

RabbitMQ 介绍

宦传建

目前公司哪些场景使用了MQ？

使用场景

- 解耦

用户支付后送积分，推送消息等

- 错峰流控

缓解 web 流量和 DB 写能力不匹配

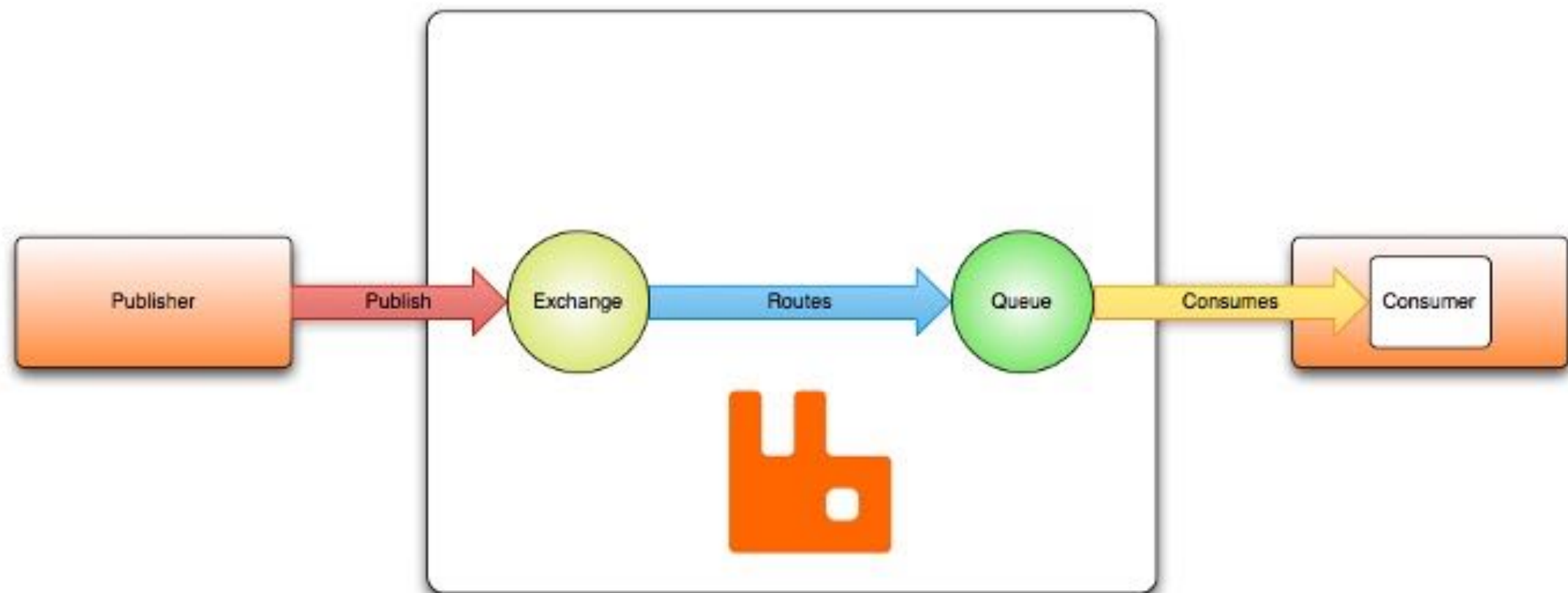
- 最终一致性

数据写入缓存，通过 MQ 持久化到 DB

- 广播

一条消息从产生到被消费在 RabbitMQ 上需要经过哪些实体？

"Hello, world" example routing



Connection & Channel

- 应用程序和 RMQ 之间建立 long-lived connection
- 应用程序对 RMQ 提交的操作都是基于 channel 的，当使用多线程处理时，每个线程应当占用一个 channel，这样是线程安全的，需要避免线程间共享，这会造成 channel 上的 frame 乱序等问题，目前不少客户端库还提供了 channel pool 机制
- Connection 会占用 OS 文件句柄资源，channel 底层数据结构是 Erlang 轻量级进程，会占用内存资源。

