原因、迁移后风险对象及SQL、应用相关的改造建议、迁移方案的成本等。此外，分析结果还包括目标数据库迁移计划。

## 数据库改造

依据智能分析阶段得到的目标数据库迁移计划，利用数据库改造迁移服务，客户可以自动化的将源库的 Schema 对象迁移到目标库，其中完全兼容和改造后兼容的对象在迁移过程中自动完成改造。对于不兼容对象，客户可以参考系统中提示来做改造并验证正确性。

### 应用改造定位

ADAM利用应用采集器采集到的应用数据（包括 SQL、调用栈等）来定位应用的SQL访问位置，同时结合数据库分析得到的对象信息及兼容改造建议，帮助客户快速定位应用的改造点，极大的提高应用改造的效率。ADAM 的数据库与应用的融合分析能力是 ADAM 的特色，将显著降低数据库迁移的门槛。

# 3.9. DTS

## 3.9.1. 产品概述

数据传输服务(Data Transmission Service，简称DTS) 支持关系型数据库、大数据(OLAP)等数据源间的数据传输。它是一种集数据迁移及数据实时同步于一体的数据传输服务。

相对于传统数据迁移/同步工具，DTS为您提供功能更丰富、传输性能更强、易用性更高且安全可靠的服务，帮助您简化复杂的数据交互工作，专注于上层的业务开发。

- DTS具备极高的链路稳定性和数据可靠性。
- 数据传输支持同/异构数据源之间的数据交互，提供数据迁移/同步交互功能。
- 支持节点的故障容灾，可实现链路的秒级恢复。
- 支持断点续传，可有效解决因硬件、网络等异常导致的传输中断。
- DTS支持RAM主子账号体系，用户可以使用子账号创建并管理DTS实例，提高企业安全性。

## 3.9.2. 产品优势

数据传输服务DTS支持关系型数据库、OLAP等多种数据源之间的数据传输。

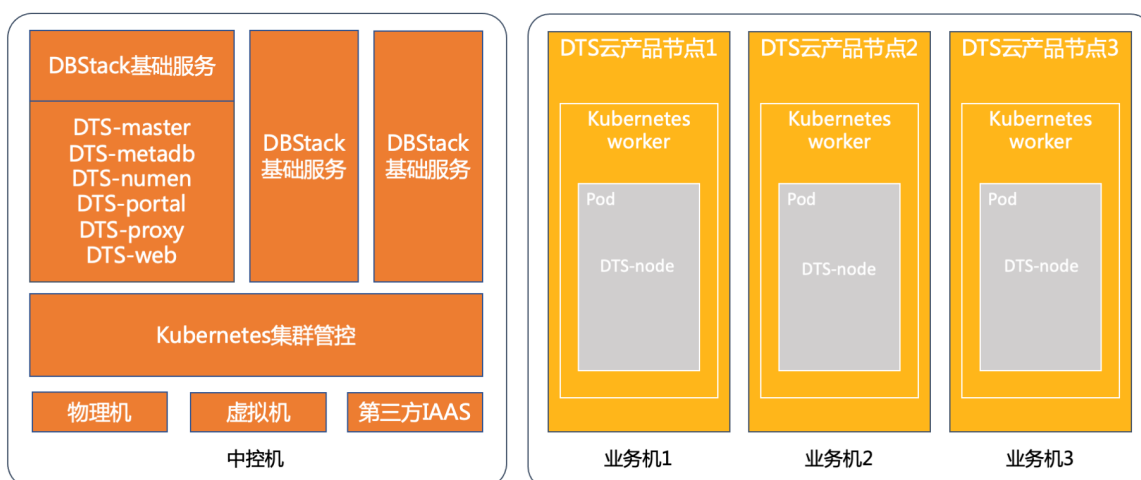
它提供了数据迁移及数据实时同步等多种数据传输方式。相对于第三方数据流工具，DTS提供丰富多样、高性能、高安全可靠的传输链路，同时它提供了诸多便利功能，极大方便了传输链路的创建及管理。

DTS的核心优势如下所示：

- 实时数据同步。
  - 支持9种数据库的日志解析，可基于非侵入式的日志解析技术，对源端数据库增量数据实时捕获，实现多种数据库实时同步，且对源数据库的性能几乎无影响。
  - 基于安全位点技术，可以实现数据的断点续传，故障恢复后，无需重新做全量的数据同步操作。
  - 基于并发解析技术，可以实现多表并发解析同步，全量性能峰值可达到100MB/s。
- 双向同步，实现业务多活。
  - 业界唯一一个云服务产品，支持MySQL数据库实时双向同步。
  - 通过在同城/异地构建多个单元(业务中心)并基于数据双向防循环秒级同步技术，可以帮助用户构建多活的数据中心。
  - 避免城市级别的故障灾害，影响服务的可持续性；解决远距离用户访问延迟高，用户体验差、单地域部署带来的基础设施的扩展限制等问题。

