消息队列 RabbitMQ 版

产品简介

腾讯云

文档目录

产品简介

产品概述

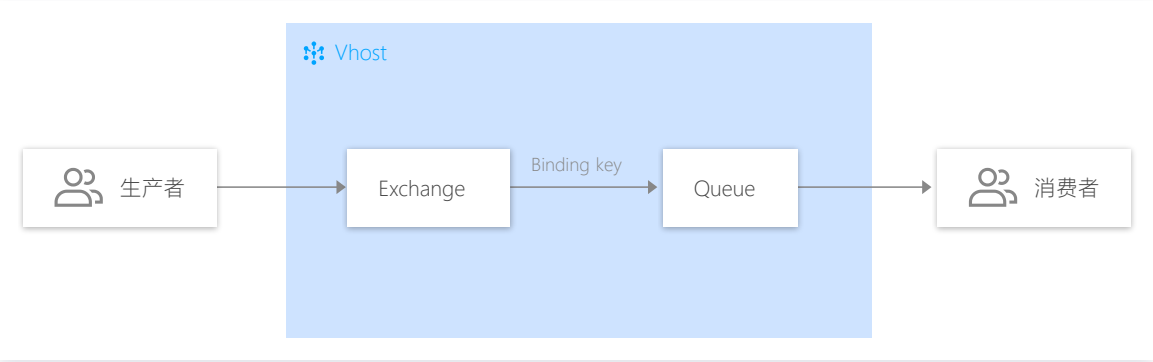
最近更新时间：2022-02-11 11:31:13

消息队列 TDMQ RabbitMQ 版（TDMQ for RabbitMQ，简称 TDMQ RabbitMQ 版）是一款腾讯自主研发的消息队列服务，支持 AMQP 0-9-1 协议，完全兼容开源 RabbitMQ 的各个组件与概念，同时具备计算存储分离，灵活扩缩容的底层优势。
TDMQ RabbitMQ 版拥有极为灵活的路由来适应各类业务的消息投递规则，能有缓冲上游流量压力的能力，保证消息系统的稳定运行。常用于系统间的异步通信和服务解耦，减轻不同服务之间的依赖，广泛应用于金融，政务等行业的分布式系统中。

基础架构

TDMQ RabbitMQ 版的基本概念如下：

- 生产者：向 Exchange 发送消息。
- Vhost：用作逻辑隔离，不同 Vhost 之间的 Exchange 和 Queue 相互隔离，互不干扰。
- Exchange：接收来自生产者的消息并将消息路由到 Queue 的组件。
- Queue：存储消息的缓冲区，供消费者消费消息。
- 消费者：从 Queue 拉取消息进行消费。



更多关于 TDMQ RabbitMQ 版的概念，请参见 相关概念 。

产品优势

最近更新时间：2022-03-02 11:37:02

兼容开源

支持 AMQP 0-9-1 版本标准协议，完全支持开源 RabbitMQ 社区和 Queue、Exchange、Vhost 组件。一键迁移开源 RabbitMQ 元数据，实现迁移上云零成本。

功能完备

TDMQ RabbitMQ 版支持原生 RabbitMQ 的各类消息模型。支持死信交换机与备用交换机，用户无需担心由于消息过期、路由失败等因素造成的消息丢失。

稳定可靠

持久化机制确保了 TDMQ RabbitMQ 版的高可靠性。设置 Exchange、Queue、消息的持久化，保证服务重启后元数据与消息内容不丢失。消息采用三副本存储策略，某台物理机故障时，能够实现数据的快速迁移，保证用户数据3个备份可用，服务可用性达99.95%。

高扩展性

TDMQ RabbitMQ 版相比于开源 RabbitMQ 支持更高的队列数量，可扩展能力强，底层系统可根据业务规模自动弹性伸缩、扩容/缩容集群规模，对用户透明。

