

SOFAStack

数据访问代理
产品简介

产品版本：AntStack Plus 1.11.0

文档版本：20220929



法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，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。

商标声明

 蚂蚁集团
ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

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。蚂蚁集团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，并在蚂蚁集团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蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
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。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 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.概述	05
2.功能特性	06
3.产品优势	07
4.应用场景	08
5.基本概念	09
6.基本原理	10
6.1. 分库分表	10
6.2. SQL 路由	10
6.3. 读写分离	12

1. 概述

数据访问代理是蚂蚁集团自主研发的金融级分布式数据库中间件，用于解决海量请求下数据库访问的瓶颈及数据库的容灾问题，提供水平拆分、平滑扩缩容、读写分离的在线分布式数据库服务。十年来专注于为海量数据库访问提供低消耗、高性能、高可用的轻量级解决方案，确保在高并发、数据库异常的情况下依然非常稳定与可靠。

数据访问代理兼容 MySQL 协议和语法，支持分库分表、平滑扩容、服务升降配、透明读写分离和分布式事务等特性，具备分布式数据库全生命周期的运维管控能力。

数据访问代理主要应用场景在大规模在线数据操作上，通过贴合业务的拆分方式，将操作效率提升到极致，有效满足用户在线业务对关系型数据库要求。

数据访问代理是一个分布式数据库系统，也是一个实现了 MySQL 协议的服务器。前端用户可以把它看作是一个数据库代理，用 MySQL 客户端工具和命令行访问，而其后端可以用 MySQL 原生协议与多个 RDS 或者 OceanBase 服务器通信。它解决了目前传统关系型数据库难以扩展、不可切分的问题，可以避免单机（单库）的性能缺陷，克服了数据存储和业务规模迅速增长情况下的数据瓶颈。

2. 功能特性

数据访问代理兼容 MySQL 协议和语法，支持分库分表、平滑扩容、服务升降配、透明读写分离和分布式事务等特性，具备分布式数据库全生命周期的运维管控能力。

分库分表

支持 RDS、OceanBase、MySQL 的分库分表。在创建分布式数据库后，只需选择拆分键，数据访问代理就可以按照拆分键生成拆分规则，实现数据水平拆分。

透明读写分离

通过使用数据访问代理只读实例或者 MySQL 备机实现读写分离，帮助应用解决事务、只读实例或者备机失效、指定主备访问等细节问题。对应用无侵入，在数据访问代理控制台即可完成读写分离相关操作。

数据存储平滑扩容

当出现数据存储容量和访问量瓶颈时，数据访问代理支持在线存储容量扩展，扩容无需应用改造，扩容进度支持可视化跟踪。

服务升降配

数据访问代理实例可以通过改变资源数量实现服务能力的弹性扩展。

全局唯一数字序列

数据访问代理支持分布式全局唯一且有序递增的数字序列。满足业务在使用分布式数据库下对主键或者唯一键以及特定场景的需求。

数据库账号权限体系

数据访问代理支持类单机 MySQL 账号和权限体系，确保不同角色使用的账号操作安全。

分布式事务

数据访问代理结合中间件分布式事务套件，可以支持分布式事务，保证分布式数据库数据一致性。

3. 产品优势

数据访问代理的主要优势如下：

- **分布式**：数据读写存储集群化，不受单机限制，业务使用无连接数限制。
- **弹性**：数据服务可升降配，数据存储扁平化 scale-up（纵向扩展）和 scale-out（横向扩展），读写分离线性提升读能力。
- **高性能**：分库分表经典方案让操作聚焦少量数据，多种拆分方式适应数据特点，并具备特定 SQL 并行执行能力，进一步提升执行效率。
- **安全**：完整的类单机 MySQL 账号体系，提供具备授权鉴权的 OpenAPI 方便集成能力到业务管控中，产品服务支持体系化。
- **简单易用**：兼容 MySQL 协议和大部分 MySQL SQL 语法，无业务侵入式使用读写分离，支持全面的运维和监控能力。
- **成熟度高**：基于蚂蚁集团内部多年的金融级数据容灾场景，十年来专注于为海量数据访问提供低消耗、高性能、高可用的轻量级解决方案，确保在高并发、数据库异常的情况下依然非常稳定与可靠。

4. 应用场景

海量数据读写

随着业务的快速增长，数据量不断的增大，就会出现单表/单库数据量太大、单台数据库服务器压力很大、读写速度遇到瓶颈等一系列问题。尽管可以通过增大数据库实例的物理配置得到一定程度的缓解，但无法根本解决数据库单机瓶颈。数据访问代理提供灵活的数据拆分机制，代码侵入性低，可以非常方便地实现数据的水平拆分与扩容，从而从根本上解决数据库单机瓶颈的问题。

