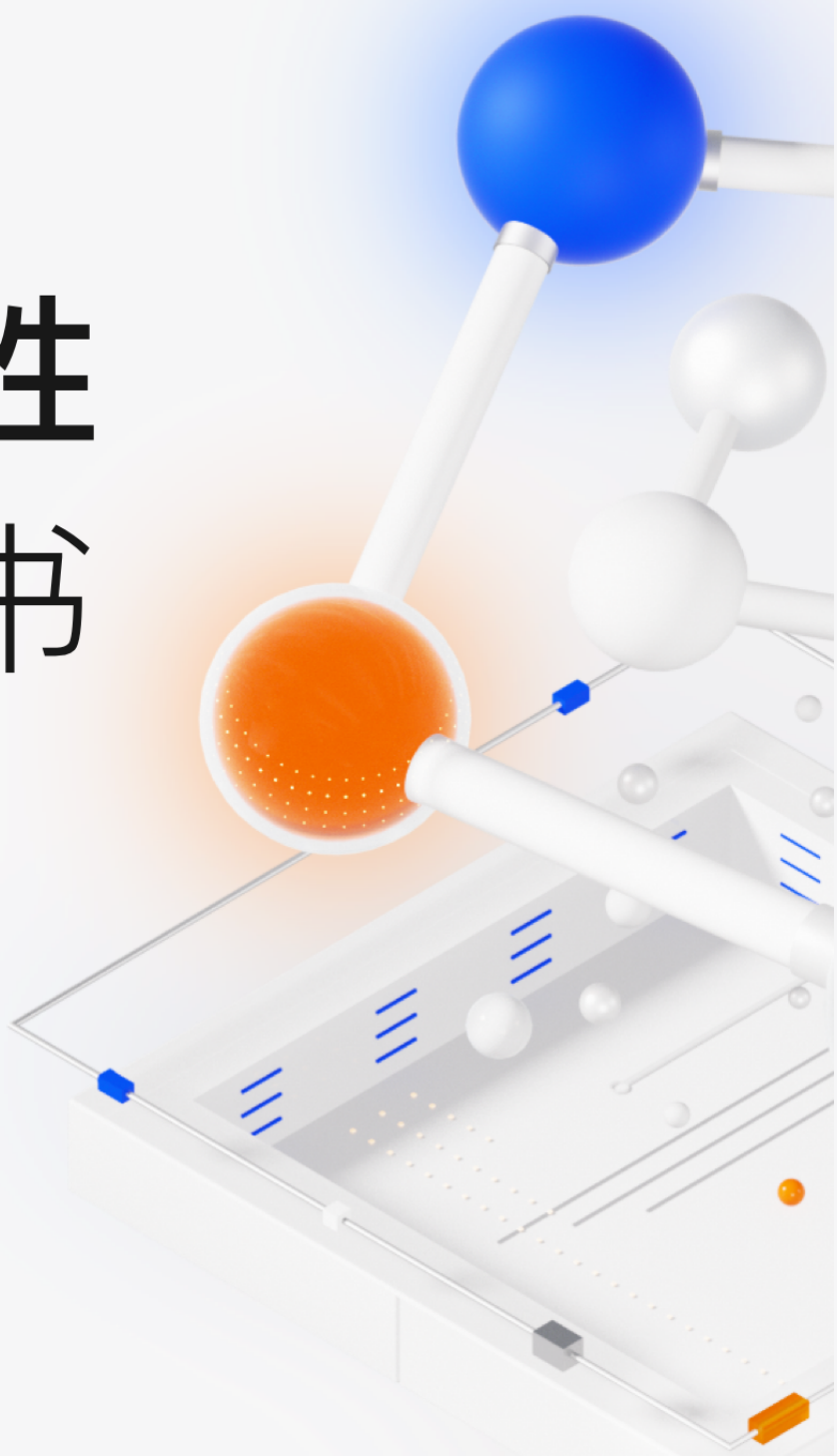


存储 可靠性 白皮书



目录

03 引言

04 对象存储 OSS 可靠性概述

OSS 简介

如何度量 OSS 可用性 SLA

07 OSS 可用性技术实践

OSS 可用性体系

异地多活架构

OSS 跨区域复制架构

管理机制

引言

数据是最宝贵的资源，而存储是数据的载体。云存储是云平台中可靠性、稳定性和安全性要求最高的部分。

随着存储到云上的数据迅猛增长，数据形式的多样化，以及对数据价值洞察和挖掘的要求越来越高，云存储面临的挑战也越来越大。阿里云存储通过系统化的架构与技术设计和智能的系统管理和运维来提供数据的高可靠性和高可用性，保障客户的业务连续性。

