

18 | 分布式存储：你知道对象存储是如何保存图片文件的吗？

李玥 · 后端存储实战课



你好，我是李玥。

我们都知道，保存像图片、音视频这类大文件，最佳的选择就是对象存储。对象存储不仅有很好的大文件读写性能，还可以通过水平扩展实现近乎无限的容量，并且可以兼顾服务高可用、数据高可靠这些特性。

对象存储之所以能做到这么“全能”，最主要的原因是，**对象存储是原生的分布式存储系统**。这里我们讲的“原生分布式存储系统”，是相对于 MySQL、Redis 这类单机存储系统来说的。虽然这些非原生的存储系统，也具备一定的集群能力，但你能感受到，用它们构建大规模分布式集群的时候，其实是非常不容易的。

随着云计算的普及，很多新生代的存储系统，都是原生的分布式系统，它们一开始设计的目标之一就是分布式存储集群，比如说 [Elasticsearch](#)、[Ceph](#) 和国内很多大厂推出的新一代数据库，大多都可以做到：

近乎无限的存储容量；

超高的读写性能；

数据高可靠：节点磁盘损毁不会丢数据；

实现服务高可用：节点宕机不会影响集群对外提供服务。

那这些原生分布式存储是如何实现这些特性的呢？

实际上不用我说，你也能猜得到，这里面同样存在严重的“互相抄作业”的情况。这个也可以理解，除了存储的数据结构不一样，提供的查询服务不一样以外，这些分布式存储系统，它们面临的很多问题都是一样的，那实现方法差不多也是可以理解。

对象存储它的查询服务和数据结构都非常简单，是最简单的原生分布式存储系统。这节课，我们就来一起来研究一下对象存储这种最简单的原生分布式存储，通过对象存储来认识一下分布式存储系统的一些共性。掌握了这些共性之后，你再去认识和学习其他的分布式存储系统，也会感觉特别容易。

