

重要的消费者参数

在 `KafkaConsumer` 中，除了第8节提及的4个默认的客户端参数，大部分的参数都有合理的默认值，一般我们也不需要去修改它们。不过了解这些参数可以让我们更好地使用消费者客户端，其中还有一些重要的参数涉及程序的可用性和性能，如果能够熟练掌握它们，也可以让我们在编写相关的程序时能够更好地进行性能调优与故障排查。下面挑选一些重要的参数来做细致的讲解。

1. `fetch.min.bytes`

该参数用来配置 `Consumer` 在一次拉取请求（调用 `poll()` 方法）中能从 `Kafka` 中拉取的最小数据量，默认值为1（B）。`Kafka` 在收到 `Consumer` 的拉取请求时，如果返回给 `Consumer` 的数据量小于这个参数所配置的值，那么它就需要进行等待，直到数据量满足这个参数的配置大小。可以适当调大这个参数的值以提高一定的吞吐量，不过也会造成额外的延迟（`latency`），对于延迟敏感的应用可能就不可取了。

2. `fetch.max.bytes`

该参数与 `fetch.min.bytes` 参数对应，它用来配置 `Consumer` 在一次拉取请求中从 `Kafka` 中拉取的最大数据量，默认值为52428800（B），也就是50MB。

如果这个参数设置的值比任何一条写入 `Kafka` 中的消息要小，那么会不会造成无法消费呢？很多资料对此参数的解读认为是无法消费的，比如一条消息的大小为10B，而这个参数的值是1（B），既然此参数设定的值是一次拉取请求中所能拉取的最大数据量，那么显然 $1B < 10B$ ，所以无法拉取。这个观点是错误的，该参数设定的不是绝

对的最大值，如果在第一个非空分区中拉取的第一条消息大于该值，那么该消息将仍然返回，以确保消费者继续工作。也就是说，上面问题的答案是可以正常消费。

与此相关的，Kafka 中所能接收的最大消息的大小通过服务端参数 `message.max.bytes`（对应于主题端参数 `max.message.bytes`）来设置。

3. `fetch.max.wait.ms`

这个参数也和 `fetch.min.bytes` 参数有关，如果 Kafka 仅仅参考 `fetch.min.bytes` 参数的要求，那么有可能会一直阻塞等待而无法发送响应给 Consumer，显然这是不合理的。`fetch.max.wait.ms` 参数用于指定 Kafka 的等待时间，默认值为500（ms）。如果 Kafka 中没有足够多的消息而满足不了 `fetch.min.bytes` 参数的要求，那么最终会等待500ms。这个参数的设定和 Consumer 与 Kafka 之间的延迟也有关系，如果业务应用对延迟敏感，那么可以适当调小这个参数。

4. `max.partition.fetch.bytes`

这个参数用来配置从每个分区里返回给 Consumer 的最大数据量，默认值为1048576（B），即1MB。这个参数与 `fetch.max.bytes` 参数相似，只不过前者用来限制一次拉取中每个分区的消息大小，而后者用来限制一次拉取中整体消息的大小。同样，如果这个参数设定的值比消息的大小要小，那么也不会造成无法消费，Kafka 为了保持消费逻辑的正常运转不会对此做强硬的限制。

5. `max.poll.records`

这个参数用来配置 Consumer 在一次拉取请求中拉取的最大消息数，默认值为500（条）。如果消息的大小都比较小，则可以适当调大这个参数值来提升一定的消费速度。

6. `connections.max.idle.ms`

这个参数用来指定在多久之后关闭闲置的连接，默认值是540000（ms），即9分钟。

7. `exclude.internal.topics`

Kafka 中有两个内部的主题： `__consumer_offsets` 和 `__transaction_state`。 `exclude.internal.topics` 用来指定 Kafka 中的内部主题是否可以向消费者公开，默认值为 `true`。如果设置为 `true`，那么只能使用 `subscribe(Collection)`的方式而不能使用 `subscribe(Pattern)`的方式来订阅内部主题，设置为 `false` 则没有这个限制。

