

在数字经济大发展的背景下，数据量和并发访问量呈现指数级增长，突发事件导致的系统不可用给企业带来的损失也越来越大。因此，在突发事件发生期间和之后，如何尽量降低突发事件对产品和服务的影响，如何确保服务可用且能够迅速恢复，成为企业数字化转型的核心诉求。

从技术上看，衡量容灾系统有两个主要指标：RPO（Recovery Point Object）当灾难发生时允许丢失的数据量、RTO（Recovery Time Object）系统恢复时间，这两大指标要求系统具备极高的可用性和可靠性。

一些常用的方案存在以下主要问题：

\* 数据丢失风险高：传统IT高可用系统的实现主要是以主备的方式进行部署， 这种方案应用广泛，已经过长时间的验证，但仍然无法很好解决例如故障发生后切换数据不丢失的需求；

\* 故障恢复时效差：传统的双活架构，由于异步复制机制问题，主中心故障发生后不敢切、不能切的情况时有发生；

\* 故障爆炸半径大：由于传统架构下业务系统整体强耦合，数据库层面也经常发生单点故障影响全站用户的情况，严重影响业务连续性。

异地多活架构和其他方案对比，有以下优势：

\* 能应对机房/城市级故障，RTO分钟级别自动恢复，RPO 最多分钟级别数据丢失；

\* 单元化设计，调度系统实现任意单元流量在城市和机房间的灵活调拨与切换，单元内故障不影响全局服务。

设计原则

1、对客户端的业务实现多活

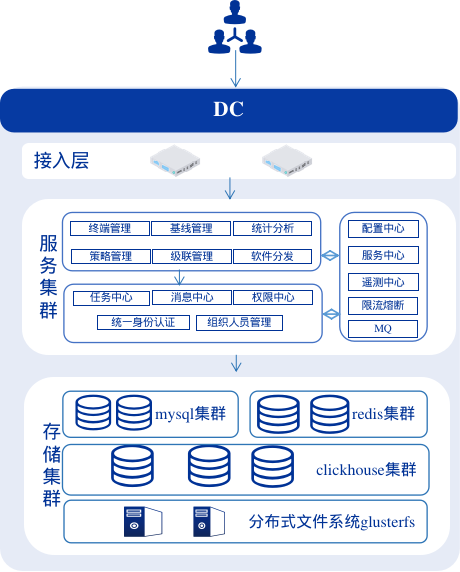


从业务类型视角分析，集中域管平台分为终端业务和管理端业务：

* 终端业务，由客户端实现策略接收、策略执行、数据上报等功能；
* 管理端业务，业务操作包括系统配置、策略、组织/人员/终端管理等。

由于客户端业务影响的用户范围大、故障发生时产生影响和损失也大，因此客户端业务需要实现多活。

2、数据同步保持最终的一致性



从平台的架构来分析，系统分为接入层、服务层和存储集群：

* 接入层提供集群层面的负载均衡
* 服务层使用K8s部署业务服务
* 存储层提供数据的读写服务

