1. 客户端
2. 概述

Redis服务器是典型的一对多服务器程序，通过使用I/O多路复用技术实现的文件事件处理器，Redis服务器使用单线程单进程的方式来处理命令请求，对于每个与服务器连接的客户端都创建了相应的redisClient结构，这个结构保存了客户端当前的状态信息。服务器状态结构的clients属性是一个链表，保存了所有与服务器连接的客户端的状态结构。

1. 客户端属性

客户端状态包含的属性有两类，一类是通用的属性，另一类是特定的属性，如操作数据库用到的db属性等等。

1. 套接字描述符

客户端状态的fd属性记录了客户端正在使用的套接字描述符，有两种类型。一种是伪客户端类型，fd属性为-1，如处理AOF文件或者Lua脚本时产生的伪客户端。另一种是普通客户端，fd属性为大于-1的整数，使用套接字来与服务器通信。使用CLIENT list命令可以列出所有已连接的普通客户端。

1. 名字

默认，连接到服务器的客户端没有名字，通过CLIENT setname命令为客户端设置一个名字，记录在客户端状态的name属性中。

1. 标志

客户端的标志属性flags记录了客户端的角色，以及目前所处的状态，可以是单个值也可以是多个标志的二进制或。

1. 输入缓冲区

客户端状态的输入缓冲区querybuf属性用于保存客户端发送的命令请求，该区域可以动态扩容，最大不能超过1GB，否则服务器会关闭这个客户端。

1. 命令与命令参数

命令请求保存在querybuf属性后，服务器将对该内容进行分析，将命令参数和命令参数的个数分别保存到客户端状态的argv属性和argc属性中。argv属性是一个数组，每个项都是一个字符串对象，其中的argv[0]是要执行的命令，而之后的其他项是参数。

1. 命令的实现函数

服务器获取命令和参数后，将根据命令的值在命令表中查找命令所对应的命令实现函数，该表是一个字典，字典的键是一个SDS结构，保存了命令的名字，字典的值是命令所对应的redisCommand结构，保存了命令的实现函数结构。当找到命令实现函数时，会将客户端状态的cmd属性指向这个结构，然后就会调用该命令实现了。

1. 输出缓冲区

执行命令所得到的命令回复会被保存在客户端状态的输出缓冲区中，每个客户端都有两个输出缓冲区，一个是固定大小的保存短字符，一个是可变大小的保存长字符。

1. 身份验证

客户端状态的authenticated属性用来记录客户端是否通过了身份验证，如果为0表示没有通过，那么除了AUTH验证命令，客户端发送的其他命令都会被服务器拒绝执行。如果为1表示通过了验证，当然这一切都需要服务器开启身份验证功能，否则，服务器不会拒绝客户端的请求。

1. 时间

ctime: 创建客户端的时间；

lastinteraction: 客户端与服务器最后一个互动时间；

1. 客户端的创建与关闭
2. 创建普通客户端

如果客户端是通过网络连接与服务器连接的普通客户端，则在使用connect函数连接到服务器时，服务器会调用连接事件处理器，为客户端创建相应的客户端状态，并添加到服务器状态的clients链表末尾。

1. 关闭普通客户端

可能因为多种原因被关闭，如客户端进程退出或被杀死，发送了不符合协议格式的命令请求，发送的命令请求大小超过了输入缓冲区的限制，命令回复超过了输出缓冲区的限制等等。服务器使用两种模式限制输出缓冲区大小，一种是硬性的，超过了就关闭，另一种是软性限制，如果超过了该大小，但是没有超过硬性限制，且超过的持续时间超过了服务器设定的时长，最后还是要关闭客户端，否则不关闭。

1. Lua脚本的伪客户端

服务器会在初始化时创建负责执行Lua脚本中包含的Redis命令的伪客户端，并关联在服务器状态的lua-client属性中，该客户端会在服务器运行周期一直存在，直到服务器关闭。

1. AOF文件的伪客户端

载入完成之后，会关闭该伪客户端。