易用免运维

提供 API 访问接口，支持开源所有语言和版本的 SDK。提供腾讯云平台整套运维服务，实时监控告警，帮助用户快速发现并解决问题，保证服务的可用性。

应用场景

最近更新时间：2024-04-12 11:13:51

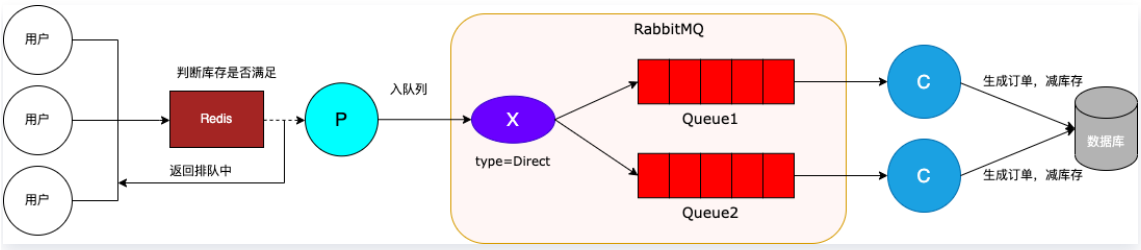
秒杀场景

秒杀是电商系统中一个非常常见的场景。解决方案比较多，但是使用消息队列 RabbitMQ 版是一个比较好的做法。

如果是复杂的扣减库存（如涉及商品信息本身或牵连其他系统），则建议使用数据库进行库存数量的扣减，可以使用异步的方式来应对这种高并发的库存更新。

1. 在用户下单时，不立刻生成订单，而是将所有订单依次放入队列。
2. 下单模块依据自身的处理速度，从队列中依次获取订单进行“下单扣库存”操作。
3. 在订单生成成功后，用户即可进行支付操作了。

这种方式是针对“秒杀”场景的，依据“先到先得”原则来保证公平公正，所有用户都可以抢购，然后等待订单处理，最后生成订单（如果库存不足，则生成订单失败）。



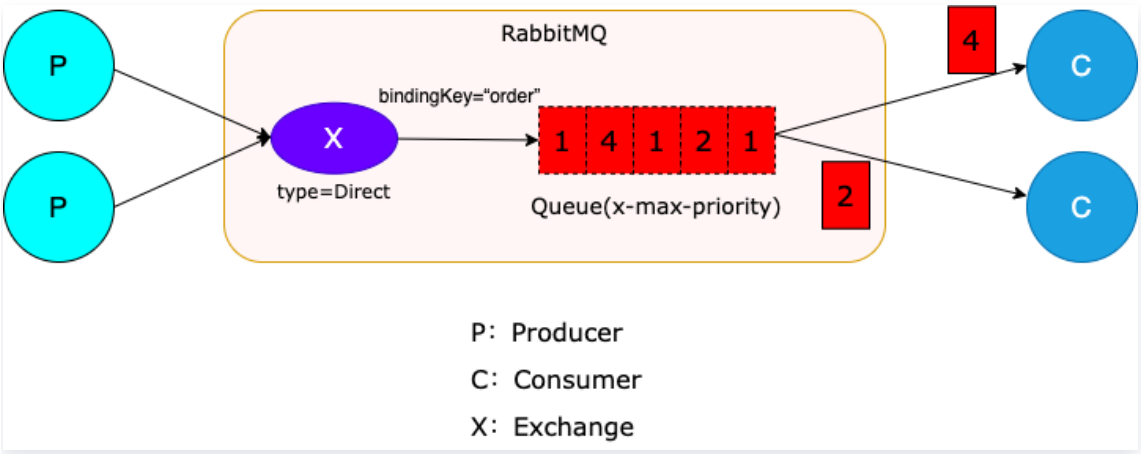
优先级消息

在消费消息时，如果消息的重要程度不同，重要性高的消息希望被优先消费，这时可以使用消息队列 RabbitMQ 版优先级队列的能力，让优先级高的消息优先被消费。

参考如下业务场景：

例如系统中有订单催付的场景，客户在电商系统下的订单，系统会及时将订单推送给客户。如果在设定的时间内未付款那么就会给客户推送一条短信提醒。但是，电商系统会分大客户和小客户，大客户的订单催收消息需要优先处理，其他的小客户的催收相对优先级会低一些。

