26 | 高可用存储架构：集群和分区

2018-06-26 李运华

上一期我讲了高可用存储架构中常见的双机架构，分别为主备复制、主从复制、双机切换和主主复制，并分析了每类架构的优缺点以及适应场景。

今天我们一起来看看另外两种常见的高可用存储架构：数据集群和数据分区。

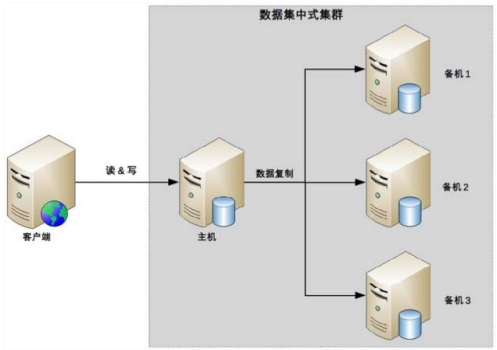
数据集群

主备、主从、主主架构本质上都有一个隐含的假设：主机能够存储所有数据，但主机本身的存储和处理能力肯定是有极限的。以PC为例，Intel 386时代服务器存储能力只有几百MB，Intel 奔腾时代服务器存储能力可以有几十GB，Intel 酷睿多核时代的服务器可以有几个TB。单纯从硬件发展的角度来看，似乎发展速度还是挺快的，但如果和业务发展速度对比，那就差得远了。早在2013年，Facebook就有2500亿张上传照片，当时这些照片的容量就已经达到了250 PB字节（250 × 1024TB），平均一天上传的图片有3亿5000万张。如此大量的数据，单台服务器肯定是无法存储和处理的，我们必须使用多台服务器来存储数据，这就是数据集群架构。

简单来说，集群就是多台机器组合在一起形成一个统一的系统，这里的“多台”，数量上至少是3台；相比而言，主备、主从都是2台机器。根据集群中机器承担的不同角色来划分，集群可以分为两类：数据集中集群、数据分散集群。

1.数据集中集群

数据集中集群与主备、主从这类架构相似，我们也可以称数据集中集群为1主多备或者1主多从。无论是1主1从、1主1备，还是1主多备、1主多从，数据都只能往主机中写，而读操作可以参考主备、主从架构进行灵活多变。下图是读写全部到主机的一种架构：



虽然架构上是类似的，但由于集群里面的服务器数量更多，导致复杂度整体更高一些，具体体现在：

- 主机如何将数据复制给备机

主备和主从架构中，只有一条复制通道，而数据集中集群架构中，存在多条复制通道。多条复制通道首先会增大主机复制的压力，某些场景下我们需要考虑如何降低主机复制压力，或者降低主机复制给正常读写带来的压力。

其次，多条复制通道可能会导致多个备机之间数据不一致，某些场景下我们需要对备机之间的数据一致性进行检查和修正。

- 备机如何检测主机状态

主备和主从架构中，只有一台备机需要进行主机状态判断。在数据集中集群架构中，多台备机都需要对主机状态进行判断，而不同的备机判断的结果可能是不同的，如何处理不同备机对主机状态的不同判断，是一个复杂的问题。

- 主机故障后，如何决定新的主机

主从架构中，如果主机故障，将备机升级为主机即可；而在数据集中集群架构中，有多台备机都可以升级为主机，但实际上只能允许一台备机升级为主机，那么究竟选择哪一台备机作为新的主机，备机之间如何协调，这也是一个复杂的问题。