8. `receive.buffer.bytes`

这个参数用来设置 Socket 接收消息缓冲区（`SO_RECVBUF`）的大小，默认值为65536（B），即64KB。如果设置为-1，则使用操作系统的默认值。如果 Consumer 与 Kafka 处于不同的机房，则可以适当调大这个参数值。

9. `send.buffer.bytes`

这个参数用来设置Socket发送消息缓冲区（`SO_SNDBUF`）的大小，默认值为131072（B），即128KB。与`receive.buffer.bytes`参数一样，如果设置为-1，则使用操作系统的默认值。

10. request.timeout.ms

这个参数用来配置 Consumer 等待请求响应的最长时间，默认值为 30000 (ms) 。

11. metadata.max.age.ms

这个参数用来配置元数据的过期时间，默认值为 300000 (ms) ，即 5 分钟。如果元数据在此参数所限定的时间范围内没有进行更新，则会被强制更新，即使没有任何分区变化或有新的 broker 加入。

12. reconnect.backoff.ms

这个参数用来配置尝试重新连接指定主机之前的等待时间（也称为退避时间），避免频繁地连接主机，默认值为 50 (ms) 。这种机制适用于消费者向 broker 发送的所有请求。

13. retry.backoff.ms

这个参数用来配置尝试重新发送失败的请求到指定的主题分区之前的等待（退避）时间，避免在某些故障情况下频繁地重复发送，默认值为 100 (ms) 。

14. isolation.level

这个参数用来配置消费者的事务隔离级别。字符串类型，有效值为“read_uncommitted”和“read_committed”，表示消费者所消费到的位置，如果设置为“read_committed”，那么消费者就会忽略事务未提交的消息，即只能消费到 LSO (LastStableOffset) 的位置，默认情况下为“read_uncommitted”，即可以消费到 HW (High Watermark) 处的位置。有关事务和 LSO 的内容可以参考 [《图解](#)

[Kafka之核心原理》](https://juejin.im/book/5c7d270ff265da2d89634e9e)

[\(https://juejin.im/book/5c7d270ff265da2d89634e9e\)](https://juejin.im/book/5c7d270ff265da2d89634e9e)的相关章节。

还有一些消费者参数在本节没有提及，这些参数同样非常重要，它们需要用单独的章节或场景中描述。部分参数在前面的章节内容中已经提及，比如 bootstrap.servers；还有部分参数会在后面的[《图解](https://juejin.im/book/5c7d270ff265da2d89634e9e)

[Kafka之核心原理》](https://juejin.im/book/5c7d270ff265da2d89634e9e)

[\(https://juejin.im/book/5c7d270ff265da2d89634e9e\)](https://juejin.im/book/5c7d270ff265da2d89634e9e)中提及，比如 heartbeat.interval.ms。下表罗列了部分消费者客户端的重要参数。

参 数 名 称	默 认 值
bootstrap.servers	“”
key.deserializer	
value.deserializer	
group.id	“”
client.id	“”
heartbeat.interval.ms	3000
session.timeout.ms	10000
max.poll.interval.ms	300000

auto.offset.reset latest

enable.auto.commit true

auto.commit.interval.ms 5000

partition.assignment.strategy org.apache.kafka.clients.consumer
interceptor.class “”

从第7节至本节主要讲述了消费者和消费组的概念，以及如何正确使用 KafkaConsumer。具体的内容还包括参数配置的讲解、订阅、反序列化、位移提交、再均衡、消费者拦截器、多线程的使用。不过本章并没有同前一章的生产者一样讲述具体的原理，因为考虑到 KafkaConsumer 内部实现相对复杂，具体的原理讲述篇幅较长，故将相关的内容编排到 [《图解Kafka之核心原理》](https://juejin.im/book/5c7d270ff265da2d89634e9e) (<https://juejin.im/book/5c7d270ff265da2d89634e9e>) 中，如果读者迫切地想要了解这部分的知识，可以直接跳转到 [《图解Kafka之核心原理》](https://juejin.im/book/5c7d270ff265da2d89634e9e) (<https://juejin.im/book/5c7d270ff265da2d89634e9e>) 相关章节进行阅读。