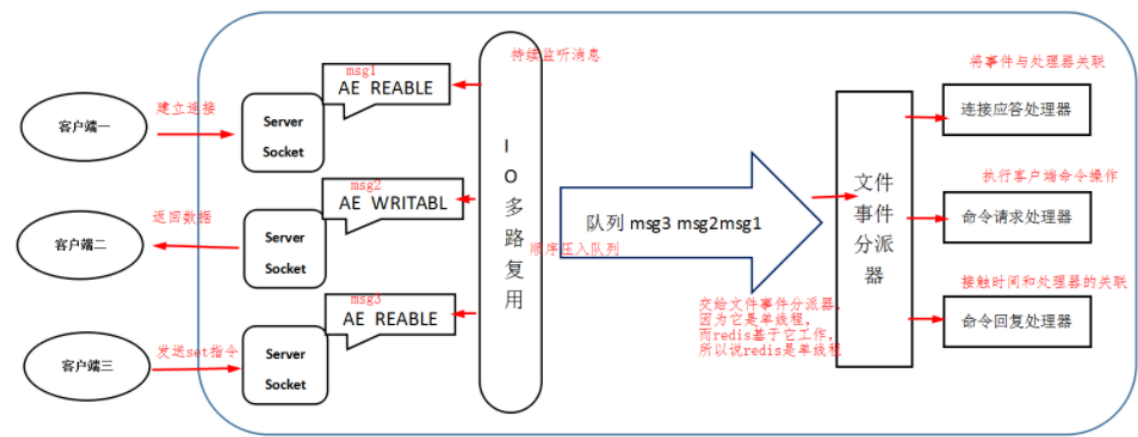


# Redis是多线程还是单线程？

没错，大家所熟知的 Redis 确实是单线程模型，指的是执行 Redis 命令的核心模块是单线程的，而不是整个 Redis 实例就一个线程，Redis 其他模块还有各自模块的线程的。



Redis基于**Reactor模式**开发了网络事件处理器，这个处理器被称为文件事件处理器。它的组成结构为4部分：多个套接字、IO多路复用程序、文件事件分派器、事件处理器。因为文件事件分派器队列的消费是单线程的，所以Redis才叫单线程模型。

## Redis 不仅仅是单线程

一般来说 Redis 的瓶颈并不在 CPU，而在内存和网络。如果要使用 CPU 多核，可以搭建多个 Redis 实例来解决。

其实，Redis 4.0 开始就有多线程的概念了，比如 Redis 通过多线程方式在后台删除对象、以及通过 Redis 模块实现的阻塞命令等。

## Redis如此快，为什么是单线程？

每秒内查询次数10万+

redis是单进程单线程模型，因为redis完全基于内存，cpu不是redis的瓶颈，redis的瓶颈可能是机器内存大小或者网络带宽。

既然单线程容易实现，并且cpu不会成为瓶颈——那就采用单线程实现，可以避免多线程的很多麻烦。

为什么单线程如此快？	redis完全基于内存，绝大多数请求都是纯粹的内存操作，数据存在内存中类似于hashmap，hashmap的优势是查找和操作的时间复杂度都为O(1)。
	数据结构简单，数据操作简单。
	利用单线程，避免了不必要的上下文切换和竞争条件，不存在多线程的cpu切换，不存在锁的添加释放以及死锁问题的出现导致的性能消耗。
	使用多路复用IO模型，非阻塞IO。

redis有部分数据存在硬盘上，这样可以保持数据的持久性，上电后自动加载。

## Redis持久化

---

redis为保证效率，数据缓存在内存中，但是会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件中以保证数据的持久化。

有两种持久化策略：redis默认使用RDB的持久化方式。

**【1】RDB：**快照形式直接将内存中的数据保存到一个dump.rdb二进制文件中，定时保存数据策略。

工作原理：redis需要持久化时会开辟一个子线程，子线程会将数据写到磁盘上的二进制临时文件中，完成文件后才去替换原有的rdb文件（写入时复制copy-on-write）

优点：适合用于备份，用于灾难恢复，随时将数据集还原到不同的版本。

缺点：可能会在服务器故障时丢失数据。

**【2】AOF：**把对redis服务器进行修改的命令都存到一个文件中，命令集合策略。

工作原理：每一个修改命令都通过write函数追加到appendonly.aof中，**redis重启时会读取AOF文件进行重放，恢复到redis关闭前的最后时刻。**

优点：可以设置不同的fsync策略，默认是每秒钟同步一次，也可以设置为always，只要有修改就会同步。

缺点：相同的数据集，AOF的文件体积通常要大于RDB文件的体积，AOF将Redis执行的每一条命令追加到磁盘中，处理巨大的写入会降低Redis的性能。根据所使用的同步策略，AOF的速度可能会慢于RDB。

两种策略如何选择：

**更关心数据，可以接受几分钟的数据丢失——RDB**

**注重完整性——AOF**

数据库备份和灾难恢复：定时生成RDB快照非常便于进行数据库备份，并且RDB恢复数据集的速度也要比AOF恢复的速度快。

redis支持同时开启RDB和AOF。系统重启后，redis会优先使用AOF来恢复数据，这样丢失的数据会最少。

<https://baijiahao.baidu.com/s?id=1660009541007805174&wfr=spider&for=pc>