

DTCC 2020 | 阿里云张鑫：阿里云云原生异地多活解决方案

原创

阿里云开发者 于 2021-01-12 17:22:55 发布 1126 收藏 1

文章标签：[SQL](#) [数据采集](#) [容灾](#) [关系型数据库](#) [OLAP](#) [OLTP](#) [数据库](#) [数据中心](#) [数据库管理](#) [Cloud Native](#)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/alitech2017/article/details/112538776>

版权

简介：异地多活，顾名思义就是分布在异地多个站点同时对外提供服务，与传统灾备最主要的区别是“多活”里所有站点都是同时在对外提供服务的。在业务不断复杂化和容灾要求不断严格化的今天，如何实现云原生的异地多活解决方案，成为了中大型企业不得不面对的挑战。在第十一届中国数据库技术大会（DTCC2020）上，阿里云高级数据库专家张鑫就为大家分享了阿里云云原生异地多活解决方案。

摘要：异地多活，顾名思义就是分布在异地多个站点同时对外提供服务，与传统灾备最主要的区别是“多活”里所有站点都是同时在对外提供服务的。在业务不断复杂化和容灾要求不断严格化的今天，如何实现云原生的异地多活解决方案，成为了中大型企业不得不面对的挑战。在第十一届中国数据库技术大会（DTCC2020）上，阿里云高级数据库专家张鑫就为大家分享了阿里云云原生异地多活解决方案。



本文内容根据演讲录音及PPT整理而成。

嘉宾介绍：

张鑫（花名：六金），阿里云高级数据库专家，之前主要作为DBA支持阿里巴巴内部包括交易、广告等在内的核心系统，近两年转战专有云市场，面向大型政企客户提供数据库解决方案。



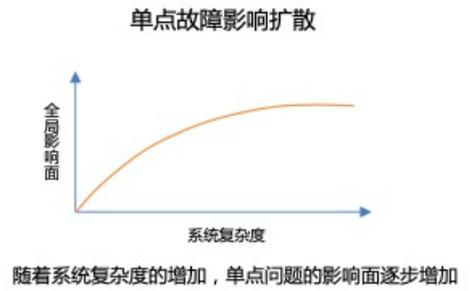
本次分享将主要分为三个方面：

1. 容灾架构分析
2. 阿里云异地多活解决方案
3. 异地多活客户案例

一、容灾架构分析

容灾必要性

容灾必要性



容灾能力
 成为企业信息化建设的**必选项**

异地多活本身是从容灾出发的，因此首先介绍一下容灾的必要性。生产系统可能会遇到三类故障，第一个是主机级故障，如单点负载过高、数据损坏等；第二类是机房级故障，如供电故障、机房网络故障等；第三类是地域级故障，如自然灾害等。对于上述三类故障而言，显然是地域级故障影响面最大，但发生概率最低，但对于主机级故障而言，却并不一定发生概率低且影响面小。阿里巴巴对于自身多年来的故障类型做了梳理，发现随着现在业务系统复杂度的增加，单点故障也可能会造成全局影响，而且当复杂度达到一定程度时，如果发生这种单点故障，排查和恢复都会非常困难，因此容灾能力成为了企业信息化建设的必选项。

容灾行业分析



容灾行业分析

全球容灾市场总额2017年约**71.3亿美元**，预计2022年上升到**115.9亿美元**

客户群体广

政府

金融

能源

互联网

通信

...

10万+
企业用户

40万+
数据库实例

- 《国家信息化领导小组关于加强信息安全保障工作的意见》中共中央办公厅、国务院办公厅（中办发[2003] 27号）；
- 《国家网络与信息安全协调小组关于做好重要信息系统灾难备份工作的通知》（信安通[2004]11号）；
- 《河北省信息安全战略》（冀办发[2007]21号）；
- 《河北省关于加强信息安全保障工作的实施意见》（冀办发[2004]12号）；
- 《河北省网络与信息安全“十二五”规划》；
- 《加强全省社保资金管理信息系统建设 推进信息共享工作方案》（办字[2012] 86号）；
- 《国家电子政务总体框架》（国信[2006]2号）；
- 《2006-2020国家信息化发展战略》（中办发[2006]11号）；
- 《关于转发〈国家信息化领导小组关于我国电子政务建设指导意见〉的通知》中共中央办公厅、国务院办公厅（中办发[2002] 17号）
- 《信息系统容灾恢复规范》（GB/T 20988-2007）