（1）mysql：业务数据库集群，存储业务数据

（2）redis：主要存储从业务数据库的查询缓存

（3）clickhouse：OLAP引擎，主要来处理分析类的工作，包括日志存储、报表分析、即席查询ad -hoc等

（4）Glusterfs：存储静态资源和日志

接入层和服务层都是无状态业务，无同步的需求：

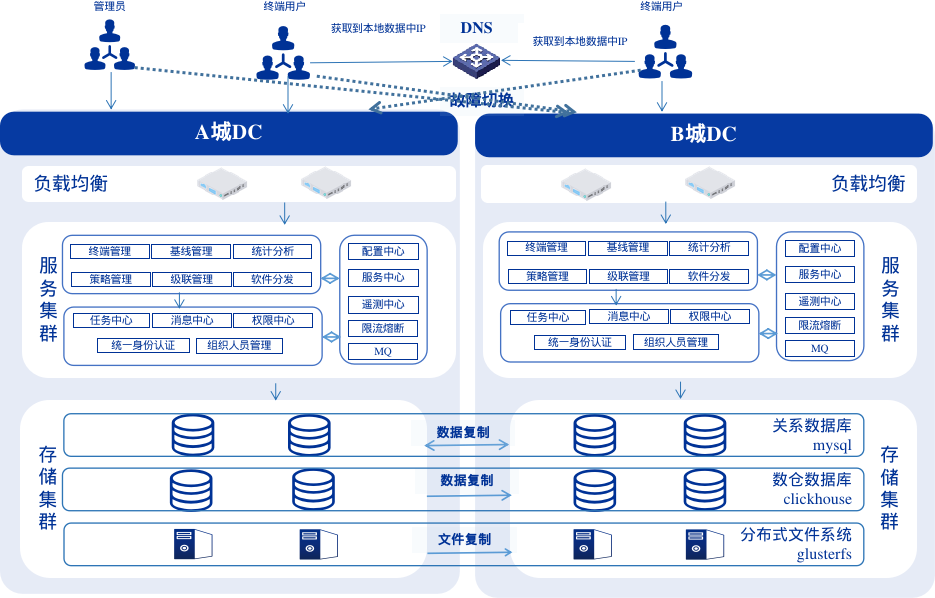
* mysql中业务数据有比较高的一致性要求，如果多活，则需要双向实时同步；
* redis缺少实时的同步手段，不同步redis数据；
* ClickHouse数据时延需求不高，但是需要同步；
* GlusterFS基本是全局数据，变化小，核心数据都是从管理终端上传，可以容忍一定时间的延迟，没有强一致性要求。

3、数据操作不容许跨中心写

在任何时间点，必须确保单行的数据在一个地方写，绝对不能在多个地方写，防止同步冲突。

                                                 改造方案

当前异地多活架构由两个异地数据中心组成，每个中心都是独立的可部署单元，两个中心之间使用专线连接，在流量分发层面，通过智能DNS对全局流量使用基于地理位置的策略进行路由，这样可以应对城市级别数据中心的故障，实现故障的快速转移，从而达到高可用和高可靠的能力。

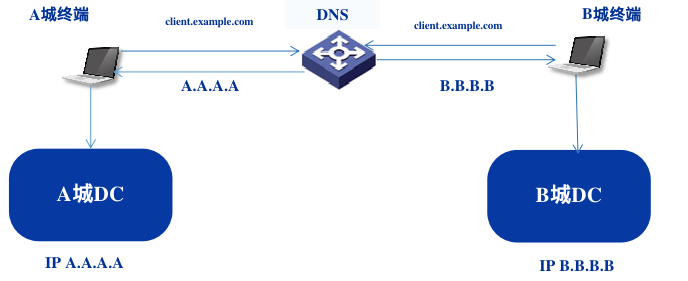


* 全局高可用：关键业务数据双向实时数据同步，低RPO、RTO；
* 单元化部署，异地多活，自动故障转移和切换；
* 就近服务的全局负载策略，业务分流，更好的用户体验；
* 单中心服务和基础组件冗余部署、故障转移和容错机制，保证单中心的高可用。

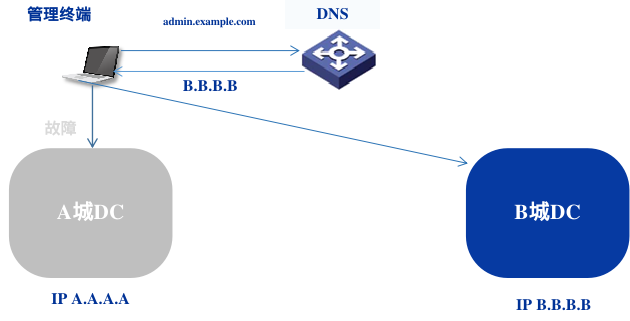
服务划分，基于地理位置切分

选择地理位置作为划分业务的单元，把地理位置上接近的用户划分到同一个机房，保证业务在一个机房完成闭环，能够保证最小的延时。在某个机房出现问题的时候，按照地理位置把用户路由到其他机房即可。

