
消费者最佳实践

一些有用的用户提示。

消费者组和订阅

第一件你需要注意的事情是不同的消费者组可以独立的消费一些 topic，并且每个消费者组都有自己的消费偏移量，请确保同一组内的每个消费者订阅相同的 topic。

消息监听器

有序的

消费者将锁定每个消息队列，以确保他们被逐个消费，虽然这将会导致性能下降，但是当你关心消息顺序的时候会很有用。我们不建议抛出异常，你可以返回 `ConsumeOrderlyStatus.SUSPEND_CURRENT_QUEUE_A_MOMENT` 作为替代。

并发的

顾名思义，消费者将并发消费这些消息，建议你使用它来获得良好性能，我们不建议抛出异常，你可以返回 `ConsumeConcurrentlyStatus.RECONSUME_LATER` 作为替代。

消费状态

对于并发的消费监听器，你可以返回 `RECONSUME_LATER` 来通知消费者现在不能消费这条消息，并且希望可以稍后重新消费它。然后，你可以继续消费其他消息。对于有序的消息监听器，因为你关心它的顺序，所以不能跳过消息，但是你可以返回 `SUSPEND_CURRENT_QUEUE_A_MOMENT` 告诉消费者等待片刻。

阻塞的

不建议阻塞监听器，因为它会阻塞线程池，并最终可能会终止消费进程。

线程数

消费者使用 `ThreadPoolExecutor` 在内部对消息进行消费，所以你可以通过设置 `setConsumeThreadMin` 或 `setConsumeThreadMax` 来改变它。

从何处开始消费信息

当建立一个新的消费者组时，需要决定是否需要消费已经存在于 Broker 中的历史消息。`CONSUME_FROM_LAST_OFFSET` 将会忽略历史消息，并消费之后生成的任何消息。

CONSUME_FROM_FIRST_OFFSET 将会消费每个存在于 Broker 中的信息。你也可以使用 CONSUME_FROM_TIMESTAMP 来消费在指定时间戳后产生的消息。

重复

许多情况会导致重复，例如：

- 生产者重复发送信息（例如，在 FLUSH_SLAVE_TIMEOUT 的情况下）
- 消费者关闭，一些偏移未及时更新到 Broker

因此，如果你的应用程序无法容忍重复，你可能需要做一些外部工作来处理此问题。例如，你可以检查你数据库的主键。

NameServer的最佳实践

在 Apache RocketMQ 中，名称服务器（name servers）被设计用来协调分布式系统中的每个组件，协调主要通过管理 topic 路由信息来实现。

管理由两部分组成：

- Brokers 定期更新每个名称服务器中保存的元数据。
- 名称服务器为客户端，包括生产者，消费者和命令行客户端提供最新的路由信息。

因此，在启动代理和客户端之前，我们需要通过提供给他们名称服务器地址列表告诉他们如何到达名称服务器。在 Apache RocketMQ 中，这可以通过四种方式完成。

程序化的方式

对于brokers，我们可以在代理配置文件中指

定 `namesrvAddr = name-server-ip1:port;name-server-ip2:port`。

对于生产者和消费者，我们可以按如下方式为他们提供名称服务器地址列表：

```
DefaultMQProducer producer = new DefaultMQProducer("please_rename_unique_group_name");
producer.setNamesrvAddr("name-server1-ip:port;name-server2-ip:port");
DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("please_rename_unique_group_name");
consumer.setNamesrvAddr("name-server1-ip:port;name-server2-ip:port");
```

如果你在 shell 中使用 admin 命令行,也可以这样指定：

```
shell sh mqadmin command-name -n name-server-ip1:port;name-server-ip2:port -X OTHER-OPTION
```

一个简单的例子是：`sh mqadmin -n localhost:9876 clusterList` 假设在名称服务器节点上查询集群信息。

如果你已经将管理工具集成到自己的控制面板中，你可以：

```
DefaultMQAdminExt defaultMQAdminExt = new DefaultMQAdminExt("please_rename_unique_group_name");
defaultMQAdminExt.setNamesrvAddr("name-server1-ip:port;name-server2-ip:port");
```

Java选项

名称服务器地址列表也可以在启动之前通过指定后续 java 选项 `rocketmq.namesrv.addr` 来提供给你的应用程序。

环境变量

你可以导出 `NAMESRV_ADDR` 环境变量。如果设置后，Brokers 和 clients 将检查并使用这个变量值。

HTTP端点

如果你未使用前面提到的方法指定名称服务器地址列表，Apache RocketMQ 将访问以下 HTTP 端点，每隔两分钟获取并更新名称服务器地址列表，初始延迟为10秒。

默认情况下，端点是：

```
http://jmenv.tbsite.net:8080/rocketmq/nsaddr
```

你可以使用 Java 选项：`rocketmq.namesrv.domain` 来覆盖 `jmenv.tbsite.net`，你也可以使用 Java 选项 `rocketmq.namesrv.domain.subgroup` 来覆盖 `nsaddr`

如果你正在生产环境中运行 Apache RocketMQ，建议使用此方法，因为它为你提供了最大的灵活性 - 你可以动态添加或删除名称服务器节点，而无需根据名称服务器的系统负载重新启动 brokers 和客户端。

优先级

首先介绍的方法优先于后者：

```
Programmatic Way > Java Options > Environment Variable > HTTP Endpoint
```