24 | 分布式数据存储系统之三要素：顾客、导购与货架

2019-11-20 聂鹏程

讲述：聂鹏程
时长 15:14　大小 13.97M

你好，我是聂鹏程。今天，我来继续带你打卡分布式核心技术。

在上一篇文章中，我们一起学习了 CAP 理论。该理论指出，在分布式系统中，不能同时满足一致性、可用性和分区容错性，指导了分布式数据存储系统的设计。

随着数据量和访问量的增加，单机性能已经不能满足用户需求，分布式集群存储成为一种常用方式。把数据分布在多台存储节点上，可以为大规模应用提供大容量、高性能、高可用、高扩展的存储服务。而，分布式存储系统就是其具体实现。

在今天这篇文章，我将带你学习分布式存储系统的关键三要素，让你对分布式数据存储系统有一个直观的理解。在后面几篇文章中，我会针对这三要素中的关键技术进一步展开，以帮

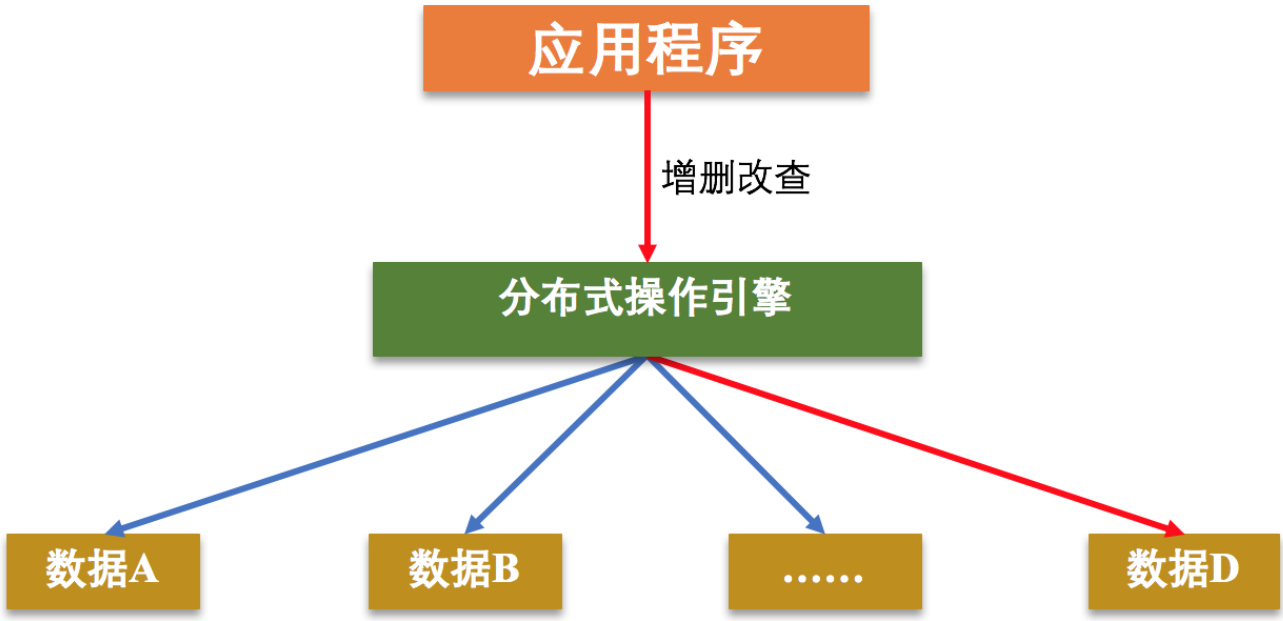
助你更深入地理解分布式数据存储系统。

接下来，我们就一起打卡分布式存储系统的三要素：顾客、导购与货架。

什么是分布式数据存储系统?

分布式存储系统的核心逻辑，就是将用户需要存储的数据根据某种规则存储到不同的机器上，当用户想要获取指定数据时，再按照规则到存储数据的机器里获取。

如下图所示，当用户（即应用程序）想要访问数据 D，分布式操作引擎通过一些映射方式，比如 Hash、一致性 Hash、数据范围分类等，将用户引导至数据 D 所属的存储节点获取数据。



静下心来想一下，获取数据的整个过程与你到商店购物的过程是不是有些类似呢?