1. 阿里云实现了多副本策略和纠删码 (Erasure Coding) 数据冗余技术，提高可靠性；
2. 通过把数据分布在 3 个可用区 (Availability Zone, AZ)，容忍数据中心级别的故障；
3. 通过端到端的数据校验和高效的周期性数据巡检，保障数据不丢不错；
4. 通过智能化系统管理和自动化系统运维，提高超大规模系统的稳定性和可用性。

基于阿里云分布式存储——盘古系统所构建的多个产品，提供了丰富的存储服务 and 解决方案，包括对象存储、块存储、文件存储、表格存储、日志服务、网盘和相册、灾备和容灾服务等，满足不同业务场景的需求。

我们以对象存储 OSS 为代表撰写本白皮书，希望读者可以系统性地了解到 OSS 的可靠性是如何定义的、阿里云做了什么来提升产品的可靠性；以及用户可以通过构建什么样的架构，充分利用对象存储 OSS 提供的可靠性能力来为自己的 IT 系统服务。

对象存储 OSS 可靠性概述

OSS 简介

对象存储 OSS (Object Storage Service) 是阿里云提供的一款海量、安全、低成本、高可靠的云存储服务, 尤其适合非结构化数据 (如图片和音视频等) 的存储, 经受住了“双 11”购物节、12306 春运、健康码这样的超大流量的考验。

OSS 的多重冗余架构设计, 为数据持久存储提供可靠保障, 提供 12 个 9 的数据可靠性和 99.995% 的 SLA (服务可用性)。同时, OSS 基于高可用架构设计, 消除单节故障, 确保数据业务的持续性。其存储类型分为标准存储、低频访问存储、归档存储、冷归档存储、深度冷归档五种存储类型, 全面覆盖从热到冷的各种数据存储场景。

如何度量 OSS 可用性 SLA

业界常见的可用性度量为年故障时长

业界对可用性的描述, 通常采用年故障时长。比如, 数据中心机房划分为不同等级, 如 T1 ~ T4 机房, 它们的可用性指标如下:

T1 机房:

可用性 99.671%、年平均故障时间 28.8 小时;

T2 机房:

可用性 99.741%、年平均故障时间 22 小时;

T3 机房:

可用性 99.982%、年平均故障时间 1.6 小时;

T4 机房:

可用性 99.995%、年平均故障时间 0.4 小时。

网络服务的可用性, 通常会折算为不能提供服务的故障时间长度来衡量, 比如典型的 5 个 9 可用性就表示年故障时长为 5 分钟, 如下表所示。

可用性指标	计算公式	年故障时长
3个9(99.9%)	$(1-99.9\%)*365*24=8.76\text{小时}$	8.76小时
4个9(99.99%)	$(1-99.99\%)*365*24=0.876\text{小时}=52.6\text{分钟}$	52.6分钟
99.995%	$(1-99.995\%)*365*24=26.2\text{分钟}$	26.3分钟
5个9(99.999%)	$(1-99.999\%)*365*24*60=5.26\text{分钟}$	5.26分钟

对象存储 OSS 可用性 SLA 采用更苛刻的度量

对象存储是资源访问型云服务, 它不提供实例而是 Serverless 化的 API 调用, 按照“年故障时长”计算可用性是不合适的。所以, 阿里云对对象存储 OSS 选择“失败请求数: 总请求数”的错误率严苛逻辑来计算可用性, 如下是详细的计算过程。

每 5 分钟粒度计算错误率

每 5 分钟错误率 = 每 5 分钟失败请求数 / 每 5 分钟有效总请求数 * 100%

计算请求错误率时, 将计算请求的时间范围拉长, 对云服务有利。因为时间越长, 总请求数越多, 会导致错误率降低。为了更好的从客户角度计算错误率, 按照 5 分钟的粒度来计算。高可用系统设计关键是冗余, 而 5 分钟是业界典型的机器故障恢复时间, 能够快速修复机器, 可以降低系统的错误率。

