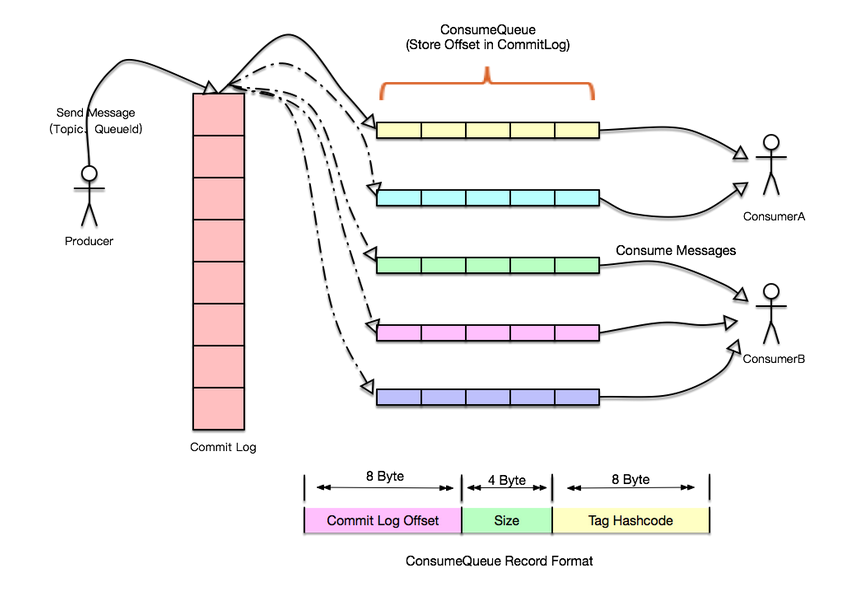
***如何在RocketMQ中支持更多的消息队列***



1. 所有的消息消息数据都存储在提交日志文件中。所有的写入都是完全顺序的，而读取是随机的。
2. ConsumeQueue存储实际的用户消费位置信息，这些信息也以顺序方式刷新到磁盘。

**优点：**

1. 每个队列都是轻量级的，并且包含有限数量的元数据。
2. 访问磁盘是完全顺序的，这样可以避免磁盘的锁竞争。

**缺点**：

1. 消费者首先读取Consume Queue,然后是提交日志。在最坏情况下，这个过程会带来一定的成本。
2. Consume Queue和提交日志需要在逻辑上保持一致，这为编程模型带来了额外的复杂性。

**设计动机：**

1. 随机读。尽可能多地读取以提高页面缓存命中率，并减少读取IO操作。如此大的内存仍然是可取的，如果累积了大量消息，读取性能是否会下降呢？答案是否定的，原因如下：

(1). 即使消息的大小仅为1KB，系统也会提前读取更多的数据，请参阅PAGE CACHE预读取参考。这意味着，对于连续数据读取，它将访问将要运行的主内存，而不是慢速的磁盘IO读取。

(2). 从磁盘随机访问提交日志。如果是在SSD的情况下将I/O调度程序设置为NOOP，则QPS将显著提升。

2. 由于Consume Queue仅存储固定大小的元数据，它主要用于记录消费进度，因此可以很好地支持随机读取。利用页面缓存预读取，访问Consume Queue与访问主内存一样快，即使是在大量消息累积的情况下也是如此。因此，ConsumeQueue不会对读取性能带来明显的损失。

3. 提交日志几乎存储所有信息，包括消息消息数据。与关系数据库中的REDO日志类似，只要提交日志存在，就可以完全恢复ConsumeQueue,消息密钥索引，和所有其他所需数据。