顾客到商店购物时，导购会根据顾客想要购买的商品引导顾客到相应的货架，然后顾客从这个货架上获取要购买的商品，完成购物。这里的顾客就是图中的应用程序，导购就相当于分布式操作引擎，它会按照一定的规则找到相应的货架，货架就是存储数据的不同机器节点。

其实，这个过程就是分布式存储系统中获取数据的通用流程，**顾客、导购和货架**组成了分布式存储系统的三要素，分别对应着分布式领域中的**数据生产者 / 消费者、数据索引和数据存储**。

接下来，我们就详细看看这三个要素吧。

分布式数据存储系统三要素

顾客就是数据的生产者和消费者，也就是说顾客代表两类角色，生产者会生产数据（比如，商店购物例子中的供货商就属于生产类顾客），将数据存储到分布式数据存储系统中，消费者是从分布式数据存储系统中获取数据进行消费（比如，商店购物例子中购买商品的用户就属于消费类顾客）；导购就是数据索引，将访问数据的请求转发到数据所在的存储节点；货架就是存储设备，用于存储数据。

顾客：生产和消费数据

顾客相当于分布式存储系统中的应用程序，而数据是应用程序的原动力。根据数据的产生和使用，顾客分为生产者和消费者两种类型。生产者负责给存储系统添加数据，而消费者则可以使用系统中存储的数据。

就像是火车票存储系统，如图所示，铁路局就相当于生产者类型的顾客，而乘客就相当于消费者类型的顾客。铁路局将各个线路的火车票信息发布到订票网站的后台数据库中，乘客通过订票网站访问数据库，来进行查询余票、订票、退票等操作。

生产者和消费者生产和消费的数据通常是多种多样的，不同应用场景中数据的类型、格式等都不一样。**根据数据的特征，这些不同的数据通常被划分为三类：结构化数据、半结构化数据和非结构化数据。**

> 结构化数据通常是指关系模型数据，其特征是数据关联较大、格式固定。火车票信息比如起点站、终点站、车次、票价等，就是一种结构化数据。结构化数据具有格式固定的特征，因此一般采用分布式关系数据库进行存储和查询。
>
> 半结构化数据通常是指非关系模型的，有基本固定结构模式的数据，其特征是数据之间关系比较简单。比如 HTML 文档，使用标签书写内容。半结构化数据大多可以采用键值

对形式来表示，比如 HTML 文档可以将标签设置为 key，标签对应的内容可以设置为 value，因此一般采用分布式键值系统进行存储和使用。

非结构化数据是指没有固定模式的数据，其特征是数据之间关联不大。比如文本数据就是一种非结构化数据。这种数据可以存储到文档中，通过 ElasticSearch（一个分布式全文搜索引擎）等进行检索。