目前开源的数据集中集群以ZooKeeper为典型，ZooKeeper通过ZAB算法来解决上述提到的几个问题，但ZAB算法的复杂度是很高的。

2.数据分散集群

数据分散集群指多个服务器组成一个集群，每台服务器都会负责存储一部分数据；同时，为了提升硬件利用率，每台服务器又会备份一部分数据。

数据分散集群的复杂点在于如何将数据分配到不同的服务器上，算法需要考虑这些设计点：

- 均衡性

算法需要保证服务器上的数据分区基本是均衡的，不能存在某台服务器上的分区数量是另外一台服务器的几倍的情况。

- 容错性

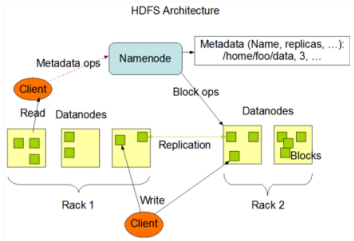
当出现部分服务器故障时，算法需要将原来分配给故障服务器的数据分区分配给其他服务器。

- 可伸缩性

当集群容量不够，扩充新的服务器后，算法能够自动将部分数据分区迁移到新服务器，并保证扩容后所有服务器的均衡性。

数据分散集群和数据集中集群的不同点在于，数据分散集群中的每台服务器都可以处理读写请求，因此不存在数据集中集群中负责写的主机那样的角色。但在数据分散集群中，必须有一个角色来负责执行数据分配算法，这个角色可以是独立的一台服务器，也可以是集群自己选举出的一台服务器。如果是集群服务器选举出来一台机器承担数据分区分配的职责，则这台服务器一般也会叫作主机，但我们需要知道这里的“主机”和数据集中集群中的“主机”，其职责是有差异的。

Hadoop的实现就是独立的服务器负责数据分区的分配，这台服务器叫作Namenode。Hadoop的数据分区管理架构如下：



(https://hadoop.apache.org/docs/r1.0.4/cn/images/hdfsarchitecture.gif)

下面是Hadoop官方的解释，能够说明集中式数据分区管理的基本方式。

> HDFS采用master/slave架构。一个HDFS集群由一个Namenode和一定数目的Datanodes组成。Namenode 是一个中心服务器，负责管理文件系统的名字空间（namespace），以及客户端对文件的访问。 集群中的Datanode一般是一个节点一个，负责管理它所在节点上的存储。HDFS暴露了文件系统的名字空间，用户能够以文件的形式在上面存储数据。从内部看，一个文件其实被分成一个或多个数据块，这些块存储在一组Datanode上。Namenode 执行文件系统的名字空间操作，比如打开、关闭、重命名文件或目录。它也负责确定数据块到具体Datanode节点的映射。Datanode负责处理文件系统客户端的读写请求。在Namenode的统一调度下进行数据块的创建、删除和复制操作。

与Hadoop不同的是，Elasticsearch集群通过选举一台服务器来做数据分区的分配，叫作master node，其数据分区管理架构是：

其中master节点的职责如下：

> The master node is responsible for lightweight cluster-wide actions such as creating or deleting an index, tracking which nodes are part of the cluster, and deciding which shards to allocate to which nodes. It is important for cluster health to have a stable master node.

(https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html)

数据集中集群架构中，客户端只能将数据写到主机；数据分散集群架构中，客户端可以向任意服务器中读写数据。正是因为这个关键的差异，决定了两种集群的应用场景不同。一般来说，数据集中集群适合数据量不大，集群机器数量不多的场景。例如，ZooKeeper集群，一般推荐5台机器左右，数据量是单台服务器就能够支撑；而数据分散集群，由于其良好的可伸缩性，适合业务数据量巨大、集群机器数量庞大的业务场景。例如，Hadoop集群、HBase集群，大规模的集群可以达到上百台甚至上千台服务器。

数据分区

前面我们讨论的存储高可用架构都是基于硬件故障的场景去考虑和设计的，主要考虑当部分硬件可能损坏的情况下系统应该如何处理，但对于一些影响非常大的灾难或者事故来说，有可能所有的硬件全部故障。例如，新奥尔良水灾、美加大停电、洛杉矶大地震等这些极端灾害或者事故，可能会导致一个城市甚至一个地区的所有基础设施瘫痪，这种情况下基于硬件故障而设计的高可用架构不再适用，我们需要基于地理级别的故障来设计高可用架构，这就是数据分区架构产生的背景。