生产者消息可靠投递

事务

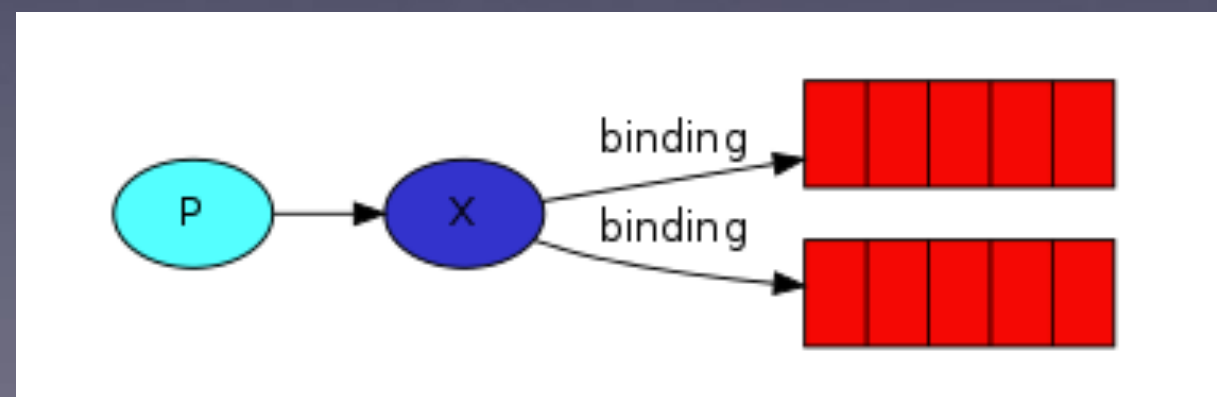
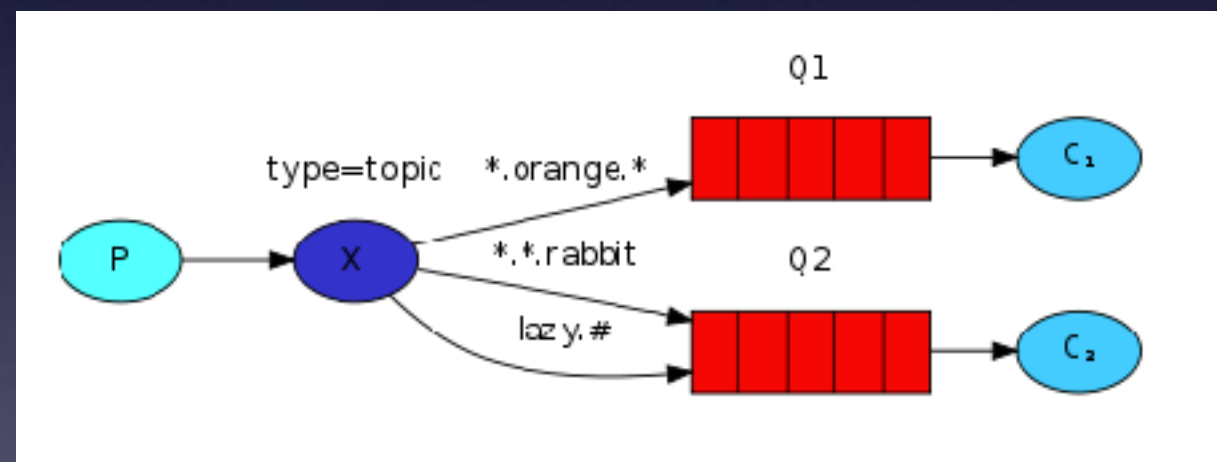
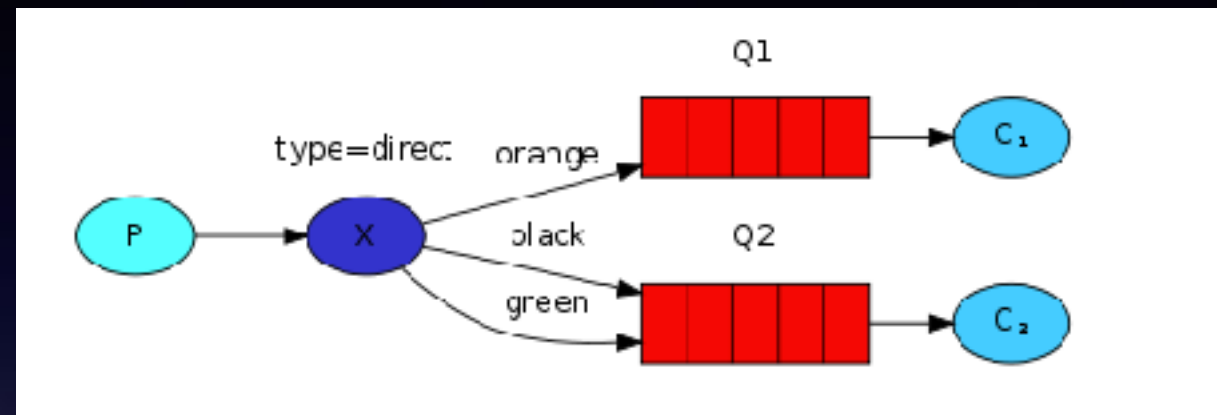
Using standard AMQP 0-9-1, the only way to guarantee that a message isn't lost is by using transactions -- make the **channel transactional**, publish the message, commit. In this case, transactions are unnecessarily **heavyweight** and decrease throughput by a factor of **250**.