导购：确定数据位置

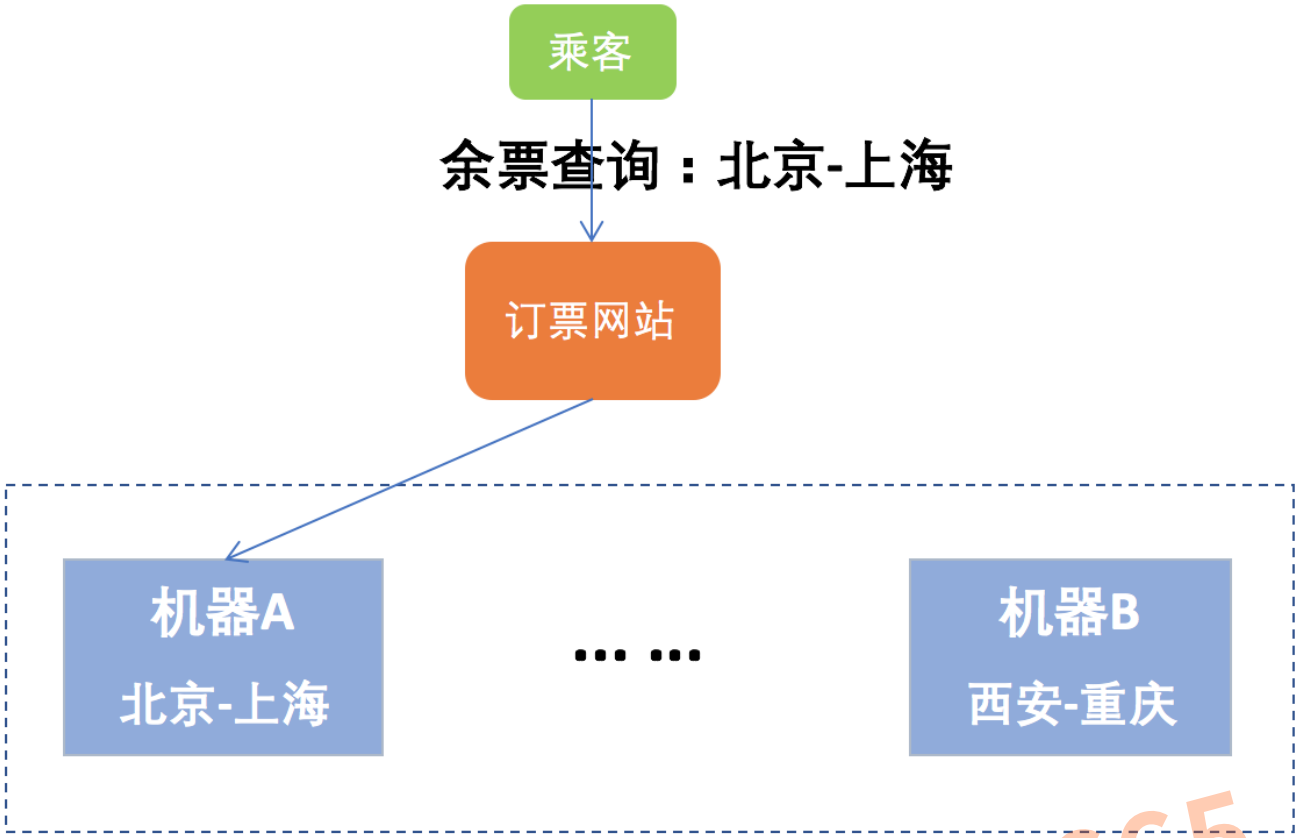
导购是分布式存储系统必不可少的要素，如果没有导购， 顾客就需要逐个货架去寻找自己想要的商品。

想象一下，如果你去订票网站订火车票，按照自己的需求点击查询车票后，系统会逐个扫描分布式存储系统中每台机器的数据，寻找你想要购买的火车票。如果系统中存储的数据不多，响应时间也不会太长，毕竟计算机的速度还是很快的；但如果数据分布在几千台甚至上万台机器中，系统逐个机器扫描后再给你响应，我相信你会对这个订票网站很失望。

这种定位数据存储位置的方式会浪费你很多时间，严重影响购票体验。因此，在分布式存储系统中，必须有相应的数据导购，否则系统响应会很慢，效率很低。为解决这个问题，**数据分片技术**就走入了分布式存储系统的大家庭。

数据分片技术，是指分布式存储系统按照一定的规则将数据存储到相对应的存储节点中，或者到相对应的存储节点中获取想要的数据，这是一种很常用的导购技术。这种技术，一方面可以降低单个存储节点的存储和访问压力；另一方面，可以通过规定好的规则快速找到数据所在的存储节点，从而大大降低搜索延迟，提高用户体验。

也就是说，当铁路局发布各个线路的火车票信息时，会按照一定规则存储到相应的机器中，比如北京到上海的火车票存储到机器 A 中，西安到重庆的火车票存储到机器 B 中。当乘客查询火车票时，系统就可以根据查询条件迅速定位到相对应的存储机器，然后将数据返回给用户，响应时间就大大缩短了。如图所示，当查询北京 - 上海的火车票相关信息时，可以与机器 A 进行数据交互。



