消息队列 CMQ

消息主题（Topic）模型

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

版权所有：腾讯云计算（北京）有限责任公司

消息主题（Topic）模型
标签匹配功能说明

最近更新时间：2020-10-13 10:38:57

CMQ 的 Topic 模式支持订阅筛选标签（tag）功能，类似于 rabbitMQ 的 direct_routing 模式，topic_pattern 模式在下个迭代提供。由于筛选标签的使用策略较复杂，本文将结合不同场景进行说明。

场景说明

场景1

有4个订阅者 A、B、C、D，A 消息标签是 apple，B 消息标签是 xiaomi，C 的消息标签是 imac+xiaomi，D 没有设置任何标签。
此时，有生产者 publish 到 Topic 100条消息，带的消息标签过滤为：apple、imac、iphone、macbook。此后 Topic 立即投递到 A、B、C、D。

场景分析：

订阅者	消息标签	消息接收说明
A	apple	由于 apple 能匹配消息过滤标签中的 apple，则能正常收到100条消息。
B	xiaomi	不能收到任何消息。
C	imac+xiaomi	由于其中一个标签 imac 能匹配，则能正常收到100条消息 。
D	-	可以收到任何消息。

场景2

Topic 有且仅有4个订阅者 A、B、C、D，四个订阅者都没有设置消息 tag。
此时，有生产者 publish 到 Topic 100条消息，带的消息标签过滤为：apple、imac、iphone、macbook。此后 Topic 立即投递到 A、B、C、D。

场景分析：
订阅者 A、B、C、D 均没有 tag，则投递时不用匹配消息， A、B、C、D 都能收到这100条消息。

场景3

Topic 有且仅有1个订阅者 A，A 的订阅标签设置为 xiaomi。
此时，有生产者 publish 到 Topic 100条消息，带的消息标签过滤为：apple、imac、iphone、macbook。此后 Topic

版权所有：腾讯云计算（北京）有限责任公司

立即投递到 A。

场景分析：

- 对于订阅者 A，订阅标签不匹配，则 A 无法收到这100条消息。此时这100条消息立即丢弃，不会堆积在 CMQ 里。
- 生产者 publish 到 Topic 时，消息过滤 tag 只能在 publish 前设置一次，跟消息 ID 绑定，不可修改。

场景4

Topic 名为 test1 ，在12点01分，调用了订阅发布的 API，定义该次操作为“发布 test1”，发布后 Topic 投递到订阅者，给 A、B、C 三个订阅者投递了200条消息。
假设结果为：A 有100条消息接收失败，B 有30条消息接收失败，C 的200条消息全部接收成功

场景分析：

- **消息堆积**：在 A、B、C 三个订阅者中，假定消息投递失败的总集合为110条（A 有100条消息失败，B 有30条失败，C全部成功，三者之间的失败消息有交错的部分），只要存在任意1个订阅者与某条消息存在关联关系，该消息都不会被立即退订。
- **阻塞策略**：以 A 为案例，Topic 投递给 A 200条消息，第101条失败时候，则后面的99条的投递会被阻塞。状态为有100条消息接收失败。
- **重试策略（退避重试）**：Topic-test1 会每隔 N 秒给 A 重新投递消息，失败的100条里，按顺序，从第一条重新尝试，若失败三次，直接就丢弃，然后投递下一条消息，失败三次后再次丢弃，按顺序排列。
- **重试策略（衰退指数重试）**：以 A 为案例，Topic 会并发投递100条消息（无法保证先后顺序）当订阅者接收的第一条消息失败后，从第一条重新尝试，若失败，继续阻塞。后续新增的投递消息，会继续阻塞。
- **消息生命周期不可延长**：假定这110条堆积在 Topic-test1 里的消息，中间无论被重试多少次，生命周期都为1天，从生产者推送到 Topic 的时间点，作为起止时间点，到期后删除
- **重新推送**：生产者不断往 Topic 中生产新的消息，客户在12点02分，又调用了订阅发布的 API，加上之前失败堆积的110条消息，假设目前 Topic 中共有110（失败堆积）+100（1分钟时间内新增的消息）=210条消息，再次投递。这时，A、B 的重试策略为衰退指数重试，处于“失联”状态，则210条消息会继续堆积。 对于 C，只接收新增的100条消息。

