1.redis的介绍

Redis 是完全开源免费的高性能的内存key-value数据库，可用作数据库、缓存和消息中间件。

支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。

不仅支持简单的key-value（string）类型的数据，同时还提供list，set，zset，hash等数据结构的存储。

支持数据的备份，即master-slave模式的数据备份。

性能极高 – 读的速度是110000次/s, 写的速度是81000次/s 。

Redis的所有操作都是原子性的。多个操作也支持事务。

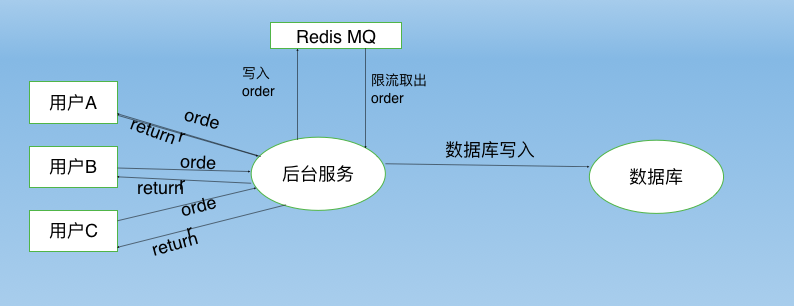
2. 阐述redis 和传统数据的不同？

与传统数据库不同的是 Redis 的数据是存在内存中的，所以存写速度非常快，因此 Redis 被广泛应用于缓存方向。Redis为分布式缓存，在多实例的情况下，各实例共用一份缓存数据，缓存具有一致性。

+ redis优点

3.为什么使用redis？

秒杀活动中用户大量的提交创建订单的请求，订单的写入需要操作数据库，而对数据库的写入数据库的操作受硬件的影响很大，硬件设备性能较低时只能支持每秒几十次的操作，大量订单写入数据库很可能会导致数据库的崩溃（缓存雪崩），因此我们需要限制服务发起数据库操作的请求的速度

解决方案：消息队列削峰

自行用语言描述这个图。

4. 阐述 电子商城缓存雪崩原因？

用户操作 +

1）缓存中大量的key设置相同的过期时间，在某一时刻同时失效

2）缓存数据库因某种原因（缓存穿透， 缓存击穿等）发生宕机或断网，此时对数据库造成的压力不可预计，所造成的雪崩较为致命

5. 阐述Redis功能？

数据缓冲

(1)Redis是将数据存到内存中，而Mysql存在磁盘中，从内存中读取数据更快

(2)访问率高但更新频率较低的数据，可以添加缓存，减少Mysql压力

(3)查询时先从Redis中查，没有再去Mysql中查，将查到的数据放缓存中一份。

数据持久化

redis中内部提供了良好的持久化的策略，保证内存中的数据不丢失，这样redis服务器重启后，依然可以获取其中的数据。

分布式锁

控制分布式系统或不同系统之间共同访问共享资源的一种锁实现，如果不同的系统或同一个系统的不同主机之间共享了某个资源时，往往需要互斥来防止彼此干扰来保证一致性。

消息队列

6. 具体描述用redis解决电子商城库存并发控制的问题？

方案一：

基于消息队列和分布式锁的方案:

1. 基于发布/订阅的消息队列模式，将各个客户端设置为订阅状态，库存量的变化将实时推送给相应客户端。将高并发的用户请求写入消息队列， 通过异步处理提高系统处理的性能，进行流量平峰。

2. Redis实现分布式锁。用户通过前端发起请求，每个请求执行前对Redis中的库存加锁，判读库存量是否充足，若充足则扣去相应库存量，进行解锁；若用户请求对库存量加锁失败，则后端拒绝该请求。

3. 预扣库存+Redis集群（主从复制）

将mysql中的库存量提前提取一部分进入Redis缓存，设置预警值，每当Redis中的库存量接近预警值，即从mysql中再提库存进入redis

使用Redis集群（主从复制）的架构，防止主redis中的数据意外丢失。

方案二：

* 1. 获取商品的锁 -》 若未获取成功，判断是否超时，超时则释放锁，再尝试获取锁
  2. 获取商品锁成功，则去判断redis中是否有库存。
  3. redis中若是没有，则去商品服务扣去预扣库存。如果实际库存为0则释放锁，订单失败。如果不为0，则设置redis。
  4. 如果redis有，则去判断是否库存小于阈值。小于则跳转3。
  5. 判断redis中商品a的数量是否小于要买的数量。
  6. 数量充足，则购买，然后释放锁，购买成功。
  7. 数量不充足则根据差值去商品服务上查询实际库存值，然后得出订单结果。

7. 如何防止两个线程请求同时访问redis造成的超卖问题（1件商品，两个请求）？

为每一个商品设置一个在redis的key,如果key存在，则线程正在访问redis,如果key不存在，则可以进行访问。