

究竟先操作缓存，还是数据库？

原创：58沈剑 架构师之路 前天

缓存存储，也是数据的冗余。

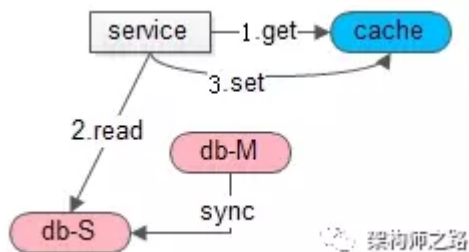
- (1) 数据库访问数据，磁盘IO，慢；
- (2) 缓存里访问数据，存操作，快；
- (3) 数据库里的热数据，可在缓存冗余一份；
- (4) 先访问缓存，如果命中，能大大的提升访问速度，降低数据库压力；

这些，是缓存的核心读加速原理。

但是，一旦没有命中缓存，或者一旦涉及写操作，流程会比没有缓存更加复杂，这些是今天要分享的话题。

读操作，如果没有命中缓存，流程是怎么样的？

答：如下图所示



- (1) 尝试从缓存get数据，结果没有命中；
- (2) 从数据库获取数据，读从库，读写分离；
- (3) 把数据set到缓存，未来能够命中缓存；

读操作的流程应该没有歧义。

写操作，流程是怎么样的？

答：写操作，既要操作数据库中的数据，又要操作缓存里的数据。

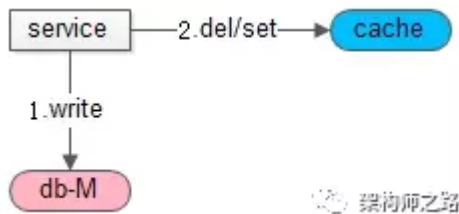
这里，有两个方案：

- (1) 先操作数据库，再操作缓存；
- (2) 先操作缓存，再操作数据库；

并且，希望保证两个操作的原子性，要么同时成功，要么同时失败。

这演变为一个分布式事务的问题，保证原子性十分困难，很有可能出现一半成功，一半失败，接下来看下，当原子性被破坏的时候，分别会发生什么。

一、先操作数据库，再操作缓存



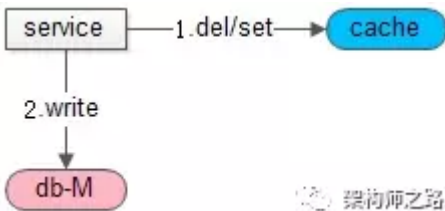
如上图，正常情况下：

- (1) 先操作数据库，成功；
- (2) 再操作缓存（delete或者set），也成功；

但如果这两个动作原子性被破坏：**第一步成功，第二步失败**，会导致，数据库里是新数据，而缓存里是旧数据，业务无法接受。

画外音：如果第一步就失败，可以返回调用方50X，不会出现数据不一致。

二、先操作缓存，再操作数据库



如上图，正常情况下：

- (1) 先操作缓存（delete或者set），成功；
- (2) 再操作数据库，也成功；

画外音：如果第一步就失败，也可以返回调用方50X，不会出现数据不一致。

如果原子性被破坏，会发生什么呢？

这里又分了两情况：

- (1) 操作缓存使用set
- (2) 操作缓存使用delete

使用set的情况：**第一步成功，第二步失败**，会导致，缓存里是set后的数据，数据库里是之前的数据，数据不一致，业务无法接受。

并且，一般来说，数据最终以数据库为准，写缓存成功，其实并不算成功。

使用delete的情况：**第一步成功，第二步失败**，会导致，缓存里没有数据，数据库里是之前的数据，数据没有不一致，对业务无影响。只是下一次读取，会多一次cache miss。

画外音：此时可以返回调用方50X。

最终，先操作缓存，还是先操作数据库？

答：

(1) 读请求，先读缓存，如果没有命中，读数据库，再set回缓存

(2) 写请求

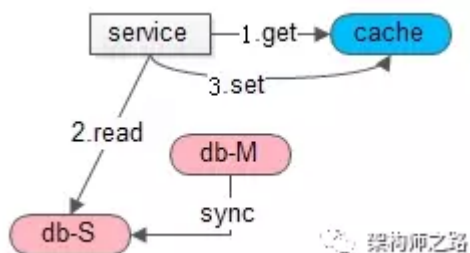
(2.1) 先缓存，再数据库

(2.2) 缓存，使用delete，而不是set

画外音：《缓存，究竟是淘汰，还是修改？》也提到了，淘汰缓存还是修改缓存的建议。

希望大家有收获，有不同方案欢迎讨论。

末了，挖个坑：



在缓存读取流程中，如果主从没有同步完成，步骤二读取到一个旧数据，可能导致缓存里set一个旧数据，最终导致数据库和缓存数据不一致。

如何解决这种情况下，缓存与数据库数据不一致的问题，是下一章要讨论的内容。



相关推荐：

《进程内缓存，究竟怎么玩？》

《选redis还是memcache，源码怎么说？》

《缓存，你真的用对了么？》

《缓存，究竟是淘汰，还是修改？》

《缓存，并发更新的大坑？》

《数据库主从不一致，怎么解？》

文字很短，希望大家有启示，帮转。

阅读原文