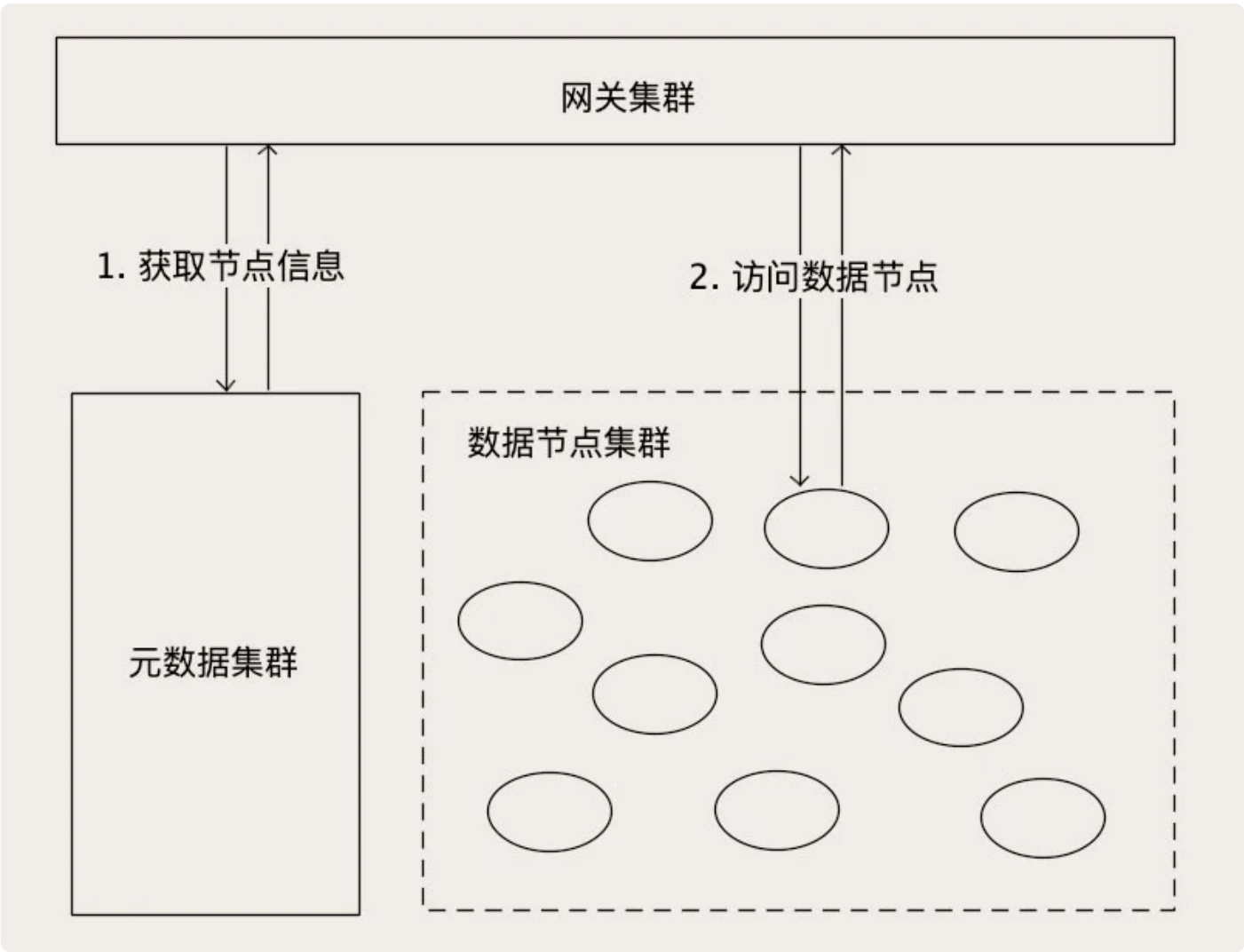
对象存储数据是如何保存大文件的？

对象存储对外提供的服务，其实就是一个近乎无限容量的大文件 KV 存储，所以对象存储和分布式文件系统之间，没有那么明确的界限。对象存储的内部，肯定有很多的存储节点，用于保存这些大文件，这个就是数据节点的集群。

另外，我们为了管理这些数据节点和节点中的文件，还需要一个存储系统保存集群的节点信息、文件信息和它们的映射关系。这些为了管理集群而存储的数据，叫做元数据 (Metadata)。

元数据对于一个存储集群来说是非常重要的，所以保存元数据的存储系统必须也是一个集群。但是元数据集群存储的数据量比较少，数据的变动不是很频繁，加之客户端或者网关都会缓存一部分元数据，所以元数据集群对并发要求也不高。一般使用类似 [🔗 ZooKeeper](#) 或者 [🔗 etcd](#) 这类分布式存储就可以满足要求。

另外，存储集群为了对外提供访问服务，还需要一个网关集群，对外接收外部请求，对内访问元数据和数据节点。网关集群中的每个节点不需要保存任何数据，都是无状态的节点。有些对象存储没有网关，取而代之的是客户端，它们的功能和作用都是一样的。



那么，对象存储是如何来处理对象读写请求的呢？这里面处理读和写请求的流程是一样的，我们来说。网关收到对象读写请求后，首先拿着请求中的 Key，去元数据集群查找这个 Key 在哪个数据节点上，然后再去访问对应的数据节点读写数据，最后把结果返回给客户端。

以上是一个比较粗略的大致流程，实际上这里面包含很多的细节，我们暂时没有展开讲。目的是让你在整体上对对象存储，以至于分布式存储系统，有一个清晰的认知。

上面这张图，虽然我画的是对象存储集群的结构，但是把图上的名词改一改，完全可以套用到绝大多数分布式文件系统和数据库上去，比如说 HDFS。

对象是如何拆分和保存的？

接下来我们说一下对象存储到底是如何来保存大文件对象的。一般来说，对象存储中保存的文件都是图片、视频这类大文件。在对象存储中，每一个大文件都会被拆成多个大小相等的**块儿 (Block)**，拆分的方法很简单，就是把文件从头到尾按照固定的块儿大小，切成一块儿一块儿，最后一块儿长度有可能不足一个块儿的大小，也按一块儿来处理。块儿的大小一般配置为几十 KB 到几个 MB 左右。