## 3.9.3. 部署架构

DTS在DBStack中的部署架构如下图所示。



DTS的相关服务包括：DTS-node（worker节点）、DTS-master（调度）、DTS-metadb（元数据库）、DTS-numen（运维平台）、DTS-portal（控制台）、DTS-proxy（代理）、DTS-web（任务配置页面）。

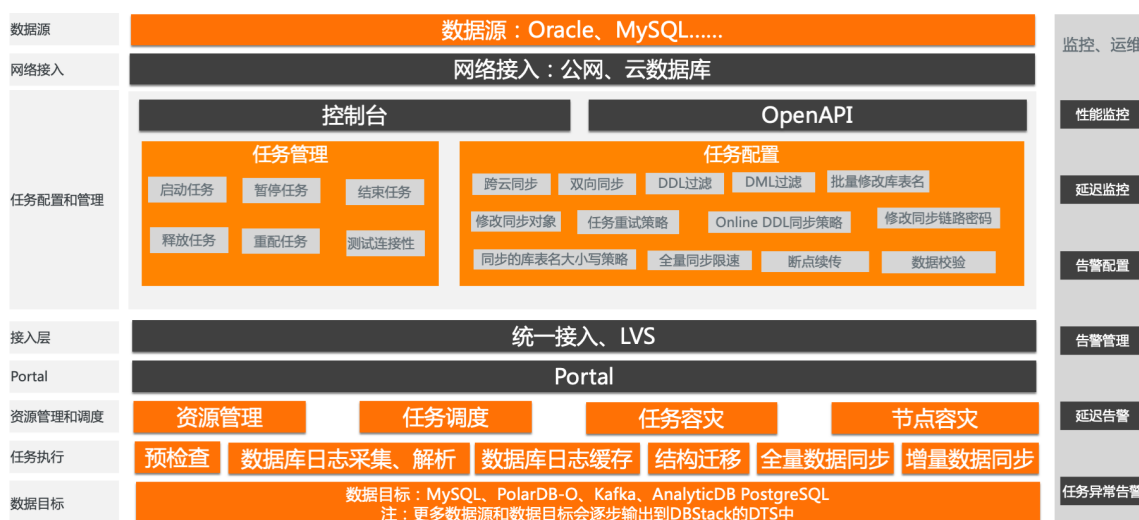
其中，DTS-node（worker节点）用于运行DTS任务，独占1台业务机部署（如图中所示），最低配置3个节点，可以运行20个数据迁移任务；也可以和其他云产品混合部署在中控机。

### 3.9.4. 功能特性

#### 3.9.4.1. 功能大图

本章节介绍了DTS的功能大图。

##### 功能大图



#### 3.9.4.2. 数据迁移

本章节介绍了DTS支持的数据迁移功能。

数据迁移功能旨在帮助用户方便、快速地实现各种数据源之间的数据迁移，适用于数据上云迁移、阿里云内部跨实例数据迁移、数据库拆分扩容等业务场景。数据传输服务提供的数据迁移功能能够支持同/异构数据源之间的数据迁移，同时提供了库表列三级映射、数据过滤多种ETL特性。

## 数据迁移类型

数据迁移支持**结构迁移**、**全量数据迁移**及**增量数据迁移**三种迁移类型。您只需配置迁移的源、目标实例及迁移对象即可，DTS会自动完成整个数据迁移过程，支持业务不停服迁移。

迁移类型	说明
结构迁移	DTS将源库中待迁移对象的结构定义迁移至目标库（例如表、视图、触发器、存储过程等）。对于异构数据库之间的结构迁移，DTS会根据源库和目标库的语法，将结构定义的语法进行转换，例如将Oracle中的number转换为MySQL中的decimal。
全量数据迁移	DTS将源库中待迁移对象的存量数据，全部迁移到目标库中。如果在配置数据迁移任务时，仅选择了结构迁移和全量数据迁移，那么在迁移过程中，源库的新增数据不会被迁移至目标库。 <span style="border: 1px solid #00aaff; padding: 2px;">? 说明 为保障数据一致性，迁移期间请勿在源库中写入新的数据。如需实现不停机迁移，您需要在配置数据迁移任务时，同时选择结构迁移、全量数据迁移和增量数据迁移。</span>
增量数据迁移	DTS会先在源库中实现静态快照，然后将快照数据迁移到目标库，最后再将迁移过程中源库产生的增量数据实时同步至目标库。 <span style="border: 1px solid #00aaff; padding: 2px;">? 说明 增量数据迁移会保持实时同步的状态，所以迁移任务不会自动结束，您需要手动结束迁移任务。</span>

## 支持的数据库和迁移类型

源数据库	目标数据库	迁移类型
<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul>
	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul>
	自建Oracle（非RAC实例、RAC或PDB实例） 9i、10g、11g、12c、18c、19c版本	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul>
	PolarDB-O 所有版本	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul>

