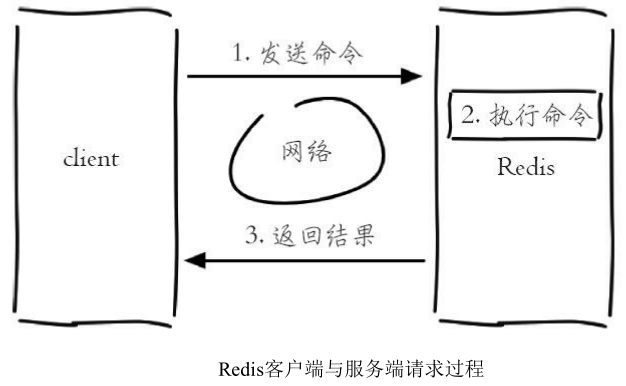
Redis使用**单线程架构**和**I/O多路复用模型**来实现高性能的内存数据库服务。

下图是redis客户端向redis服务发送请求的过程：

1.客户端发送命令

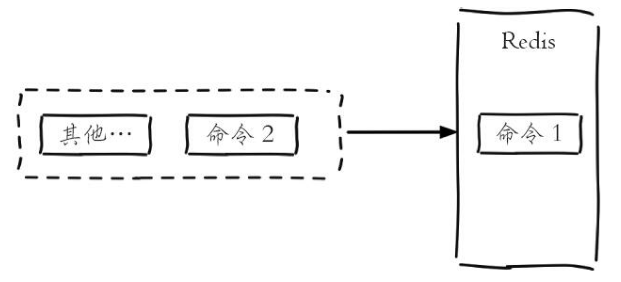
2.redis执行命令

3.返回结果



关键点在第2步：

redis是单线程来处理命令的，一个命令到达redis服务后，并不会立即执行，而是进入一个**队列**中，然后逐个执行。这就保证了线程安全的问题了，执行的顺序是以队列的顺序为主。



redis之所以快的原因：

1.纯内存访问，Redis将所有数据放在内存中，内存的响应时长大约为100纳秒，这是Redis达到每秒万级别访问的重要基础。

2.非阻塞I/O，Redis使用epoll作为I/O多路复用技术的实现，再加上Redis自身的事件处理模型将epoll中的连接、读写、关闭都转换为事件，不在网络I/O上浪费过多的时间.

3.单线程避免了线程切换和竞态产生的消耗。

**缺陷**

单线程会有一个问题。每个命令的执行时间是有要求的。如果某个命令执行过长，会造成其他命令的阻塞，对Redis这种高性能的服务来说是致命的，所以Redis是面向快速执行场景的数据库。