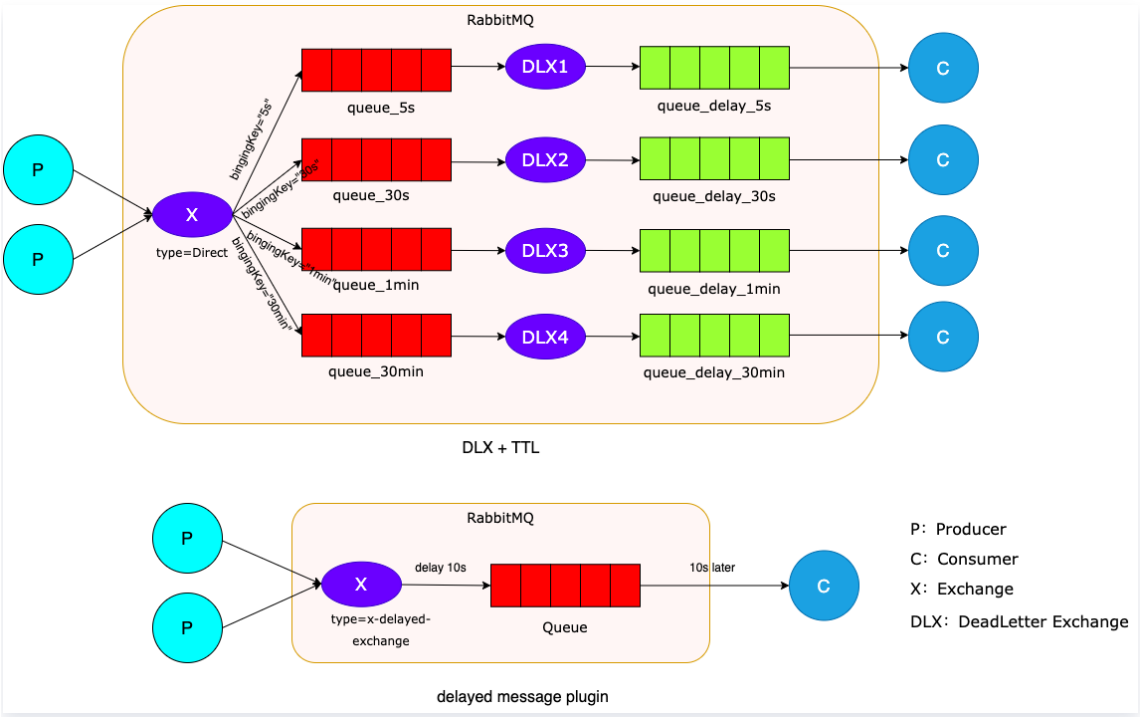
使用消息队列 RabbitMQ 版的优先级队列可以很好的支持这个场景，让优先级高的消息不会积压太久。如果发现是大客户的订单给一个相对比较高的优先级，优先被处理；否则就是默认优先级。



延迟消息场景

实际业务系统中，有发送延迟消息的需求。如果自己实现延迟逻辑，可靠性和延迟精度很难得到保障。使用消息中间件，如消息队列 RabbitMQ 版可以很好的处理此类需求。延迟消息的使用场景有：

- 订单系统中，用户下单后有30分钟时间进行支付。如果30分钟内没有支付成功，那么这个订单将进行异常处理。这里可以使用消息队列 RabbitMQ 版延迟消息能力来处理这些超时订单。
- 物联网系统中，用户希望通过手机遥控智能设备在指定的时间工作，这时就可以将指令发到延迟队列，当指令设定的时间到了，再将控制指令推送到智能设备。

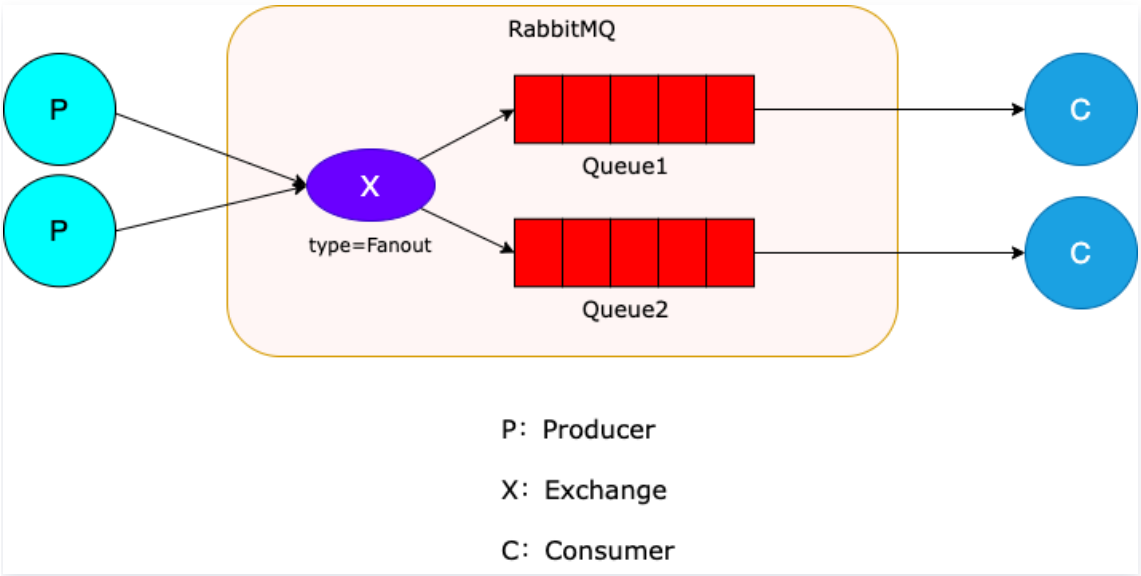


消息广播

很多业务系统需要将信息广播给下游的系统。如果通过 RPC 的方式，耦合比较重，对上游业务系统的压力也比较大。此时，可以使用消息队列 RabbitMQ 版的 Fanout Exchange 来处理此类需求。可以使用的场景：