把大对象文件拆分成块儿的目的有两个：

1. 第一是为了提升读写性能，这些块儿可以分散到不同的数据节点上，这样就可以并行读写。
2. 第二是把文件分成大小相等块儿，便于维护管理。

对象被拆成块儿之后，还是太过于碎片化了，如果直接管理这些块儿，会导致元数据的数据量会非常大，也没必要管理到这么细的粒度。所以一般都会再把块儿聚合一下，放到块儿的容器里面。这里的“容器”就是存放一组块儿的逻辑单元。容器这个名词，没有统一的叫法，比如在 [ceph](#) 中称为 Data Placement，你理解这个含义就行。容器内的块儿数大多是固定的，所以容器的大小也是固定的。

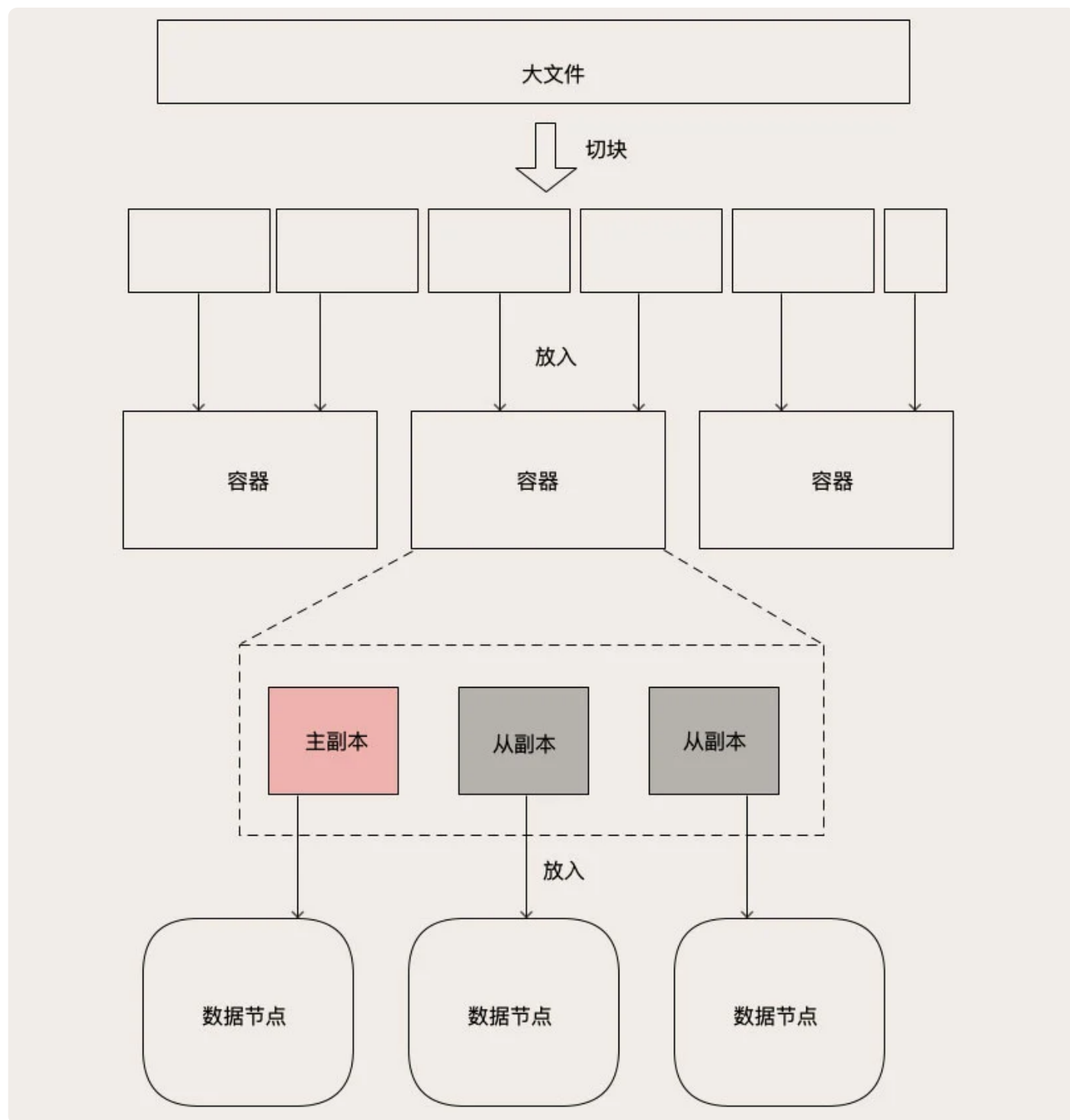
到这里，这个容器的概念，就比较类似于我们之前讲 MySQL 和 Redis 时提到的“分片”的概念了，都是复制、迁移数据的基本单位。每个容器都会有 N 个副本，这些副本的数据都是一样的。其中有一个主副本，其他是从副本，主副本负责数据读写，从副本去到主副本上去复制数据，保证主从数据一致。