基于服务周期内的 5 分钟错误率计算可用性

服务可用性 = $(1 - \text{服务周期内} \sum \text{每 5 分钟错误率} / \text{服务周期内 5 分钟总个数}) * 100\%$

对象存储 OSS 按月收费, 因此服务周期就是自然月。服务可用性, 就是将服务周期内的每 5 分钟错误率求和, 然后除以服务周期内 5 分钟总个数 (按照自然月 30 天算, 该值为 $30 * 24 * 60 / 5 = 8640$), 然后用 1 减去平均错误率, 就可得到该月的可用性。

根据此公式, 如果每 5 分钟错误率过高, 将会导致可用性下降; 因此, 提升每 5 分钟的请求成功率, 将是提升可用性关键。

模型对比

按照全年 26 分钟故障为基础, 年故障时长模型可用性为 99.995%。而 OSS 的请求错误率模型下, 全年 26 分钟平均到每月大约故障 2.16 分钟, 按每 5 分钟错误率来算, 如果请求在 2.16 分钟内则全部失败, 按比例来说错误率为 $(2.16/5) * 100\%$, 此时可用性为:

$$1 - \{(2.16/5) * 100\} / 8640 = 99.995\%$$

可以看出, 它和年故障市场模型的可用性相同, 但 OSS 上主要是带宽型应用、大请求居多, 因此在故障的 2.16 分钟前后的请求都会受影响, 导致可用性会稍微小于 99.995%, 从某种意义上讲 OSS 的请求错误率模型略微严格。

	计算公式	实际可用性
年故障时长模型	$1 - \{26 \text{ 分钟} / (365 * 24 * 60) \text{ 分钟}\}$	99.995%
OSS 请求错误率模型	$1 - \{(2.16/5 * 100\}) / 8640$ * 年故障 26 分钟, 按月折算为 2.16 分钟	99.995% * OSS 主要是带宽型应用、大请求居多, 月故障 2.16 分钟前后的请求都可能受影响, 因此略微小于 99.995%

OSS 可用性 SLA 目标

基于上述计算模型, OSS 经过长期的技术打磨, 可用性 SLA 提升到标准型存储(本地冗余存储)为 99.99%, 标准型存储(同城冗余存储)为 99.995%。该目标比原有值提升了 10 倍, 为此 OSS 构筑了大量核心技术并形成体系, 下一章将进行详细解读。

OSS 可用性技术实践

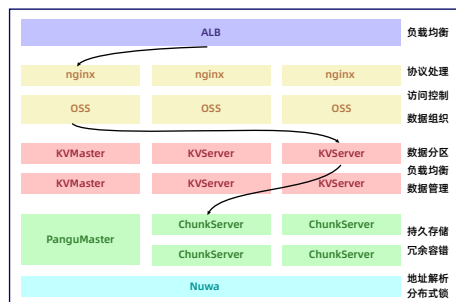
阿里云对象存储 OSS 是历经十余年研发的云服务,始终把可用性作为核心竞争力来构建,从而形成了如下的可用性体系。



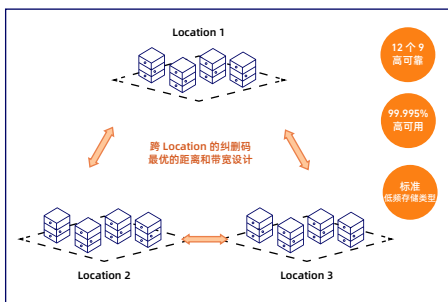
整个体系从架构、IDC 建设、分布式系统、安全防护、管理机制等维度展开,可用性的核心就是冗余和容错能力,同时要做好安全防护和管理工作。

OSS 可用性体系

对象存储提供的本地冗余存储是部署在同一个可用区 (Availability Zone, AZ) 的。同城冗余存储则部署在三个可用区 (3AZ), 它们的逻辑架构相同。同城冗余存储是在物理架构上提供机房级别的容灾能力, 将用户数据副本分散到同城多个可用区。



