

Redis 从入门到精通

黄健宏(huangz)

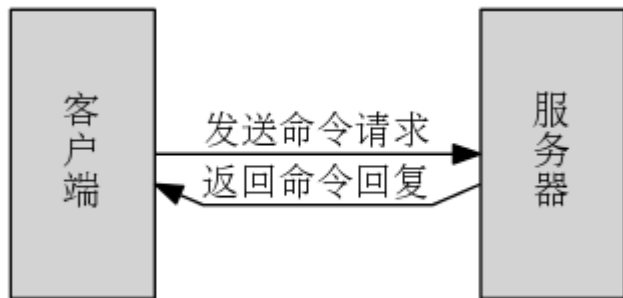


流水线功能

通过减少客户端与服务器之间的通信次数来提高程序的执行效率



在一般情况下，用户每执行一个 Redis 命令，客户端与服务器都需要进行一次通信：客户端会将命令请求发送给服务器，而服务器则会将执行命令所得的结果返回给客户端。



当程序执行一些复杂的操作时，客户端可能需要执行多个命令，并与服务器进行多次通信。



多次通信示例(1/2)

举个例子，假设我们正在构建一个为图书打标签(tag)的网站，这个网站上的每本图书都可以被打上任意多个标签。

并且为了记录哪些标签的图书是最多人阅览的，我们会为每个标签创建一个点击计数器，每当用户浏览网站上的某本书时，程序就会对该书拥有的各个标签的点击计数器执行增一操作。

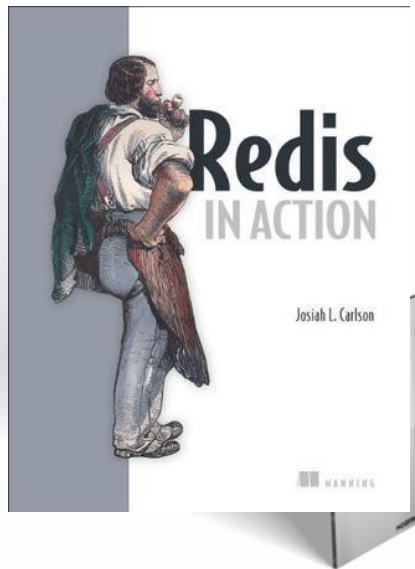
举个例子，对于《Redis in Action》这本书，用户可能会给它打上“计算机”、“编程”、“数据库”和“Redis”这四个标签，每次当《Redis in Action》的页面被访问时，程序就会对这四个标签的点击计数器执行增一操作：

```
INCR tag::计算机::click_counter
```

```
INCR tag::编程::click_counter
```

```
INCR tag::数据库::click_counter
```

```
INCR tag::Redis::click_counter
```



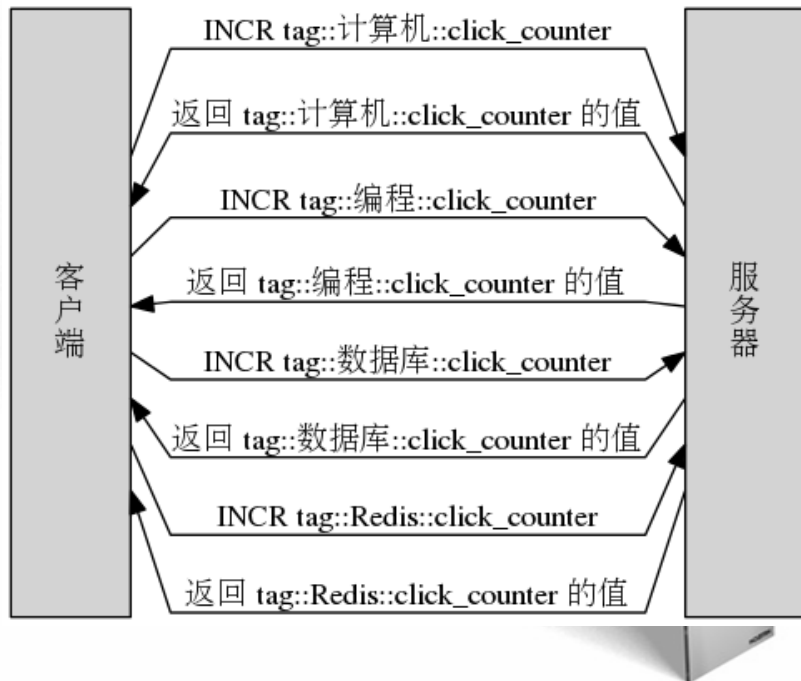
多次通信示例(2/2)

