



缓存的正确使用方式



半分舒适 [关注](#)

2020.08.29 13:00:01 字数 1,159 阅读 25



半分舒适
总资产1 (约0.05元)

[关注](#)



image

前言

开发中不同的业务场景有不同的缓存使用方式，如对于单机的、数据量不大的场景，用 `HashMap` 就能实现一个简单的缓存，也可以借助 `Guava cache` 等来实现。对于分布式缓存，常用的如 `Redis`、`Tair`。下面探讨下平时在项目中特别是高并发场景下使用缓存时需要考虑的一些问题以及其应对的策略。

缓存的使用

缓存模式

- Cache Aside Pattern
- Read/Write Through Pattern
- Write Behind Caching Pattern

Cache Aside Pattern

应用先去缓存中找数据，若命中缓存则直接返回。如果未命中缓存，则需要先去数据库中查询数据，并将查询到的数据存储到缓存中。在这种模式下，没有单独的缓存组件，缓存和DB的读写操作由应用方负责。

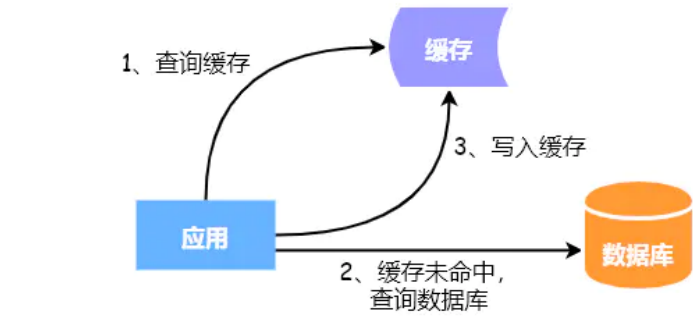


写下你的评论...

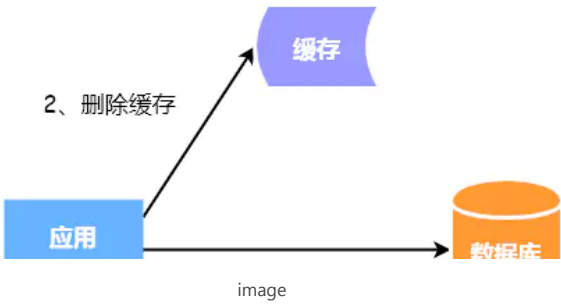
评论0

赞

...