自建Oracle（非RAC实例、RAC或PDB实例） 9i、10g、11g、12c、18c、19c版本	自建Oracle（非RAC实例、RAC或PDB实例） 9i、10g、11g、12c、18c、19c版本	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	<ul style="list-style-type: none"> <li>• RDS MySQL 所有版本</li> <li>• 自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	自建Kafka 0.10.1.0~2.7.0版本	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
PolarDB-O 所有版本	PolarDB-O 所有版本	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	Oracle（非RAC实例、RAC或PDB实例） 9i、10g、11g、12c、18c、19c版本	<ul style="list-style-type: none"> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
PolarDB-X ④ 说明 PolarDB-X需兼容MySQL 5.7版本。	PolarDB-X ④ 说明 PolarDB-X需兼容MySQL 5.7版本。	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	<ul style="list-style-type: none"> <li>• RDS MySQL 所有版本</li> <li>• 自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>
	自建Kafka 0.10.1.0~2.7.0版本	<ul style="list-style-type: none"> <li>• 库表结构迁移</li> <li>• 全量数据迁移</li> <li>• 增量数据迁移</li> </ul>



自建SQL Server	自建SQL Server	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><span style="color: #00aaff;">?</span> 说明 如源自建SQL Server为2005版本时，则不支持增量数据迁移。</p> </div>
DB2 for LUW 9.7~11.5版本	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul>
DB2 for i 7.3、7.4版本	<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> <li>增量数据迁移</li> </ul>
Teradata 17版本	云原生数据仓库AnalyticDB PostgreSQL版 4.3、6.0版本	<ul style="list-style-type: none"> <li>库表结构迁移</li> <li>全量数据迁移</li> </ul>

### 多种ETL特性

数据迁移具备多种ETL特性，如下表所示：

特性	说明
库表列三级对象名映射	可以实现对源跟目标实例的库名、表名或列名不同的两个对象之间进行数据迁移。
迁移数据过滤	客户可以对要迁移的表设置SQL条件过滤要迁移的数据，例如客户可以设置时间条件，只迁移最新的数据。

### 3.9.4.3. 数据同步

本章节介绍了DTS支持的数据同步功能。

数据同步功能旨在帮助客户实现数据源之间的数据实时同步。

#### 支持的数据库

源数据库	目标数据库	同步类型	同步拓扑
<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	库表结构同步 全量数据初始化 增量数据同步	单向同步 双向同步
	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	库表结构同步 全量数据同步 增量数据同步	单向同步
自建Oracle (非RAC实例、RAC或PDB实例) 9i、10g、11g、12c、18c、19c版本	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	库表结构同步 全量数据同步 增量数据同步	单向同步
PolarDB-X ? 说明 PolarDB-X需兼容MySQL 5.7版本。	PolarDB-X ? 说明 PolarDB-X需兼容MySQL 5.7版本。	库表结构同步 全量数据同步 增量数据同步	单向同步
	<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	库表结构同步 全量数据同步 增量数据同步	单向同步
	云原生数据仓库AnalyticDB PostgreSQL 4.3、6.0版本	库表结构同步 全量数据同步 增量数据同步	单向同步
	自建Kafka 0.10.1.0~2.7.0版本	库表结构同步 全量数据同步 增量数据同步	单向同步
自建SQL Server 2008 R2、2012、2016或2017版本	自建SQL Server 2008 R2、2012、2016或2017版本	库表结构同步 全量数据同步 增量数据同步	单向同步
DB2 for LUW 9.7~11.5版本	云原生数据仓库AnalyticDB PostgreSQL版 4.3、6.0版本	库表结构同步 全量数据同步 增量数据同步	单向同步

DB2 for i 7.3、7.4版本	<ul style="list-style-type: none"> <li>RDS MySQL 所有版本</li> <li>自建MySQL 5.1、5.5、5.6、5.7、8.0版本</li> </ul>	库表结构同步 全量数据同步 增量数据同步	单向同步
PolarDB O引擎集群	云原生数据仓库AnalyticDB PostgreSQL版 4.3、6.0版本	库表结构同步 全量数据同步 增量数据同步	单向同步
<ul style="list-style-type: none"> <li>云数据库Redis社区版实例（集群架构、标准架构或读写分离架构） 4.0、5.0、6.0版本</li> <li>云数据库Redis企业版实例（集群架构、标准架构或读写分离架构） 5.0版本</li> </ul>	<ul style="list-style-type: none"> <li>云数据库Redis社区版实例（集群架构、标准架构或读写分离架构） 4.0、5.0、6.0版本</li> <li>云数据库Redis企业版实例（集群架构、标准架构或读写分离架构） 5.0版本</li> </ul>	库表结构同步 全量数据同步 增量数据同步	单向同步