客户端请求的服务配置单独的域名，在智能DNS上配置基于地理位置的域名解析机制，当客户端发起域名解析请求时，通过客户端IP所在的地理位置将流量转发到对应的机房。

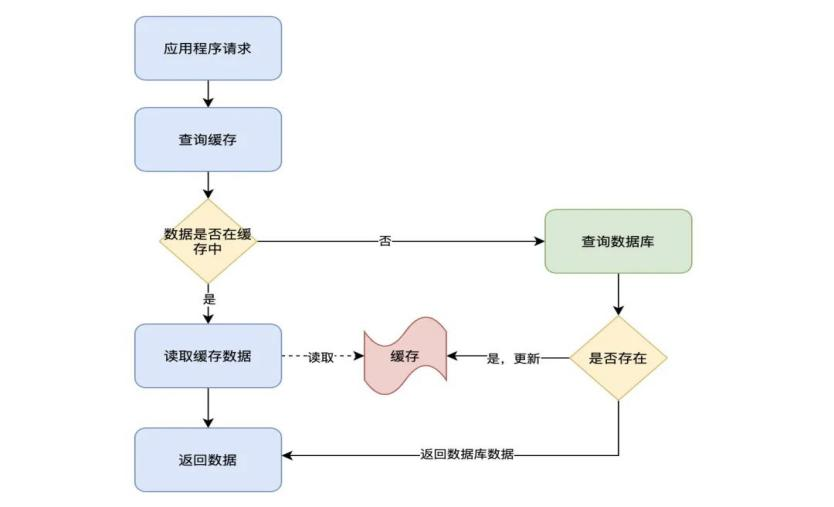


管理端业务域名解析通过智能DNS配置全局的故障转移策略，默认情况下解析主机房的IP，当机房不可用时，解析备中心机房，能保证故障转移，提高可用性。



redis缓存使用改造

由于不同步redis的缓存，对于全局性的缓存数据，采用Cache-Aside Pattern。



读流程



写流程

如果管理端修改了某个中心数据库的业务数据，同时删除缓存中的旧数据，等到再次查询时加载到内存，保证当前数据中心的缓存是最新的，然而另外一个中心无法主动感知管理员发起的操作，那么这个数据中心缓存中的数据没法删除更新，因此需要对缓存设置过期时间，利用过期时间淘汰旧数据。过期时间设置的规则按业务变化的频度和对不一致性的容忍度，对缓存对象设置过期时间。