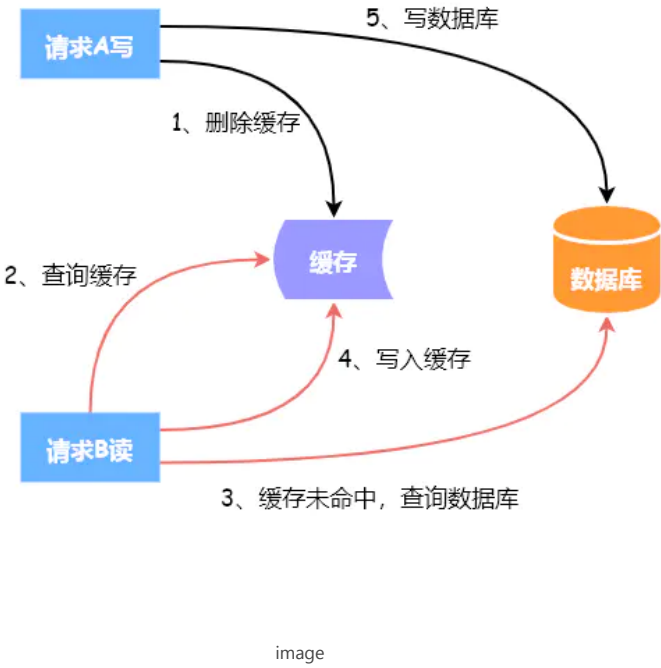


写操作



先删除缓存，再写数据库有什么问题？

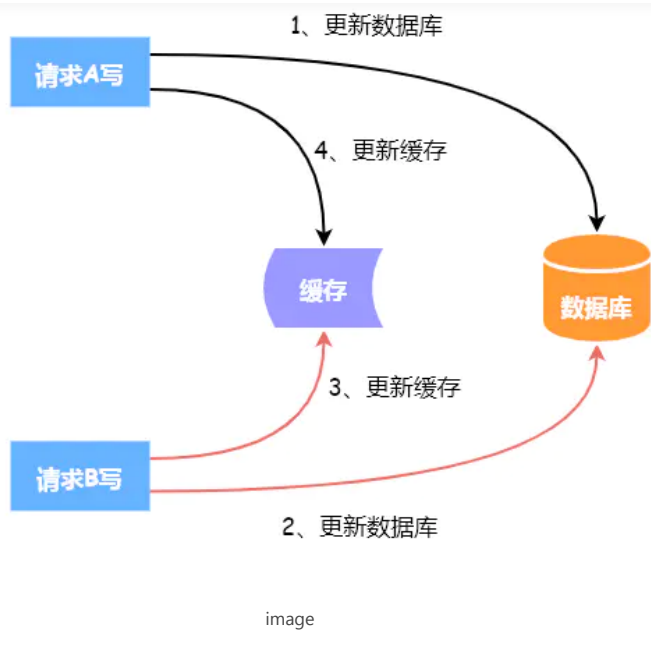
我们先来看一下场景：



在请求A淘汰缓存后，这时候请求B查询缓存，未命中，再查询数据库，此时请求尚未写完成，请求B读取的老数据，再将其写入缓存中，缓存中的数据就会一直脏下去。

为什么会出现脏数据？

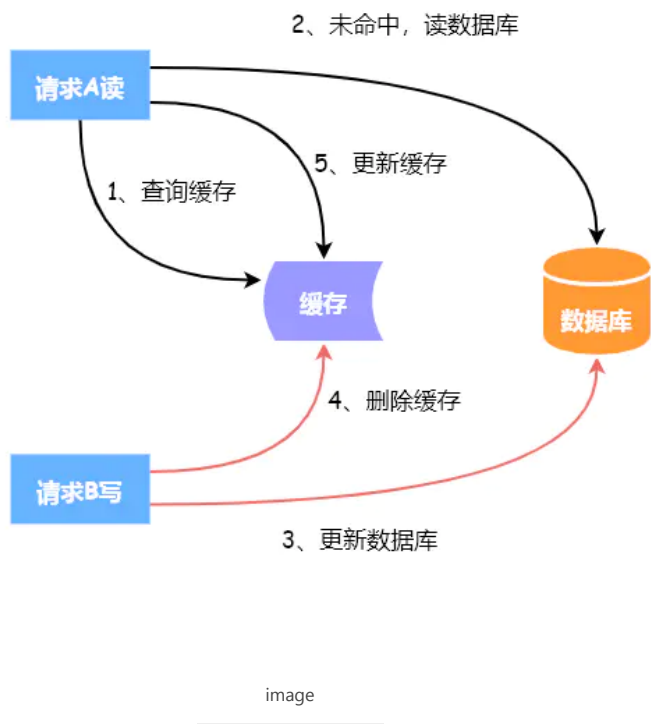
写下你的评论... 评论0 赞 ...



写请求B在请求A之前更新了缓存，这时候请求B再更新缓存，则会导致缓存中的数据就会一直脏下去。

若缓存需要繁杂的计算才能得到，且写操作频繁，则会浪费大量的计算，可以通过请求来缓存数据，也符合数据懒加载。

先更新数据，再删除缓存，同样存在数据不一致，只是概率相对小很多



若读请求B在写请求A写数据成功之前读取数据库中的数据，同时更新了缓存中的数据，

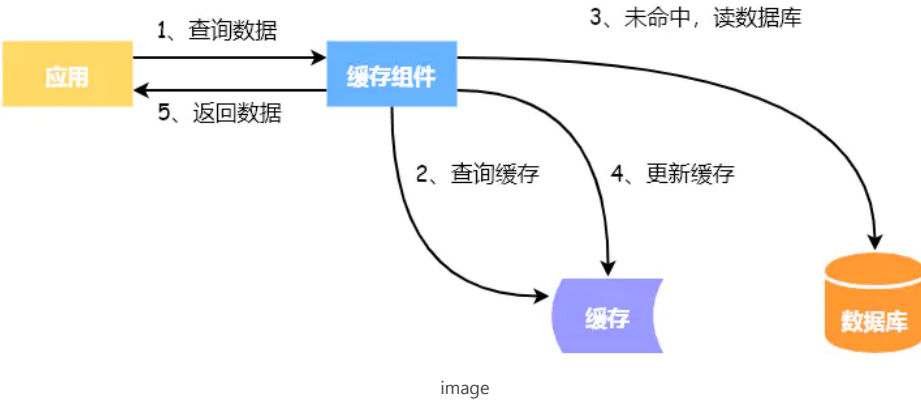
概率。

Read/Write Through Pattern

Cache Aside Pattern 是由应用程序来更新缓存中的数据，导致应用方数据库和缓存的维护设计侵入代码，数据层的耦合增大，而 Read/Write Through Pattern 则是通过缓存自身来更新数据，调用方直接和缓存管理组件打交道，避免应用和数据库之间的直接连接。