OSS 本地冗余逻辑架构



OSS 同城冗余 (3AZ) 物理架构

当出现火灾、台风、洪水等容易断电、断网的灾难事件, 导致某个机房不可用时, 对象存储仍然可以提供一致性的服务能力, 即故障切换过程中用户业务不中断、数据不丢失, 满足关键业务系统对于低 RTO 和 RPO 的强需求。为提供“12 个 9”的数据可靠性及 99.995% 的服务可用性, 同城冗余存储在物理层的冗余设计如下:

同城冗余存储多可用区的距离和延时设计:

在公共云部署时遵循阿里云 IDC 与网络架构设计规则及可用区选址的相关要求, 特别是要满足多可用区设计要求, 可用区间典型延时为毫秒级, 典型距离在几十千米内。

供电、制冷冗余:

对象存储是多地域部署的云服务, 几乎每年都会遇到自然灾害、供电异常、空调设备故障等问题, 在建设数据中心时要做好对双路市电和柴油发电机备电, 以及连续制冷能力的设计。

网络冗余:

对象存储作为公共云服务, 既有外部的互联网访问、虚拟私有云 (Virtual Private Cloud, VPC) 访问, 还有分布式系统的内部网络连接, 它们都需要做好冗余设计。对于外部网络, 互联网接入多运营商的边界网关协议 (Border Gateway Protocol, BGP) 和静态带宽, 实现公网访问的冗余。同时, VPC 的接入则通过阿里云网络的边界网关协议 (VPC 网关) 实现冗余。对于内部网络, 对象存储是分布式存储, 由多台服务器组成, 采用内部网络将多台服务器连通起来, 通过数据中心 ASW-PSW-DSW 的分层设计实现冗余, 保证即使某台网络设备故障, 系统仍然能够正常工作。

服务器冗余:

对象存储采用经性价比优化后的服务器, 基于分布式系统和软件定义存储的需求, 硬件上采用通用服务器, 并提供冗余的网络接口, 无须采用传统存储阵列双控冗余设计的定制硬件。

对象存储服务层聚焦数据组织和功能实现。借助底层飞天盘古的分布式能力, 对象存储服务层按照无状态方式设计, 可在业务发生故障时快速切换。由于对象存储是多租户模型设计, 所以做好服务质量 (Quality of Service, QoS) 的监控和隔离是保障租户可用性的关键。

流控分类	流控项	描述
单机流控	整体 QoS	CPU、内存资源限额
	桶级 QoS	桶的流量和请求限额
	用户级 QoS	用户的流量和请求限额
	后台任务 QoS	后台任务的流量和请求限额
	镜像回源 QoS	镜像回源特性的流量和请求限额
分布式流控	桶级 QoS	集群层面的桶流量和请求限额
	用户级 QoS	集群层面的用户流量和请求限额

对象存储的 QoS 流控

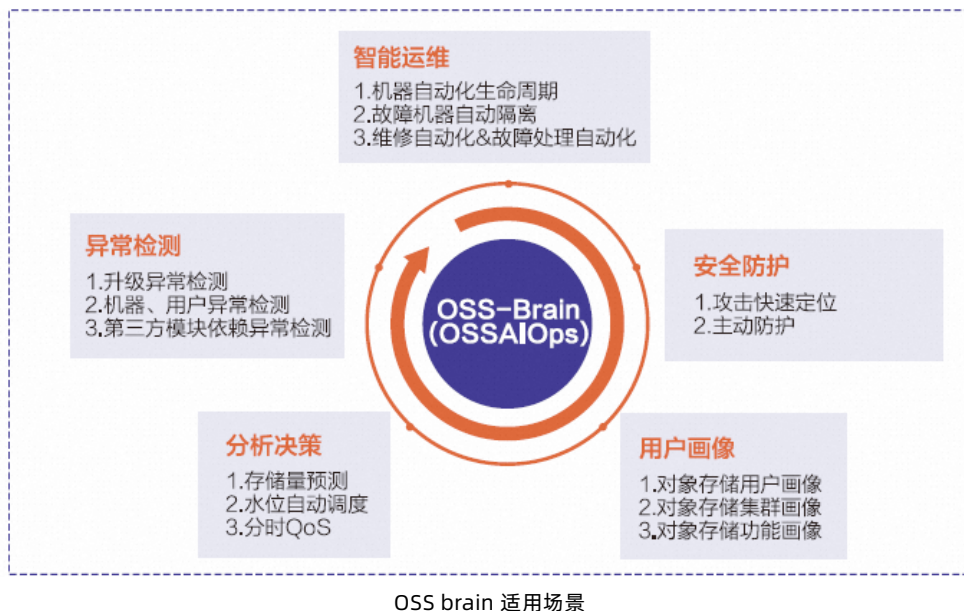
对象存储要承接海量的访问请求, 在接入层采用了负载均衡, 通过绑定 VIP 提供高可用服务, 并且和前端机集群对接, 任何模块故障都能快速切换, 保证可用性, 同时基于阿里云的应用型负载均衡 (Application Load Balancer, ALB) 技术, 具备大流量、高性能访问能力。

