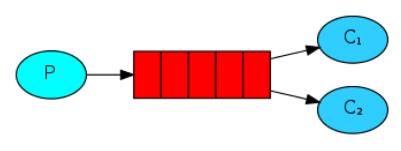
# RabbitMq

RabbitMq是一个消息中间级，主要有四种工作模式：

* **Work queues**



生产者向队列中添加消息，消费者会从队列中获取消息，如果生产者提供了100条消息，那么在默认的情况下这100条消息会平均分发给C1，C2。

ACK：如果C1在处理消息的过程中突然宕机了，那么分配给C1的消息都会丢失，为了避免这样的问题，RabbitMq使用了ACK机制，队列会暂存分配出去的消息，如果C1接受并且处理完成了该消息，那么C1会给RabbitMq发送一个ACK来进行提醒RabbitMq可以删除该消息了。如果C1挂了（channel关闭了，连接丢失了）没有发送ack，那么RabbitMq会重新讲那些分发给C1，并且没有收到ack的消息重新分发给其他的消费者（如果有的话）。这样就可以保证不会有消息丢失。

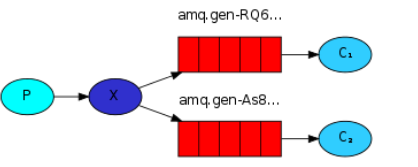
能者多劳：默认情况下，消息队列是平均分发给消费者，但是有的消息处理的比较快，有的消息处理的表较慢，那么就会存在一个很忙，一直没有空闲，而另一个很闲，发过来的都处理完成了。为了避免闲等的情况，可以添加如下的语句：



这句话的意思是一次只发送一条消息给一个消费，换句话说就是如果没有收到消费者的ack，那么就不发送新的消息给该消费者。取而代之的是给下一个闲着的消费者发送消息。

该模式可以应用在下单场景，只要有一个服务器接收到下单这个操作就行。

* **Publish/Subscribe**



之前的work模式一个消息只有一个消费者收到，如果想要多个消费者收到消息，那么就需要使用Publish/Subscrib模式。从上述图中可以看出，生产者不直接将消息发送给队列，而是发送给一个exchange，中文名是交换机（区别于计网中的交换机），然后再通过该交换机转发给各个和交换机绑定的队列。

Exchange（交换机）:交换机一头接受从生产者发送过来的消息，一头将他们推送给队列。交换机必须清楚的知道怎样处理它所受到的消息，将其添加到一个队列还是多个队列还是直接丢弃，不同的操作就对应着不同交换机类型——direct，topic，headers，fanout。这边我们只关注fanout类型。fanout类型仅仅是将接受到的消息发送给所有与之相关联的队列。