

分布式 Redis 缓存

Redis 与 Ehcache 的缓存的区别

Redis 的缓存属于分布式的缓存，数据存放在服务端中内存中，也会持久化到硬盘上。

Ehcache 属于本地 jvm 内置的缓存，可以支持集群数据共享，但是内部之间支持数据的同步，效率非常低。

你有用过 Redis 吗？Redis 有那些应用场景

Token 令牌的生成

短信验证码 Code

缓存热点数据

实现网页计数器

分布式锁

延迟操作（订单超时时间设计）

Redis 线程为什么效率这么高

从内存、IO 多路复用原则

1. 数据存放在内存中
2. IO 多路复用原则，使用一个线程维护多个不同 Redis 客户端请求的操作
3. 基于跳跃表数据增删改效率非常高

Redis 线程安全吗？

Redis 的线程是安全的

Redis 线程安全为什么效率还非常高

1. Redis 的底层采用 Nio 中的多路 IO 复用的机制，能够非常好的支持这样的并发，从而保证线程安全问题；
2. Redis 单线程，也就是底层采用一个线程维护多个不同的客户端 io 操作。
但是 Nio 在不同的操作系统上实现的方式有所不同，在我们 windows 操作系统使用 select

实现轮训时间复杂度是为 $O(n)$ ，而且还存在空轮训的情况，效率非常低，其次是默认对我们轮训的数据有一定限制，所以支持上万的 tcp 连接是非常难。

3. 所以在 linux 操作系统采用 `epoll` 实现事件驱动回调，不会存在空轮训的情况，只对活跃的 `socket` 连接实现主动回调这样在性能上有大大的提升，所以时间复杂度是为 $O(1)$
4. 注意：windows 操作系统是没有 `epoll`，只有 linux 系统才有 `epoll`
5. 所以为什么 `nginx`、`redis` 都能够非常高支持高并发，最终都是 linux 中的 IO 多路复用机制 `epoll`

Redis 分为分为多少个库？为什么要分成 16 个库？

1. Redis 默认的情况下分为 16 个库
2. 为什么要分成 16 个库，在单个库中不允许存在重复的 `key`。
3. 有可能不同的业务逻辑如果连接同一个 Redis 的相同的 `db`，有可能 `key` 是相同，所以可以采用 Redis 数据库实现区分。

Redis 的数据类型有那些

String
List
Hash
Zset
Set

Redis 中如何存放对象

1. 基于 JSON 序列化存放 优点：阅读性强、可以跨语言 缺点：明文的不安全
2. 基于 String 二进制直接存放我们的对象 不可以跨语言，阅读性差、比较安全。

二进制与 json 存放对象的区别

1. 基于 JSON 序列化存放 优点：阅读性强、可以跨语言 缺点：明文的不安全

2. 基于 String 二进制直接存放我们的对象 不可以跨语言，阅读性差、比较安全。

全量同步与增量同步实现的区别

- 1.全量同步：每天定时避开高峰期，将所有数据全部实现备份同步，优点：数据可以不用丢失，效率高、但是可能会产生数据同步的延迟。
- 2.增量同步：对行为的操作实现对数据的同步，数据同步延迟的概率比较多，因为比较频繁效率效率低。

Redis 内存满了，如何处理？

Redis 内存满了，可以采用扩容我们的服务器内存，但是这种方案缺点：治标不治本，扩容硬件设施成本也比较高，这种形式不推荐。
官方推荐采用 Redis 内存淘汰策略，Redis 如果内存满的情况下，删除经常不被使用的 key。

Redis 宕机了？数据会丢失吗？

不会丢失，因为 Redis 有持久化的机制,采用 RDB 和 AOF 持久化方案

Redis 持久化中有那些方案

RBD 和 AOF

RDB 与 AOF 同步存在那些区别

RBD 采用定时全量同步，将所有 Redis 缓存的数据存入到硬盘中；
AOF 采用行为增量同步，以日志文件（Redis 的执行命令）的形式存放到硬盘追加中。
RDB 恢复的速度比 AOF 高。

Redis 事务支持回滚吗？

Redis 中存在事务，但是没有回滚只有取消事务。

为什么 Redis 没有回滚事务？

MySQL 中的事务 如果在 mysql 对某行数据开启了事务情况下，采用行锁的机制，其他线程客户端不能够对该行数据做写的写的操作。

但是在 Redis 中的事务，如果 Redis 对某个 key 开启了事务，其他线程可以对 key 做写的错误。

Redis 中 SetnX 与 Set 命令的区别

Setnx 可以返回该 key 是否存在 存在返回 0 不存在返回 1

如果该 key 存在的情况下，是不能做修改的。

Set 每次直接覆盖该 key 对应的 value

Redis 主从复制整个原理过程

- 1.如果使用单台 Redis 的情况下，如果该 Redis 宕机后可能会导致数据的丢失，可以采用 Redis 的主从复制保持数据备份。
- 2.多个 Redis 中分为主从关系，写的请求转发到主的节点，主节将数据写完之后在同步给每个从节点。从节点主要负责读取。

