小象科技 ChinaHadoop.cn

# Redis 从入门到精通

黄健宏（huangz）

小象科技
ChinaHadoop.cn

# 发布与订阅

定义与模型

# 发布与订阅功能

小象科技 ChinaHadoop.cn

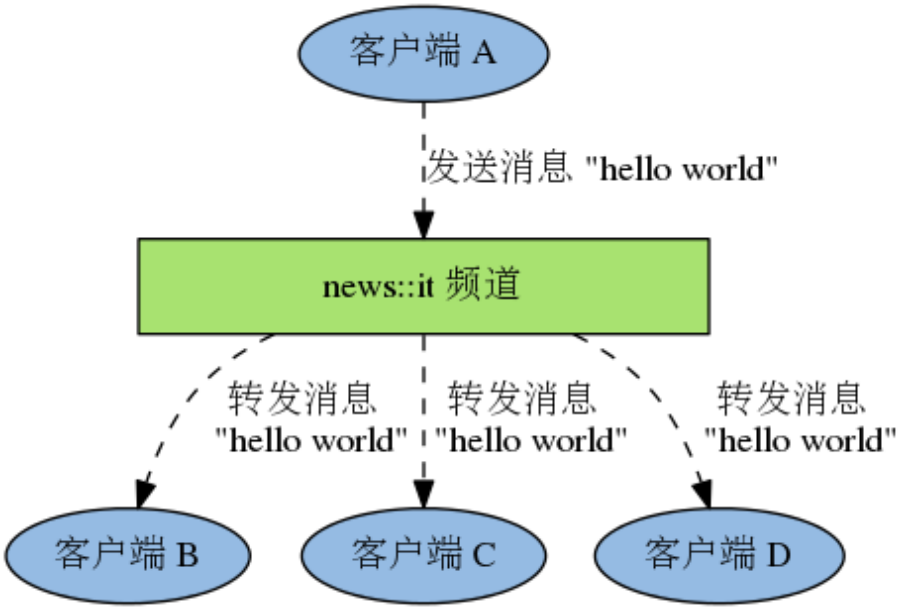
Redis 的发布与订阅功能可以让用户将消息同时发送给多个客户端。

这个功能由几个不同的角色 协作组成:

- 发布者(publisher):发布消息的客户端。
- 频道(channel):构建在服务器内部，负责接收发布者发送的消息，并将消息转发给频道的订阅者。
- 模式(pattern):构建在服务器内部，负责对频道进行匹配，当被匹配的频道接到消息时，模式也会将消息转发给模式的订阅者。
- 订阅者(subscriber):通过订阅频道或者模式来获取消息的客户端。每个频道或者模式都可以有任意多个订阅者。

