初探 RabbitMQ

作者：kevin2020
原文链接：https://ld246.com/article/1516298085214
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

<h2 id="优势">优势</h2>
RabbitMQ 是除了 Qpid 之外，唯一实现了 AMPQ 标准的代理服务器
正式由于 Erlang，RabbitMQ 集群不可思议的简单
RabbitMQ 性能更高一些
<h3 id="术语">术语</h3>
AMPQ 高级消息队列协议
<h3 id="消费者与生产者">消费者与生产者</h3>
生成者发送消息到 RaabitMQ,消息包含 2 个部分
有效载荷 payload 传输的数据内容
标签 label 一个交换器的名称和可选的主题标记
这是一种发后即忘得单向方式 fire-and-forget
消费者只能拿到有效载荷，想要知道发送方是谁，
除非发送方将信息放入有效载荷中
P/S 必须先建立一条信道 channel
会在应用程序与 RabbitMQ 间建立 TCP 连接，一旦连接打开
应用程序可以创建一条 amqp 信道。
信道是建立在真实地 tcp 连接内的虚拟连接
AMQP 命令都是通过信道发送出去的。
每条信道会被指派唯一一个 ID
<blockquote>
为什么不直接使用 TCP 连接？
操作系统建立 tcp 连接的代价是十分缓慢的
线程启动后，会在现有连接上创建一条信道，也获得了连接到 RQ 的私密通道
不会给操作系统带来负担，每秒成百上千次的创建信道也不会影响操作系统
一条 TCP 连接上创建多少信道是没有限制的，可以把信道想象成一束光纤电缆
电缆相当于 TCP 连接 光纤束相当于信道
</blockquote>
<h2 id="队列">队列</h2>
AMPQ 消息路由必须有三部分：交换器，队列，绑定
消息最先到达交换器
绑定决定了消息从路由器路由到特定队列
AMPQ 栈 交换器 绑定 队列
消费者通过以下方式接受消息
basic.consume 命令订阅 会将信道设置为接受模式直到取消订阅
basic.get 获得单条消息
<blockquote>
将 basic.get 放入循环中即模拟 basic.consume 不过性能大打折扣
</blockquote>
无人订阅的队列，消息就在队列中暂存
一旦有人订阅，消息会被发送
多个消费者，以循环的方式发送给消费者 每条消息只发送给一个消费者
消息投递方式如下
消息 A 到达队列
RQ 将消息 A 发送给 user1
user1 确认收到消息 A
RQ 将消息 A 从队列删除
消费者必须通过 basic.ack 命令显示的向 RQ 确认
默认 aut_ack 为 true 一旦消费者接受消息自动视为其确认接收
<blockquote>
确认消息 与 消息被接受并不等价 确认命令可以告诉 RQ 消息被正确接受
</blockquote>
如果消费者收到一条消息，确认前程序崩溃了，RQ 会认为该消息未被分发
将传给下一位消费者
再上一条消息被确认之前，RQ 不会为该消费者继续发送消息

明确拒绝消息而不是确认消息
1 消费者从 RQ 断开连接 好处所有版本都支持
坏处开销比较大
2 RQ2.0.0 及以上版本 可使用 basic.reject 命令
reject 命令参数设为 true 消息会被发送给下一位消费者
设为 false RQ 会将其从队列中移除
<blockquote>
当你发现消息格式有误时，也可以直接确认接受来忽略消息
这种忽略消息的方式所有版本都支持
为什么不直接确认来忽略消息？RQ 未来版本会有一个死信 dead letter 队列
专门存放被拒绝不被重入的消息 用来帮助用户分析
</blockquote>
<h3 id="如何创建队列">如何创建队列</h3>
消费者生产者都能使用 queue.declare 命令来创建队列
消费者在同一条信道上订阅了另一个队列时，就无法在声明队列了
必须先取消队列 将信道置为传输模式。
当创建队列时，需要指定队列名称 消费者订阅时需要队列名称
绑定时也需要队列名称 如不指定 RQ 会分配一个随机名称并在
declare 命令响应中返回
参数 exclusive 设置为 true 队列私有只有你的应用程序可以消费队列信息 限制一个队列只有一个消者
auto-delete 当最后一个消费者取消订阅时 队列会自动移除
如果尝试声明一个已经存在的队列
声明参数若完全相同 什么都不做成功返回，就像这个队列创建成功一样
参数不匹配 队列会声明尝试失败
将 queue.declare passive 设置为 true
队列存在 成功返回
不存在 不会创建队列 并返回一个错误
<h3 id="谁来创建队列">谁来创建队列</h3>
生产者消费者都应该尝试创建队列
如果发出去的消息路由到不存在的队列那么 RQ 会将其忽略
<blockquote>
如果能接受消息丢失，或者实现一种方法来重新发布未处理的消息 可以只让消费者声明队列
</blockquote>
<h3 id="队列总结">队列总结</h3>
为消息提供了住所 消息在这等待消费
对负载均衡来说 队列简直完美，附加一堆消费者，让 RabbitMQ 以循环方式均匀分配消息
队列时 Rabbit 中消息的最后终点 除非消息进了黑洞。
<h2 id="交换器与绑定">交换器与绑定</h2>
消息到达交换器后，将按照绑定的方式路由到队列
<h3 id="四种交换器">四种交换器</h3>
<h4 id="headers">headers</h4>
允许匹配 AMQP 消息而非路由键，除此之外与 direct 交换器完全一致
但性能会差很多。不是很实用，基本用不到了
<h4 id="direct">direct</h4>
包含一个空白字符串名称的默认交换器，当你声明一个队列时，
它会自动绑定到默认的交换器，并以队列名称作为路由键
basic_publish($msg,'','queue-name');
第一个参数为发送内容，第二个为空字符串，指定默认交换器，第三个为路由键，
这个路由键就是之前用来声明队列的名称
exchange.declare 命令发送自己的交换器
<h4 id="fanout">fanout</h4>
会将收到的消息广播到绑定的队列上
<h4 id="topic">topic</h4>

按类型匹配到指定的队列上
*将.视为分隔符；#没有分割概念将.视为匹配的部分
<h3 id="虚拟机与隔离">虚拟机与隔离</h3>
vhost 虚拟主机 拥有自己的权限机制
vhost 之于 rabbitmq 就如虚拟机之于服务器上一样
默认为开箱即用的 vhost:/
创建时，若有集群，集群中都会创建该 host
rabbitmqctl add_vhost[vhost_name]
rabbitmqctl delete_vhost[vhost_name]
rabbitmqctl list_vhosts
<blockquote>
指定-n rabbit@[server_name] 远程管理 必须确保服务器与控制台安装了相同的 Erlang cookie
/p>
</blockquote>
<h3 id="消息持久化">消息持久化</h3>
队列与交换器 durable 属性 默认 false
消息投递模式 delivery mode 设置为 2
最终要的一点 发送后必须到达持久化队列
保存在日志文件中一旦持久化队列消费了一条持久化信息时 日志会标记该消息等待垃圾收集
开启持久化后会严重影响性能吞吐量至少下降 10 倍 使用 ssd 可以极大提高持久化性能
事务 把信道设置为事务模式
劣势不但性能大幅度下降 而且会使生产者应用程序产生同步
<blockquote>
方案代替 将信道设置为 confirm 模式 只有重新创建信道来关闭该设置
成功后会返回 id 号默认 1 丢失返回 nack
</blockquote>
<h3 id="流程">流程</h3>
连接到 rq 获得信道 声明交换器 声明队列 绑定队列与交换器 消费消息 关闭信道 关闭连接