这个例子中按照数据起点、终点的方式划分数据，将数据分为几部分存储到不同的机器节点中，就是数据分片技术的一种。当查询数据时，系统可以根据查询条件迅速找到对应的存储节点，从而实现快速响应。

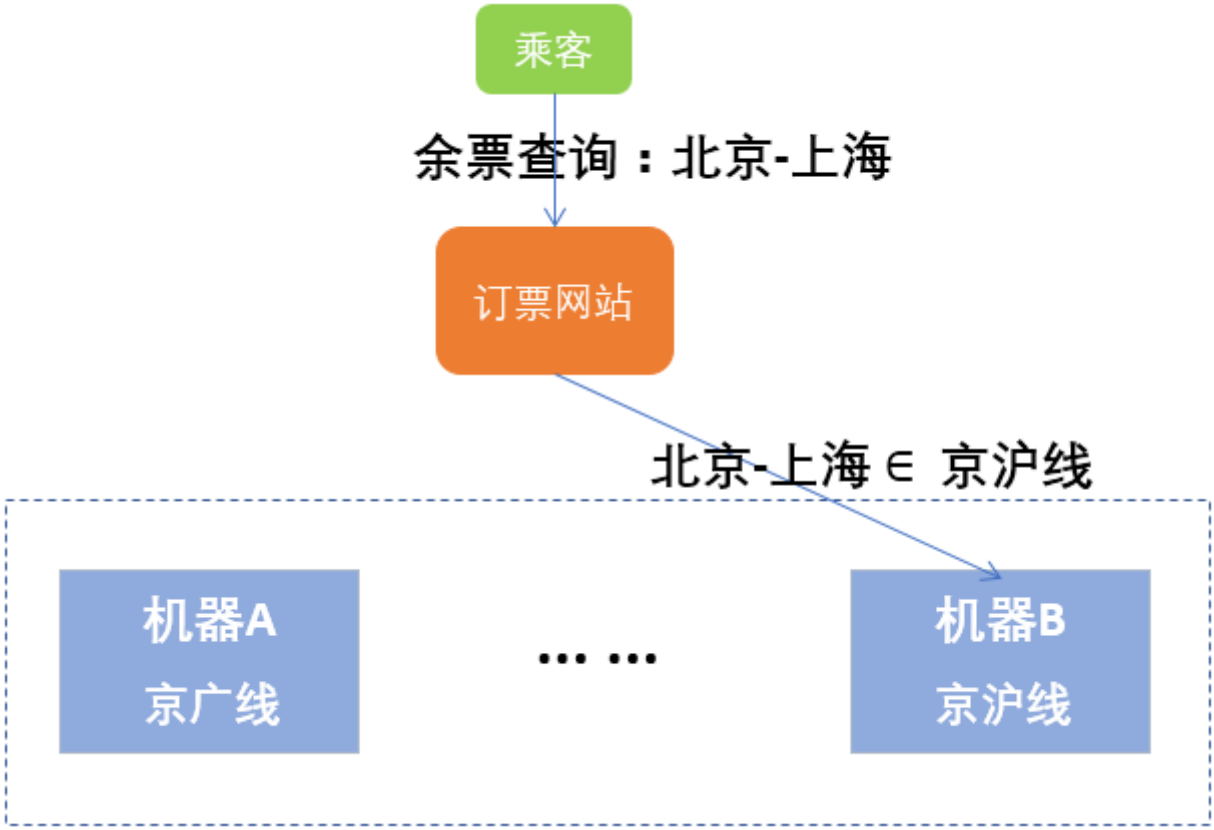
上述的例子中，按照数据特征进行了数据分片，当然，还有其他很多数据分片的方案。比如，按照数据范围，采用哈希映射、一致性哈希环等对数据划分。我会在下一篇文章中，与你详细讲述哈希和一致性哈希的内容。

接下来，我就**针对数据范围这种数据分片方案做一个具体介绍**吧。

针对数据范围的数据分片方案是指，按照某种规则划分数据范围，然后将在这个范围内的数据归属到一个集合中。这就好比数学中通常讲的整数区间，比如 1～1000 的整数，[1,100] 的整数属于一个子集、[101,1000] 的整数属于另一个子集。

