

redis是单线程，而单线程还能保证效率的原因是：

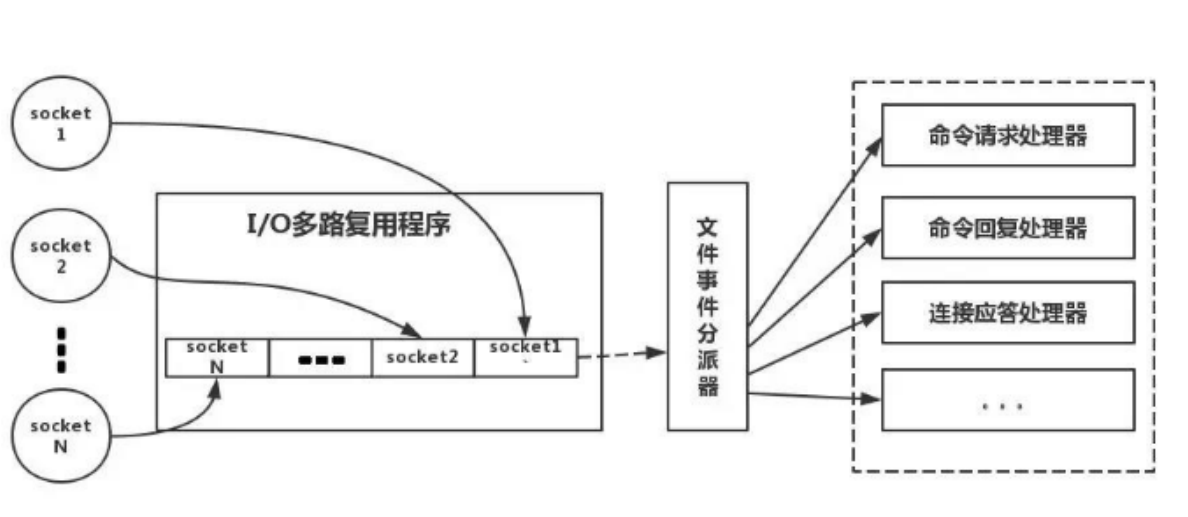
1. 绝大部分请求是纯粹的内存操作（非常快速）
2. 采用单线程，避免了不必要的上下文切换和竞争条件
3. 非阻塞IO

内部实现采用epoll，采用了epoll+自己实现的简单的时间框架。epoll中的读、写、关闭、连接都转化成了时间，然后利用epoll的多路复用特性，绝不在io上浪费一点时间。

这3个条件不是相互独立的，特别是第一条，如果请求都是耗时的，采用单线程吞吐量及性能可想而知了。应该说redis为特殊的场景选择了合适的技术方案。

redis实际上是采用了线程封闭的观念，把任务封闭在一个线程，自然避免了线程安全问题，不过对于需要依赖多个redis操作的复合操作来说，依然需要锁，而且有可能是分布式锁。

下面类比到真实的redis线程模型，如图所示



参照上图，简单来说，就是。我们的redis-client在操作的时候，会产生具有不同事件类型的socket。在服务端，有一段I/O多路复用程序，将其置入队列之中。然后，文件事件分派器，依次去队列中取，转发到不同的事件处理器中。

redis利用队列技术将并发访问变成串行访问，消除了传统数据库串行控制的开销。