> 每一条消息的 ID，作为 key，value 是关联的订阅者、代表每个订阅者消费成功与否。

场景5

Topic 名为 test2 ，在12点01分，调用了订阅发布的 API，定义该次操作为“发布 test2”，给 A、B、C 三个订阅者投递了200条消息。
假设结果为：A、B、C 的200条消息全部接收成功。

场景分析：

若 Topic-test2，有且仅有 A、B、C 三个订阅者，且都确定消费成功后，Topic 会立即删除200条消息。

规律总结

通过以上不同场景，标签匹配规律如下：

订阅者	订阅者有无 tag	有无消息 tag	消息接收说明
A	有	无	订阅者不匹配，不能收到消息。
B	无	有	投递时消息不用匹配，订阅者都能收到消息。
C	有	有	两者匹配的，才能收到消息。支持 N:M 匹配，如消息有10个 tag，订阅者有4个 tag，其中有1个 tag 相互能匹配上，则订阅者能收到消息。
D	无	无	投递后，所有订阅者都能收到消息。

版权所有：腾讯云计算（北京）有限责任公司

创建主题

最近更新时间：2020-08-17 14:54:39

操作场景

该任务指导您通过控制台，创建一个主题（Topic）。

操作步骤

> ⚠ **注意：**
>
> 主题订阅需要使用主账号创建，否则会导致投递消息失败。

1. 登录 消息队列 CMQ 控制台，单击左侧菜单栏的【主题订阅】。
2. 在主题订阅页面左上角，选择地域。
3. 在主题订阅页面，单击【新建】，根据页面提示填写信息。
4. 单击【提交】，即可在指定地域下创建一个主题（Topic）。

创建 Topic 时，用户需要指定以下属性值：

属性	说明
地域	设定 Topic 的地域属性，在主题订阅页面顶部选择。
主题名	输入 Topic 名称，该名称一旦创建后不能修改。 只能包含字母、数字、短横线 `-` 和下划线 `_` ；名称长度限制在3 - 64字节之间，长于64字节将被自动截取；名称严格区分大小写，为了防止混淆，不允许创建大小写同名主题。
消息最大长度	Topic 的 MaximumMessageSize 属性，允许发送到该主题的消息体的最大长度，单位为 byte，有效值范围为：1KB - 1024KB，默认值为64KB。
消息生命周期	消息在本主题中最长的存活时间，从发送到该队列开始经过此参数指定的时间后，不论消息是否被取出过都将被删除；单位为秒，默认值为86400s，不允许修改。
消息堆积	默认开启。存在生产者的消息，还未触发推送到订阅者，或订阅者接收消息失败，暂时堆积到 Topic 中。 在 Topic 列表中，可查看当前消息堆积的总数（近似值）。

版权所有：腾讯云计算（北京）有限责任公司

属性	说明
消息过滤类型	标签：请参考 标签匹配功能说明。 路由匹配：请参考 路由键匹配功能说明。

版权所有：腾讯云计算（北京）有限责任公司

修改主题属性

最近更新时间：2020-08-17 11:29:36

操作场景

该任务指导您通过控制台，修改已创建的主题属性。

前提条件

已完成 创建主题。

操作步骤

1. 登录 消息队列 CMQ 控制台，单击左侧菜单栏的【主题订阅】。
2. 在主题订阅列表中，单击操作列的【修改配置】，即可修改消息最大长度。
3. 单击【提交】，完成修改。

- 主题的地域属性，不可修改。
- 主题名称、主题 ID，两者都不可修改。
- 消息最大长度：可修改，该项目为允许发送到该主题的消息体的最大长度，有效值范围为1KB - 1024KB，默认值为64KB。
- 创建时间、修改时间，两者都不可修改。