Publisher Confirms 机制

1. Client -> confirm.select (channel 处于 confirm mode)
2. Broker -> confirm.select-ok
3. Broker -> basic.ack (返回包含 delivery-tag 来标识确认的消息)

Exchange

- Direct exchange 需要精准匹配 routing_key
- Topic exchange 允许模糊匹配 routing_key, 注意 * 和 #, 可以用来实现组播
- Fanout exchange 没有 routing_key 的概念, 广播到所有 binding 的 queue 里



Mirror Queue

- RMQ 通过镜像队列机制实现高可用
- RMQ 通过设置 policy 来控制复制规则

Features	durable: true
Policy	ha-all
Node	VM-229-137-ubuntu
Slaves	VM-229-139-ubuntu VM-229-138-ubuntu

Policies					
▼ All policies					
Filter:		<input type="checkbox"/> Regex (?) (?)			
Virtual Host	Name	Pattern	Apply to	Definition	Priority
vhost-fmasc	ha-all	^ha\.	all	ha-mode: all	0

Consumer utilisation

Details

Features	durable: true	State	<div><div></div>running</div>	Total	Ready	Unacked	In memory	Persistent
Policy	ha-all	Consumers	4	Messages (?)	0	0	0	0
Node	VM-229-138-ubuntu	Consumer utilisation (?)	100%	Message body bytes (?)	0B	0B	0B	0B
Slaves	VM-229-137-ubuntu VM-229-139-ubuntu			Process memory (?)	141kB			

Consumers


Channel	Consumer tag	Ack required	Exclusive	Prefetch count	Arguments
10.105.229.138:43473 (1)	amq.ctag-5z7L5UJmROPQaB34HqxbXw	•	○	5	
10.105.229.138:43479 (1)	amq.ctag-fhiqNHBS7muEvG5SQjuImq	•	○	5	
10.105.229.138:43485 (1)	amq.ctag-_XvSqN-O81cfYpWAqPpVmA	•	○	5	
10.105.229.138:43525 (1)	amq.ctag-XoqpCq9r3xOoD00CNp3tKg	•	○	5	

- 消费者使用率：一段时间内处理消息的能力，反映了消费端的性能，越接近100% 越佳
- Prefetch count: 在网络延时较大时，适当调大该值效果会比较明显