对于前面讲的火车票的案例，按照数据范围分片的话，可以将属于某条线的所有火车票数据划分到一个子集或分区进行存储，比如机器 A 存储京广线的火车票数据，机器 B 存储京沪线的火车票数据。也就是说，数据范围的方案是按照范围或区间进行存储或查询。

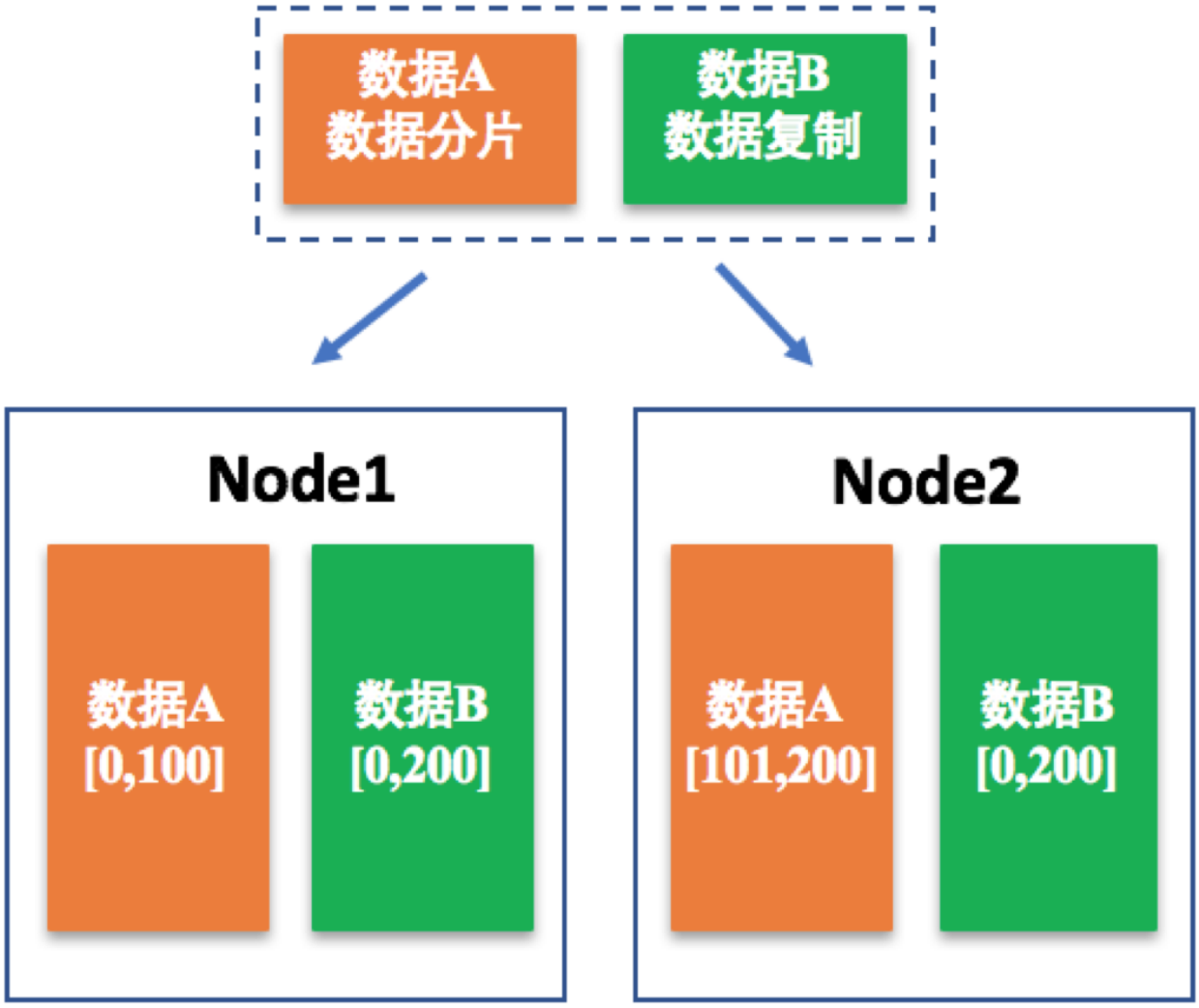
如图所示，当用户查询北京 - 上海的火车票相关信息时，首先判断查询条件属于哪个范围，由于北京 - 上海的火车线路属于京沪线，因此系统按照规则将查询请求转到存取京沪线火车票数据的机器 B，然后由机器 B 进行处理并给用户返回响应结果。