订阅主题

最近更新时间：2020-08-04 15:52:26

您可以通过指定以下属性订阅一个主题：

- 输入 Topic 名称。
- 输入 Topic 资源 ID。
- 填写订阅名称，该名称一经填写无法修改。
- 填写订阅终端协议，选项有：Queue 消息服务、URL 地址。
- 订阅地址，填写 URL、Queue 队列名称，目前只允许 Topic 发送给同一个账户下的 Queue。
- 重试策略：订阅的 NotifyStrategy 属性，向接收端推送消息出现错误时的重试策略。该策略默认开启。您需要在以下两个选项中选择一个：
  - 退避重试：重试3次，间隔时间10 - 20s之间的一个随机值，超过3次后，该条消息对于该订阅者丢弃，不会再重试。
  - 衰退指数重试（默认勾选）：重试176次，总计重试时间为1天，间隔时间依次为：2^0，2^1，…，512，512，…，512秒。
- 重试验证：若 HTTP 返回码为200，则认为成功。
- 添加订阅者标签：添加订阅者时，可增加 FilterTag。增加 FilterTag 后，该订阅者仅能收到带该 FilterTag 的消息。单个 tag 为不超过16个字符的字符串，单个订阅者最多可添加5个 tag。只要其中某个 tag 能匹配 Topic 的过滤标签，订阅者即可收到该次 Topic 投递的消息，若消息不带任何标签，则该订阅者无法收到该类型消息。
- 单个 Topic 订阅者上限 ：单个 Topic 下最多允许关联500个订阅者。
- 该订阅者关联的消息总数量：该数量为近似数值，说明某 Topic 等待投递、重试投递到订阅者的消息数量。

版权所有：腾讯云计算（北京）有限责任公司

生产者发布消息至主题

最近更新时间：2021-06-29 18:08:43

生产者通过指定以下信息向主题推送消息：

- Topic 名称。
- Topic 资源 ID。
- 发布主题，由用户自定义填写。
- 发布内容：为消息的主体内容，客户自定义填写，CMQ 不会进行任何编码和修改。
- 添加消息过滤标签：Tag，即消息标签、消息类型，用来区分某个 CMQ 的 Topic 下的消息分类。允许消费者按照 Tag 对消息进行过滤，确保消费者最终只消费到他关心的消息类型。该功能默认不开启，未开启时，所有消息向所有订阅者发送，当订阅者设置了 Tag 时，由于不匹配，该订阅者无法收到消息。消息过滤标签描述了该订阅中消息过滤的标签（标签一致的消息才会被推送）。单个 Tag 不超过16个字符的字符串，单条消息可最多添加5个 Tag。

版权所有：腾讯云计算（北京）有限责任公司

路由键匹配功能说明

最近更新时间：2020-08-04 15:38:47

CMQ 的路由键匹配功能类似于 rabbitMQ 的 exchange queue，可以用于消息过滤，根据不同的条件使订阅者获取不同的消息。创建 topic 时，可开启【路由匹配键】。

使用说明

Binding key、Routing key 是组合使用的，提供类似于 RabbitMQ 的消息过滤能力。发消息时配的 Routing key 是客户端发消息带的。创建订阅关系时配的 Binding key 是 topic 和 订阅者的绑定关系。

使用限制

- Binding key 的数量不超过5个。单个 binding key 的长度 <= 64字节，用于表示发送消息的路由路径，最多含有15个“.”，即最多16个词组。
- Routing key 的数量由1个字符串组成。单个 Routing key 的长度<=64 字节，用于表示发送消息的路由路径，最多含有15个“.”，即最多16个词组。