这里面有一点儿和我们之前讲的不一样的是，对象存储一般都不记录类似 MySQL 的 Binlog 这样的日志。主从复制的时候，复制的不是日志，而是整块儿的数据。这么做有两个原因：

1. 第一个原因是基于性能的考虑。我们知道操作日志里面，实际上就包含着数据。在更新数据的时候，先记录操作日志，再更新存储引擎中的数据，相当于在磁盘上串行写了 2 次数据。对于像数据库这种，每次更新的数据都很少的存储系统，这个开销是可以接受的。但是对于对象存储来说，它每次写入的块儿很大，两次磁盘 IO 的开销就有些不太值得了。

2. 第二个原因是它的存储结构简单，即使没有日志，只要按照顺序，整块儿的复制数据，仍然可以保证主从副本的数据一致性。

以上我们说的对象（也就是文件）、块儿和容器，都是逻辑层面的概念，数据落实到副本上，这些副本就是真正物理存在了。这些副本再被分配到数据节点上保存起来。这里的数据节点就是运行在服务器上的服务进程，负责在本地磁盘上保存副本的数据。



了解了对象是如何被拆分并存储在数据节点上之后，我们再来回顾一下数据访问的流程。当我们请求一个 Key 的时候，网关首先去元数据中查找这个 Key 的元数据。然后根据元数据中记录的对象长度，计算出对象有多少块儿。接下来的过程就可以分块儿并行处理了。对于每个块儿，还需要再去元数据中，找到它被放在哪个容器中。

我刚刚讲过，容器就是分片，怎么把块儿映射到容器中，这个方法就是我们在《[15 | MySQL 存储海量数据的最后一招：分库分表](#)》这节课中讲到的几种分片算法。不同的系统选择实现的方式也不一样，有用哈希分片的，也有用查表法把对应关系保存在元数据中的。找到容器之后，再去元数据中查找容器的 N 个副本都分布在哪些数据节点上。然后，网关直接访问对应的数据节点读写数据就可以了。

小结

对象存储是最简单的分布式存储系统，主要由数据节点集群、元数据集群和网关集群（或者客户端）三部分构成。数据节点集群负责保存对象数据，元数据集群负责保存集群的元数据，网关集群和客户端对外提供简单的访问 API，对内访问元数据和数据节点读写数据。

