

第二篇 最佳实践指南之生产者

一些实用的使用提示

发送状态

发送一条消息后，将会返回一个SendResult对象，该对象包含了SendStatus。首先，假设消息对象中的isWaitStoreMsgOK=true（默认true）。如果isWaitStoreMsgOK=false，当没有捕获到异常，那么将返回SEND_OK。以下对每个状态进行了说明：

刷盘超时（FLUSHDISKTIMEOUT）

消息存储配置参数中可以设置刷盘方式和同步刷盘时间长度，如果Broker服务器设置了刷盘方式为同步刷盘即FlushDiskType=SYNC_FLUSH（默认为异步刷盘方式），当Broker服务器未在同步刷盘时间内（默认为5s）完成刷盘，则将返回该状态——刷盘超时。

数据同步到Slave服务器超时（FLUSHSLAVETIMEOUT）

如果Broker服务器的角色是同步Master即SYNCMASTER（默认是异步Master服务器即ASYNCMASTER），并且从属Broker服务器未在同步刷盘时间（默认为5秒）内完成与主服务器的同步，则将返回该状态——数据同步到Slave服务器超时。

无Slave服务器可用（SLAVENOTAVAILABLE）

如果Broker服务器的角色是同步Master即SYNCMASTER（默认是异步Master服务器即ASYNCMASTER），但没有配置slave Broker服务器，则将返回该状态——无Slave服务器可用。

发送成功（SEND_OK）

消息发送成功也不意味着它是可靠的。要确保不会丢失任何消息，还应启用同步Master服务器或同步刷盘即SYNCMASTER或SYNC_FLUSH。

重复或丢失（DuplicationorMissing）

如果返回值是刷盘超时（FLUSHDISKTIMEOUT），同步Slave服务器超时（FLUSHSLAVETIMEOUT）并且Broker服务器正好关闭，那么丢失的消息是可以找到的。此时，有两个选择，一个是放弃该消息，这可能会导致此消息丢失；另一种方法是重新发送消息，这可能会使消息重复。通常推荐重新发送消息，在客户端消费消息时过滤重复消息，除非某些消息不重要，能够接受这些消息的丢失。但请记住，当返回值为无Slave服务器可用（SLAVENOTAVAILABLE）时，重新发送消息是无用的。如果发生这种情况，用户应该保留场景并提醒群集管理器。

超时

客户端向Broker服务器发送请求，并等待响应，但如果最大等待时间已过，并且未返回任何响应，则客户端将抛出请求远程超时异常即RemotingTimeoutException，默认等待时间为3秒。用户还可以使用方法send (msg, timeout) 而不是send (msg) 传入超时参数。请注意，不建议等待时间太小，因为Broker服务器需要一些时间来刷新磁盘，并同步到从设备。如果该值远超于同步刷盘时间（syncFlushTimeout），则该值则没有太大意义，因为Broker服务器可能会返回同步到Slave服务器超时（FLUSHSLAVETIMEOUT），或者在请求超时之前返回同步到Slave服务器超时（FLUSHSLAVETIMEOUT）。

消息大小

建议消息的大小不要超过512K。

异步发送（AsyncSending）

默认方法send (msg) 会在得到返回响应之前一直被阻塞。因此，如果用户重视性能，那么建议使用send (msg, callback)，这样消息发送将以异步方式进行。

生产者组

通常而言，生产者组没有什么作用。但是如果关系到事务，那么用户就需要关注到生产者组。默认情况下，在同一个JVM中在一个生产者组只能创建一个相同的生产者，这通常就足够了。

线程安全

生产者是线程安全的，可以在业务中使用。

性能

如果希望在一个JVM中有多个生产者来进行大数据处理，有以下建议： * 通过一些生产者（3~5个足以）使用异步发送 * 为每个生产者的设置实例名即setInstanceName