

Redis缓存雪崩面试题

题目标签

学习时长：20分钟

题目难度：中等

知识点标签：缓存、二级缓存、限流

题目描述

什么是缓存雪崩，如何解决缓存击穿

题目解决

缓存雪崩

缓存层承载着大量的请求，有效保护了存储层。但是如果由于缓存大量失效或者缓存整体不能提供服务，导致大量的请求到达存储层，会使存储层负载增加，这就是缓存雪崩的场景。

解决缓存雪崩

1.保持缓存层的高可用性

使用Redis 哨兵模式或者Redis 集群部署方式，即便个别Redis 节点下线，整个缓存层依然可以使用。除此之外，还可以在多个机房部署 Redis，这样即便是机房死机，依然可以实现缓存层的高可用。

2.限流降级组件

无论是缓存层还是存储层都会有出错的概率，可以将它们视为资源。作为并发量较大的分布式系统，假如有一个资源不可用，可能会造成所有线程在获取这个资源时异常，造成整个系统不可用。降级在高并发系统中是非常正常的，比如推荐服务中，如果个性化推荐服务不可用，可以降级补充热点数据，不至于造成整个推荐服务不可用。常见的限流降级组件如 Hystrix、Sentinel 等。

3.缓存不过期

Redis 中保存的 key 永不失效，这样就不会出现大量缓存同时失效的问题，但是随之而来的就是Redis 需要更多的存储空间。

4.优化缓存过期时间

设计缓存时，为每一个 key 选择合适的过期时间，避免大量的 key 在同一时刻同时失效，造成缓存雪崩。

5.使用互斥锁重建缓存

在高并发场景下，为了避免大量的请求同时到达存储层查询数据、重建缓存，可以使用互斥锁控制，如根据 key 去缓存层查询数据，当缓存层为命中时，对 key 加锁，然后从存储层查询数据，将数据写入缓存层，最后释放锁。若其他线程发现获取锁失败，则让线程休眠一段时间后重试。对于锁的类型，如果是在单机环境下可以使用 Java 并发包下的 Lock，如果是在分布式环境下，可以使用分布式锁（Redis 中的 SETNX 方法）

分布式环境下使用Redis 分布式锁实现缓存重建，优点是设计思路简单，对数据一致性有保障；缺点是代码复杂度增加，有可能会造成用户等待。假设在高并发下，缓存重建期间 key 是锁着的，如果当前并发 1000 个请求，其中 999 个都在阻塞，会导致 999 个用户请求阻塞而等待。

6.异步重建缓存

在这种方案下构建缓存采取异步策略，会从线程池中获取线程来异步构建缓存，从而不会让所有的请求直接到达存储层，该方案中每个Redis key 维护逻辑超时时间，当逻辑超时时间小于当前时间时，则说明当前缓存已经失效，应当进行缓存更新，否则说明当前缓存未失效，直接返回缓存中的 value 值。如在Redis 中将 key 的过期时间设置为 60 min，在对应的 value 中设置逻辑过期时间为 30 min。这样当 key 到了 30 min 的逻辑过期时间，就可以异步更新这个 key 的缓存，但是在更新缓存的这段时间内，旧的缓存依然可用。这种异步重建缓存的方式可以有效避免大量的 key 同时失效。

练习题

1.基于Spring boot 模拟并解决缓存穿透