合规严格

DTCC
2020

北京国际会议中心 2020/12/21-12/23

IT168.com

ChinaUnix

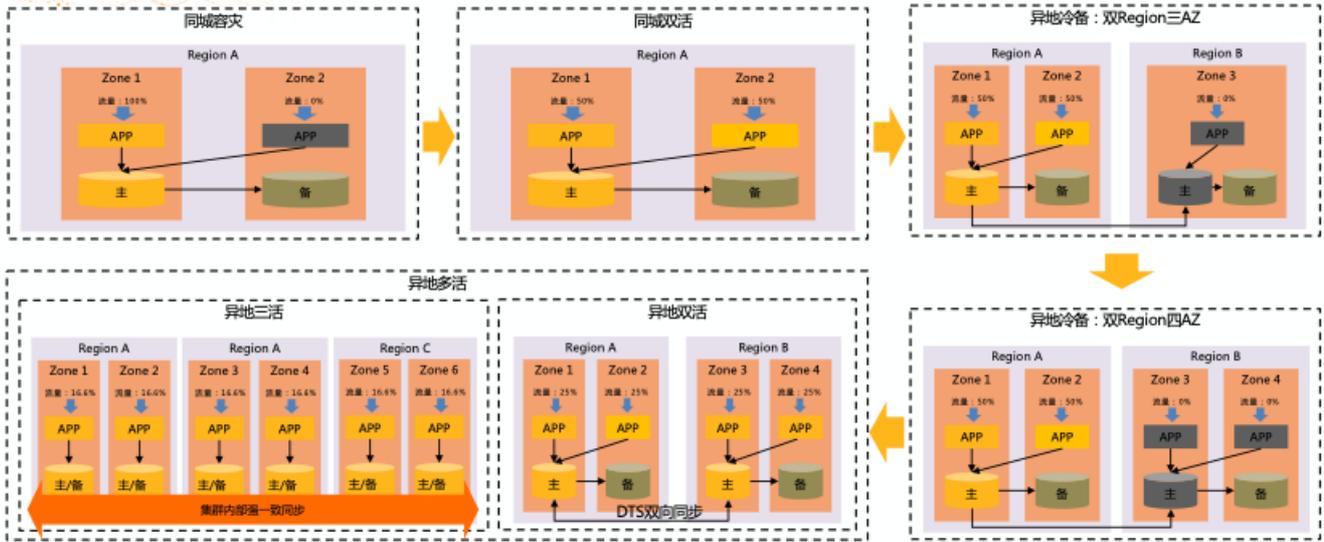
ITPUB

从行业分析来看，容灾的市场还是比较可观的。根据权威报告预测：在2020年全球容灾市场份额将达到115.9亿美元，并且客户群体非常广泛，比如政府、金融、能源、互联网、通信等，基本上只要有信息化系统就有容灾需求。阿里云目前拥有十万家企业用户和四十万个数据库实例，这些都需要容灾能力保障。而在国家层面，也具有严格的合规要求，尤其是现在大型的政企客户都需参照《信息系统容灾恢复规范》GB/T 20988进行容灾建设。

容灾架构演进

容灾架构演进

➢ Region : 地域级别
 ➢ Zone : 机房级别



容灾架构的演进主要分成几个阶段。同城容灾最为简单，即在同一个地域内有一个IDC并部署了业务，容灾时再部署一个机房备份系统和数据库，在中间实现异步或者同步的数据同步，业务流量集中在一边，另外一边只做灾备。后来逐渐演进出了同城双活，其借用了同城内两个数据中心地理距离比较近，网络延迟较短的优势，可以将业务部署到两端，因为物理距离较短，延迟等问题都可以接受。再往后就是异地双活，即两点三中心以及其衍生出的两地四中心等，主要就是在同城双活的基础之上再增加一个灾备中心，这个灾备中心常态下是不接收流量的，只有发生地域级故障时才会切换。

传统的容灾方案

传统的容灾方案



重新梳理一下传统的容灾方案，对于同城容灾或者同城双活而言，优势在于部署简单，并且接入成本非常低；缺点在于仅提供同城保护，在GB/T 20988中只能达到1级能力，因此对于大型客户而言，无法选择该方案。对于异地冷备而言，优势同样在于部署简单，对业务侵入比较少，并且异地部署的灾备能力相对而言会高一些，能够达到2到5级；缺点在于冷备单元冗余成本较高，造成一定的资源浪费，此外因为灾备单元常年不接流量，因此真正发生故障的时候切换是否可用是一个未知数。对于两地三中心而言，其实就是同城双活和异地冷备两种方案的结合，其优势就是上述两个方案的优势，缺点则是冷备中心成本浪费和地域级故障发生时不敢进行切换。

二、阿里云异地多活解决方案

阿里云异地多活架构



阿里云异地多活架构



异地多活的本质：异地多活本质上是通过对业务做自顶向下的流量隔离来实现的

接入层

- 实现入口流程的识别与纠错
- 按照DNS权重分配流量
- 自定义的分流规则
- 纠错流量

服务层

- 部署客户的应用系统，对业务服务进行划分，不同类型服务采用不同纠错策略
- 单元化服务
- 中心化服务
- 普通服务

数据层

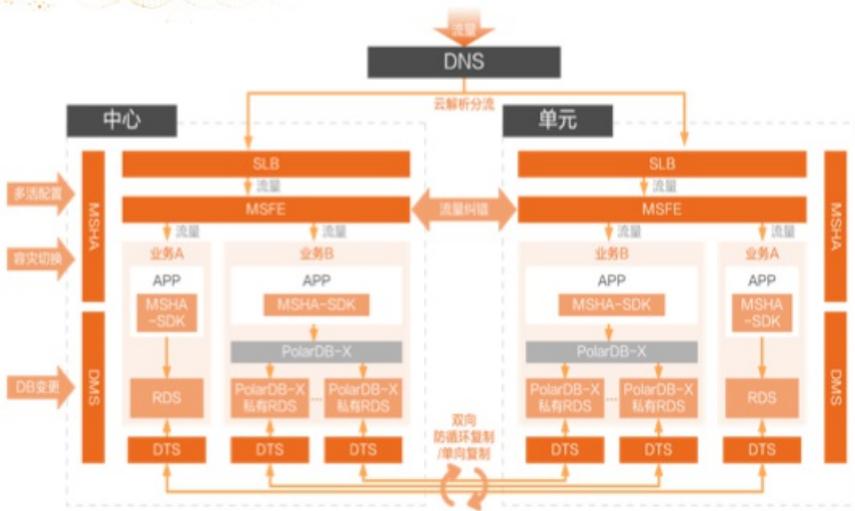
- 数据存储和数据实时同步，并在切流时进行数据保护
- 跨域实时同步
- 防循环
- 数据质量保障