对象存储提供 HTTP/HTTPS 的数据访问服务, 会受到来自互联网和 VPC 网络的安全攻击, 典型代表为分布式阻断服务 (Distributed Denial of Service, DDoS), 做好安全攻击防护是保障可用性的重要工作。

安全攻击的一个目的, 就是让对象存储之上的业务遭受损失, 让整体的可用性降低。安全攻击的两种策略, 就是拥塞对象存储有限的带宽入口 (拥塞带宽)、耗尽计算资源。细化的安全攻击分类, 如下表所示。

攻击分类	流量/资源型攻击	慢速攻击
网络层攻击	ICMP/IGMP Flood	
传输层攻击	UDP-Flood TCP连接-Flood PSH+ACK-Flood ACK 反射攻击 RST-Flood SSL-Flood	
应用层攻击	DNS QUERY Flood DNS NXDOMAIN Flood DNS 放大攻击 NTP 放大攻击 SNMP 放大攻击 HTTP/HTTPS-Flood(CC)	Slowloris攻击 慢速POST/READ请求攻击

OSS-Brain 是应对安全攻击的智能运维平台, 其目标和使命是用数据结合算法来保障系统稳定运行, 赋能线上运维及运营。它通过分析线上数据, 提供智能决策, 包括机器隔离、线上主动预警、用户画像、异常检测、资源调度、用户隔离等, 如下图所示。



对象存储是区域级服务，区域故障将会导致服务不可用。为了提供更高的业务可用性，对象存储提供了异地多活架构的高可用解决方案，利用跨地域复制能力，将数据复制到备区域，从而使得备区域有全量的数据。

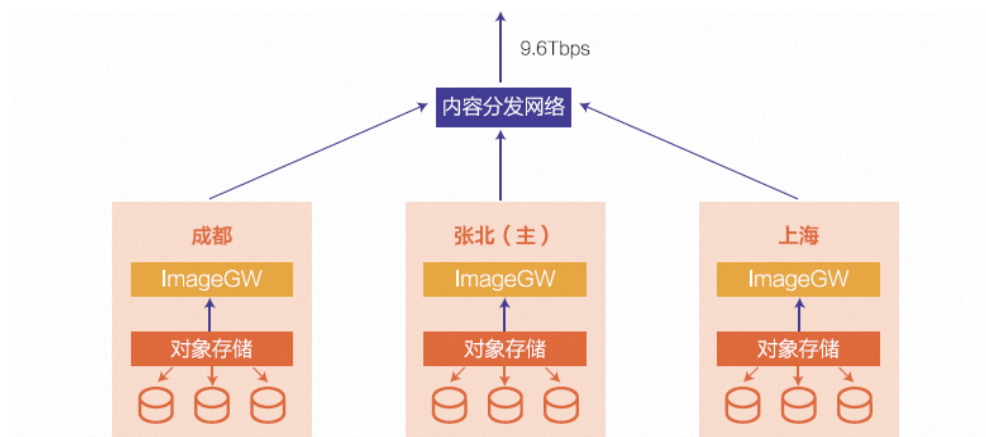
读取时可根据地域就近读取，降低延时。由于写入时只写数据到主区域，数据是异步复制到备区域的，所以用户读取备区域的数据时，可能数据还未复制完成，此时可通过对象存储镜像回源功能从主区域读取数据，在不同的区域级故障场景中快速实现数据读取，达到秒级复原点目标，保证业务应用的连续性。