为了提高分布式系统的可用性与可靠性，**除了通过数据分片减少单个节点的压力外，数据复制也是一个非常重要的方法。**数据复制就是将数据进行备份，以使得多个节点存储该数据。

想象一下，当某个存储节点出现故障时，如果只采用数据分片技术，那这个节点的数据就会丢失，从而给用户造成损失。因此，数据复制在分布式存储系统中是不可或缺的。关于数据复制技术，我会在第 26 篇文章中与你详细讲解。

接下来，我与你说说数据复制和数据分片技术的区别吧。关于它们之间的区别，你可以先看看下面这张图片：



数据 A 被拆分为两部分存储在两个节点 Node1 和 Node2 上，属于数据分片；而对数据 B 来说，同一份完整的数据在两个节点中均有存储，就属于数据复制。

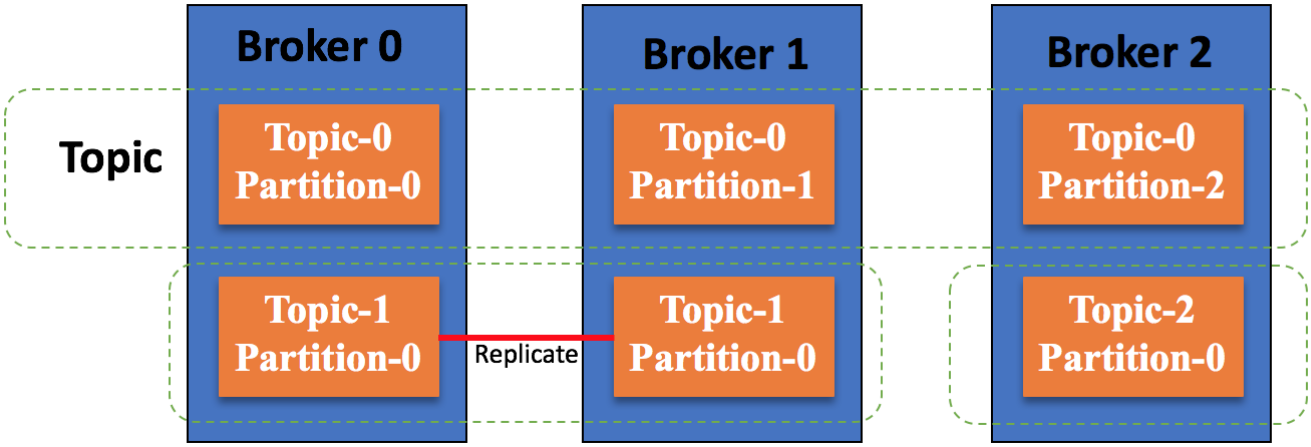
在实际的分布式存储系统中，数据分片和数据复制通常是共存的：

> 数据通过分片方式存储到不同的节点上，以减少单节点的性能瓶颈问题；
>
> 而数据的存储通常用主备方式保证可靠性，也就是对每个节点上存储的分片数据，采用主备方式存储，以保证数据的可靠性。其中，主备节点上数据的一致，是通过数据复制技术实现的。

讲到这里，我们再回忆下第 20 篇文章中涉及的 Kafka 集群的总体架构图吧。我从中抽取出 Kafka 集群消息存储架构图，如下所示。

消息数据以 Partition（分区）进行存储，一个 Topic（主题）可以由多个 Partition 进行存储，Partition 可以分布到多个 Broker 中；同时，Kafka 还提供了 Partition 副本机制（对