# 频道的订阅与消息发布



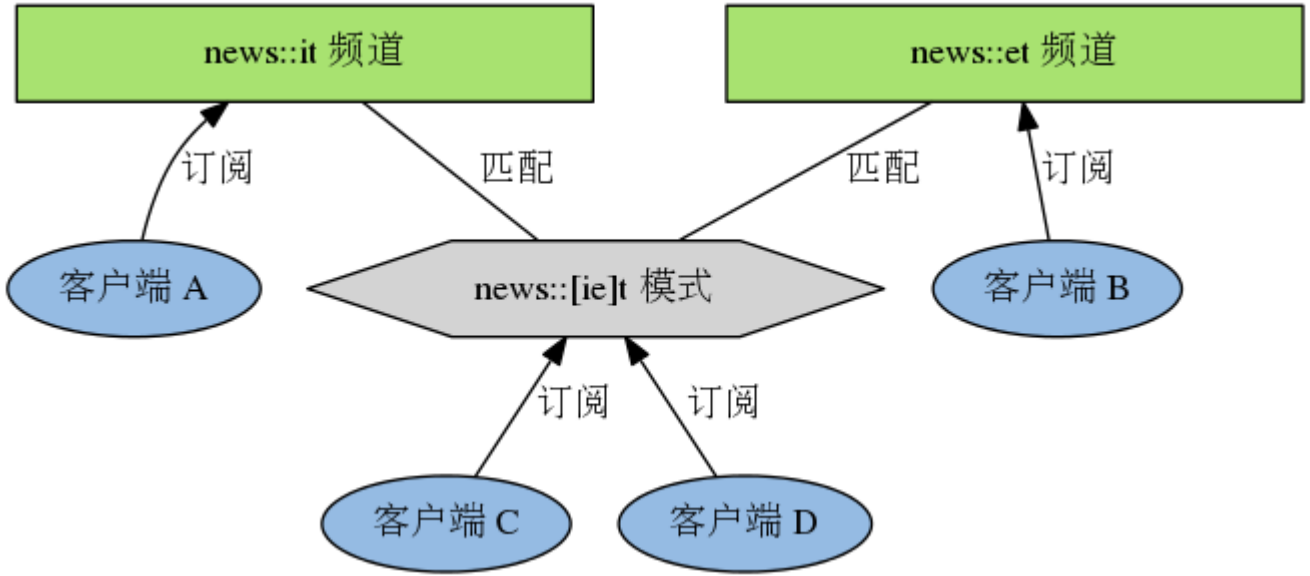



B 、C 、D 三个客户端正在订阅 news::it 频道。

当客户端 A 向 news::it 频道发送消息 “hello world” 时，该消息将被频道转发至 B 、C 、D 三个客户端。

# 模式的订阅与消息发布

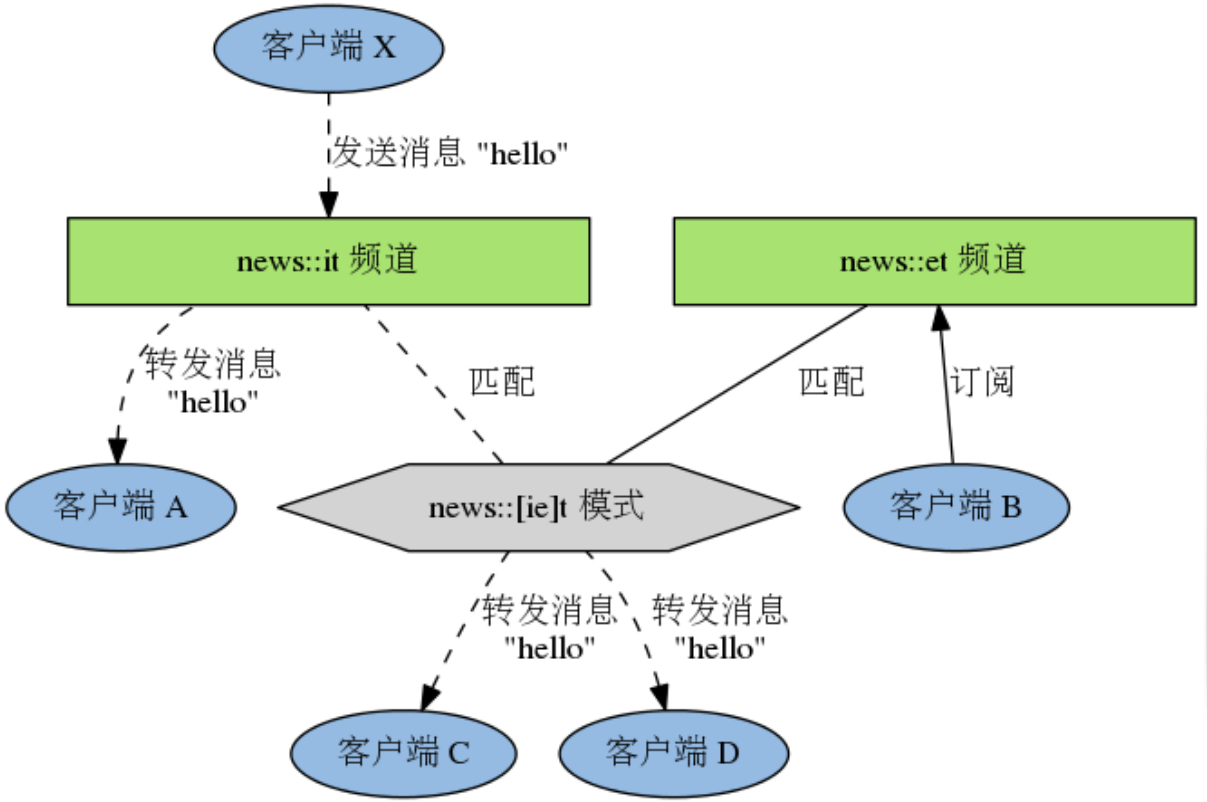


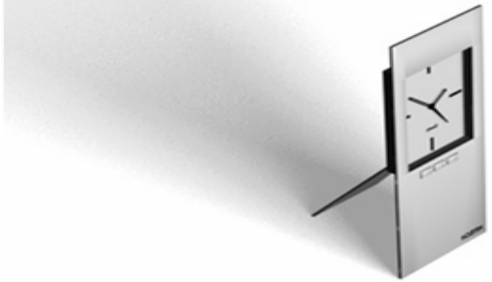



客户端 C 和 D 订阅了 news::[ie]t 模式，而这个模式又和 news::it 、news::et 两个频道匹配，当 news::it 频道或者 news::et 频道接收到消息的时候，这些消息不仅会被转发给频道的订阅者，也会被转发给客户端 C 和 D 。