- 大型多人在线游戏（MMO）可以将其用于排行榜更新或其他全局事件；

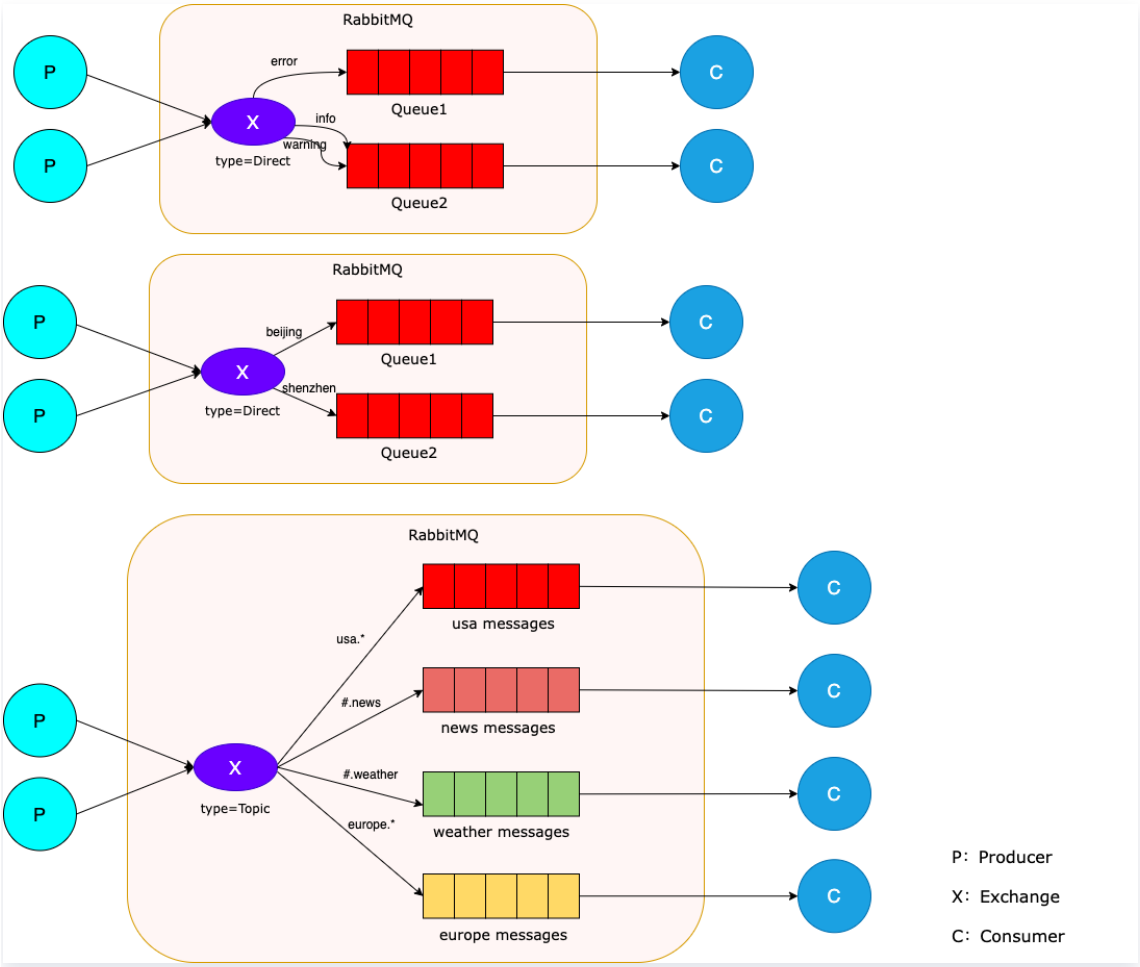
- 体育新闻网站可以使用扇形交换机向客户端近乎实时的分发比分信息；
- 分布式系统使用它来广播各种状态和配置更新；
- 群聊可以使用它在参与者之间分发消息。



灵活路由场景

随着微服务架构的流行，服务拆分得较细，服务的数据会以消息的形式，使用精心设计的分发策略发送到不同的队列中去。这时可以充分的利用消息队列 RabbitMQ 版灵活的消息路由的能力，将消息分发到目标 Queue 中。可以使用的场景：