分区存储的信息进行备份，比如 Broker 1 中的 Topic-1 Partion-0 是对 Broker 0 上的 Topic-1 Partition-0 进行的备份），从而保证了消息存储的可靠性。



这就是数据分片和数据复制共存的一个典型应用场景。

货架：存储数据

货架是用来存储数据的，因为数据是由顾客产生和消费的，因此货架存储的数据类型与顾客产生和消费的数据类型是一致的，即包括结构化数据、半结构化数据和非结构化数据。

针对这三种不同的数据类型，存储“货架”可以大致划分为以下三种：

分布式数据库，通过表格来存储结构化数据，方便查找。常用的分布式数据库有 MySQL Sharding、Microsoft SQL Azure、Google Spanner、Alibaba OceanBase 等。

分布式键值系统，通过键值对来存储半结构化数据。常用的分布式键值系统有 Redis、Memcache 等，可用作缓存系统。具体的缓存技术我将在第 27 篇文章“分布式数据之缓存技术：‘身手钥钱’随身带”中与你详细介绍。

分布式存储系统，通过文件、块、对象等来存储非结构化数据。常见的分布式存储系统有 Ceph、GFS、HDFS、Swift 等。

而**对货架材料也就是存储介质的选择，本质就是选择将数据存储在磁盘还是内存（缓存）上**：

磁盘存储量大，但 IO 开销大，访问速度较低，常用于存储不经常使用的数据。比如，电商系统中，排名比较靠后或购买量比较少、甚至无人购买的商品信息，通常就存储在磁

盘上。

内存容量小，访问速度快，因此常用于存储需要经常访问的数据。比如，电商系统中，购买量比较多或排名比较靠前的商品信息，通常就存储在内存中。

知识扩展：业界主流的分布式数据存储系统有哪些？

在前面介绍货架的时候，我有提到针对结构化数据、半结构化数据和非结构化数据，分别对应不同的“货架”，即分布式数据库、分布式键值系统和分布式文件系统进行存储。

对于分布式键值系统，我会在第 27 篇文章中进行讲解，并与你介绍和分析主流存储系统。

所以在这里，我就重点与你对比分析分布式数据库和分布式文件系统的几款主流的系统，以便于你理解和选型。

首先，我们看一下主流的分布式数据库，主要包括 MySQL Sharding、SQL Azure、Spanner、OceanBase 等，具体对比分析如下表所示。

分布式数据库	MySQL Sharding	SQL Azure	Spanner	OceanBase
起源	瑞典MySQL AB公司开发，目前属于Oracle公司	微软在SQL Server技术基础上发展出来的云端关系型数据库服务	Google研发的可扩展的、全球分布式的数据库	阿里研发的高性能分布式数据库系统
是否开源	是	否	否	是
数据格式	结构化数据	结构化数据	结构化数据	结构化数据
应用场景	最流行的关系型数据库管理系统；在Web应用方面，是最好的关系数据库管理系统应用软件之一	云端数据库平台，适用于云端存储大规模结构化数据的场景	全球分布式的数据库，适用于全球性、大规模的结构化数据存储场景	支持海量数据存储和操作，比如 实现了上千亿条记录和上百TB数据上的跨行跨表事务，目前用于 用于存储淘宝具体的商品、店铺等信息

然后，我们看一下主流的分布式存储系统，主要包括 Ceph、GFS、HDFS 和 Swift 等，具体对比分析如下所示。

分布文件存储系统	Ceph	GFS	HDFS	Swift
起源	最初起源于塞奇 · 韦伊（Sage Weil）就读博士期间的工作	Google的分布式文件存储系统	Hadoop的核心子项目，为类似Hadoop这样的云计算而生	最初由Rackspace公司开发，2010 年贡献给 OpenStack 开源社区
是否开源	是	否	是	是
系统架构	去中心化	集中式架构	集中式架构	去中心化
数据格式	非结构化数据	非结构化数据	非结构化数据	非结构化数据
应用场景	通用的实时存储系统，适合频繁读写场景	基于Linux的专有大规模分布式文件系统，大文件读写场景	大数据场景，擅长处理离线批量大数据；	openstack对象存储场景