# 模式订阅者接收消息示例(1)

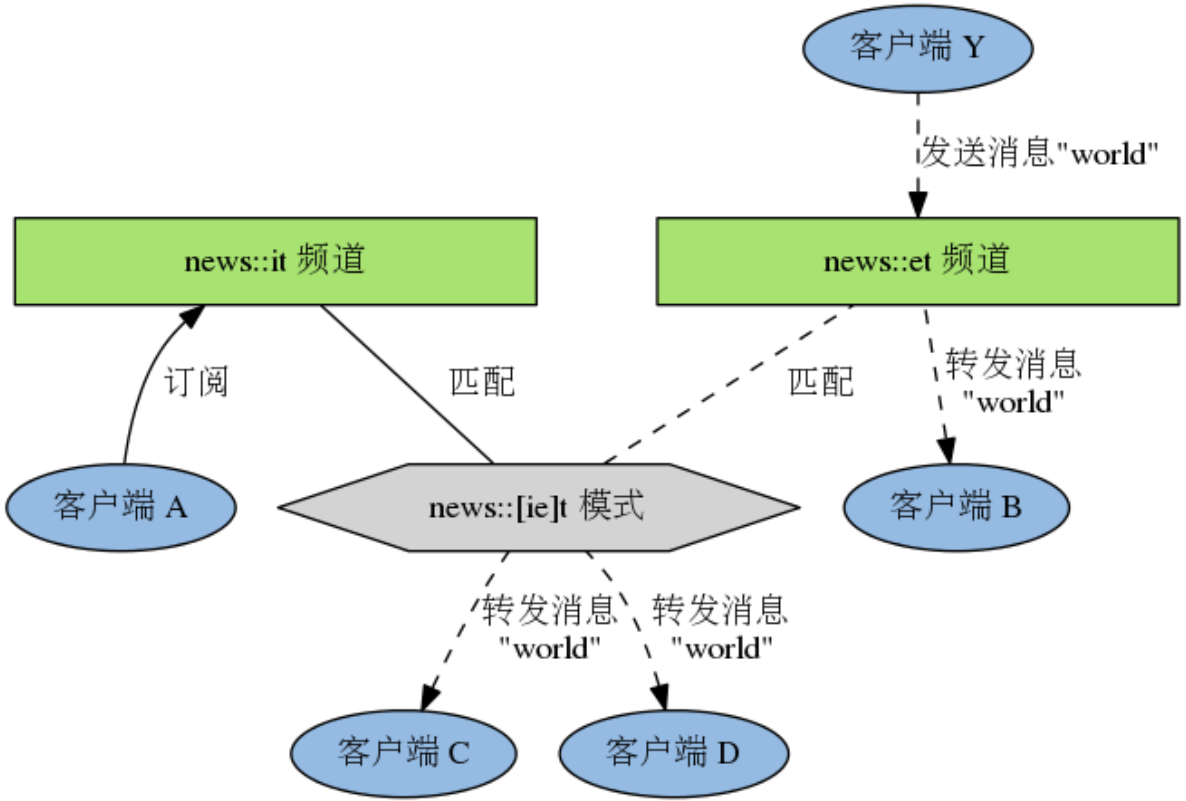


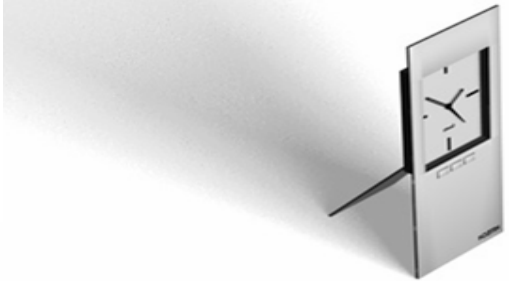


# 模式订阅者接收消息示例(2)





# 订阅命令与发布命令

订阅频道或模式、退订频道或模式、发布消息。

# 订阅频道

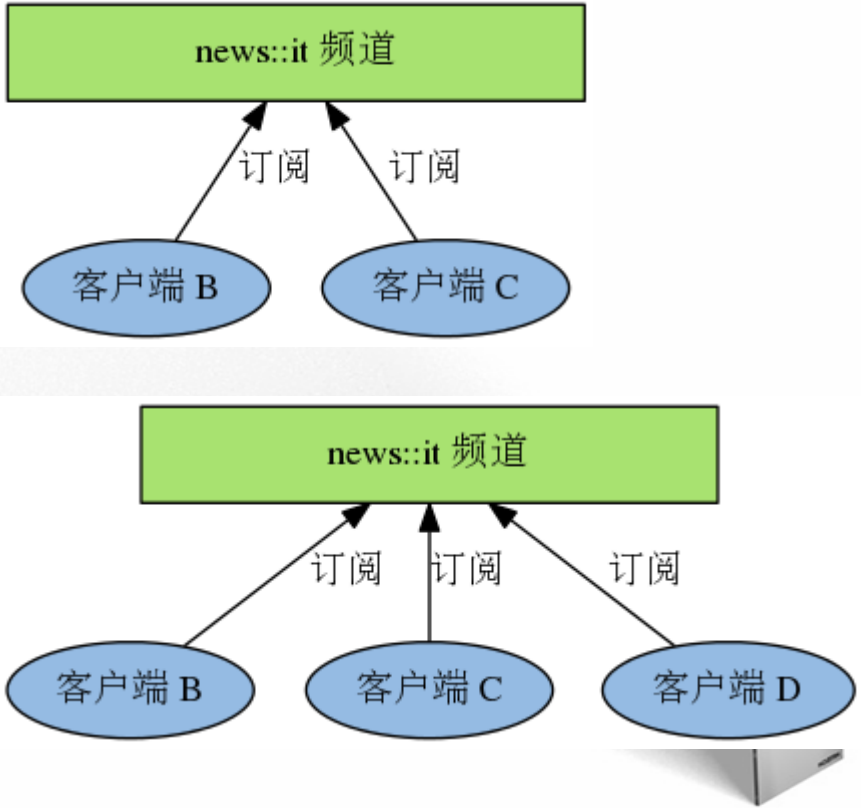
SUBSCRIBE channel [channel ...]

订阅给定的一个或多个频道。

复杂度为 O(N)， N 为被订阅频道的数量。

redis> SUBSCRIBE news::it
Reading messages... (press Ctrl-C to quit)
1) "subscribe"  # 订阅频道时返回的消息
2) "news::it"  # 被订阅的频道
3) (integer) 1  # 客户端目前订阅频道的数量

1) "message"  # 这是从频道接收到的消息
2) "news::it"  # 消息来源的频道
3) "hello"  # 消息内容



# 订阅模式



PSUBSCRIBE pattern [pattern ...]

