

RocketMQ架构

2021年11月4日 星期四 上午10:29

从结构上看，RocketMQ分为四个部分；

前面讲的生产者和消费者是两组独立的机器

其中我们说的消息服务器主要分为NameServer和Broker

Broker

Broker是RocketMQ的核心模块，所有消息的存储、处理、分发都是由Broker来负责的

为了保证Broker的性能，我们可以在这个系统中配置多个Broker

所有的Broker都可以配属一个主从的结构，即一台主Broker、一台从Broker

如果主Broker挂掉了，消费这就去消费从Broker，这样使得发送上去的消息都被消费掉

在所有的Broker中，我们所有的主Broker都是相互独立的，它们之间不会做数据的交换，它们之间的协调靠的是NameServer。所以Broker启动之后它会把自己注册到一台或者多台NameServer上，然后每隔30s去向NameServer上去发自己的Topic信息，称之为Topic的路由信息

NameServer

NameServer是注册中心，因为所有的Broker在启动时都要把自己注册到NameServer上，且每隔30s要去发送自己的topic信息，这样所有的消息都存在于NameServer上，这样不管生产者消费者是要发送消息还是获得消息，都是去NameServer上去做查询

为了保持系统的稳定性，我们可以设多个NameServer，但NameServer之间是互相不通信的

如果架设了多台NameServer，在Broker上，每一台Broker都要设定自己的NameServer组，所有的NameServer都无法感知它们的存在，Broker和NameServer保持连接和心跳

对于生产者

它是通过NameServer来获取所有Broker的topic信息的，即看topic在哪一个Broker上面，然后根据负载均衡的原则来选择某一个NameServer的节点来保持和Broker的长连接，来定时的查询Topic的信息。如果一个NameServer挂掉了，就自动选择下一个NameServer，直到有连接为止

每一个生产者都会和跟他关联的所有的组的Broker去保持一个长连接，并保持心跳。因为它发送消息时，会发送到其中的任何一台Broker。

当每一次发送成功后，为了使所有Broker的负载得到均衡，它在下一次时一般会选择下一个Broker

对于消费者

它是从NameServer上去获取Broker信息，来看topic在哪一个Broker上。它也要和NameServer保持长连接，如果NameServer挂了，自动选择下一个。

通过NameServer可以查到topic在哪个Broker上，这叫做Topic路由信息。然后向Broker去发送Pull请求去获取消息的数据

单个的消费者会和所有的Broker保持一个长连接，并保持心跳。如果失去了心跳，他就会向同组的消费者发送通知，把原来分发给这个消费者的消息再去重发给同组的另外一个消费者

对于Topic

Topic是消息服务器概念中最重要的的一个概念模型。

它是消息的逻辑分类，一个Topic是可以分布在不同的Broker上的，并不限制在某一个服务器上。一台服务器上的Topic的子集称之为一个Topic的分片，每一个分片中消息是以队列来存放的，我们可以设定每一个Topic的分片有多少个队列（默认为4）

当一个Broker被创建时，它的队列就会被创建出来

队列是我们消息资源分配的一个最基本资源，如果一个Topic有五个队列，消费者组中有五个消费者，那么我们会给第一个消费者三个队列，给第二个消费者两个队列。那么发送到前三个队列中的消息会分配给第一个消费者，发送到后两个队列中的消息会分配给第二个消费者，这是集群消费的情况。如果是广播消费，则发给所有的消费者

因此，我们在设置时应该注意，消费者的数量应该小于队列的数量。如果消费者数量大于队列数量，就会有消费者没有分到任何队列，也就不会收到任何消息。

消息

消息是消息服务器中的载体，一个消息会被发送给一个Topic，Topic相当于消息的寄信地址，消息会被发送到Topic中的一个队列中

我们可以给消息指定Topic，也可以给Topic增加一个Tag，这个Tag的作用是过滤器，让队列可以有选择性的接受某些消息进行消费

部署

我们在 RocketMQ 时，要以一定的顺序去启动以上的这些部分

首先要启动NameServer

然后启动Broker，在启动时要注意Broker注册到哪一个NameServer

然后运行生产者和消费者。生产者在启动时会连到一个NameServer，获取NameServer上所有Broker上的地址，所以生产者不需要知道Broker是谁，只需要知道NameServer的地址与端口。拿到Broker之后再和Broker建立连接，然后往Broker上发送信息

生产者只会连接一个NameServer，但会连接所有的Broker，这样就可以轮流的向Broker上发送数据

消费者同样是在启动后连接一个NameServer，获取NameServer上的Broker的地址，然后去订阅Broker上的Topic地址

如果生产者发送的一条消息被放到了某一个队列中，订阅了Topic且分配到这个队列的消费者就会接收到这一条消息