对于这个点击计数程序来说, 客户端要执行的 INCR 命令次数取决于书本的标签数量, 而标签的数量越多, 要执行的 INCR 命令就越多。

右图就展示了《Redis in Action》的页面被查看时, 客户端和服务端之间进行的通信情况。

因为 Redis 服务器的性能非常高, 所以当有一个命令请求抵达服务器之后, 服务器通常很快就会将命令执行完毕, 并向客户端返回命令的执行结果。

在多数情况下,
客户端在执行 Redis 命令时,
大部分等待时间都耗费在了
发送命令请求和接收命令回复上面。



流水线功能

对于前面展示的点击计数器这种需要执行多个命令的程序来说，如果我们能减少程序执行时，客户端与服务器之间的通信次数，就能够有效地提升程序的性能，而 Redis 的流水线功能(pipeline)就是为此而设置的。

Redis 的流水线功能允许客户端一次将多个命令请求发送给服务器，并将被执行的多个命令请求的结果在一个命令回复中全部返回给客户端，使用这个功能可以有效地减少客户端在执行多个命令时需要与服务器进行通信的次数。



流水线应用示例

以前面列出的点击计数器为例，我们可以将多条 INCR 命令都包裹到一个流水线里面执行，使得程序在执行 N 个 INCR 命令时所需的通信次数从 N 次降低为一次。

下图展示了流水线功能是如何将更新《Redis in Action》各个标签的点击计数器所需的通信次数从四次降低为一次的。



Python 客户端中的流水线功能(1/2)

各个 Redis 客户端使用流水线功能的方法都不一样，以下代码展示了在使用 Python 客户端的情况下，开启流水线功能来更新《Redis in Action》各个标签的点击计数器的方法：

```
from redis import Redis
# 创建 Redis 客户端
client = Redis()
# 创建一个流水线对象，包裹四个 INCR 命令
# transaction=False 表示关闭事务功能，只使用流水线功能
# 关于事务方面的详细信息我们稍后就会介绍
pipeline = client.pipeline(transaction=False)
pipeline.incr('tag::计算机::click_counter')
pipeline.incr('tag::编程::click_counter')
pipeline.incr('tag::数据库::click_counter')
pipeline.incr('tag::Redis::click_counter')
# 以流水线形式发送被包裹的四个命令
pipeline.execute()
```



Python 客户端中的流水线功能(2/2)

以流水线方式执行的多个命令的结果会在一个命令回复中被返回， 以下是上面的计数器更新代码的执行结果：

```
# 多个命令的结果会以列表的形式返回  
# 列表的第一个项 10087 是 tag::计算机::click_counter 的值  
# 第二个项 5001 是 tag::编程::click_counter 的值  
# 第三个项 3421 是 tag::数据库::click_counter 的值  
# 第四个项 1001 是 tag::Redis::click_counter 的值  
[10087, 5001, 3421, 1001]
```



复习

回顾一下流水线功能的相关知识。



在一般情况下，用户每执行一个 Redis 命令，客户端和服务端都需要进行一次通信：客户端向服务器发送命令请求，而服务器则会将执行命令所得的命令回复返回给客户端。

在大多数情况下，执行命令时的绝大部分时间都耗费在了发送命令和接收回复的过程上，因此减少客户端与服务端之间的通信次数，可以有效地提高程序的执行效率。

流水线可以将多条命令打包一起发送，并且在一次回复中接收所有被执行命令的结果，使用这个功能可以有效地提升程序在执行多条命令时的效率。