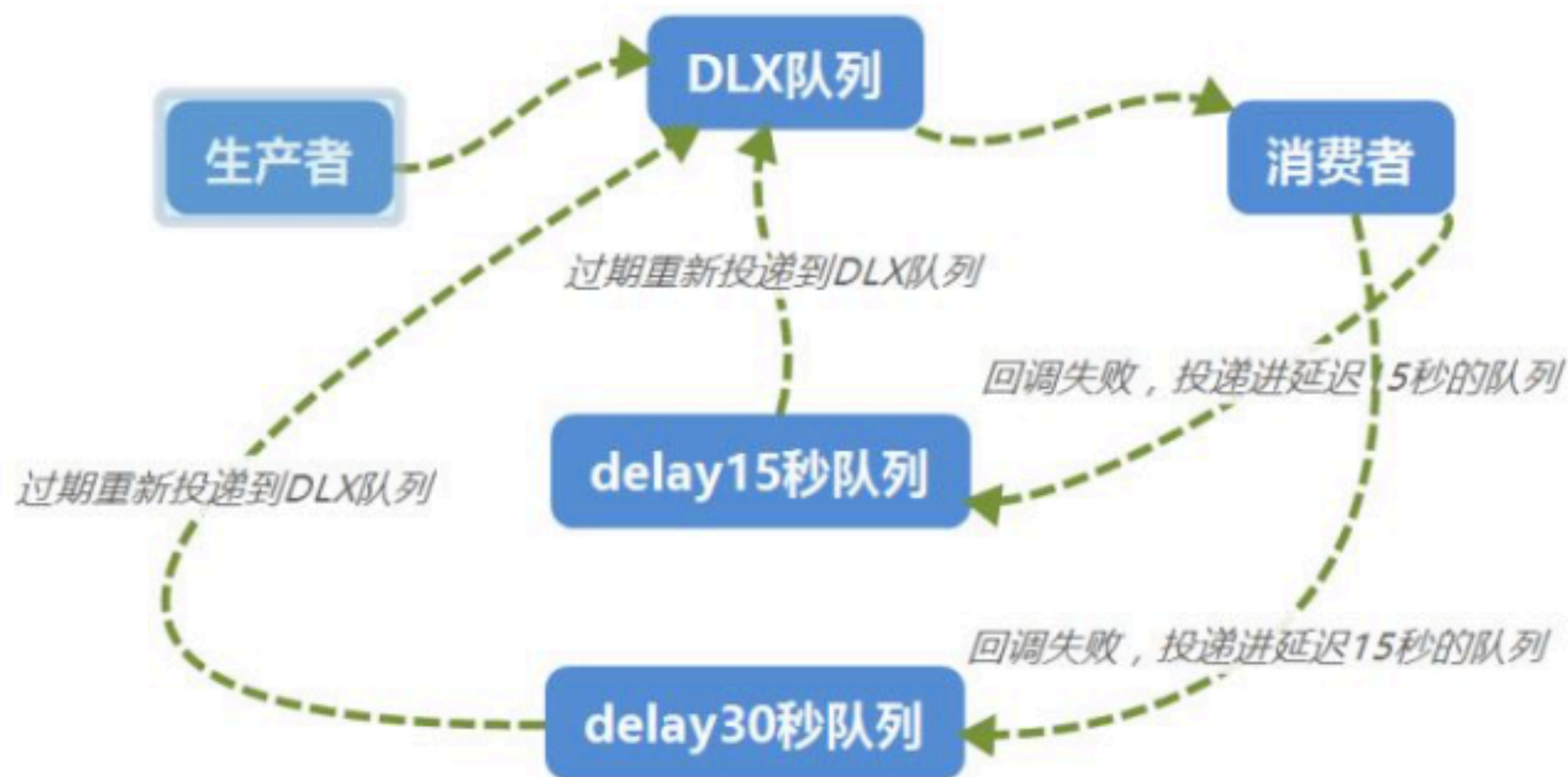
Dead Letter (死信机制)

- 声明死信队列属性 DLX (x-dead-letter-exchange), DLK (x-dead-letter-routing-key)
- TTL 可以是队列级别或消息级别
- 死信队列不需要消费者
- **注意**: 进入死信队列的消息依旧遵循 FIFO 进行重新分发, 即使已经到了过期时间。

VM-229-137-ubuntu	+2	D	TTL	DLX	DLK	ha-all	running
VM-229-137-ubuntu	+2			D		ha-all	idle
VM-229-137-ubuntu	+2			D		ha-all	idle
VM-229-137-ubuntu	+2			D		ha-all	idle
VM-229-137-ubuntu	+2	D	DLX	DLK		ha-all	idle

Details			
Features	x-dead-letter-exchange: <i>PaymentPush</i>	State	 running
	x-dead-letter-routing-key: <i>Push</i>	Consumers	0
	x-message-ttl: <i>120000</i>	Consumer utilisation (?)	0%
	durable: <i>true</i>		
Policy	ha-all		
Node	VM-229-137-ubuntu		
Slaves	VM-229-139-ubuntu		
	VM-229-138-ubuntu		
▼ Consumers			
... no consumers ...			

延迟队列



Any Questions?

谢谢大家！