关于缓存

作者：antswl
原文链接：https://ld246.com/article/1535446391453
来源网站：链滴

什么是缓存

凡是位于速度相差较大的两种数据存取介质之间，用于协调两者数据传输速度差异的结构，均可称之缓存（Cache）。

命中和回源

当向存储介质获取数据时, 当缓存中有对应的数据, 则直接讲缓存中的数据返回给数据读取端, 这个过就是缓存命中。

如果在缓存中没有获取到对应的数据，则向缓存下层的存储获取数据，这个过程就是缓存回源。

在设计和使用缓存的过程中, 关注的比较多的就是缓存的命中率。

缓存未命中

如果缓存中还有空间，那么没有命中的数据也会被存储到缓存当中；如果缓存中没有了空间，而缓存没命中，那么就会按照一种策略把缓存中的旧对象踢出，把新的对象加入到缓存中，而那些策略就被称为缓存淘汰算法。

缓存淘汰算法

缓存淘汰算法决定了到底应该将哪些对象从缓存中移除。

引入缓存淘汰算法, 通常需要综合考虑存储成本和缓存的命中率。

常见的缓存淘汰算法有：

1. LFU(Least Frequently Used)

根据缓存访问的历史访问频率来淘汰数据, 其核心思想是 "如果数据过去访问的多次, 那么将来被访问频率也会很高"。

LFU 算法中，每个缓存数据块都有一个引用计数，所有数据块按照引用计数排序，如果引用计数相同则按照时间排序。

LFU 的实现通常使用队列, 初始化状态缓存数据块引用计数的数量都为0。当由数据块被访问过之后，往队头移动（前提要保证队列头到队列尾按照引用计数降序排列），并且记录下最后访问时间。

2. LRU(Latest Recently Used)

根据缓存的历史访问记录来淘汰数据, 其核心思想是 "如果数据最近被访问过, 那么将来被访问的可能也很高"。

LRU 算法的实现通常使用链表，新数据插入到链表头部；每当缓存命中（即缓存数据被访问），则将据移到链表头部；当链表满的时候，将链表尾部的数据丢弃。

当存在热点数据时，LRU 的效率很好，但偶发性的、周期性的批量操作会导致 LRU 命中率急剧下降缓存污染情况比较严重。

3. LRU-K (Latest Recently Used K times)

相比于 LRU, LRU-K 需要多维护一个队列, 用于记录所有缓存访问的历史, 只有当数据的访问次数达到 K 次时, 才将数据放入缓存。当淘汰数据时，LRU-K 会淘汰第 K 次访问时间距离现在最大的数据。