- 日志处理场景，可以将日志按类型投递到不同的 Queue。例如 Error 单独一个处理队列，优先处理。
- 电商物流系统的物流信息按地域分发给不同的消费端进行处理。
- 复杂路由场景。


RabbitMQ
P
P
X
type=Direct
error
info
warning
Queue1
Queue2
C
C
RabbitMQ
P
P
X
type=Direct
beijing
shenzhen
Queue1
Queue2
C
C
RabbitMQ
P
P
X
type=Topic
usa.*
#.news
#.weather
europe.*
usa messages
news messages
weather messages
europe messages
C
C
C
C
P：Producer
X：Exchange
C：Consumer

使用限制

最近更新时间：2024-06-03 14:23:41

本文列举了 TDMQ RabbitMQ 版专享集群对集群和字符的限制，请您在使用产品时注意不要超出对应的限制值，避免出现异常。

集群

限制类型	限制说明
集群名称长度	3个字符 - 64个字符
集群名称规范	不能为空，只能包含数字、字母、“-”和“_”

Vhost

限制类型	限制说明
Vhost 名称长度	1个字符 - 64个字符
Vhost 名称规范	不能为空，只能包含字母、数字、“.”、“-”及“_”
Vhost 数量	每个集群限制20个

Exchange

限制类型	限制说明
Exchange 名称长度	1个字符 - 64个字符
Exchange 名称规范	不能为空，只能包含字母、数字、“.”、“-”及“_”
Exchange 路由类型	可选项包含direct，fanout，topic及headers，创建时必选其一
Exchange 数量	每个集群限制1000个

Queue

限制类型	限制说明
Queue 名称长度	1个字符 - 64个字符
Queue 名称规范	不能为空，只能包含字母、数字、“.”、“-”及“_”
Queue 数量	每个集群限制1000个

Channel

限制类型	限制说明
Connection 通道数	每个 Connection 的最大通道数为1024，如果用户在这个连接上创建的通道数超过这个限制则无法再创建通道

相关概念

基础概念

最近更新时间：2024-04-08 16:51:21

本文主要介绍了在使用 TDMQ RabbitMQ 版中常见的名词及解释说明。

Binding

RabbitMQ 中通过 Binding 将 Exchange 与 Queue 关联起来，这样 RabbitMQ 就知道如何正确地将消息路由到指定的 Queue了。

Binding key

- 在绑定（Binding）Exchange 与 Queue 的同时，一般会指定一个 binding key；消费者将消息发送给 Exchange 时，一般会指定一个 routing key；当 binding key 与 routing key 相匹配时，消息将会被路由到对应的 Queue 中。
- 在绑定多个 Queue 到同一个 Exchange 的时候，这些 Binding 允许使用相同的 binding key。
- binding key 并不是在所有情况下都生效，它依赖于 Exchange Type，例如 fanout 类型的 Exchange 就会无视 binding key，而是将消息路由到所有绑定到该 Exchange 的 Queue。

Channel

在客户端的每个物理 TCP 连接里，可建立多个 Channel，每个 Channel 代表一个会话任务。

Connection

TCP 连接，生产者或消费者与 TDMQ RabbitMQ 版间的物理 TCP 连接。

Exchange

生产者将消息发送到 Exchange，由 Exchange 将消息路由到一个或多个 Queue 中（或者丢弃）。
Exchange 根据消息的属性或内容路由消息。

Exchange Types

RabbitMQ 常用的 Exchange Type 有 fanout、direct、topic 、header 等。

- Fanout：fanout 类型的 Exchange 会把所有发送到该 Exchange 的消息路由到所有与它绑定的 Queue 中。
- Direct：Direct 类型的 Exchange 会把消息路由到那些 binding key 与 routing key 完全匹配的 Queue 中。
- Topic：该类型与 direct 类型相似，Topic 类型 Exchange 支持多条件匹配和模糊匹配，即使用 Routing Key 模式匹配和字符串比较的方式将消息路由至与其绑定的 Queue 中。

- Header：与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个 map 键值对，通过这个map 对象实现 Queue 和 Exchange 的绑定。当消息发送到 RabbitMQ 时会取到该消息的 Headers 与 Exchange 绑定时指定的键值对进行匹配；如果完全匹配则消息会路由到该队列，否则不会路由到该队列。

Message acknowledgment

消息回执，消费者在消费完消息后发送一个回执给 RabbitMQ，RabbitMQ 收到回执后才将消息从队列中删除，如果没有收到回执并检测到消费者的 RabbitMQ 连接断开，那么会将该消息发送给其他消费者进行处理。