数据分区指将数据按照一定的规则进行分区，不同分区分布在不同的地理位置上，每个分区存储一部分数据，通过这种方式来规避地理级别的故障所造成的巨大影响。采用了数据分区的架构后，即使某个地区发生严重的自然灾害或者事故，受影响的也只是一部分数据，而不是全部数据都不可用；当故障恢复后，其他地区备份的数据也可以帮助故障地区快速恢

复业务。

设计一个良好的数据分区架构，需要从多方面去考虑。

1.数据量

数据量的大小直接决定了分区的规则复杂度。例如，使用MySQL来存储数据，假设一台MySQL存储能力是500GB，那么2TB的数据就至少需要4台MySQL服务器；而如果数据是200TB，并不是增加到800台的MySQL服务器那么简单。如果按照4台服务器那样去平行管理800台服务器，复杂度会发生本质的变化，具体表现为：

- 800台服务器里面可能每周都有一两台服务器故障，从800台里面定位出2台服务器故障，很多情况下并不是一件容易的事情，运维复杂度高。
- 增加新的服务器，分区相关的配置甚至规则需要修改，而每次修改理论上都有可能影响已有的800台服务器的运行，不小心改错配置的情况在实践中太常见了。
- 如此大量的数据，如果在地理位置上全部集中于某个城市，风险很大，遇到了水灾、大停电这种灾难性的故障时，数据可能全部丢失，因此分区规则需要考虑地理容灾。

因此，数据量越大，分区规则会越复杂，考虑的情况也越多。

2.分区规则

地理位置有近有远，因此可以得到不同的分区规则，包括洲际分区、国家分区、城市分区。具体采取哪种或者哪几种规则，需要综合考虑业务范围、成本等因素。

通常情况下，洲际分区主要用于面向不同大洲提供服务，由于跨洲通讯的网络延迟已经大到不适合提供在线服务了，因此洲际间的数据中心可以不互通或者仅仅作为备份；国家分区主要用于面向不同国家的用户提供服务，不同国家有不同语言、法律、业务等，国家间的分区一般也仅作为备份；城市分区由于都在同一个国家或者地区内，网络延迟较低，业务相似，分区同时对外提供服务，可以满足业务异地多活之类的需求。

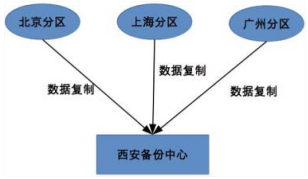
3.复制规则

数据分区指将数据分散在多个地区，在某些异常或者灾难情况下，虽然部分数据受影响，但整体数据并没有全部被影响，本身就相当于一个高可用方案了。但仅仅做到这点还不够，因为每个分区本身的数据量虽然只是整体数据的一部分，但还是很大，这部分数据如果损坏或者丢失，损失同样难以接受。因此即使是分区架构，同样需要考虑复制方案。

常见的分区复制规则有三种：集中式、互备式和独立式。

集中式

集中式备份指存在一个总的备份中心，所有的分区都将数据备份到备份中心，其基本架构如下：

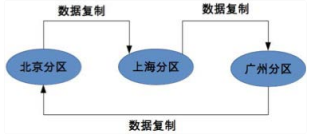


集中式备份架构的优缺点是：

- 设计简单，各分区之间并无直接联系，可以做到互不影响。
- 扩展容易，如果要增加第四个分区（例如，武汉分区），只需要将武汉分区的数据复制到西安备份中心即可，其他分区不受影响。
- 成本较高，需要建设一个独立的备份中心。