## 同步对象

- 数据同步的同步对象的选择粒度为：库、表、列。用户可以根据需要同步某几个表的数据。
- 数据同步支持库、表、列名映射，即用户可以进行两个不同库名的数据库之间的同步，或两个不同表名的表之间的数据同步。
- 数据同步支持列选择，即用户可以根据业务需求，只同步表中的某几列数据。

### 3.9.4.4. 数据一致性

DTS通过日志读取和回放模块以及数据校验、断点续传的功能，保障源库和目标库的数据一致性。

#### 日志读取和回放模块

DTS通过日志读取和回放模块，实现增量数据的一致性。

日志读取模块：在全量数据迁移或初始化前，DTS会从源实例读取原始数据，经过解析、过滤及标准格式化，最终将数据在本地持久化。

**说明** 日志读取模块通过数据库协议连接并读取源实例的增量日志。如果源数据库为RDS MySQL，那么数据抓取模块通过Binlog dump协议连接源库。

日志回放模块：日志回放模块从日志读取模块中请求增量数据，并根据同步对象的配置进行数据过滤，然后在保证事务时序性及事务一致性的前提下，将包含增量数据的日志记录同步到目标实例。

图 1. 迁移底层模块

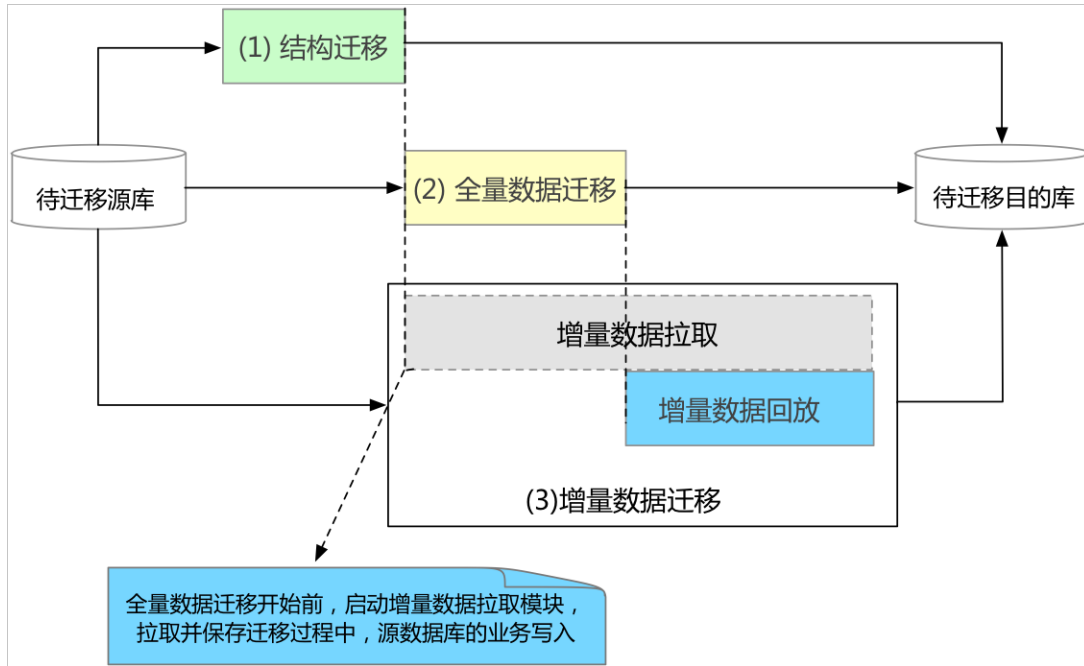
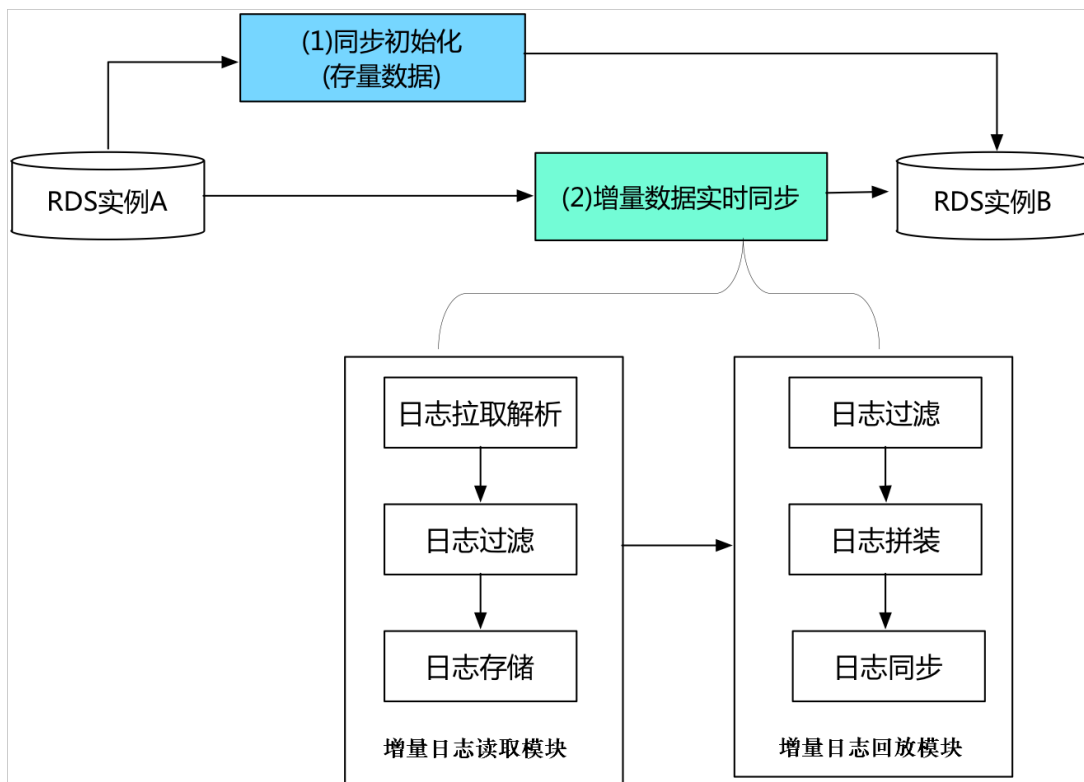