若备地域不可用，上层业务则快速切换到另外两个地域，并将流量均分，业务能立即恢复，切换也非常方便。

若主地域不可用，则选择新的主地域，并开通跨域复制，从而业务可以将写请求切换到新的主地域，读请求也切换到剩下的地域。同时，基于对象存储的版本控制和业务无更新写等功能，实现了主地域故障切换的数据一致性。

异地多活架构

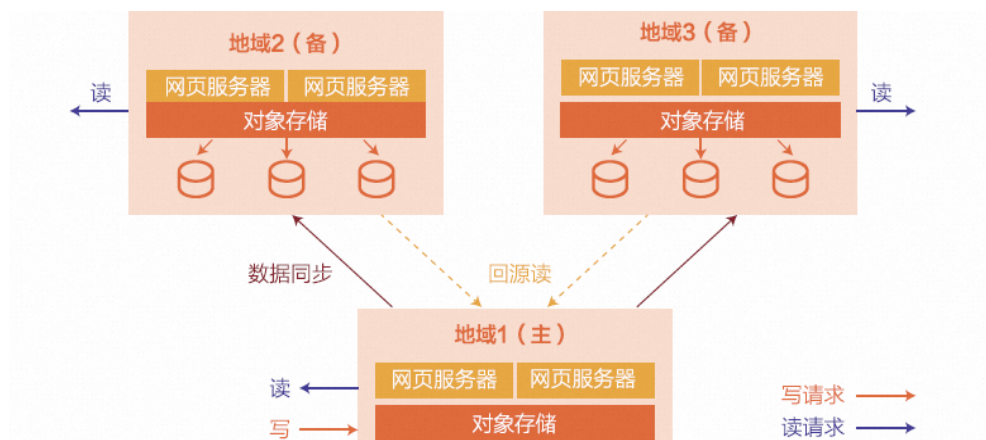
作为一个分布式的存储，对象存储底层是采用多副本方式实现的，以便在一个可用区内，保证每个写入的对象都有极高的可靠性。虽然硬盘故障、单机异常等情况不会影响到数据可用性和可靠性，但是在面对机房级别故障（如停电、网络异常）或自然灾害（如地震、海啸）等导致的一个数据中心无法提供服务时，使用该数据中心的客户服务还是会受到影响的，因此异地容灾能力不可或缺，当某集群发生异常时，需要快速切换服务，保障业务的可用性。



淘宝图片业务异地多活架构

上图所示的是淘宝图片业务异地多活架构，数据中心分别位于张北、上海、成都三地，对于淘宝上的所有商品图片，三地都会有一份全量的数据。正常情况下，用户浏览图片，内容分发网络（Content Delivery Network, CDN）就近回源到各个地域（Region）业务层图片空间 ImageGW 应用，ImageGW 通过从同地域对象存储读取数据进行业务逻辑处理后返回内容分发网络，展示到客户端。这个过程中三地都是多活提供服务的。当某个地域发生异常时，通过切换内容分发网络回源，快速将流量调度到其他两地，保障服务的高可用。

