

## redis架构

- 单主：只有一台redis
- 主从：一主多从，主redis负责写，从redis负责读
- 哨兵：在主从基础上，增加哨兵机制，来选择主redis，但是还是主redis负责写，从redis负责读
- 集群：多主，16384个hash slot确定往哪台主redis上写

## Redis分布式锁的实现

- 1.setnx + expire
  - 如果key不存在 set成功返回1，失败返回0
  - expire防止锁的长期持有或死锁
- 2.比较删除解锁
  - 删除前判断是否是加锁的客户端
- 3.Lua实现原子性
- 4.zookeeper实现分布式锁

## Redis Sentinel选择机制

1. 过滤掉主观下线的节点
2. 选择优先级最高的节点，如果有则返回没有就继续选择
3. 选择出复制偏移量最大的节点，因为复制便宜量越大则数据复制的越完整，如果有就返回了，没有就继续
4. 选择run\_id最小的节点

## 数据存储的核心算法

- 1.hash算法

$\text{hash}(\text{key}) \% N$  取模分配key到各个master中。一旦一个master宕机，那么大量的数据就需要重新计算。导致大量的请求无法拿到数据，大量的流量就会冲击数据库（一般不用）
- 2.一致性hash算法（memcached）

$\text{hash}(\text{key})$  后的值落在圆环上，顺时针寻找距离最近的一个节点

  - 缓存热点问题

可能集中在某个hash区间内的值特别多，那么会导致大量的数据都涌入同一个master内，造成master的热点问题，性能出现瓶颈
  - 解决方案：圆环上设置虚拟节点
- 3.redis cluster（hash slot / slot）算法

$\text{CRC16}(\text{key}) \% 16384$  取模后分配到hash slot（16384）

## Redis Cluster注意问题

- Redis-Cluster主从节点不要在同一台机器部署，一对主从所在机器：不跨机房、要跨机架、可以在一个机柜
- Redis-Cluster误判节点fail进行切换

## 布隆过滤器 (BloomFilter)

Bitmap : 位图是支持按 bit 位来存储信息，可以用来实现 布隆过滤器 (BloomFilter) ；

### Redis事务：

最后一个功能是事务，但 Redis 提供的不是严格的事务，Redis 只保证串行执行命令，并且能保证全部执行，但是执行命令失败时并不会回滚，而是会继续执行下去。