图 2. 同步底层模块



### 数据校验

数据迁移和同步完成后，为保障数据可用性，DTS在运维平台提供数据校验功能，用于确保源端迁移、同步到目标端数据的完整性、一致性。

通过数据校验可以检测出源端和目标端不一致的数据条数，单击不一致的数量可以查源端和目标端的校验对象以及两者不一致的数据，为不一致的数据订正提供支持。

图 3. 数据校验

共 168 个，其中运行 0 个，警告 0 个，失败 0 个，未启动 0 个，完成 168 个，46 个表不一致

请输入对象名称

Q 查询

对象名称	源库	目的库	总数 (条)	已完成 (条)	状态	Diff Row	耗时
agt_customer_agent_apply	ycy	newycy	5	5	完成	0	5秒
agt_grade	ycy	newycy	3	3	完成	0	5秒
agt_profit	ycy	newycy	5,397,833	5,397,833	完成	4575 !	1分钟10秒
agt_profit_rate	ycy	newycy	15	15	完成	0	1分钟10秒
agt_update_apply	ycy	newycy	2,411	2,411	完成	0	1分钟10秒
api_auth	ycy	newycy	481,877	481,877	完成	630 !	2分钟12秒
api_auth_acl	ycy	newycy	0	0	完成	0	2分钟12秒
api_doc_item	ycy	newycy	2,361	2,361	完成	0	2分钟12秒
api_doc_message	ycy	newycy	338	338	完成	0	2分钟12秒
api_doc_scheme	ycy	newycy	36	36	完成	0	2分钟12秒
api_doc_scheme_desc	ycy	newycy	0	0	完成	0	2分钟12秒
api_doc_scheme_service	ycy	newycy	339	339	完成	0	2分钟12秒

## 数据断点续传

在数据迁移和同步过程中，DTS会记录迁移和同步到目标端的位点信息，用于确保故障发生时数据的一致性。如遇网络故障，DTS会记录下故障发生时已迁移或同步到目标端的位点信息，当故障恢复时，DTS将继续从故障前的位点迁移或同步数据，确保数据的一致性。

# 3.10. DMS

## 3.10.1. 产品概述

数据管理（Data Management Service，简称DMS）是一种高效、安全、全面的数据库开发工作环境，提供多种数据库类型、多种环境统一的web数据库管理终端；可以为企业用户快速复制搭建与阿里集团同等安全、高效、规范的数据库DevOps研发流程解决方案。

## 3.10.2. 产品优势

DMS支持丰富的数据源、操作流程安全可控、细粒度的权限管控等功能，极大提升了数据管理的便捷性和安全性。

### 丰富的数据源支持

- 关系型数据库：
  - MySQL：云数据库RDS MySQL、云原生数据库PolarDB-X、其他云厂商MySQL、自建MySQL
  - SQL Server：云数据库RDS SQL Server、其他云厂商SQL Server、自建SQL Server
  - PostgreSQL：云数据库RDS PostgreSQL、其他云厂商PostgreSQL、自建PostgreSQL
  - 自建达梦数据库DamengDB
  - 自建Oracle
  - 云数据库OceanBase、自建OceanBase
- NoSQL数据库：Redis：云数据库Redis、其他云厂商Redis、自建Redis MongoDB：云数据库MongoDB、其他云厂商MongoDB、自建MongoDB 图数据库GDB（Graph Database）
- OLAP数据库：云原生数据仓库AnalyticDB MySQL云原生数据仓库AnalyticDB PostgreSQL