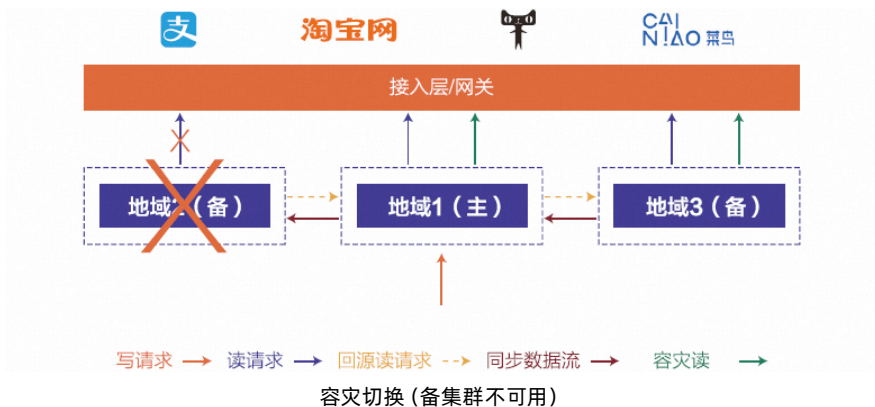
1. 异地多活容灾实现细节



一写多读模式

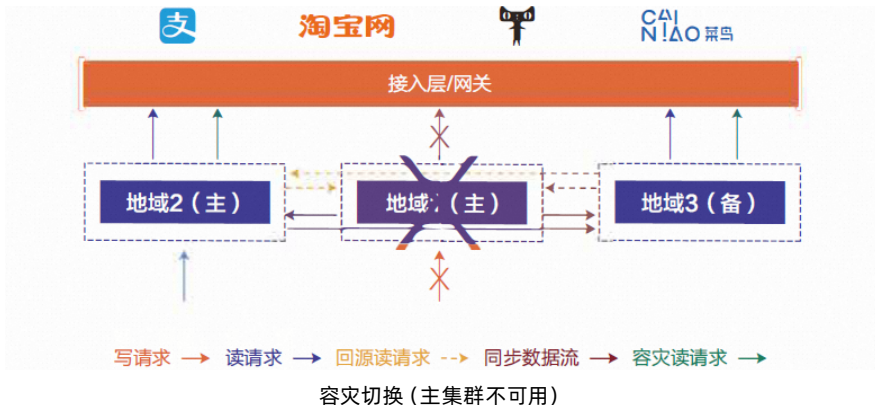
读取时,根据地域就近读取,降低延时。由于写入时只写一份数据到主集群,数据是异步复制到备集群的,所以用户读备集群数据时,可能还有数据没来得及复制到备集群,导致读取不到。这时,通过镜像回源读功能,可以直接从主集群回源读取数据,达到主备集群都能实时读取数据的目的。

2. 容灾切换（备集群不可用）



当备集群不可用时,对于内容分发网络读流量,只需要将内容分发网络回源从地域 2 备集群切换到地域 1 和地域 3,流量均分,业务就能立刻恢复。同理,将业务读对象存储的桶数据切换到其他两个集群,也能达到容灾的目的。

3. 容灾切换（主集群不可用）



当主集群不可用时,这时就需要切换写了。将地域 2 设置为目标主集群,首先需要开通从地域 2 到地域 3 的跨地域复制,客户端通过更换写入的桶配置,切换写到地域 2,通过数据复制技术,将数据同步到地域 3 了。对于读来说,由于地域 1 主集群已不可用,镜像回源配置就需要切换到新主集群地域 2 了。

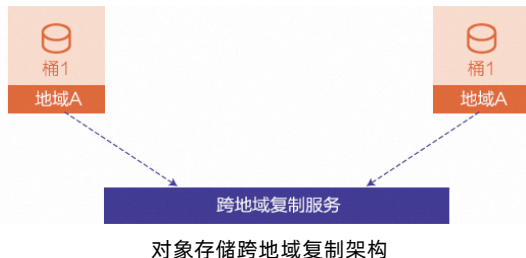
此时和上述备集群不可用时切换读方式一样,通过切换内容分发网络回源配置和用户读对象存储的桶配置,即可将业务读切换到其他两个正常的集群。对于线上淘宝图片业务,阿里云已经预先开通好了各个地域互相复制的能力。在切换写时,客户端可直接切换写入的桶,不需要再配置跨地域复制关系,将切换过程尽量简化。

在主集群故障,但并没有受到毁灭性灾难摧毁时(如只是停电、断网导致暂时不可用,而不是地震、海啸等导致数据中心损毁的情况),对于刚写入主集群还未被复制到备集群的数据,在故障期间并不会丢失,集群恢复后这部分数据还是会被复制到备集群。

由于容灾切换主集群后,用户可能在新集群写入了数据,这部分数据可能会和容灾切换前写入主集群的数据有相同的名称,所以就可能会产生冲突。对象存储系统同步只保证最终一致性(按照文件写入的先后顺序),如果用户对同一个文件有覆盖操作,并且对中间结果有依赖,那就要么建议用户开启多版本功能,要么建议不要让文件名重复。在淘宝图片空间场景中,文件名由客户端唯一生成,不会重复,容灾切换和恢复时也不会存在文件冲突的场景。