互备式

互备式备份指每个分区备份另外一个分区的数据，其基本架构如下：

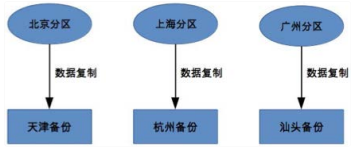


互备式备份架构的优缺点是：

- 设计比较复杂，各个分区除了要承担业务数据存储，还需要承担备份功能，相互之间互相关联和影响。
- 扩展麻烦，如果增加一个武汉分区，则需要修改广州分区的复制指向武汉分区，然后将武汉分区的复制指向北京分区。而原有北京分区已经备份了的广州分区的数据怎么处理也是个难题，不管是做数据迁移，还是广州分区历史数据保留在北京分区，新数据备份到武汉分区，无论哪种方式都很麻烦。
- 成本低，直接利用已有的设备。

独立式

独立式备份指每个分区自己有独立的备份中心，其基本架构如下：



有一个细节需要特别注意，各个分区的备份并不和原来的分区在一个地方。例如，北京分区的备份放到了天津，上海的放到了杭州，广州的放到了汕头，这样做的主要目的是规避同城或者相同地理位置同时发生灾难性故障的极端情况。如果北京分区机房在朝阳区，而备份机房放在通州区，整个北京停电的话，两个机房都无法工作。

独立式备份架构的优缺点是：

- 设计简单，各分区互不影响。
- 扩展容易，新增加的分区只需要搭建自己的备份中心即可。
- 成本高，每个分区需要独立的备份中心，备份中心的场地成本是主要成本，因此独立式比集中式成本要高很多。

小结

今天我为你讲了大数据量存储的两种高可用存储架构：集群架构和分区架构，并介绍了其中的关键设计点，希望对你有所帮助。

这就是今天的全部内容，留一道思考题给你吧，既然数据集群就可以做到不同节点之间复制数据，为何不搭建一个远距离分布的集群来应对地理位置级别的故障呢？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。（编辑乱入：精彩的留言有机会获得丰厚福利哦！）

XNorth

2018-06-26

远距离集群，需要考虑带宽影响。数据量较大，复制成本较高。

郭涛

2018-06-26

数据集群中的节点间需要通信，远距离通信网络延迟高，无法保证集群中节点间的数据一致性；长距离网络传输带宽不比局域网，大量数据复制带宽容易成为瓶颈；且一旦出故障，排查的成本也高。数据集群架构适合应对硬件级的故障，远距离、地理范围的可用性保障采用数据分区。

生活就是态度

2018-06-28

可以讲讲es,kafka这些中间件优化及应用不

作者回复

2018-06-28

专栏的目的在于提炼架构设计中本质和通用的内容，这样以后你看到一个系统就知道怎么去分析和研究，如果只是详细讲解某个系统，换个系统这些东西可能都没用了

空档滑行

2018-06-27

1.远距离集群网络延时较高，而且网络出问题的几率加大，导致数据复制的逻辑会比较复杂
2.成本过高，数据全量复制，等于存储多份
所以更好的办法是从业务端对数据做分区，出现地理故障时只形象一部分用户或者功能的使用

作者回复

2018-06-28

分析到位��

feifei

2018-06-26

1，远距离分布的网络延迟高，将导致集群中的数据数据同步时延很高，如果出现业务远距离的数据同步，业务的时延不然很高，某些极端情况，可能用户得不到响应，影响用户的使用
2，远距离的网络可靠性很难保证，支付宝就因为挖断光缆导致用户不可用
3，成本高，数据中心稀缺的资源是宽带资源，因为是远距离分布，所以宽带的费用会很高

Geek_8242cb

2018-07-17

老师，双中心搭建一个Redis集群不一定能做到高可用，因为redis cluster不能自动区分两个中心，虽然能做到主备节点不在同一台机器，但是做不到主备节点分布于两个中心。或者您有什么好的方法推荐吗？