总结

今天，我主要与你分享的是分布式数据存储系统的三要素，即顾客、导购和货架，对应到分布式领域的术语就是数据生产者 / 消费者、数据索引和数据存储。

其中，顾客包括产生数据的顾客和消费数据的顾客两类；导购，就是数据索引引擎，包括数据存储时确定数据位置，以及获取数据时确定数据所在位置；货架，负责数据存储，包括磁盘、缓存等存储介质等。

不同应用场景中，顾客产生的数据类型、格式等通常都不一样。根据数据的特征，这些不同的数据可以被划分为三类：结构化数据、半结构化数据和非结构化数据。与之相对应的，货架也就是数据存储系统，也包括三类：分布式数据库、分布式键值系统和分布式文件系统。

针对分布式数据库和分布式文件系统的主流框架，我在“知识扩展模块”进行了对比分析，以方便你理解、记忆与应用。而对于分布式键值系统，我将在第 27 篇文章中进行详细介绍。

最后，我再通过一张思维导图来归纳一下今天的核心知识点吧。

相信通过今天的学习，你对分布式数据存储有了更深入的理解，对其中的核心角色和关键技术也有个更清晰的认识。加油，和我一起学习后面的章节，一起揭开分布式数据存储系统的神秘面纱吧！

思考题

传统单机关系型数据库与分布式数据库的区别是什么？

我是聂鹏程，感谢你的收听，欢迎你在评论区给我留言分享你的观点，也欢迎你把这篇文章分享给更多的朋友一起阅读。我们下期再会！

上一篇　23 | CAP理论：这顶帽子我不想要

下一篇　25 | 数据分布方式之哈希与一致性哈希：“掐指一算”与“掐指两算”的事

精选留言 (6)

写留言

Eternal
2019-11-23

老师的问题：传统单机关系型数据库与分布式数据库的区别是什么?

回答老师的问题，我任务首先需要需要梳理几个维度，然后通过这些维度来比较单机数据库和分布式数据库的区别，其实课程内容中老师已经
回描述了很多了，我这里总结一下自己的思考：...

展开

1

leslie
2019-11-20

传统的单机其实问题在于处理并发的能力弱，早期的网络速度和使用用户没那么多；单机足以支撑，现在网速、连接数、需求更加多样化了，单机就难以支撑了。生产环境不可能

是读写1：1，大多数场景还是会是70%的读30%的写或者更高，故而后期的做法基本都是用读写分离去平衡这种问题。

用擅长的东西去做擅长的事情吧，故而MYSQL会有MYISAM和INNODB两种存储引...

展开

1

Jackey

2019-11-20

刚好公司目前在拆分数据库，我从使用的角度聊一聊单机和分布式的区别吧。首先影响最大的可能是联查，分布式数据库的联查是一件很麻烦的事情，因为要关联的数据可能不在同一个库里。然后是取前n个值，由于数据分散在各个库中，如果不是根据sharding字段排序，要得到准确排序，可能需要把m（数据库数量）倍的数据都查出来，放到内存中排序，这个效率会非常低。

展开

1

1

阿西吧

2019-11-23

老师会说单主节点、多主节点、无主节点复制的相关内容吗

随心而至

2019-11-20

这篇文章真是太赞了，之前我总是觉得Mongodb，Redis，Elasticsearch，Kafka等等关于数据存储都好像好像。

下面是我之前的留言：

分布式数据存储，好像都是利用某种hash算法将数据存在不同的机器上。Kafka： hash（消息的键）来确定分区；Redis：hash（key）来确定slot；ES：hash（docId）来确...

展开

xingoo

2019-11-20

在存储位置方面，传统数据库一般很少做集群，点也很少。分布式数据库一般会由几十，几百，几千个节点组成。而数据的查询分布都是靠数据分片的规则来定。

在存储格式上，传统数据库一般采用b+树，适合增删改查重建索引；分布式数据库一般采用lsm树，顺序插入，合并更新机制。...

展开