Message durability

消息持久化，即 RabbitMQ 服务重启的情况下，也不会丢失消息，我们可以将 Queue 与 Message 都设置为（durable），这样可以保证绝大部分情况下我们的 RabbitMQ 消息不会丢失。

Prefetch count

消费者在开启 acknowledge 的情况下，对接收到的消息可以根据业务的需要异步对消息进行确认，prefetch 允许为每个 consumer 指定最大的 unacked messages 数目。例如：设置prefetchCount=10，则 Queue 每次给每个消费者（假设一共2个AB）发送10条消息，消费者无需马上回应，消费者将10条消息缓存本地客户端；当一个消费者处理完1条消息后 Queue 会再给该消费者发送一条消息。如果两个消费者都没有回复任何一条 ack，则 queue 就不会继续发送消息。

Queue

Queue（队列）是 RabbitMQ 的内部对象，用于存储消息。每个消息都会被投入到一个或多个 Queue 里。

Routing key

- 生产者在将消息发送到 Exchange 的时候，一般会指定一个 routing key，来指定这个消息的路由规则，而这个 routing key 需要与 Exchange Type 及 binding key 联合使用才能最终生效。
- 在 Exchange Type 与 binding key 固定的情况下（在正常使用时一般这些内容都是固定配置好的），我们的生产者就可以在发送消息给 Exchange 时，通过指定 routing key 来决定消息流向哪里。

Vhost

虚拟主机（Virtual Host，Vhost），用作逻辑隔离，分别管理各自的 Exchange、Queue 和 Binding，使得应用安全的运行在不同的 Vhost 实例上，相互之间不会干扰。一个实例下可以有多个 Vhost，一个 Vhost 里面可以有若干个 Exchange 和 Queue。生产者和消费者连接消息队列 RabbitMQ 版需要指定一个 Vhost。

Exchange

最近更新时间：2023-02-08 14:28:32

本文介绍 TDMQ RabbitMQ 版中 Exchange 的概念、类型和使用方式。

概念

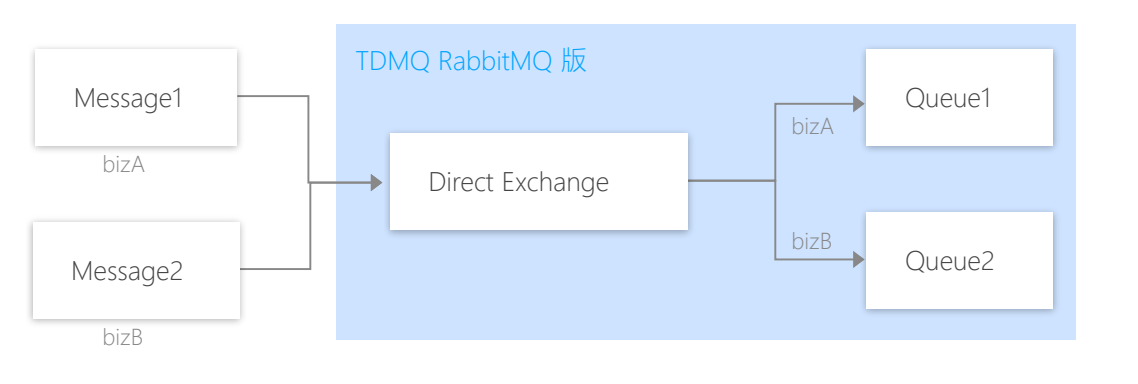
Exchange 是 TDMQ RabbitMQ 版的消息路由代理，Producer 将消息发送到 Exchange 中，Exchange 根据消息的属性或内容将消息路由到一个或多个 Queue 中（或者丢弃），Consumer 从 Queue 中拉取消息进行消费。

TDMQ RabbitMQ 版目前支持 Direct、Fanout、Topic 和 Header 四种类型的 Exchange。

- Direct：该类型 Exchange 会把消息路由到 RoutingKey 和 BindingKey 完全匹配的 Queue 中。
- Fanout：该类型 Exchange 会将消息路由到所有与其绑定的 Queue 中。
- Topic：该类型 Exchange 支持多条件匹配和模糊匹配，即使用 RoutingKey 模式匹配和字符串比较的方式将消息路由至与其绑定的Queue中。
- Header：该类型 Exchange 与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个map键值对，通过这个map对象实现消息队列和交换机的绑定。