所以 Redis 的主从可以实现

- 1.Redis 的数据的备份
- 2.简单化集群模式
- 3.读写分离过程

传统一主多从复制存在那些问题

如果主宕机之后，需要手动的重启我们主的服务器或者是手动实现服务器节点的选举。从节点如果越多的情况下，主节点复制给从节点的过程压力比较大，可以采用二叉树的形式实现一主多从复制过程。

哨兵集群架构设计原理

哨兵监控我们的主 Redis，如果主 Redis 宕机之后从新在其他的从节点选举一个新的主节点，所以哨兵主要帮助我们实现自动化选举的过程。

Redis 缓存穿透概念与解决方案

概念：频繁查询不存在的 key，导致会查询数据库，对数据库访问压力比较大。

解决方案：

1. 记录空查询 key，设置简单的过期时间
2. 采用布隆过滤器
3. 对我们 api 接口实现限流、黑名单、白名单的机制。

Redis 缓存雪崩概念与解决方案

概念：在同是一时间大量的 key，同时失效，同时查询我们的数据库，对数据库访问压力比较大。

解决方案：

1. 对我们 Redis 的 key 过期时间都是随机的
2. 或者 Rediskey 不过期

Redis 缓存击穿的解决方案

在高并发的情况下，多个请求同时访问一个过期的 key，多个请求会同时查询我们的数据库，对数据库查询的压力非常大。

解决方案：

1. 基于分布式锁实现
2. 设置软过期时间

缓存穿透： 查询不存在的

雪崩：多个 key 同时失效

击穿：单个 key 失效

Redis 集群有那些方式

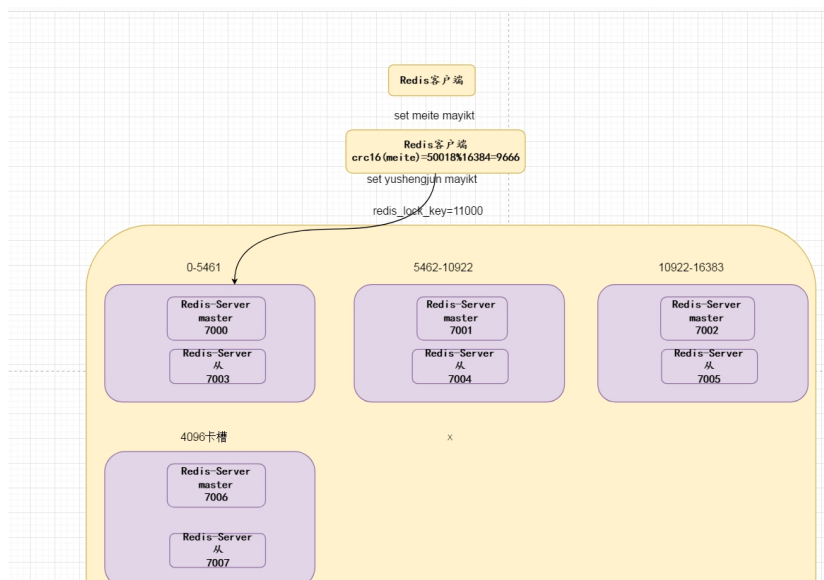
三种集群方案

1. 直接主从复制集群模式 缺陷：如果主宕机的话，需要人工修改选举的配置
2. 哨兵机制模式 优势自动帮助我们实现选举，不需要人工的修改配置 缺点：数据会冗

余

3. Redis 的 Cluster 集群模式 动态实现扩容和缩容 而且保证每个节点的数据不冗余存放

RedisCluster 分片集群实现原理



余胜军 java架构面试宝典 ms.mayikt.com

原理：

在我们的 Redis 集群模式中分为 16384 个卡槽，类似于数据库的中分表模式。

当我们在写入一个 key 的时候，会对该 key 计算 crc16 算法得住一个数字 16384=卡槽的位置。每个卡槽对应具体节点存放的位置，这样的话就可以将我们的数据可以均摊的存放各个节点。

每个卡槽是否可以存放多个不同的 key？必须可以的

Mayikt/16384=52

Yushengjun%16384=52

类似推出 数据库表结构

该模式集群中最多只能支持 Redis 主的节点 16384 个，也技术每个节点对应一个卡槽位置。

RedisCluster 如何快速扩容和缩容

重新分配每个卡槽对应的 Redis 的存放节点。