订阅一个或多个模式，pattern 参数可以包含 glob 风格的匹配符，比如：

- news::* 模式可以匹配 news::bussiness 、news::it 、news::sports::football 等频道；
- news::[ie]t 模式可以匹配 news::it 频道或者 news::et 频道；
- news::?t 模式可以匹配 news::it 、news::et 、news::at 等频道；

诸如此类。

复杂度为 O(N)，N 为被订阅模式的数量。

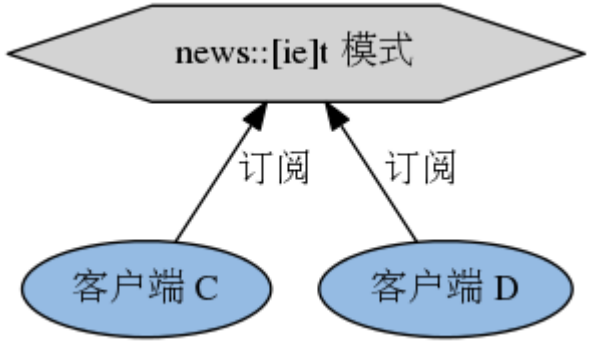
# 订阅模式示例

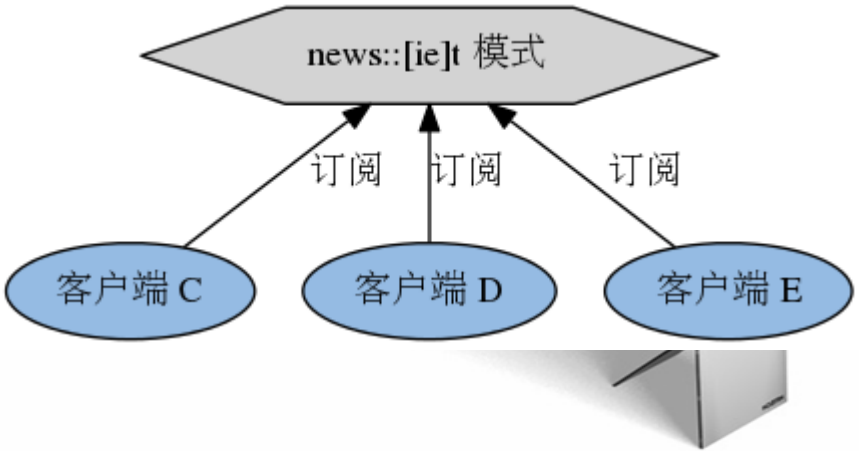


```
redis> PSUBSCRIBE news::[ie]t
Reading messages... (press Ctrl-C to quit)
1) "psubscribe"    # 订阅模式时返回的信息
2) "news::[ie]t"    # 被订阅的模式
3) (integer) 1      # 客户端目前订阅的模式数量

1) "pmessage"    # 这是从模式接收到的消息
2) "news::[ie]t"    # 被匹配的模式
3) "news::it"       # 消息的来源频道(被匹配的频道)
4) "hello"          # 消息正文

1) "pmessage"
2) "news::[ie]t"
3) "news::et"
4) "world"
```



# 退订频道和退订模式



| 命令 | 作用 | 复杂度 |
| --- | --- | --- |
| UNSUBSCRIBE [channel [channel ...]] | 退订指定的频道。<br>如果执行时没有指定任何频道，<br>那么退订已订阅的所有频道。 | O(N)，N 为被退订的频道数量。 |
| PUNSUBSCRIBE [pattern [pattern ...]] | 退订指定的模式。<br>如果执行时没有指定任何模式，<br>那么退订已订阅的所有模式。 | O(M)，M 为服务器中被订阅模式的数量。 |

退订命令的行为在各个客户端的表现都不同，比如 redis-cli 客户端就是通过直接退出客户端来进行退订的，而 Python 和 Ruby 的客户端则需要显示地执行退订命令。

# 退订示例



redis-cli 客户端

```
redis> SUBSCRIBE news.it
Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "news::it"
3) (integer) 1
^C
```