为了便于维护和管理，大的对象被拆分为若干固定大小的块儿，块儿又被封装到容器（也就分片）中，每个容器有一主 N 从多个副本，这些副本再被分散到集群的数据节点上保存。

对象存储虽然简单，但是它具备一个分布式存储系统的全部特征。所有分布式存储系统共通的一些特性，对象存储也都具备，比如说数据如何分片，如何通过多副本保证数据可靠性，如何在多个副本间复制数据，确保数据一致性等等。

希望你通过这节课的学习，不仅是学会对象存储，还要对比分析一下，对象存储和其他分布式存储系统，比如 MySQL 集群、HDFS、Elasticsearch 等等这些，它们之间有什么共同的地方，差异在哪儿。想通了这些问题，你对分布式存储系统的认知，绝对会上升到一个全新的高度。然后你再去看一些之前不了解的存储系统，就非常简单了。

思考题

我们刚刚说到过，对象存储并不是基于日志来进行主从复制的。假设我们的对象存储是一主二从三个副本，采用半同步方式复制数据，也就是主副本和任意一个从副本更新成功后，就给客

户端返回成功响应。主副本所在节点宕机之后，这两个从副本中，至少有一个副本上的数据是和宕机的主副本上是一样的，我们需要找到这个副本作为新的主副本，才能保证宕机不丢数据。

但是没有了日志，如果这两个从副本上的数据不一样，我们如何确定哪个上面的数据是和主副本一样新呢？欢迎你在留言区与我交流讨论。

感谢你的阅读，如果你觉得今天的内容对你有帮助，也欢迎把它分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (27)



李玥 置顶

2020-04-07

Hi，我是李玥。

这里回顾一下上节课的思考题：

如果出现缓存不同步的情况，在你负责的业务场景下，该如何降级或者补偿？

这个问题我看到留言区有一些同学给出了非常好的答案，比如说，设置一个合理的缓存过期时间，这样即使出现缓存不同步，等缓存过期后就会自动恢复。再比如，识别用户手动刷新操作，强制重新加载缓存数据（但要注意防止大量缓存穿透）。还可以在管理员的后台系统中，预留一个手动清除缓存的功能，必要的时候人工干预。

共 5 条评论 >

👍 34



L

2020-04-10

老师，我问一个小问题。公司是做素材库的，现在自建对象服务器，对象服务器里面大多都是图片素材，场景是读多写少。选择Ceph可以用于生成环境吗？或者有没有更好的方案选择？谢谢老师