通配符说明

- *（星号），可以替代一个单词（一串连续的字母串），不能为空。
- #（井号）：可以匹配零个或多个字符。

示例：

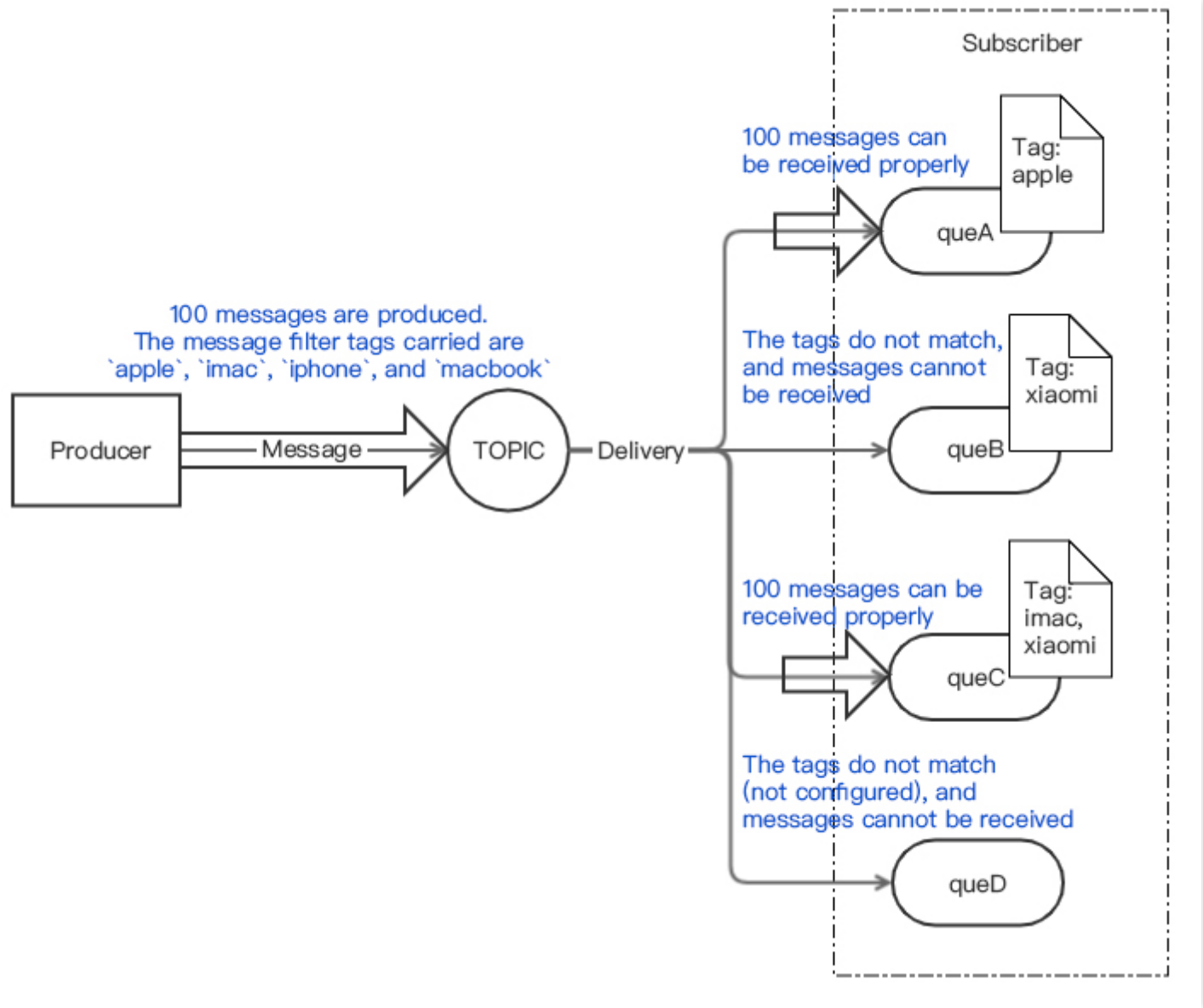
- 订阅者是『1.*.0』，此时消息为『1.任意字符.0』，则订阅者都能收到消息。
- 订阅者是『1.#.0』，此时消息为『1.2.3.4.4.2.2.0』，『1.0』则订阅者都能收到消息（消息中间元素随意）。
- 订阅者是『#』，则所有消息订阅者都能收到。

版权所有：腾讯云计算（北京）有限责任公司

主题向订阅者投递消息

最近更新时间：2020-07-23 16:20:14

主题（Topic）向订阅者投递消息的模型如下：



Topic 向订阅者投递消息时，遵循以下原则：

- Topic 会尽最大努力将生产者 publish 的消息，投递（notification）到订阅者。
- 当投递重试多次仍失败后，消息会堆积在 Topic 中，等待下一次投递；若持续失败，将在消息最大生命周期（1天）结束后，丢弃该消息。

Topic 向订阅者投递消息失败时，可能的原因如下：

- 投递消息时需要提供 SecretId 和 SecretKey（需要持久密钥），可在 API 密钥管理 中获取。
- 订阅需要使用根账号创建，不能用子账号。
- 如果队列名称不存在，请在 CMQ 控制台 检查队列是否存在。

版权所有：腾讯云计算（北京）有限责任公司