如上图所示的是阿里云异地多活整体架构。实际上，异地多活的本质是通过对业务做自顶向下的流量隔离来实现的。阿里云将整个异地多活架构分为三层，第一层是接入层，实现异地双活首先需要为业务制定一个分流策略，如按照地域或用户维度分配流量，一旦定义好分流策略，即可在接入层实现流量拆分，属于本单元的流量可以继续向下透传执行，如果不属于则会将其转入正确的单元。第二层是服务层，就是对外提供服务的业务系统，针对于提供能力的不同划分为单元化服务、中心化服务和普通服务三种类型。第三层是数据层，这一层所需要解决的是数据库所需要具备的双向跨域同步能力、防循环能力，并且需要保障切流时的数据质量。

阿里云针对OLTP和OLAP两种业务场景对于多活架构方案进行了细化，接下来逐个介绍。

OLTP业务多活架构

OLTP业务多活架构



围绕TP数据库RDS、POLARDB-X构建：

多活配置 (MSHA)

- 通过MSHA进行一站式多活配置

- 同步对象
- 分流策略
- 安全策略

多活流量控制 (MSFE)

- 根据既定规则通过MSFE进行分流

- 流量识别
- 流量分发
- 流量校正

多活数据同步 (DTS)

- 通过DTS实现数据单向/双向复制

- 防循环
- 网络优化
- 切流联动

多活容灾切换 (MSHA)

- 通过MSHA方便进行容灾切换

- 状态检查
- 规格推送

多活场景运维 (DMS)

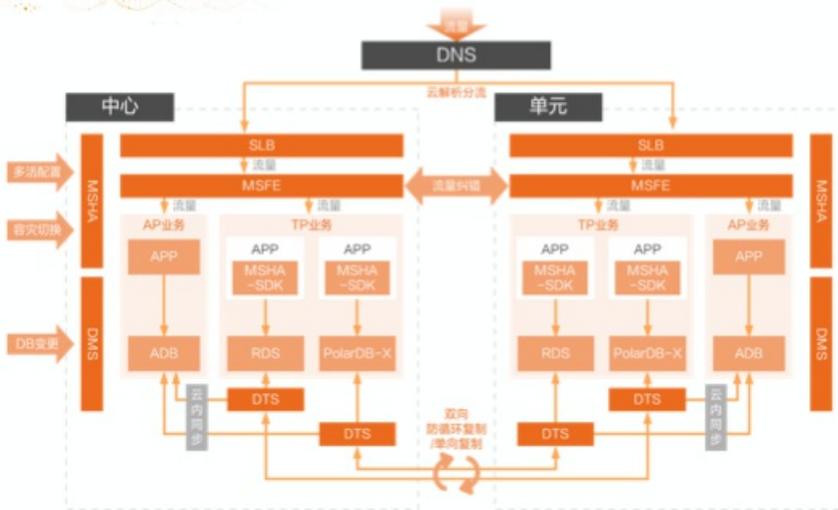
- 借助DMS进行日常的运维变更

- DDL变更
- 数据运维

针对于OLTP业务，阿里云提供了一套相应的多活架构，其中包含了几个关键要素。第一，多活配置，主要通过MSHA进行一站式多活配置，其负责制定流量划分策略、决定哪些数据库需要进行多活。第二，多活流量控制，主要根据既定规则通过MSFE进行分流，其负责流量识别、流量分发以及流量校正。第三，多活数据同步，主要是通过DTS实现，DTS本身是数据同步工具，其针对多活场景增加了很多新功能，如防循环、网络优化和切流联动等。第四，多活容灾切换，也是通过MSHA实现，主要负责将规格下推到各层，并对多活切换之前的状态进行全局检查。第五，多活场景运维，通过DMS实现，多活场景下实现DDL变更和数据运维存在双写问题，并存在同步延迟的可能，因此执行DDL和DML变更的策略是不同的，DMS针对于多活场景进行了能力适配。