金融级数据容灾

基于蚂蚁集团内部多年的金融级数据容灾场景，数据访问代理针对不同业务场景提供了多种机房级数据容灾解决方案，能够保障数据的稳定性与业务的连续性，为企业发展保驾护航。

数据库流量分配

当系统提供了大量查询服务，且稳定性要求较高，单数据库实例进行读写已经不能满足业务层面的要求。数据访问代理提供了基于规则的流量分发机制，通过部署多个数据库实例（如一写多读）的方式来满足业务需求。

5. 基本概念

中文	英文	释义
实例	instance	实例是数据访问代理的集群，相当于一个逻辑的物理数据库实例。实例基于 MySQL 标准接口，作为统一的 SQL 请求入口，所有 SQL 都需要请求给实例进行处理，实例会负责将访问请求发到正确的物理数据库上。
数据库	database	表示数据访问代理的逻辑数据库。以这个作为访问的入口，通过分库分表、读写分离等规则可以请求到后端真实的物理数据库。
数据表	data table	表示数据访问代理的逻辑表，应用访问数据访问代理时 SQL 中的表即数据表。一个逻辑表会对应多个物理表，数据访问代理在路由时，会将逻辑表名替换成物理表名。
逻辑表	logic table	同数据表。
物理数据库	physical database	真实的物理数据库。可能是 RDS (MySQL)、OceanBase、Oracle 等。目前支持 RDS (MySQL) 和 OceanBase 两种物理数据库。
物理数据源	physical data source	同物理数据库。
物理表	physical table	物理库内存放的真实表。
数据库分片	database sharding	表示分库分表中的数据库分库。分库是一个逻辑上的概念，物理上可能是一个物理数据库代表一个“分库”，也可能是多个物理数据库组成一个“分库”，在数据访问代理里面统一概念称分片。
分片	shard	同数据库分片。
数据节点	data node	代理层后端的用户物理数据节点实例，类型可以是 MySQL (RDS) 或者 OceanBase。

6. 基本原理

6.1. 分库分表

数据库访问代理在后端将数据量较大的数据表水平拆分到各个 RDS 数据库中，后端的这些 RDS 数据库被称为分库，分库中的表被称为分表。

拆分后，每个分库负责一份数据的读写操作，从而有效的分散了整体访问压力。在系统扩容时，只需要水平增加分库的数量，并且迁移相关数据，就可以提高数据库访问代理系统的总体容量。

数据拆分

数据库访问代理支持库级拆分，表级拆分和分库分表拆分，通过数据库访问代理 DDL 语句指定，具体操作参见 [DDL 语法](#)。

数据库访问代理根据指定拆分键的值，采用特定的算法进行计算，然后根据计算结果将数据存储到对应的分库/分表中。拆分键即分库/分表字段，因此分为分库键和分表键。目前，拆分键只支持单个字段。

- 分库键：数据库访问代理根据分库键的值将数据水平拆分到后端的各个 RDS 分库里。键值相同的数据一定位于同一个 RDS 数据库。
- 分表键：每一张逻辑表都可以定义自己的分表键，键值相同的数据一定位于同一个 RDS 数据表。



说明

在执行带有 WHERE 条件的 UPDATE、DELETE、SELECT 语句时，如果 SQL 语句中没有使用拆分键，或者虽然指定了拆分键但是范围太广，会导致 SQL 语句被分发到所有分库上执行（即全表扫描），且执行结果会在数据库访问代理中进行合并。全表扫描响应较慢，在高并发业务场景中应尽量避免使用。

6.2. SQL 路由

在分库分表模式下，数据库访问代理会根据拆分键（即拆分字段）以及 SQL 语义把 SQL 语句分发到底层中各个存储数据的分表进行执行。执行结束后，数据库访问代理会将各个分表获取的数据合并，然后返回给用户。本文介绍在分库分表场景中数据库访问代理执行 SQL 语句时的路由原理以及数据合并。