Direct Exchange

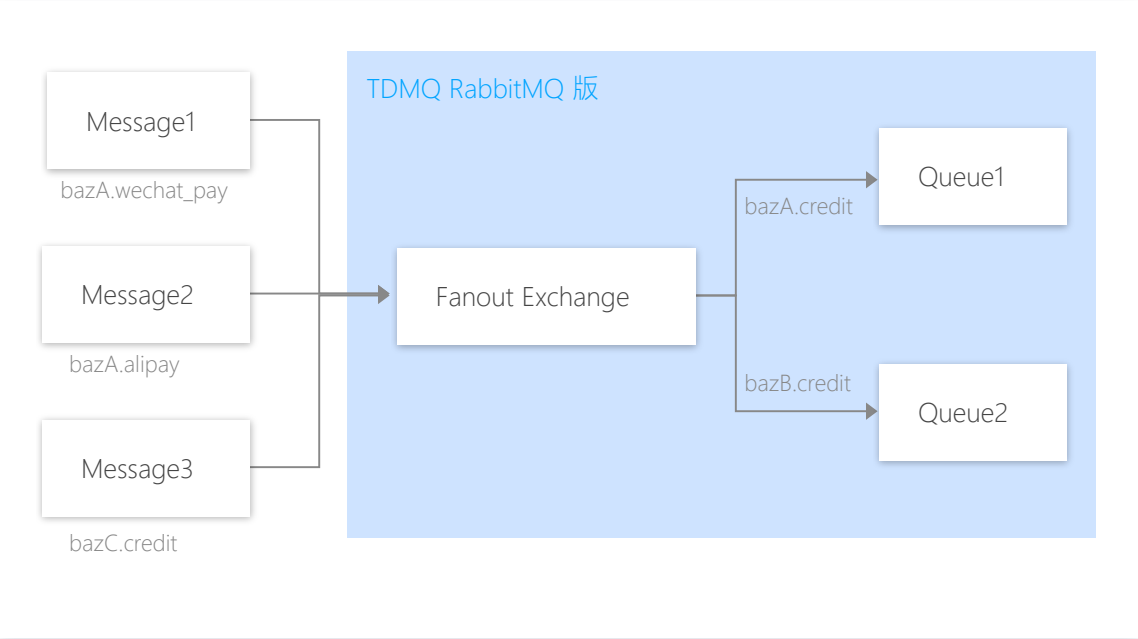
路由规则：Direct Exchange 会把消息路由到 RoutingKey 和 BindingKey 完全匹配的 Queue 中。
应用场景：该类型 Exchange 适用于通过简单字符标识符过滤消息的场景，常用于单播路由。
使用示例：



Message	Routing Key	Binding Key	Queue
Message 1	bizA	bizA	Queue 1
Message 2	bizB	bizB	Queue 2

Fanout Exchange

路由规则： 该类型 Exchange 会将消息路由到所有与其绑定的 Queue 中。
应用场景：该类型 Exchange 适用于广播消息的场景。例如分发系统使用 Fanout Exchange 来广播各种状态和配置更新。
使用示例



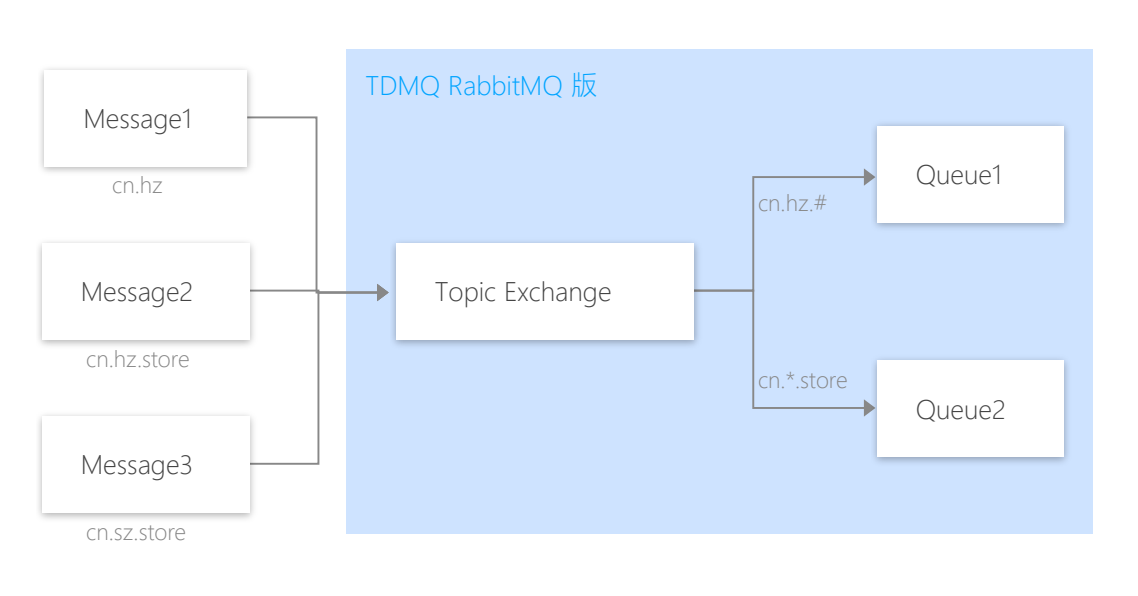
Message	Routing Key	Binding Key	Queue
Message 1	bazA.wechat_pay	bazA.credit , bazB.credit	Queue 1，Queue 2
Message 2	bazA.alipay	bazA.credit , bazB.credit	Queue 1，Queue 2
Message 3	bazC.credit	bazA.credit , bazB.credit	Queue 1，Queue 2