### 统一的操作入口且支持审计

- 管理员录入数据库实例后，数据库的查询、变更结构、变更数据等需求均可以在DMS平台内完成。
- 所有操作可按照人员、数据库、表、时间等多维度进行搜索审计。

### 细粒度的权限管控

普通用户不再接触数据库账号和密码，只需根据实际需要在DMS平台内申请目标库、表或字段的查询、导出、变更权限，权限到期后DMS会自动回收。

### 可定制的审批流程

每个数据库实例可定义满足业务实际需要的各模块审批流程，可同时满足效率、安全等维度的要求。例如：

- 测试环境轻管控，可减少流程或不设置流程。
- 生产环境重管控，可配置具体操作经过指定人员逐层审批后方可生效执行。

### 自定义表结构设计规范

可按照需求自定义MySQL表结构的设计规范，例如字段类型、索引类型、索引个数、字段名长度、表名长度、发布流程等相关规范。

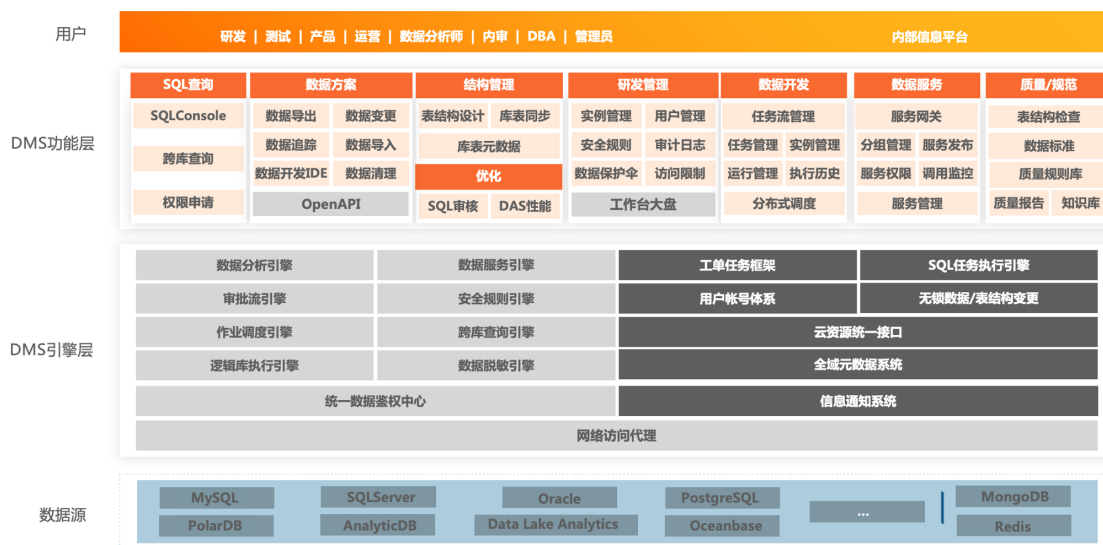
### 便捷的周期任务编排调度

可快速搭建各种数据库SQL任务节点编排、周期调度，实现历史数据转储、周期报表产出等便捷的数据价值挖掘。

## 3.10.3. 产品架构

本章节介绍了DMS的产品架构和功能大图。

### 产品大图



### 主要功能

- 提供研发从线下环境根据设计规范全自动进行表结构设计、上线前代码内使用到增删改查的SQL审核、结构按需生产发布到指定环境的完整数据库研发流程支持。
- 提供库、表、字段级的细粒度操作权限管控，所有用户通过产品产生的数据库操作可在线化、可溯源审计支持。
- 提供根据业务需要可灵活配置结构设计、数据变更、数据导出、权限申请等各个模块的操作规范与审批流程支持。
- 统一研发与数据库交互的入口，任何用户都不再直接接触数据库账号密码也不需要频繁切换数据库链接进