OLAP业务多活架构

OLAP业务多活架构



AP业务并不独立存在，AP业务的数据来源是TP业务实时产生的增量数据。

多活流量控制

- 通过MSFE进行分流，为AP业务定义不同的分流策略

多活数据同步

- 借助TP多活的数据同步能力
- 通过DTS实现本单元的TP向AP数据同步

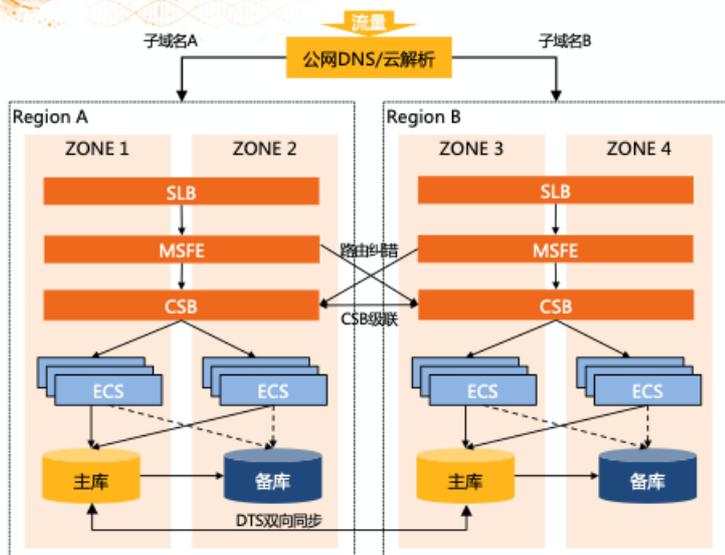
多活容灾切换

- AP业务以读为主，对数据同步时延不敏感
- 通过MSHA控制台上对分流策略进行调整

OLAP业务多活架构与OLTP区别不大，要素也基本一样，唯一不同在于在OLTP业务多活架构中在底层实现了双向的数据同步，在OLAP业务多活架构中，则不建议做这样的工作。主要有两个原因，其一，跨地域数据同步的带宽成本非常高，如果OLTP已经将数据同步了一份，那么尽量选择在云内同步，而不是OLAP同步；其次，还需要保证数据一致性，在OLTP上同步了一次，如果在OLAP上还需要同步一次，那么保证数据一致性就会比较困难。因此，阿里云建议不在OLAP上做数据同步，而应该全部在OLTP上做，并且在云内可以实现数据同步能力的补齐。

双活典型架构：双Region四AZ

双活典型架构：双Region四AZ



架构说明

- Region A、Region B均为生产单元，提供在线服务
- 每个单元对外暴露一个子域名
- MSFE跨AZ部署，进行流量归属判断
- 两Region的CSB级联部署，提供服务纠错和中心化服务的转发
- 数据库主备架构，分别部署在本Region的两个AZ中

分流策略

- 入口流量按照权重转发到不同cname下，可根据实际情况调控
- 流量进入接入层MSFE进行归属判断，归属本单元则继续向下，否则转发到对端单元
- CSB通过级联可以暴露本单元服务到对端，对于中心化服务的流量转发到中心。

数据同步策略

- 数据在Region A和Region B之间进行DTS双向同步

成本

- 应用冗余200%，AZ1 50%，AZ2 50%，AZ3 50%，AZ4 50%。
- 数据库冗余4份数据

容灾能力（计划外切换）

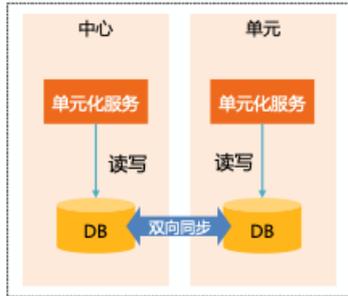
- 支持AZ级故障，RTO 分钟级，RPO > 0
- Region级故障，RTO 分钟级，RPO > 0

上图所示的是双活典型架构，分为两个Region，每个Region里面各有两个AZ，首先具备AZ级别的容灾能力，如果真的发生了地域级故障，再将Region级别的容灾能力用起来。在这个架构下，MSFE以及具体业务系统等是跨AZ部署的，在云内具备AZ级高可用。数据库在AZ1和AZ2、AZ3和AZ4可以进行主备部署，底层通过DTS实现双向同步。数据是四份副本冗余，业务冗余达到200%，每个AZ冗余到达50%，但真正承接流量时可实现每个AZ只有25%，业务可以自行调配。对于计划外的切换而言，可以达到分钟级RTO。

多活中不同的服务类型

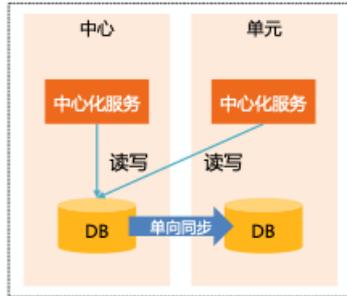
多活中不同的服务类型

单元化服务



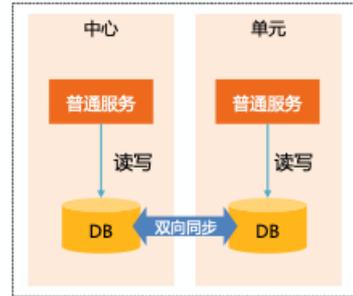
- 多活主要面向的服务类型
- 单元内封闭调用，不依赖其他单元
- 非本单元流量纠错到对端单元

