**1.本身性能很高，每秒处理10万次的读写。**

**2代码更清晰，处理逻辑更简单**

**3.不用去考虑各种锁的问题，不存在加锁释放锁操作，没有因为可能出现死锁而导致的性能消耗（下面详细说）**

**4.不存在多进程或者多线程导致的切换而消耗CPU**

**5.事件处理类型是IO多路复用（同一个线程内同时处理多个IO请求的目的）**

**6.将并行事件，串行化发生。**

还要考虑Redis操作的是内存中的数据结构。如果在多线程中操作，那就需要为这些对象加锁。所以使用多线程可以提高性能，但是每个线程的效率严重下降了，**而且程序的逻辑严重复杂化。Redis的数据结构并不全是简单的Key-Value，**

还有list，hash等复杂的结构，这些结构有可能会进行很细粒度的操作，比如在很长的列表后面添加一个元素，在hash当中添加或者删除一个对象，这些操作还可以合成MULTI/EXEC的组（[MULTI](http://redis.cn/commands/multi.html) 开启事务，总是返回OK，[EXEC](http://redis.cn/commands/exec.html) 提交事务，[DISCARD](http://redis.cn/commands/discard.html)放弃事务（放弃提交执行），[WATCH](http://redis.cn/commands/watch.html)监控

QUEUED是将命令加入执行的队列）。**这样一个操作中可能就需要加非常多的锁，导致的结果是同步开销大大增加。**