行管理。

- 提供在线任务编排功能，实现数据库SQL任务周期编排调度管理，轻松解决历史数据转储、周期报表分析产出等功能。

## 3.11. DAS

### 3.11.1. 产品概述

数据库是所有企业业务的基座，但数据库运维和管理常常受到如下多种挑战。

- 业务快速迭代，数据库故障频发  
企业内的研发、测试、运营、运维等人员每天都需要对数据库进行操作或者查询。但使用数据库的人员对数据库的了解程度参差不齐，导致数据库的稳定性不断受到如下挑战。
  - 在业务发布时产生了大量慢SQL。
  - 对业务短期高峰时的容量预估不足。
  - 表结构或者索引设计错误。
  - 未经审核的SQL或者表被发布到线上。
- 缺少数据支撑，问题根因排查困难  
数据库的问题排查和性能优化一直都是数据库领域的专业问题。即使是最专业的DBA在排查某些问题的时候，往往也会出现耗费了很长时间，但仍然无法定位到根因的情况。问题根因排查主要有如下难点。
  - 获取信息难。问题诊断和性能优化都需要依赖于大量的系统数据，甚至是长期的历史数据。只有基于完备的信息数据才能给出准确的解法。
  - 分析信息难。DBA通常需要有多年的经验才能在获取信息后给出准确的解法，同时也需要多样的场景才能比较全面的覆盖问题类型。但经验与场景往往难以沉淀于传承。
  - 优化手段难。找出问题和对应解法后，并不意味着问题就能马上解决，因为有些解法要求深入到数据库引擎层进行代码优化，短期内想要完成比较困难。
- 管理成本高  
随着云计算的普及，企业可以更加便捷地根据不同的业务类型使用不同的数据库，或将数据库部署在不同的环境中。多环境和多数据库管理的挑战也随之而来。
  - 精通多种数据库的专业数据库管理员是稀缺人才，招聘难。
  - 管理部署在多种环境的多种数据库的难度大。
  - 管理经验沉淀和传承的难度大。
- 数据库安全风险大  
随着数据价值的提升，企业的数据库面临着越来越多的内部或者外部的攻击，数据泄漏、数据丢失等问题层出不穷。全球重大数据泄漏事件几乎每月都有发生，小型泄漏事件更是不胜枚举。
  - 未授权或者不可预期或者错误的数据库访问和使用。
  - 数据泄漏。
  - 数据损坏。
  - 黑客攻击。
  - 软、硬件缺陷导致数据异常。
  - 误操作导致数据丢失。

### 我们的解决方案

数据库自治服务DAS (DatabaseAutonomyService) 是一种基于机器学习和专家经验实现数据库自感知、自修复、自优化、自运维及自安全的云服务，帮助用户避免数据库管理的复杂性及人工操作引发的服务故障，有效保障数据库服务的稳定、安全及高效。



该服务已经在阿里巴巴集团所有的数据库上验证了2年多，截止到2020年4月已取得如下成果：

- 自动优化了超过4000万的SQL。
- 自动回收了4PB的空间。
- 自动优化了20%的内存。

### 3.11.2. 产品优势

数据库自治服务DAS具备节约成本、提升稳定性、持续可用、安全高效等优点。

- **节约成本**
  - DAS提供统一监控功能、统一告警功能，您无需耗费人力和资源搭建性能监控平台和告警平台。
  - DAS提供统一的管理平台，您无需在多个管理平台上切换，帮助您提升工作效率、节省人力成本。
- **提升稳定性**
  - DAS提供丰富的数据库性能监控和告警功能，可以快速发现和定位数据库异常，提升数据库的稳定性。
  - 运维和管理一站式，无需多平台间切换，显著减少误操作概率。
- **持续可用**

DAS基于机器学习和专家经验实现数据库自感知、自修复、自优化、自运维及自安全的云服务，保障您的数据库持续可用。
- **安全高效**
  - DAS提供高危SQL识别、SQL注入检测、新增访问来源识别、敏感数据访问发现等服务，能够快速识别数据库是否存在异常访问、拖库等行为，有效保障数据库安全。
  - DAS采用无侵入式设计，不会侵入您的数据库环境，您也无需在数据库实例上安装客户端。
  - DAS采用安全的数据链路，数据库的信息利用KMS进行加密存储，加密压缩后才进行传输，保障数据安全。

### 3.11.3. 产品功能

本文主要介绍当前阶段DAS的各个功能模块。



## 统一管理

对多种类数据库进行集中管理，统一监控，能够为您节省一半以上的管理成本，显著减少因人为误操作导致故障的概率。

- 统一监控

通过DAS平台，即可查看所有环境、所有集群、所有实例的性能趋势情况和实时性能情况。

- 低成本

用户无需耗费人力开发和部署采集、计算、存储程序，直接使用DAS即可监控数据库。

- 指标丰富

支持数据库各项关键指标的采集、计算和展示。

- 细粒度的监控

您可以按业务需要设置细粒度的监控，最小支持秒级监控，帮助您快速发现异常。

- 统一告警

支持自定义告警规则的触发条件和告警信息的发送人等。

- 默认告警模版

基于阿里巴巴的数据库运维经验，为各种数据库引擎定义了默认的告警模版，方便您直接使用。

- 灵活配置

支持各种告警规则、告警模版、告警联系人、告警联系组的灵活配置，用户可以为企内不同的使用者定义不同的告警模版。

- 异常发现

自动发现没有定义告警的数据库实例，避免出现因为告警信息发送不及时，导致业务受损的情况。

## 自治服务（智能诊断和优化）

DAS基于机器学习和细粒度的监控数据，能够提供7\*24小时的异常检测，支持自动SQL限流、异常快照、自动SQL诊断和优化、存储空间自动扩展、计算资源自动扩展等服务。DAS还能通过异常发现、根因分析、进行止损或优化、效果跟踪、回滚或沉淀知识库等手段，实现诊断流程的闭环和优化效果的可量化，确保数据库的持续可用。我们将数据库的自动驾驶能力分为如下5个层级：