中心化服务



- 数据有强中心要求，通常提供全局业务服务
- 仅中心提供服务，各单元读写请求均路由到中心
- 单元数据仅提供灾备服务

普通服务



- 不做任何改造的服务，就近访问本地
- 能容忍同步延迟，写入后往往不需要立即读取
- 主要面向读服务，不建议写场景使用（缺少单元写保护）

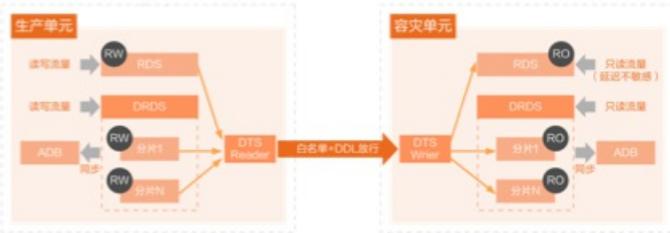
前面提到服务层分为三种服务类型，第一种是单元化服务，这是多活架构下主要面向的服务类型，比如淘宝买家的信息修改就是典型的单元化服务，其根据买家的用户ID进行流量分流，在这个维度下，可以实现单元内封闭调用，不依赖于对端数据，而底层的数据同步只是在数据切换时确保对端数据是完整的，能够将数据补齐的，这样切换之后能够让业务直接运行。第二种是中心化服务，主要面向全局配置或者业务具有强中心读写要求的场景，如库存扣减，不允许在多个地方同时扣减同个库存，这种场景一定会访问中心数据库，底层通过单向同步来同步数据，这样的服务提供的并不是多活能力，而是容灾能力。第三种是普通服务，所针对的是如果业务按照某一个维度进行了流量划分，那么一些耦合的边缘服务可能无法按照相同维度进行划分，这类业务可能会选择普通服务，比如淘宝交易按照买家ID进行划分，那么卖家就无法按照这一维度进行划分。普通服务能够容忍同步延迟，也就是最终一致，但是无法接受访问延迟，因此主要面向读服务，不建议写场景使用。

跨云数据同步



跨云数据同步

COPY类型/非多活类型



- ✓ 应对中心化服务和普通服务
- ✓ 数据单向同步，单元只可读不可写
 - 同步任务配置使用白名单+DDL放行方式
 - 跨城同步异步复制

UNIT类型

- ✓ 适配单元化服务和普通服务
- ✓ 数据双向同步，各单元均可读写
 - 防循环机制通过事务表/THREAD_ID方式实现
 - 通过全局Sequence避免冲突



DTCC
2020



北京国际会议中心

2020/12/21-12/23



ChinaUni云

ITPUB

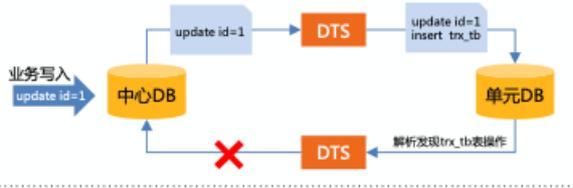
上述三种服务类型在底层的数据同步方式不同，因此给出了两种跨云数据同步方式。第一种是COPY类型的数据同步方式，主要面向中心化服务和普通服务，数据是单向同步的，单元只可读不可写，同步任务配置通过白名单+DDL放行方式实现。第二种是UNIT类型的数据同步方式，主要面向单元化服务和普通服务，数据是双向同步的，各单元均可读写，此时就需要通过事务表等解决防循环问题，并且通过全局Sequence避免冲突。

防循环&Sequence

防循环&Sequence

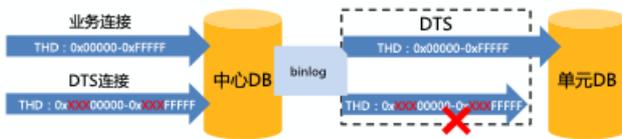
防循环

事务表方式：



THREAD_ID方式：

AliSQL内核具备设置THREAD_ID功能



Sequence

分配全局唯一的Sequence，避免数据冲突。

- > 内步长：单次申请的可用于分配的ID个数
- > 外步长=内部长*单元个数

以内步长10万，单元个数2为例：

create table TEST(

ID int primary key auto_increment unit count 2 index 0)

	中心	单元
外步长	1 ~ 100000	100001 ~ 200000
	200001 ~ 300000	300001 ~ 400000
	400001 ~ 500000	500001 ~ 600000
	内步长	

阿里云POLARDB和RDS数据库等针对于防循环和Sequence两个能力进行了实现。在防循环部分，主要提供了两种方式，第一种是事务表方式，也就是业务在写入数据库的时候，即事务提交完成，生成Binlog，Binlog被DTS拿走并解析完成后会发现向目标单元DB写入的时候会在事务表里面产生一个自定义记录，这样一来在单元里面落地的事务实际上除了原始业务逻辑之外还会多一个小Event。通过目标端的DTS解析之后就会发现Binlog里面还多了一个事务操作，就会知道这个操作是来自于DTS的，而不是来自于业务系统的，因此可以将该操作过滤掉，进而防止数据循环。第二种是通过THREAD_ID的方式，这是AliSQL内核定制的优化功能，将原生MySQL内核的THREAD_ID从8字节改到了5字节，因此业务生成连接只能是0x00000到0xFFFFF之间，而高位则留给DTS连接使用，这样中心DB就能够区别两类连接，Binlog会记录所有的THREAD_ID，因此DTS也能够很清晰地解析出来操作来自于业务还是DTS，如果来自业务就同步过去，如果来自DTS就中断掉，从而达到防循环的功能。第一种方式对业务具有一定的侵入性，第二种则是完全原生的能力，对用户或者内核没有太大影响。

对于Sequence功能而言，其实就是在两边同时写入数据，需要保证数据不能冲突。因此，阿里云针对于POLARDB-X做了全局唯一Sequence的能力，在原生的DDL上面增加了标识去控制当前单元个数以及每个单元的Index。基于这种方式创建出来的表，以内步长为10万，单元数为2举例，产生结果如上图所示，从而达到全局Sequence的能力。