Python 的 Redis 客户端

```
>>> from redis import Redis
>>> client = Redis()
>>> pubsub = client.pubsub()
>>> pubsub.subscribe('news::it')    # 订阅频道
>>> pubsub.channels    # 列出已订阅的频道
set(['news::it'])
>>> pubsub.unsubscribe('news::it')  # 退订频道
>>> pubsub.channels
set([])
```

# 发布消息



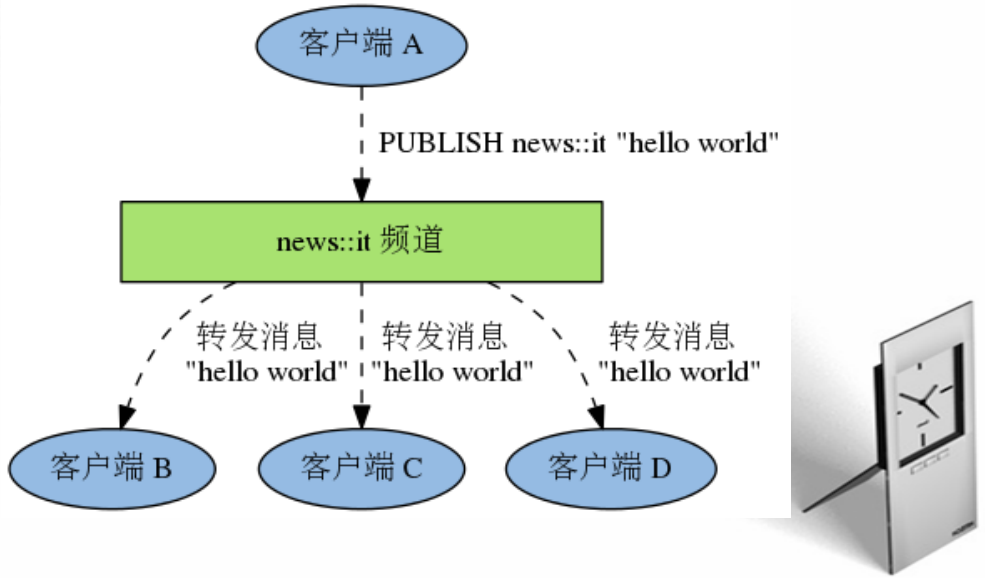
PUBLISH channel message

将消息发送至指定的频道，命令返回接收到消息的 订阅者数量。

复杂度为 O(N)，N 为接收到消息的订阅者数量(包括通过订阅频道来接收消息的订阅者和通过订阅模式来接收消息的订阅者)。

redis> PUBLISH news::it "hello world"
(integer) 2

redis> PUBLISH news::et "hello again"
(integer) 1



# 订阅状态命令

查看被订阅的频道、频道的订阅数量以及模式的订阅数量。

# 查看被订阅的频道



PUBSUB CHANNELS [pattern]

列出目前至少有一个 订阅者的频道。

如果给定了可选的 pattern 参数，那么只列出与模式相匹配的 频道。

复杂度为 O(N)，N 为服务器中被订阅频道的总数量。

```
redis> PUBSUB CHANNELS    # 没有任何频道被订阅
(empty list or set)

redis> PUBSUB CHANNELS    # 有客户端正在订阅 news::et 和 news::it 频道
1) "news::et"
2) "news::it"
```

# 查看频道的订阅者数量



PUBSUB NUMSUB [channel-1 ... channel-N]

返回给定频道的订阅者数量。

复杂度为 O(N)，N 为给定频道的数量。

```
redis> PUBSUB NUMSUB news::it news::et
1) "news::it"     # 有两个客户端正在订阅 news::it 频道
2) "2"
3) "news::et"     # 有一个客户端正在订阅 news::et 频道
4) "1"
```