能力层级	描述	决策主体	关键能力
Level-0	全人工	人	
Level-1	辅助工具、看板	人	负载信息采集、监控、告警、脚本以及工具支持。
Level-2	输出诊断和优化建议	人和DAS	异常检测、SQL优化建议、容量评估、健康诊断等。
Level-3	分场景全自动化	DAS和人	自动异常修复、自动SQL优化、自动参数配置、自动弹性伸缩等等，少量场景仍然由人做主要决策或动作，系统辅助。
Level-4	自动驾驶	DAS	全部场景实现端到端决策，进入自动驾驶模式。

自治服务主要包含如下功能：

- SQL诊断和优化

- 慢SQL分析
- 空间分析
- 性能趋势
- 会话管理
- 全量SQL分析
- 诊断报告
- 7\*24小时异常检测
- 自动SQL限流
- 自动SQL优化
- 自动SQL Review和优化
- 容量评估和规格推荐
- 弹性伸缩

## 企业级数据库服务

### • Dashboard

DAS总结阿里巴巴数据库团队多年的数据库运维和管理经验，提供多种监控场景，支持查看跨实例、跨集群、跨环境、跨功能模块的护航大盘、实例大盘等。

### • 多环境、多集群管理

DAS满足企业管理多套环境多套集群的需求，支持环境级别、集群级别的性能监控指标的聚合和下钻，贴近企业级管理视角。

### • 巡检评分

DAS支持巡检评分，自动对用户接入DAS的所有数据库实例进行巡检，包括基础巡检以及SQL、容量、性能、安全巡检等，并给出健康评分，帮助用户一目了然地确认数据库运行情况。

## 数据库安全审计

DAS提供高危SQL识别、SQL注入检测、新增访问来源识别、敏感数据访问发现等服务，具备实时检测、全量审计、快速识别数据库异常访问和拖库等行为的能力，有效保障数据库安全。数据库安全审计包含如下功能：

- SQL注入识别
- 高危SQL识别
- 新增访问来源识别

## 3.11.4. 应用场景

数据库自治服务DAS (Database Autonomy Service) 支持统一管理、批量管理、智能诊断、安全审计等功能，帮助您实现多种典型应用场景。

### • 典型使用场景——统一管理

DAS支持多种引擎的数据库统一管理，零成本构建监控和告警平台。

#### ◦ 统一管理

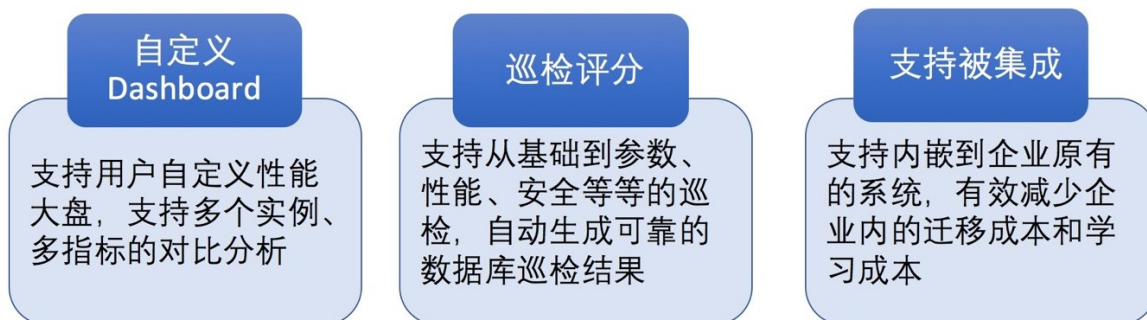
3分钟即可接入阿里云上数据库，实现统一监控、统一告警。

#### ◦ 支持多类型的数据库

DAS支持RDS MySQL、云数据库Redis版和云数据库MongoDB版。

### • 典型使用场景——批量管理

DAS提供企业级数据库管理服务，贴近业务管理视角，支持全局、应用组、实例的多种管理维度，并且提供了自定义大盘、批量管理、巡检等企业级能力，同时支持与企业原有数据库管理系统集成。



• 典型使用场景——智能诊断

- DAS基于机器学习和细粒度的监控数据，实现7\*24小时的异常检测，相比传统的基于阈值的告警方式，能够更为及时的发现数据库的异常变化，并自动进行诊断、止损或者优化，保证数据库的稳定。
- DAS通过异常发现、根因分析、进行止损或优化、效果跟踪、回滚或沉淀知识库等手段，实现诊断流程的闭环和优化效果的可量化，确保数据库持续可用。

• 典型使用场景——安全审计

DAS提供高危SQL识别、SQL注入检测、新增访问来源识别、敏感数据访问发现等服务，能够快速识别数据库受否存在异常访问、拖库等行为，有效保障数据库安全。

