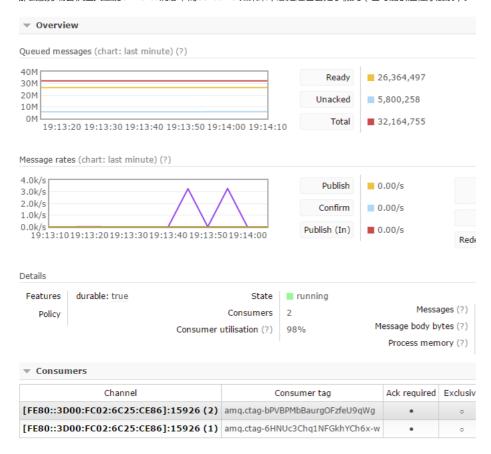
消费RabbitMQ时的注意事项,如何禁止大量的消息涌到Consumer,保证线程安全

转载 2018年03月01日 10:57:54

按照官网提供的订阅型写法(Retrieving Messages By Subscription ("push API")) 我发现,RabbitMQ服务器会在短时间内发送大量的消息给Consumer,然后,如果你没那么服务端会积压大量的UnAcked消息,而Consumer如果来不急处理也会处于假死(也可能引起程序崩溃)。



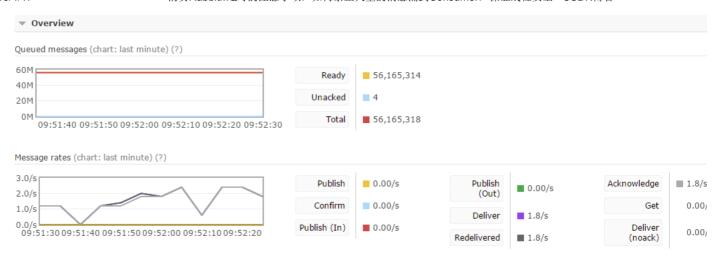
仅有两个Channel,结果积压了大量的UnAcked消息。

这明显是与我们的目的不一致,我们不能保证Consumer一定会急时快速的处理消息。所以这种方式带来的后果就是Consmer崩溃后,UnAcked消息又ReQueue,这肯定会源。

我试图在官网上找到一种方法,让每条消息明确的Ack后再接受下一条。但是好没有。好在在 gitbooks.io/rabbitmq-quick/ 这儿找到了,通过设置Channel的QOS即可

```
var channel = Connect.CreateModel();
channel.BasicQos(0,1,false); //RabbitMQ客户端接受消息最大数量
```

设置后的结果:



在开启4个Consumer的情况下,每条消息处理要耗时2秒。然后问题解决了。Unacked的消息只有4个。