定时任务改造

对于全局性的数据操作处理的定时任务，比如各类报表的统计任务等，如果在多个中心都执行，会导致最终生成数据在业务上的冲突和重复，对于这类定时任务，只在主中心执行。

数据冲突的解决

改造时，需要双向同步两个中心的数据，使用自增ID时会导致双中心同一个表的主键冲突，需要使用全局的UUID解决自增ID的冲突。

* 生成规则：snowflake（时间戳+机房ID+服务器IP+服务在1s产生的自增序列）；
* UUID的生成封装成lib，由服务调用获取UUID。

数据同步

由于两个中心都有用户数据的插入、更新，为了保证机房发生故障时，别的中心可以完全接管流量，需要双向数据同步，保证不同机房之间数据的一致性。

mysql数据的双向同步

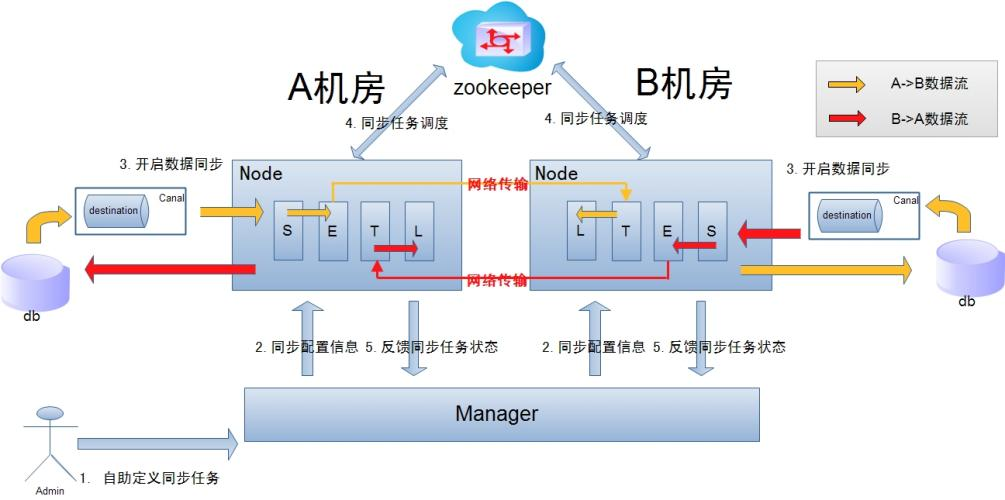
核心诉求：

* 数据不能丢失 (变更数据一定要成功应用到目标库)
* 数据最终一致性 (双向两边记录要保证最终一致性)

otter

基于数据库增量日志解析，准实时同步到本机房或异地机房的mysql/oracle数据库：

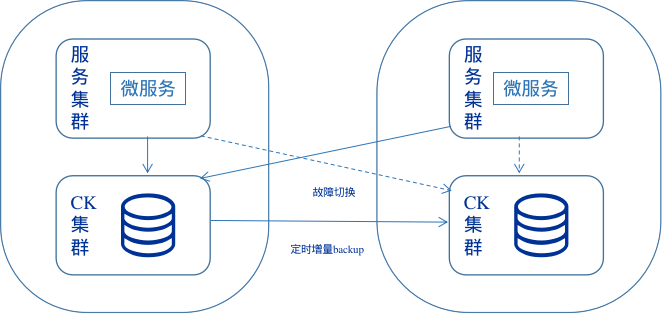
* 异构库同步
* 单机房同步 (数据库之间RTT < 1ms)
* 异地机房同步
* 双向同步



从otter的特性看，完全能满足异地同步的需求。

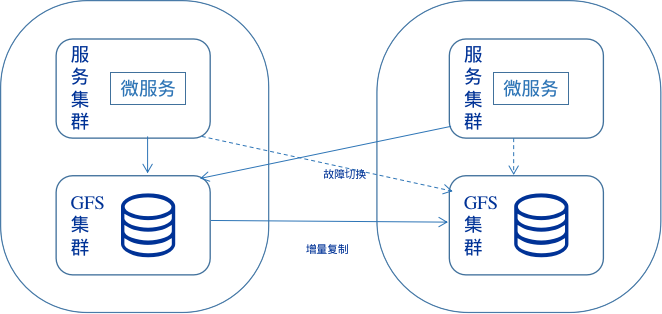
ClickHouse同步

对于OLAP的数据报表业务来说，对延迟的要求并不高，这类业务也不需要多活，毕竟多活的成本太高，因此，两个数据中心1主1备，主备之间使用同步工具增量数据复制，业务自己基于gorm dsn做故障转移，并且具备高可用性。



GlusterFS文件同步

采用一种持续、异步、增量数据备份策略，可以通过局域网、广域网、因特网来进行。使用同步工具能够在存储环境建立数据冗余，提供数据的灾难恢复功能，通过GFS的挂载卷实现故障的转移。



总结

异地多活采用单元化的设计，将业务单元化和用户单元化保证路由一致性和快速的故障转移。对于企业来说，通过评估故障的影响范围和损失来选择哪些业务多活需要单元化，对于不同的一致性要求和时延要求选择不同的数据同步方案，从而减少系统的复杂度和维护实施成本，降本增效。

©统信软件技术有限公司。访问者可将本网站提供的内容或服务用于个人学习、研究或欣赏，以及其他非商业性或非盈利性用途，但同时应遵守著作权法及其他相关法律的规定，不得侵犯本网站及相关权利人的合法权利。除此以外，将本网站任何内容或服务进行转载，须备注：**该文档出自【faq.uniontech.com】统信软件知识分享平台**。否则统信软件将追究相关版权责任。