Topic Exchange

路由规则：该类型 Exchange 支持多条件匹配和模糊匹配，即使用 Routing Key 模式匹配和字符串比较的方式将消息路由至与其绑定的 Queue 中。

- Topic Exchange 支持的通配符包括星号“*”和井号“#”。
- 星号“*”代表一个英文单词，例如 sh。
- 井号“#”代表零个、一个或多个英文单词，单词间用英文句号"."分隔，例如 cn.hz。

应用场景
该类型 Exchange 常用于多播路由，例如需要使用 Topic Exchange 分发有关于特定地理位置的数据。
使用示例



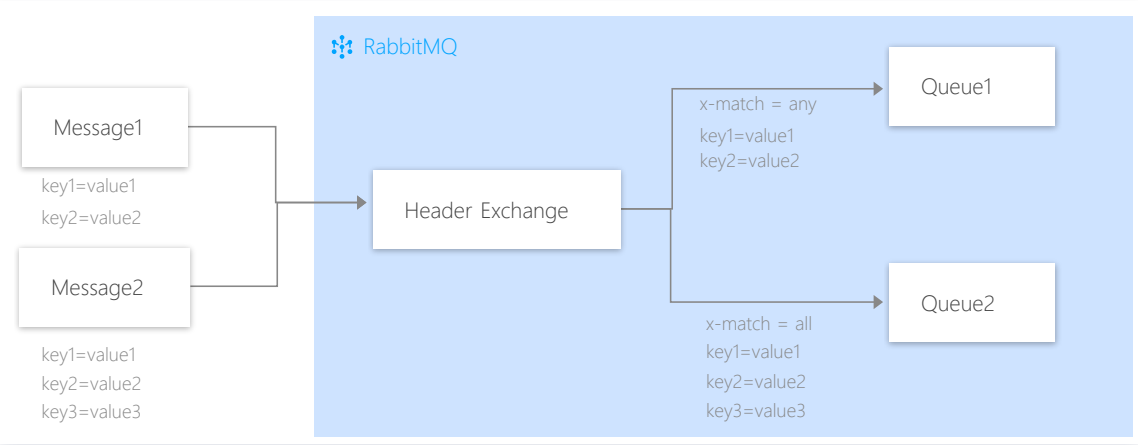
Message	Routing Key	Binding Key	Queue
Message 1	cn.hz	cn.hz.#	Queue 1
Message 2	cn.hz.store	cn.hz.# 、 cn.*.store	Queue 1、Queue 2
Message 3	cn.sz.store	cn.*.store	Queue 2

Header Exchange

与 Routing Key 无关，匹配机制是匹配消息中的 Headers 属性信息。在绑定 Queue 与 Headers Exchange 之前声明一个map键值对，通过这个map对象实现Queue 和 Exchange 的绑定。当消息发送到 RabbitMQ 时会取到该消息的 Headers 与 Exchange 绑定时指定的键值对进行匹配；如果完全匹配则消息会路由到该队列，否则不会路由到该队列。

匹配规则x-match有下列两种类型：

- x-match = all ： 表示所有的键值对都匹配才能接受到消息
- x-match = any ： 表示只要有键值对匹配就能接受到消息



Message	消息Headers属性	Binding Headers 属性	Queue
Message 1	key1=value1 key2=value2	x-match = any key1=value1 key2=value2	Queue 1
Message 2	key1=value1 key2=value2 key3=value3	x-match = any key1=value1 key2=value2 、 x-match = all key1=value1 key2=value2 key3=value3	Queue 1、 Queue 2