多活场景数据保护

数据质量核心问题：数据双写

单元保护-日常态

数据层，应用和PolarDB-X对写操作的多活分流规则进行路由逻辑校验，若非本单元流量，触发单元保护机制，不进行写操作。

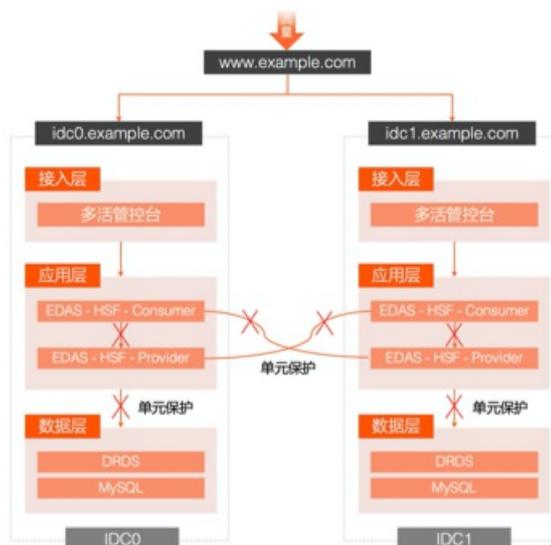
单元保护-变更态

DMS定制化的数据运维功能，订正前对进行各单元数据备份，并对同步链路状态进行检查，小于1S才能执行。

单元保护-切流态

切流前进行预检查，切流中通过多种手段保障局部切流用户的数据在各单元的一致性。

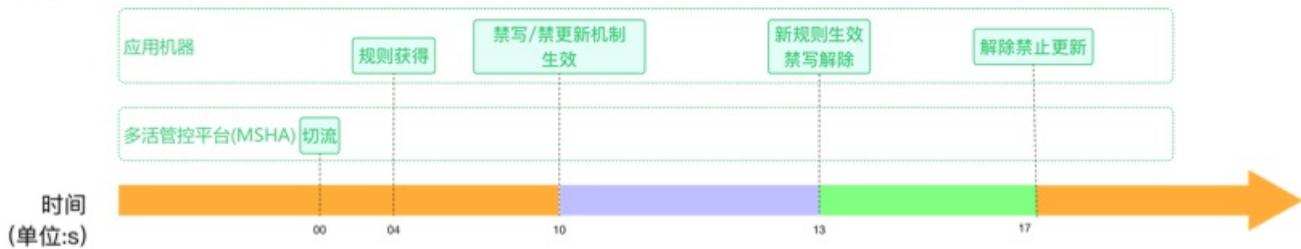
- 绝对禁写
- 延迟禁写
- 前镜像匹配同步
- 延迟检查



在多活场景下，和原生最大的区别就是不需要关注可用性，但是却多了数据质量的问题，该问题在单数据中心场景下可能不容易发生，但是在多活场景下因为业务需要双写，因此容易出现数据质量的冲突问题。归根结底，所有的数据质量问题都是由于数据双写导致的，因此需要针对于这种场景制定一定的保护措施。阿里云制定了三个维度的单元保护措施，第一个是日常态，针对接入层、应用层和数据层提供相应的方法多写操作的多活分流规则进行路由逻辑校验，如果非本单元流量，则在接入层和应用层将流量转走，但如果在数据层，则直接阻塞掉。第二个是变更态，主要针对数据运维变更，比如批量数据订正，阿里云提供了事前检查和事后补充的能力，在DMS上面针对于多活场景下的数据变更任务提前检查变更情况，如果同步延迟很大则会被阻塞掉，降低了数据双写的概率，同时在变更前和变更后通过检查保持数据的一致。第三个是切流态，是在数据多活切流过程中做的保护策略，包括了绝对禁写、延迟禁写、前镜像匹配同步以及延迟检查等功能。

多活切流流程

多活切流流程



举例说明，如上图，在某个时刻有切流需求，用户在多活管控平台进行切流操作。

1. 在0s，多活管控平台(MSHA)下发切流命令，并打开DTS前镜像匹配功能。
2. 在04s，应用机器获得切流操作命令及新的流量分发规则，此时机器上运行的仍旧是老的流量分发规则，实际业务无影响。
3. 在10s，在规定的某个时刻(例如10s)启动绝对禁写，此刻正式开始切流，切流用户的写操作和更新操作均被拒绝，当前请求失败。
4. 在13s，持续3s(可配置)后，规则在各层完成收敛，此刻解除绝对禁写，启动延迟禁写，禁写时长取决于数据同步速度，若未同步完成，多 db 同时的更新会带来数据脏写问题。
5. 当数据同步完成后，机器上解除延迟禁写，关闭DTS前镜像匹配，此刻切流用户的更新操作可正常进行。

在多活切流时，首先会打开前镜像匹配功能。一般认为，在多活场景下业务写入的数据比同步过来的数据更重要，因此需要保证业务写入的数据不被同步的数据覆盖掉，所以如果切流过程中，数据同步有延迟，为了不覆盖掉业务数据，则需要将Binlog里面前镜像拿出来拼到SQL里面去执行。前镜像匹配功能开启之后会将新的流量分发规则在各层进行下发，在规则下发完成之后会开启绝对禁写的动作，在此过程中，所有参与切流的用户流量是无法执行的。在禁写过程中首先需要判断三层规则是否全部收敛成功，其次还需要判断每层内各个节点的规则是否收敛成功，最终目标是让所有服务器上的规则保持一致，这样才能保证不出现双写。上述条件满足之后，解除绝对禁写，开启延迟禁写，这一点可由用户配置。当数据同步完成之后，解除禁写和前镜像匹配，切流过程至此完成。

