

Redis并发竞争key的解决方案详解

题目标签

学习时长：20分钟

题目难度：中等

知识点标签：高并发、并发竞争、分布式锁、消息队列

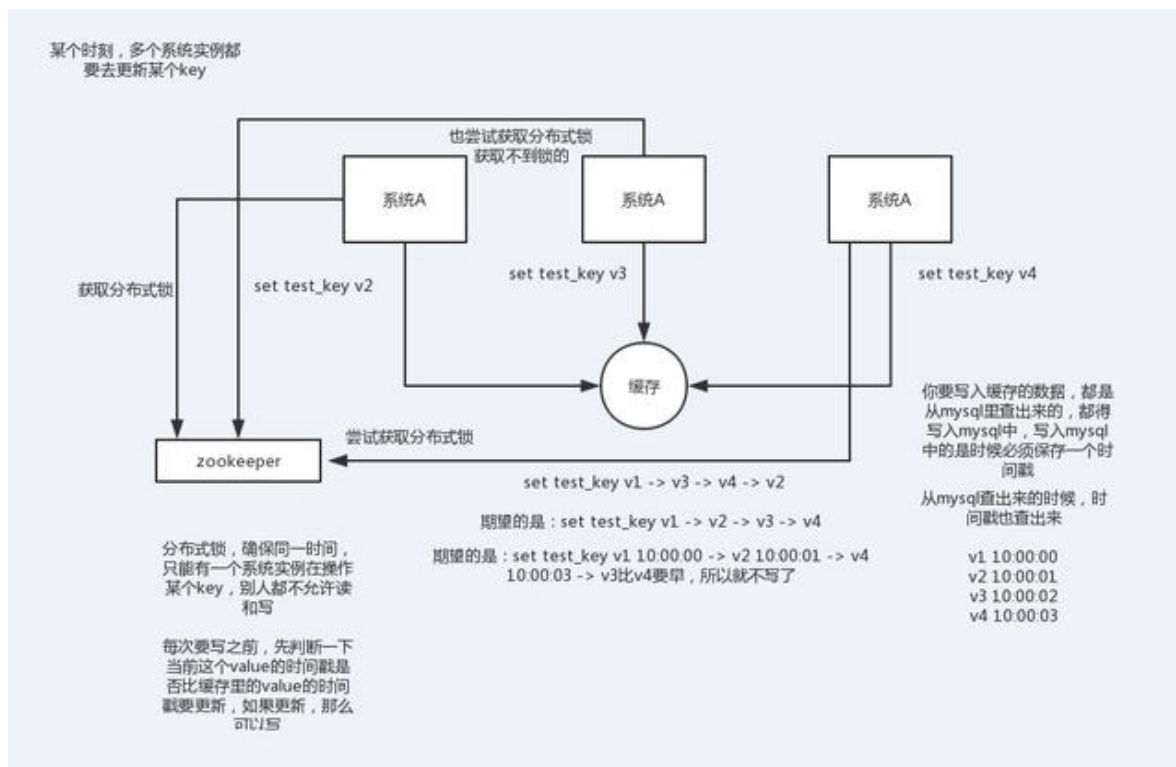
题目描述

Redis并发竞争key的解决方案详解？

1.面试题分析

Redis缓存的高性能有目共睹，应用的场景也是非常广泛，但是在高并发的场景下，也会出现问题，数据一致性方案，如何解决Redis缓存雪崩、缓存穿透、缓存并发，以及今天要谈到的Redis并发竞争问题，这里的并发指的是多个redis的client同时set key引起的并发问题。

比如：多客户端同时并发写一个key，一个key的值是1，本来按顺序修改为2,3,4，最后是4，但是由于并发设置的原因，最后顺序变成了4,3,2，最后变成的key值成了2。



2.解决Redis的并发竞争key问题

第一种方案：分布式锁

1.整体技术方案

这种情况，主要是准备一个分布式锁，大家去抢锁，抢到锁就做set操作。

2.为什么是分布式锁

因为传统的加锁的做法（如java的synchronized和Lock）这里没用，只适合单点。因为这是分布式环境，需要的是分布式锁。

当然，分布式锁可以基于很多种方式实现，比如zookeeper、redis等，不管哪种方式实现，基本原理是不变的：用一个状态值表示锁，对锁的占用和释放通过状态值来标识。

3.分布式锁的要求

- 互斥性：在任意一个时刻，只有一个客户端持有锁。
- 无死锁：即便持有锁的客户端崩溃或者其他意外事件，锁仍然可以被获取。
- 容错：只要大部分Redis节点都活着，客户端就可以获取和释放锁

4.分布式锁的实现方式

- 数据库
- Memcached（add命令）
- redis（setnx命令）
- Zookeeper（临时节点）

第二种方案：利用消息队列

在并发量过大的情况下,可以通过消息中间件进行处理,把并行读写进行串行化。

把Redis.set操作放在队列中使其串行化,必须的一个一个执行。

这种方式在一些高并发的场景中算是一种通用的解决方案。