

1 Redis 相比 memcached 有哪些优势？

- (1) memcached 所有的值均是简单的字符串，redis 作为其替代者，支持更为丰富数据类型
- (2) redis 的速度比 memcached 快很多
- (3) redis 可以持久化其数据

2 Redis 支持哪几种数据类型？

String、List、Set、Sorted Set、hashes

3 Redis 集群方案应该怎么做？都有哪些方案？

1.twemproxy，大概概念是，它类似于一个代理方式，使用方法和普通 redis 无任何区别，设置好它下属的多个 redis 实例后，使用时在本需要连接 redis 的地方改为连接 twemproxy，它会以一个代理的身份接收请求并使用一致性 hash 算法，将请求转接到具体 redis，将结果再返回 twemproxy。使用方式简便(相对 redis 只需修改连接端口)，对旧项目扩展的首选。问题：twemproxy 自身单端口实例的压力，使用一致性 hash 后，对 redis 节点数量改变时候的计算值的改变，数据无法自动移动到新的节点。

4 Redis 回收使用的是什么算法？

LRU 算法

5 为什么要做 Redis 分区？



更多关注 Java 大后端公众号

分区可以让 Redis 管理更大的内存，Redis 将可以使用所有机器的内存。如果没有分区，你最多只能使用一台机器的内存。分区使 Redis 的计算能力通过简单地增加计算机得到成倍提升，Redis 的网络带宽也会随着计算机和网卡的增加而成倍增长。

6 Redis 的内存占用情况怎么样？

给你举个例子：100 万个键值对（键是 0 到 999999 值是字符串“hello world”）在我的 32 位的 Mac 笔记本上 用了 100MB。同样的数据放到一个 key 里只需要 16MB，这是因为键值有一个很大的开销。在 Memcached 上执行也是类似的结果，但是相对 Redis 的开销要小一点点，因为 Redis 会记录类型信息引用计数等等。

7 Memcached 服务特点及工作原理是什么？

- a、完全基于内存缓存的
- b、节点之间相互独立
- c、C/S 模式架构，C 语言编写，总共 2000 行代码。
- d、异步 I/O 模型，使用 libevent 作为事件通知机制。
- e、被缓存的数据以 key/value 键值对形式存在的。
- f、全部数据存放于内存中，无持久性存储的设计，重启服务器，内存里的数据会丢失。
- g、当内存中缓存的数据容量达到启动时设定的内存值时，就自动使用 LRU 算法删除过期的缓存数据。
- h、可以对存储的数据设置过期时间，这样过期后的数据自动被清除，服务本身不会监控过期，而是在访问的时候查看 key 的时间戳，判断是否过期。
- j、memcache 会对设定的内存进行分块，再把块分组，然后再提供服务。



更多关注 Java 大后端公众号

8 如何实现集群中的 session 共享存储?

Session 是运行在一台服务器上的，所有的访问都会到达我们的唯一服务器上，这样我们可以根据客户端传来的 sessionID，来获取 session，或在对应 Session 不存在的情况下（session 生命周期到了/用户第一次登录），创建一个新的 Session；但是，如果我们在集群环境下，假设我们有两台服务器 A, B，用户的请求会由 Nginx 服务器进行转发（别的方案也是同理），用户登录时，Nginx 将请求转发至服务器 A 上，A 创建了新的 session，并将 SessionID 返回给客户端，用户在浏览其他页面时，客户端验证登录状态，Nginx 将请求转发至服务器 B，由于 B 上并没有对应客户端发来 sessionId 的 session，所以会重新创建一个新的 session，并且再将这个新的 sessionID 返回给客户端，这样，我们可以想象一下，用户每一次操作都有 $1/2$ 的概率进行再次的登录，这样不仅对用户体验特别差，还会让服务器上的 session 激增，加大服务器的运行压力。

为了解决集群环境下的 session 共享问题，共有 4 种解决方案：

1. 粘性 session

粘性 session 是指 Nginx 每次都将同一用户的所有请求转发至同一台服务器上，即将用户与服务器绑定。

2. 服务器 session 复制

即每次 session 发生变化时，创建或者修改，就广播给所有集群中的服务器，使所有的服务器上的 session 相同。

3. session 共享

缓存 session，使用 redis、memcached。

4. session 持久化

将 session 存储至数据库中，像操作数据一样才做 session。

9 memcached 与 redis 的区别？

1、Redis 不仅仅支持简单的 k/v 类型的数据，同时还提供 list, set, zset, hash 等数据结构的存储。而 memcache 只支持简单数据类型，需要客户端自己处理复杂对象

2、Redis 支持数据的持久化，可以将内存中的数据保持在磁盘中，重启的时候可以再次加载进行使用（PS：持久化在 rdb、aof）。



更多关注 Java 大后端公众号

3、由于 Memcache 没有持久化机制，因此宕机所有缓存数据失效。Redis 配置为持久化，宕机重启后，将自动加载宕机时刻的数据到缓存系统中。具有更好的灾备机制。

4、Memcache 可以使用 Magent 在客户端进行一致性 hash 做分布式。Redis 支持在服务器端做分布式(PS:Twemproxy/Codis/Redis-cluster 多种分布式实现方式)

5、Memcached 的简单限制就是键（key）和 Value 的限制。最大键长为 250 个字符。可以接受的储存数据不能超过 1MB（可修改配置文件变大），因为这是典型 slab 的最大值，不适合虚拟机使用。而 Redis 的 Key 长度支持到 512k。

6、Redis 使用的是单线程模型，保证了数据按顺序提交。Memcache 需要使用 cas 保证数据一致性。CAS（Check and Set）是一个确保并发一致性的机制，属于“乐观锁”范畴；原理很简单：拿版本号，操作，对比版本号，如果一致就操作，不一致就放弃任何操作

cpu 利用。由于 Redis 只使用单核，而 Memcached 可以使用多核，所以平均每一个核上 Redis 在存储小数据时比 Memcached 性能更 高。而在 100k 以上的数据中，Memcached 性能要高于 Redis 。

7、memcache 内存管理：使用 Slab Allocation。原理相当简单，预先分配一系列大小固定的组，然后根据数据大小选择最合适的块存储。避免了内存碎片。（缺点：不能变长，浪费了一定空间）memcached 默认情况下下一个 slab 的最大值为前一个的 1.25 倍。

8、redis 内存管理：Redis 通过定义一个数组来记录所有的内存分配情况，Redis 采用的是包装的 malloc/free，相较于 Memcached 的内存 管理方法来说，要简单很多。由于 malloc 首先以链表的方式搜索已管理的内存中可用的空间分配，导致内存碎片比较多

更多请关注微信公众号：Java 技术栈，回复：资料



更多关注 Java 大后端公众号



更多关注 Java 大后端公众号