异地多活价值总结

异地多活价值总结

业务即容灾	业务连续性保障	业务高速发展支撑	流量有效隔离	降本增效
各个数据中心常态承载业务流量，切流有保障	核心单元化业务异地多活，且秒级切流到异地	实现业务级的流量闭环	各个数据中心承载的流量大小可灵活调配	有效分摊各个数据中心成本，实现成本小于200%冗余
	全局业务异地热备，分钟级切换	具备跨地域的水平扩展能力	在最小单元内进行风险可控的技术演进	
	一体化多活运维管控平台，支撑流量、数据库一键切流、零脏写			

简单总结下异地多活的价值。首先，多活本身是做容灾的，但是现在来看异地多活已经不像是传统容灾那样放置一个灾备单元了。现在业务即容灾，业务系统和容灾系统紧密地连接到了一起。其次，业务连续性有了保障，为业务提供了高可用能力。第三，为业务的高速发展提供了支撑，在多活场景下划分了很多原子单元，可以根据原子单元合理配比相关资源，达到最优效果，最终具有跨地域的水平扩展能力。第四，流量有效隔离，基于阿里云的异地多活解决方案可以非常灵活地调配流量，可以按照不同维度设置规格，也可以按照不同的权重配比设置，实现流量大小的灵活调配，并可实现在最小单元内进行风险可控的技术试验。第五，降本增效，传统容灾方案无法突破200%的冗余成本问题，而通过三活、四活的方案可以实现冗余成本小于200%。

用户自行实施异地多活的难点

用户自行实施异地多活的难点



流量管理难度高

- ✓ 需要对接入层、服务层、数据层等的**流量规则进行统一管理**。
- ✓ 在分发规则时，需要保障众多节点**规则的一致性**。
- ✓ 需要具备**多维的分流能力**，和**动态调配能力**。



数据同步策略复杂

- ✓ 针对不同服务类型需要**支持多种同步类型**
- ✓ 双向数据同步中需要具备**同步防循环能力**。
- ✓ **远距离数据同步**，对同步性能、同步带宽有很高要求。



容灾切换数据质量保障难

- ✓ 为了保障RPO要求，容灾切换过程中需要对业务架构的各层进行**状态检查的能力**
- ✓ 对**规则分发的收敛情况**以及跨数据中心的**同步情况**进行准确评估，难度较大。



多数据中心统一管控难度大

- ✓ 需要具备**快速建站**和**多单元统一管理**的能力
- ✓ 需要**对接众多基础设施**

用户自行实施异地多活所需面对很多难点，如流量管理难度高、数据同步策略复杂、容灾切换数据质量保障难，以及多数据中心统一管控难度大等，这也是阿里巴巴将异地多活能力沉淀为产品级解决方案的推动力。基于阿里云的异地多活方案，用户只需要关系如何对流量进行分割即可。