# 查看被订阅模式的数量



PUBSUB NUMPAT

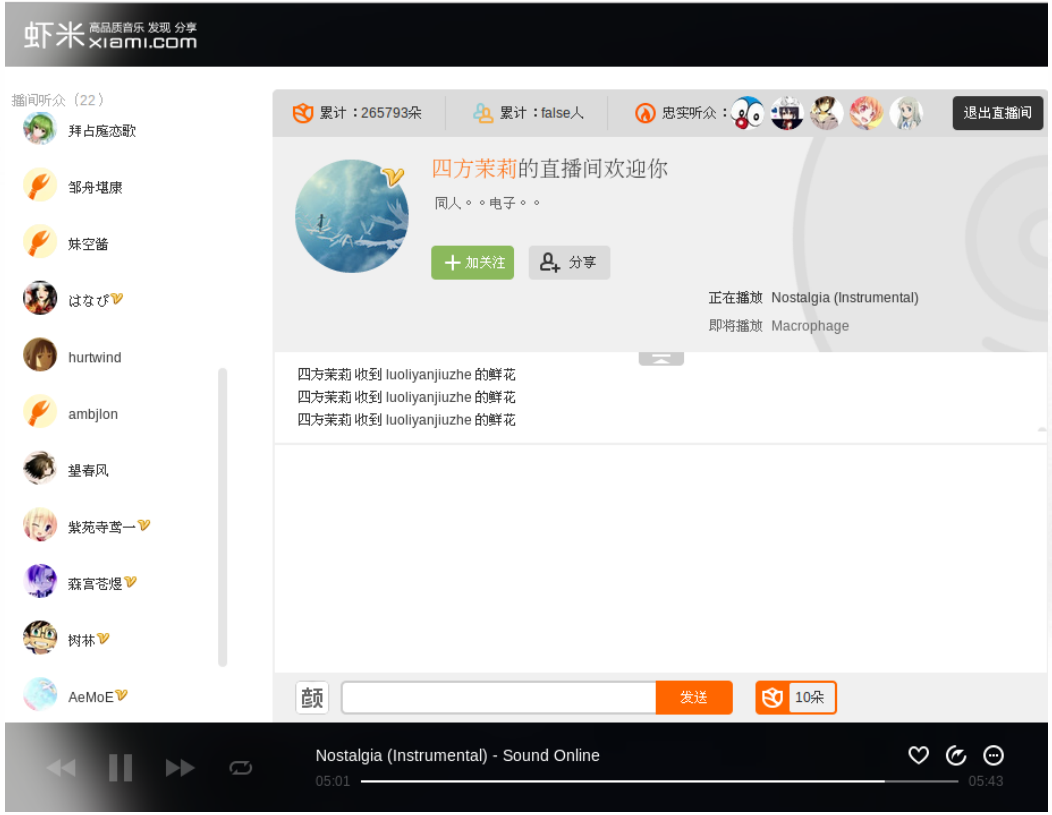
返回服务器目前被订阅的模式数量。

复杂度为 O(1) 。

```
redis> PUBSUB NUMPAT
(integer) 3    # 服务器目前有三个模式被订阅
```

# 示例:在线直播间

小象科技 ChinaHadoop.cn




左图是虾米网站上的一个直播间，每个直播间都有一个播主和多个听众。

播主只要在直播间播放一首歌，那么所有听众都会听到相同的歌曲。

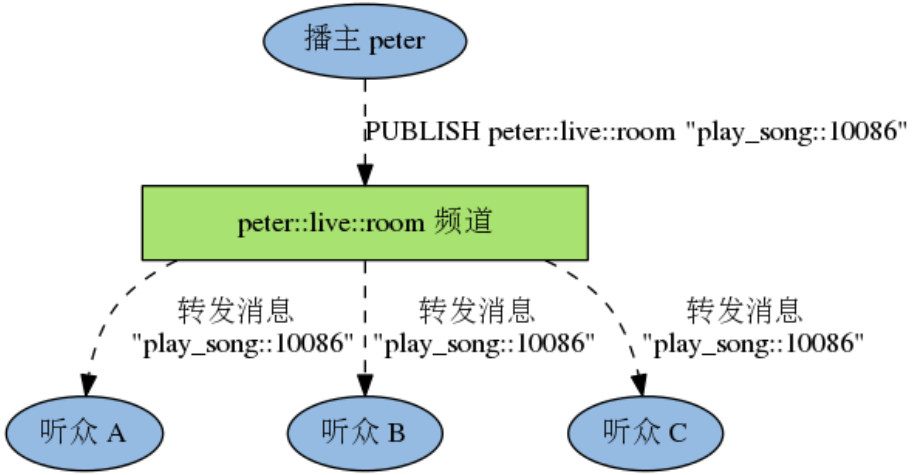
# 使用发布与订阅功能实现直播间



要使用 Redis 实现类似的直播间，我们可以为每个直播间设置一个频道，每个直播间的听众都是频道的订阅者，而播主则通过向频道发送消息来告诉大家要播放哪首歌。

比如说，假设用户 peter 正在进行直播，他的直播间的频道为 peter::live::room，当 peter 要播放 ID 为 10086 的歌曲时，他会执行命令 PUBLISH peter::live::room “play_song::10086”。

而所有听众在接收到这条消息时，只需要分析出里面的指令（play_song）和要播放的歌曲 ID（10086），就可以播放 peter 指定的歌曲了。




# 直播间的 API 及其实现



| API | 作用 | 实现原理 |
| --- | --- | --- |
| LiveRoom(client, name) | 设置直播间的客户端和名字。 | 名字会被用作频道名字。 |
| LiveRoom.play(song_id) | 播主功能，播放指定歌曲。 | 调用 PUBLISH 命令，向频道发送播放指令。 |
| LiveRoom.listen() | 听众功能，收听播主播放的歌曲。 | 调用 SUBSCRIBE 订阅指定频道，并根据频道转发的消息来播放指定的歌曲。 |

这个直播间的实现代码可以在 live_room.py 看到。

# 直播间的播主示例

小象科技 ChinaHadoop.cn

```
>>> from live_room import LiveRoom
>>> live = LiveRoom(client, "peter::live::room")    # 指定直播间
