

简单说项目做了什么

这是一个分布式的数据存储项目，其中用到了消息队列RabbitMQ、ES。可以将客户端上传的数据随机保存到任意一个储存节点，并且根据文件的大小选择不同的上传策略，包括分片存储、断点续传等。

消息队列扮演了什么角色

RabbitMQ在项目中起到了解耦的作用，通过增加一层使业务API和实际操作的数据存储API分离开，中间通过消息队列连接。消息队列相当于一个银行服务员，当客户端来的时候，业务员查询后台可以做存储的服务器节点，并且随机挑选一个将地址拿到，然后从客户端手中将要办理存储业务的数据也拿到，按照刚才拿到的地址找到存储服务器进行存储。斌

并且RabbitMQ可以通过设计一个心跳消息，不断管理存储节点，如果发现节点挂掉，及时将节点从列表中剔除，保证服务的可用性。

ES有什么用？

因为我们在上传的时候要避免重复上传，并且完成数据校验保证上传数据的完整性。所以需要一个东西去保存哈希值与其名字的对对应关系

显然不能存在内存里，所以需要一个分布式数据库去进行存储，然后在GET对象时候通过查询ES拿到其存储信息。

Redis也可以实现你说的功能，为什么用ES？

Redis也可以实现分布式存储，但它与ES的侧重点不同，首先Redis是基于内存的，虽然也有持久化机制但仍然存在潜在问题，比如短时间内存入大量数据是否会导致内存不足？而且Redis主要做的是key-value存储，并不是传统意义的数据库，ES可以通过一个key存储更多的消息，这更适合我们的场景。而且ES在搜索上比Redis速度更快

数据校验是怎么做的

我们在上传的时候并不是直接将对象存入节点中，而是将其首先设置成一个缓存，这是逻辑上的缓存，我们在上传前，客户端要提前通知其哈希值、size等信息，然后等存储节点将数据接受完毕以后，计算接收到的哈希值和size，与其将客户端告知的进行对比，如果一致，说明数据没问题，将其转正。如果有问题那么立即删除对象。

如果有一个存储节点挂掉了怎么办？

没有影响，因为我们虽然是随机选取一个节点进行存储，但实际上还做了数据冗余防止节点挂掉数据丢失。即在其他节点进行一些副本的存储作为备用，为了进一步节省空间，采用了RS纠删码进行实现。（RS纠删码自己查一下）

分布式数据库和分布式集群数据库什么区别?

分布式数据库是为了把数据分散到不同的节点上，节点的数据不同，是为了均摊分散

分布式集群是将同样的数据复制好多份到不同节点上，是为了容灾

本文采用分布式数据库，搭配RS纠删码也可以通过数据恢复实现容灾机制

CAP理论

它是指一个分布式系统最多只能同时满足一致性（Consistency）、可用性（Availability）和分区容错性（Partition tolerance）这三项中的两项。

参考资料：<https://cloud.tencent.com/developer/article/1860632>

分布式集群中选举机制

选举算法主要是根据RAFT、ZAB和Paxos进行设计，各个框架的具体实现不同

（这里可以聊这三个算法的流程，资料很多自己查一下）

Raft相关问题

leader挂掉怎么办？

leader挂掉又好了？

同时两个leader怎么办？

出现分区怎么办？

心跳机制的设计原则？

这些问题raft协议论文里写的都很明白。