rabbitmq堆积消息后生产速率降低的问题分析及应对措施

### 问题描述：

在rabbitmq没有消费者的情况下，生产者持续向mq发消息，使得消息在mq中大量堆积，发送速率不受影响，但当有新的消费者连接上mq并开始接收消息时，生产速率大幅降低。

### 问题分析：

Rabbitmq的中处理队列收发逻辑的是一个有穷状态机进程，它对消息的处理流程可以概括为下图所示的流程：



1. 当MQ既有生产者也有消费者时，该状态机的处理流程为 ：接收消息->持久化->发送消息->接收消息 –> … ->。在流控机制的控制下，收发速率能够保持基本一致，队列中堆积的消息数会非常低。
2. 当没有消息者时，处理流程如图中橙色线条所示，MQ会持续接收消息并持久化直到磁盘被写满，因为没有发送逻辑，这时可以达到更高的生产速率。
3. 当MQ中有消息堆积时，处理流程如图中绿色线条所示，MQ会持续从队列中取出堆积的消息将其发送出去，直到没有了堆积消息，或者消费者的qos被用光，或者没有消费者，或者消费者的channel被阻塞。如果一直没有满足上述4个条件之一，MQ就会持续的发送堆积消息，不去处理新来的消息，在流控机制的作用下，发送端就被阻塞了。

总结：从上述描述可以看出，消息堆积后，发送速率降低是MQ的处理流程使然，不是bug。这样的流程设计基于以下两个原因：

1. 让堆积的消息更快的被消费掉，降低消息的时延。
2. MQ中堆积的消息越少，每个消息处理的平均开销就越少，可以提高整体性能，所以需要尽快将堆积消息发送出去。

### 应对措施：

1. 打破发送循环条件。
2. 设置合适的qos值，当qos值被用光，而新的ack未被mq接收时，就可以跳出发送循环，去接收新的消息。
3. 消息者到主动block接收进程，消费者感知到接收消息的速度过快时，主动block，利用block与unblock方法调节接收速率。当接收进程被block时，mq跳出发送循环。
4. 建立新的队列  
   若服务器cpu资源有较多剩余，而又不需要保证消息的顺序的情况下可以通过建立新的vhost，在该vhost下创建queue，生产者将消息发送掉新的queue，消费者同时订阅新旧queue。
5. **使用缓存**

在生产者端使用缓存，当生产速率受到流控限制时，缓存数据。在堆积的消息被处理完后，生产速率恢复正常时，此时将缓存的数据发送给MQ。

1. 更新rabbitmq版本

在新版2.8.4中，在有大量消息堆积时，生产速率会受到抑制，但生产者不会完全被阻塞。

1. 加机器。