阿里云云原生方案优势

阿里云云原生方案优势



阿里多年沉淀

- ✓ 阿里从12年开始实施异地多活
- ✓ 有超过300+业务，上万数据库实例的实践经验



流量精细化管理

- ✓ 实现多维的流量管理策略，满足灵活性
- ✓ 流量管理与底层存储分布解耦，流量随心所欲切换



一体化的解决方案

- ✓ 统一管理和路由规则分发
- ✓ 从多活建站到容灾演练实现能力全覆盖



数据质量保障

- ✓ 提供多种数据质量保障手段
- ✓ 有效控制切流态的数据质量问题



分钟级切换

- ✓ 保障容灾切换时的RPO最高可达到秒级
- ✓ 通过“一键切换”能力对各层规则统一管理，可以达到切换RTO分钟级



成熟的多活生态

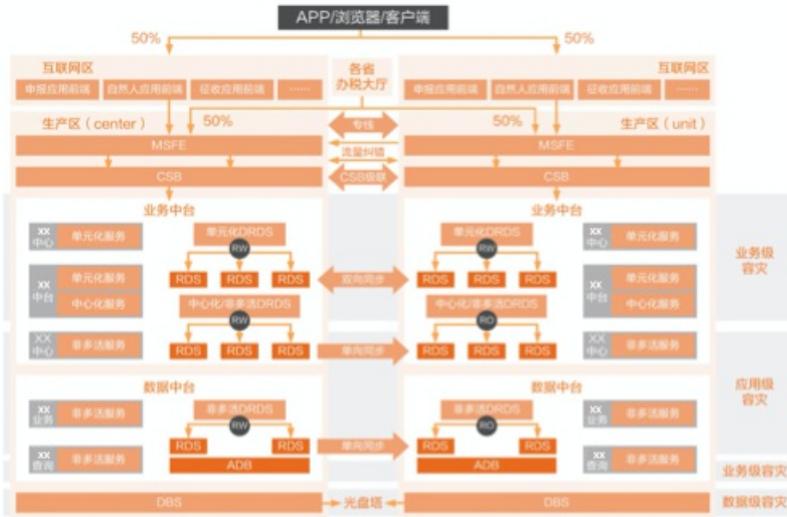
- ✓ 生态工具对多活场景进行充分适配

目前能够实现产品级异地多活能力的厂商极少，阿里云经过8年的积累和沉淀，在异地多活的云原生方案上具有诸多优势。

三、异地多活客户案例



客户案例-某税务核心系统



方案概述

接入层：MSFS做流量纠错，支持按省份分流和按自然人档案号分流

服务层：CSB级联支持普通服务跨云调用

数据层：针对不同服务类型（单元化、中心、非多活）实施不同容灾级别的数据同步架构。

多活实施效果

- ✓按地域和自然人两个维度实施多活
- ✓秒级切换能力，达到国标6级效果
- ✓两单元接流，充分利用资源
- ✓灰度放量能力

DTCC
2020



北京国际会议中心

2020/12/21-12/23



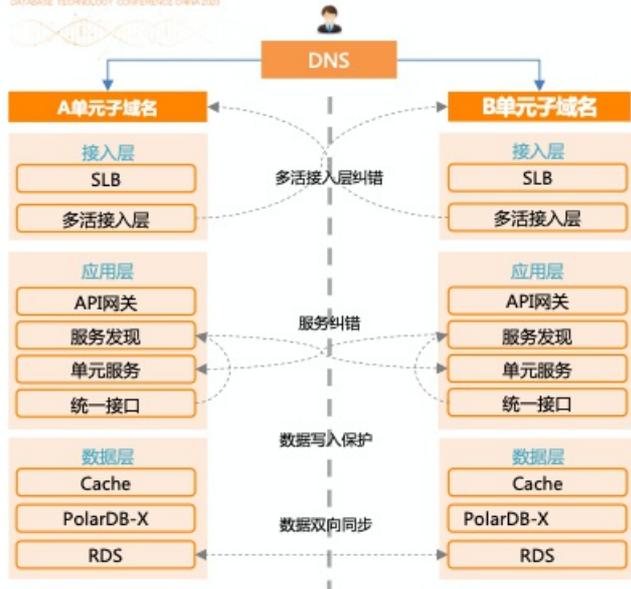
ChinaUnix



某税务核心系统的异地多活方案也是按照三层架构实现的，在接入层，支持按照两个维度流量拆分，即省份和自然人档案号。在服务层利用CSB产品实现普通服务的跨云调用。在数据层，针对不同服务类型实施不同容灾级别的数据同步。最终实现了两个维度的多活，秒级切换能力，达到了国标6级效果，因为基于两单元接流，因此在成本上具有优势，并且具有灰度放量能力。

客户案例-某运营商客服系统

客户案例-某运营商客服系统



分流标识: 按省分

方案概述

接入层统一控制: DNS按地域分流, 接入层按照路由规则判断并纠错

应用层单元化部署: 服务发现实现双中心的服务同步能力

数据层双向同步: 通过DTS双向同步实现数据最终一致性

多活实施效果

- ✓ 客服系统的多个业务实现按地域多活分流
- ✓ 实现多次容灾演练, 秒级完成切换, 数据0丢失
- ✓ 客户常态两个单元均承载业务流量, 充分利用两单元的资源

DTCC
2020

北京国际会议中心 2020/12/21-12/23

IT168

ChinaUnix

ITPUB

某运营商客服系统实现了按省份分流能力, 即按照DNS的地域分流接入南北两个中心, 接入层按照路由规则进行判断和纠错, 在业务层对于客户原有系统进行了适配改造, 实现了双中心的服务同步。在数据层, 则通过POLARDB-X和DTS实现双向数据同步。最终使得该运营商客服系统的多个业务按地域多活分流, 在多次容灾演练中, 可以完成秒级切换, 并保障了数据0丢失。此外, 由于常态由两个单元承载业务流量, 因此成本也有所降低。

[点击下载本场演讲PPT](#)

相关阅读

【内含干货PPT下载】DTCC 2020 | 阿里云叶正盛: 数据库2025

<https://developer.aliyun.com/article/780725>

【内含干货PPT下载】DTCC 2020 | 阿里云赵殿奎: PolarDB的Oracle平滑迁移之路

<https://developer.aliyun.com/article/780749>

【内含干货PPT下载】DTCC 2020 | 阿里云朱洁: NoSQL最新技术发展趋势

<https://developer.aliyun.com/article/780746>

【内含干货PPT下载】DTCC 2020 | 阿里云王涛: 阿里巴巴电商数据库上云实践

<https://developer.aliyun.com/article/781001>

DTCC 2020 | 阿里云梁高中: DAS之基于Workload的全局自动优化实践

<https://developer.aliyun.com/article/781036>

【内含干货PPT下载】DTCC 2020 | 阿里云程实: 云原生时代的数据库管理

<https://developer.aliyun.com/article/780992>

【内含干货PPT下载】DTCC 2020 | 阿里云吉剑南: 在线分析进入Fast Data时代的关键技术解读

<https://developer.aliyun.com/article/780747>

原文链接: <https://developer.aliyun.com/article/781031?>

版权声明：本文内容由阿里云实名注册用户自发贡献，版权归原作者所有，阿里云开发者社区不拥有其著作权，亦不承担相应法律责任。具体规则请查看《阿里云开发者社区用户服务协议》和《阿里云开发者社区知识产权保护指引》。如果您发现本社区中有涉嫌抄袭的内容，填写侵权投诉表单进行举报，一经查实，本社区将立刻删除涉嫌侵权内容。