>>> live.play(10086)    # 播放 id 为 10086 的歌曲
playing song, id = 10086
>>> live.play(12345)    # 播放 id 为 12345 的歌曲
playing song, id = 12345
```

# 直播间的听众示例

小象科技 ChinaHadoop.cn

```
>>> from live_room import LiveRoom
>>> live = LiveRoom(client, "peter::live::room")    # 指定直播间
>>> live.listen()    # 开始收听 peter 的直播间
playing song, id = 10086    # 播主播放了 id 为 10086 的歌曲
playing song, id = 12345    # 播主播放了 id 为 12345 的歌曲
```

# 复习

回顾一下本节学习的发布与订阅相关知识。

# 复习(1/2)

小象科技 ChinaHadoop.cn

Redis 的发布与订阅功能由四个不同的角色组成:

- 发布者(publisher):发布消息的客户端。
- 频道(channel):构建在服务器内部,负责接收发布者发送的消息,并将消息转发给频道的订阅者。
- 模式(pattern):构建在服务器内部,负责对频道进行匹配,当被匹配的频道接到消息时,模式也会将消息转发给模式的订阅者。
- 订阅者(subscriber):通过订阅频道或者模式来获取消息的客户端。每个频道或者模式都可以有任意多个订阅者。

# 复习(2/2)

小象科技 ChinaHadoop.cn

| 类型 | 命令 |
| --- | --- |
| 订阅频道和退订频道 | SUBSCRIBE 和 UNSUBSCRIBE |
| 订阅模式和退订模式 | PSUBSCRIBE 和 PUNSUBSCRIBE |
| 发布消息 | PUBLISH |
| 查看订阅状态 | PUBSUB CHANNELS [pattern]<br>列出至少有一个 订阅者的频道，给定 pattern 时只列出和模式匹配的频道。<br>PUBSUB NUMSUB [channel-1 ... channel-N]<br>返回给定频道的订阅者数量。<br>PUBSUB NUMPAT<br>返回服务器目前被订阅的模式数量。 |