作者回复

2018-07-18

一般不建议这样做，除非同城双机房，网络延迟能够保证，异地双中心不建议

枫晴 andy

2018-07-08

远距离搭建集群复制理论上可行，但是网络延迟太大，对数据中心的硬件资源，尤其是网络要求较高，成本太大。

Geek_8242cb

2018-07-06

李老师，Redis Cluster算是设计比较好的数据分散集群。目前我们在两个双活的同城中心都搭建了一个Redis集群，但是只启用了其中一个，因为没有找到好的集群同步方法，其实我们希望两个中心的集群数据准实时的同步，您有什么推荐吗？另外，我们的文件服务器也是单中心启动，有没有好的工具可以用来同步？

作者回复

2018-07-09

双活同城中心数据库都可以同步，redis直接也可以同步的呀，直接搭建跨双活同城中心的redis集群就可以。

类似文件服务器，用hdfs这类搭建集群即可，你们可以把双活同城中心当成同机房，除非你们的网络建设做不到这点，那就是伪双活

衣申人

2018-07-04

老师，数据分散集群，数据应该是不一样的吧？关键是分区吧，虽然可能节点会互备数量数据，但不等于数据都相同，不然和数据集中集群有和区别？

作者回复

2018-07-05

嗯，数据分散集群中的节点数据不同，我之前的回复混淆了，参考es就知道了

衣申人

2018-07-03

请问数据分散集群不就是分区吗？文章在论述时的分类和层次上是不是有点重合？还是我理解错了呢？请老师指导

作者回复

2018-07-04

分散集群是地理位置上在同一个机房，集群中的数据一样；分区分布在不同地理位置，且数据不一样

衣申人

2018-07-03

思考题里的跨地域复制集群，不就是异地多活吗？请问我有理解错吗？

作者回复

2018-07-04

异地多活是从业务角度来看的，跨地域复制是异地多活的基础，但不等于异地多活

大光头

2018-07-02

远距离分布式集群网络延迟是一个很大的问题，所以一般不建议

@漆~心endless

2018-06-29

能否引入区块链的相关思想去做数据备份，将非机密数据进行“去中心化”操作，这样能降低硬件成本，以及提高容灾能力。

作者回复

2018-06-30

区块链性能太差��

成功

2018-06-29

远距离大规模集群有以下问题：Ⅰ)成本，硬件成本，组网成本，人力维护成本过高2)性能较之本地集群会有降低3)安全性很难保证

Tom

2018-06-27

用分布式集群和数据分区备份主要区别应该是复制时间点不一样。集群一般是有写入就开始复制，这个时候一般都是业务繁忙的时候，此时进行远程复制会占用太多带宽会影响正常业务。而数据分区备份可以选择非业务繁忙时段比如深夜进行复制。

作者回复

2018-06-28

分区备份也是实时的，不然丢1天的数据，影响也还是很大

问题究竟系边度

2018-06-27

如果采用远距离集群，网络抖动和延时就会对整个集群性能造成影响。 如果出现网络分区，数据出现差别。 同时副本复制的控制会比较复杂

作者回复

2018-06-28

������

云学

2018-06-26

远距离备份的带宽成本很高，复制时间延迟，网络稳定性也是挑战

小小笑儿

2018-06-26

远距离的分布式集群主要的问题可能是延迟吧？毕竟一个集群里的节点都是要提供服务的，而且数据复制的时候应该也有影响？像kafka，数据写入分区的时候可能要等待写入复制分区，如果复制分区的延迟高会影响吞吐。

星火燎原

2018-06-26

远距离的网络场景容易造成丢包和数据延迟，数据丢失和延迟让用户体验度极差

---

narry

2018-06-26

应该是网络延迟太大，并且集群系统不太好使用异步或者批量的的方式来通信，这样对于cp的系统就会出现算法执行效率低，并且容易出现分区，导致系统不可用，对于ap的集群，达到最终一致性的时间长，系统设计变的复杂

若水清茵

2018-06-26

远距离分布式离群机房之间的延迟无法满足集群各个node之间数据传输要求，同时也不方便管理.