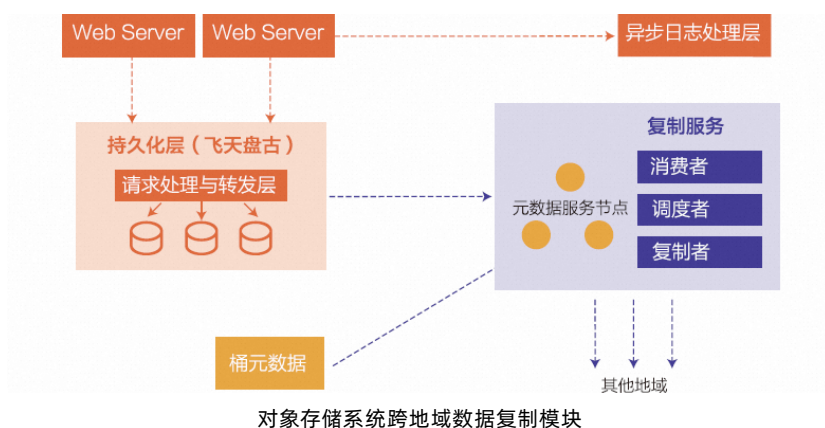
OSS 跨区域复制架构

系统中最重要的部分是数据同步部分,也就是对象存储跨地域复制(Cross-Region Bucket Replication),其架构如下图所示,通过跨不同数据中心,针对用户的数据,进行异步的复制,将对源端数据的改动,如新建、覆盖、修改、删除等,同步到另一个地域中去。目标地域的数据是源端数据的精确副本,具有相同的名称、创建时间、拥有者、用户自定义的元数据、对象内容等。



简而言之,一旦用户为桶配置了跨地域复制功能之后,任何对象在上传到开启这个功能的桶时,对象存储都会自动地为其在用户所指定的另一个地域中的桶里进行备份。当用户修改对象内容或属性时,对象存储也会自动复制,始终保证源桶和目标桶中的对象一致,且完全不需要用户干预。对象存储数据复制(如下图所示)

包含四大模块：请求处理与转发层 (Process Layer)、持久化层 (飞天盘古)、异步日志处理层 (Scanner)、复制服务 (Replication Service)。



用户的请求从 Web Server 进入之后,经过处理(如协议解析、权限验证、切片、校验)持久化到飞天盘古系统。同时,对象存储在写入数据时,会记录一条日志,后续由异步日志处理层会对这些日志进行异步扫描,产生复制事件。

复制服务会在后台运行,它是一个分布式的服务,分为主节点 (Master) 和工作节点 (Worker)。主节点处理桶元数据信息,调度任务到不同的工作节点上执行。工作节点分为三种:一种是消费复制事件的工作节点,叫作消费者 (Consumer);一种是调度复制事件的工作节点,叫作调度者 (Scheduler);一种是复制数据的工作节点,叫作复制者 (Replicator)。当有复制事件产生时,消费者会取出该事件,解析验证,交由调度者负责调度;调度者会分配任务到某个复制数据的复制者去处理,同时负责资源控制、均衡压力;复制者就将数据复制到另外一个地域中去,它会负责数据的切分、传输及校验。

那么,在这样的架构下,如何实现海量数据的快速复制呢?

首先,系统中所有工作节点,包括消费者、调度者、复制者都是无状态的,当发生失效转移 (Failover) 时,不用依赖于上下文信息,可以迅速地重新加载。如果没恢复,主节点会将其上的任务调度到其他工作节点上执行,不会导致某些任务无法运行。

其次,包括扫描日志,以及对复制任务做分发到复制任务的执行在内的所有流程都是解耦并异步执行的,充分利用了系统资源的能力。

再次,所有的任务都是哈希打散开的,包括事件消费、分发和复制任务都是均匀分布到多个工作节点上执行,没有单点瓶颈。

最后,数据复制采用的是流式复制方式,数据不落盘,从源端获取到一个缓存后就开始复制,进一步提升效率。

管理机制

除了上述的各种技术保障外, 还有如下的管理机制来提升可用性。

库存管理

公共云服务需要自己管理供应链库存, 智能预测资源需求, 按需提供服务是可用性的基本保证。

水位管理

对象存储是云存储服务, 监控容量水位、带宽水位、每秒查询率 (Queries-per-second, QPS) 等水位能力, 进行动态的智能调度, 可以优化系统的可用性。

稳定性文化

从开发、设计、测试、运维等环节制定稳定性制度, 追求卓越的可用性能力。