作者回复：建议你使用公有云的对象存储服务，小规模的公司自建对象存储维护成本太高，不是太划算。

共 3 条评论 >

👍 20



每天晒白牙

2020-04-07

如果是基于 Raft 协议的，会根据任期的编号大小决定谁是领导者



👍 18



喆里

2020-05-25

请教个问题，为什么分块后又聚合到容器中，直接一个容器一个块不行吗？这块设计思路没看明白

作者回复: 一个容器就是一个分片，是数据复制的基本单位。也就是说，每个分片都有n个 副本。

分片不能做的太小的原因是，分片越小，意味着存储同样容量的数据，分片的数量越多。分片数量过多，查找分片时，需要查找的元数据就会太多，影响查找效率。

另外，对于数据复制，同样要有一定的开销，比如记录日志位置，维护数据一致性的开销。分片太小，相对的，这些开销就会比较大。



👍 17



于海涛

2020-05-15

老师，有一个问题想问您，对象存储的cdn缓存是怎么做的？是每次要访问这些元数据，还是直接把这些源数据所有都放在内存里？数据量这么大感觉不适合放内存里吧？感谢老师

作者回复: CDN缓存的文件一般是保存在CDN节点的磁盘上，当然不排除某些CDN会用节点的内存缓存文件，加速访问。

共 2 条评论 >

👍 9



Kevin Wang

2020-07-31

老师讲得很好，提一点建议。

数据冗余技术主要由两种：

1. 传统副本复制

2. 纠删码，基于纠删算法，时间换空间（著名开源对象存储MinIO就是基于纠删码的）

建议文中也提一下第2种。

共 2 条评论 >

👍 7



一步

2020-04-07

对象存储 不是一般都有版本控制的吗？ 最新的版本就是最新的数据

共 1 条评论 >

👍 6



世纪猛男

2020-04-08

Zookeeper transaction id



👍 3



凯文小猪

2021-12-15

有些同学对于为什么不适用操作日志可能有疑惑 这里简单说一下：

1.文件系统为了保证一致性与原子性 需要使用journal（又被称为write ahead logging,即WAL）技术来实现。

写入时或者更新时 要先写index node日志 数据日志，而这些对于文件系统来说会有两个问题：

a)通常文件系统保存的文件都很大 所以无论怎么分割块 index node 天生就多，这样journal 日志写入反而是个负担

b)分布式存储不像文件系统需要支持很多的搜索场景 所以内部存储不是用B树 而是KV，故写入反而简单，如果同步失败也可以快速发现重做即可

2.binlog文件的增大 会导致每次写操作必须保证原子性 因为宿主机linux文件系统还是以一页16KB来做一个原子写，这就会衍生出其他的问题。

最后回答下老师的问题

可以使用版本号来维护 将其作为epoch，在根据其大小用raft选主即可



👍 3



xzy

2020-10-26

请假下老师，对象存储和分布式文件系统的区别在哪？另外对象存储底层是否也是依赖的文件系统，谢谢

共 1 条评论 >

👍 2



J.Smile

2020-04-17

老师，有个地方不清楚，这里的每个容器一般只存储对象的一部分块，那么首先应该是从元数据集群中根据key找到这个对象对应的所有容器，第二步，因为多个容器中的块组成一个完整对象，而每个容器又被存到某个数据节点中了，所以此时应该再去元数据集群中找多个容器所对应的数据节点。总之，只要找到了容器的存储位置，容器内块就找到了。文章中提到的块的聚合指的是容器来聚合块吗？这样元数据集群就是管理容器了。容器内存的是块的索引还是实际数据呢？

作者回复: 容器内存放的就是实际的数据。

共 2 条评论 >

👍 2



Jxin

2020-04-07

每次操作都更新操作时间。谁的操作时间最新谁就是最接近主节点的副节点。

作者回复: 那这个时间从哪儿来呢？如果都从本机的时钟获取，怎么保证集群中所有节点的时钟是一致的呢？

共 2 条评论 >

👍 3



ifelse

2022-12-13 来自浙江

对象存储之所以能做到这么“全能”，最主要的原因是，对象存储是原生的分布式存储系统。——记下来



👍 2



林铭铭

2021-07-29

通俗易懂



👍 1



特种流氓

2020-04-07

容器里面的主副本信息存放在元数据集还是数据管理集群
对于容器的概念还是有点模糊



Mq

2020-04-07

可以设置2个从节点监听主节点的心跳时间不同，可以把跟主节点一致的设置的小些，当监控到主节点挂了后发起选举升级成主节点。



ifelse

2022-12-13 来自浙江

学习打卡



Geek_dc693e

2022-04-19

没看懂呀，这个容器是一些块的集合？类似分片，就是说每个容器都有全量数据？那这个容器有和一个副本是啥区别



Jay

2022-03-28

为何需要网关，这个能否解释一下



千锤百炼领悟之极限

2022-03-03

一个大文件会先拆分为多个大小相等的块，最后一块长度不足，也当做一个相同长度的块。

把这些文件块，按指定数量，放到若干个分片中。

每个分片为了高可用，都会有主副本，从副本，每个副本对应数据节点。

读取文件时，先用key找到所有的分片，然后通过分片找到分片副本，再通过分片副本找到对应的数据节点获取数据。最后把多个分片的数据整合，返回。

