并行：服务器可以同时执行几个事情

并发：几个计算单元处理来自多个IO流的流请求

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a sign

Description automatically generated

A close up of a sign

Description automatically generated

A white sign with black text

Description automatically generated

A close up of a sign

Description automatically generated



先问范围

Key k1\*

**如何通过Redis实现分布式锁**

1. **SETNX key value: 如果key不存在，则创建并赋值**

**时间复杂度O(1)**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**如何使用Redis做异步队列**

* 使用List作为队列，RPUSH生产消息，LPOP消费消息
* 缺点：不会等待队列中有值采取消费的
* 弥补：可以在应用层中编写Sleep逻辑去调用LPOP重试。
* 或者使用BLPOP key [key …] timeout: 阻塞知道队列中有消息或超时。(比如blpop testlist 30)
* 缺点：只能供一个消费者消费
* Pub/sub：主题订阅者模式实现一对多的消费队列
* 发送者（pub）发送消息，订阅者（sub）接受消息
* 订阅者可以订阅任意数量的频道
* Subscribe myTopic
* Publish myTopic “text”
* 缺点：消息发布是无状态的，无法保证可达 （如果要解决的话就要使用专业的消息队列 如Kafka）

**Redis 如何做持久化**

<https://segmentfault.com/a/1190000002906345>

A screenshot of a cell phone

Description automatically generated

A close up of a piece of paper

Description automatically generated

**最后再看 考的概率不大**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**使用bgrewriteaof来触发aof**

**Redis和mysql的事务区别**

**A screenshot of a social media post

Description automatically generated**