有关数据库访问代理的数据拆分原理，请参考文档 [分库分表](#)。

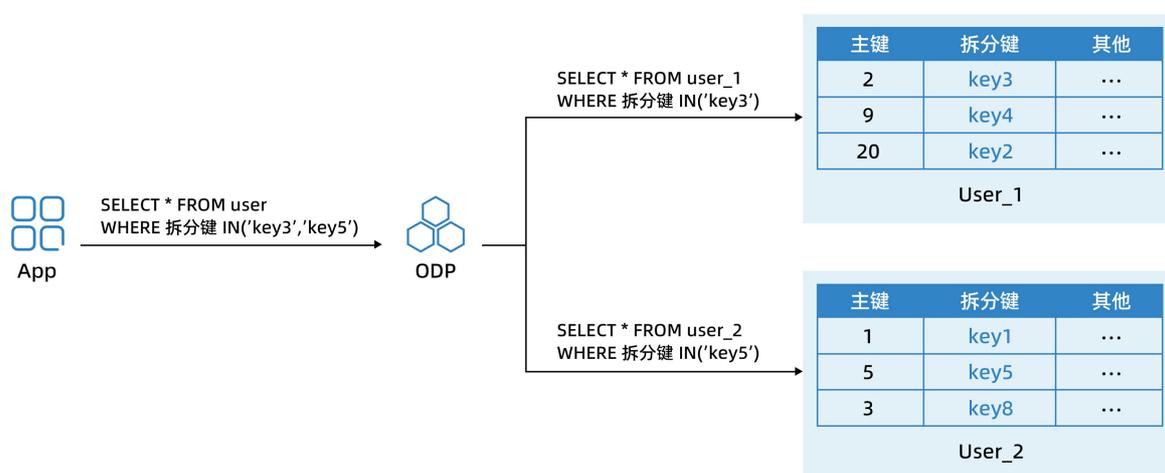
拆分键

分库分表过程中，数据库访问代理按照指定的拆分键，采用特定的算法进行计算，然后根据计算结果将数据存储到对应的分表中。拆分键是数据库访问代理中数据分布和 SQL 路由的凭证。

主键	拆分键	其他
1	key1	...
3	key2	...
4	key1	...
5	key1	...
9	key2	...
11	key1	...
20	key2	...

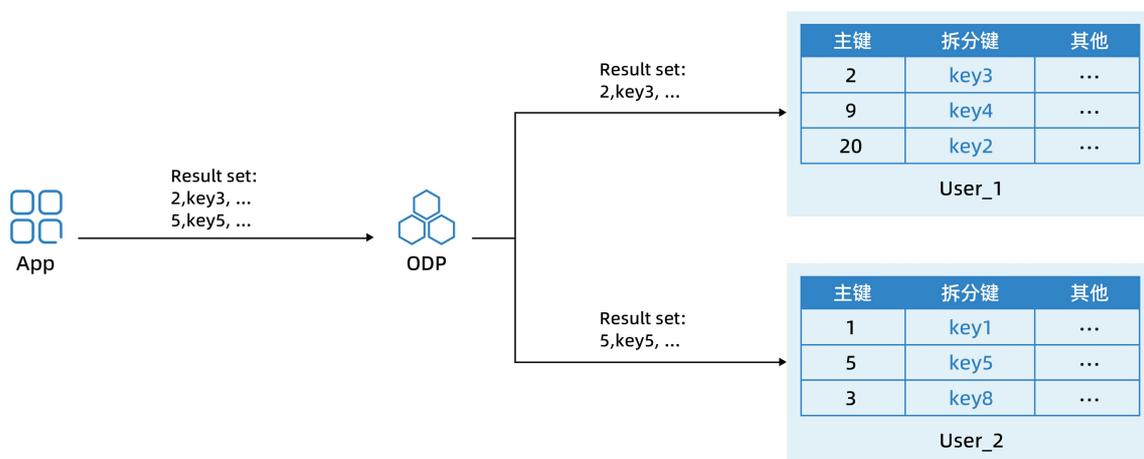
SQL 路由

当用户发起执行 SQL 语句的请求时，数据访问代理会理解 SQL 语句的含义，然后按照拆分键的值和执行策略将 SQL 语句路由到对应分区进行执行，如下图所示：



数据合并

如果一个 SQL 语句被路由到多个分表执行，数据访问代理会将各个分表返回的数据按照原始 SQL 语义进行合并，并将最终结果返回给用户。



6.3. 读写分离

在主实例的读请求较多、读压力较大的时候，可以通过数据访问代理读写分离功能对读流量进行分流，减轻 RDS 主实例的读压力。

数据访问代理的读写分离功能是对应用透明的设计。在不修改任何应用代码的情况下，只需要在数据访问代理控制台中调整读权重，即可将读流量按配置的比例在主 RDS 实例与多个 RDS 只读实例之间进行分流；写流量则全部到主实例，不做分流。

设置读写分离后，主 RDS 实例读取过程是强读，即实时强一致读，而只读实例上的数据是从主实例上异步复制的，存在毫秒级的延迟，因此只读 RDS 实例读取过程是弱读，属于非强一致性读。在金融级业务场景下，当需要实时、强一致读时，可以通过数据访问代理控制台的 [数据库设置](#) 页面来关闭读写分离功能，这时数据访问代理仅能通过强读的方式读取主 RDS 实例，保证访问该库的 SQL 语句只在主 RDS 实例上执行。