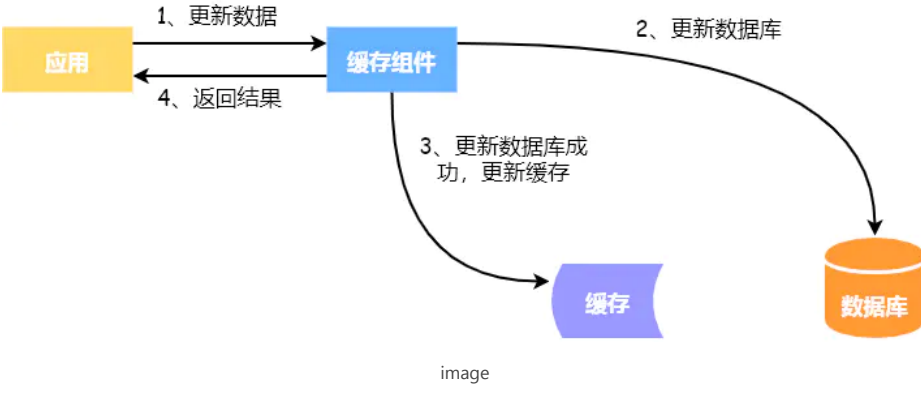
Read Through

应用向缓存管理组件发送查询请求，由缓存管理组件查询缓存，若缓存未命中，查询数据库，并将查询的数据写入缓存，并返回给应用。



Write Through

Write Through 套路和 Read Through 相仿，当更新数据的时候，将请求发送给缓存管理组件，由缓存管理组件同步更新数据库和缓存数据。



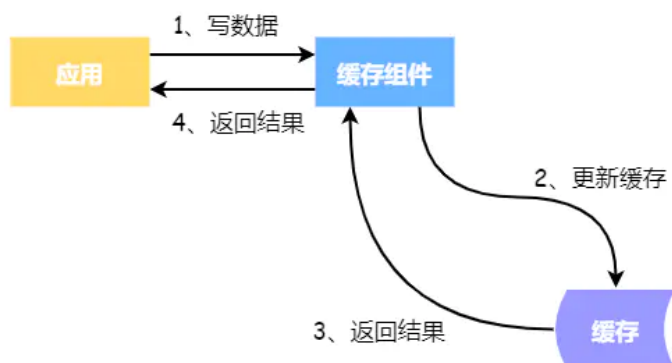
Write Behind Caching Pattern

Write Behind 模式和 Write Through 模式非常相似，不同点在于 Write through 将数据更新到数据库中是同步的方式，而 Write Behind 是通过异步方式。

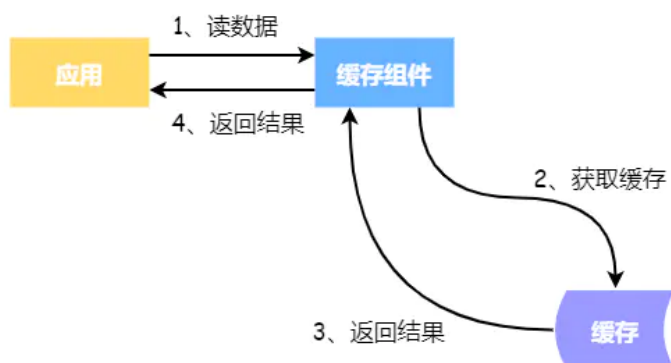




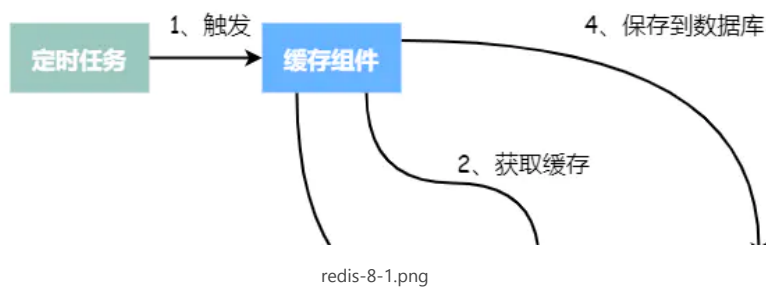
与缓存



读缓存



刷新缓存



redis-8-1.png

请求直接将数据写入缓存，并没有更新到数据库。而通过定时或者一定阈值的方式，将数据异步的同步到数据库中。这种方式优势在于有效减少了更新数据的频率，读写响应非常快，吞吐量也会有明显的提升。

由于数据操作不是强一致性的，而此时数据存放在缓存中，如果缓存在同步到数据库的过程中宕机了，则这部分数据就会有丢失的风险。



Read/Write Through 引入了缓存组件来统一管理缓存和数据库，对于使用方来说是透明的，无代码侵入，但相对于Cache Aside实现会负责一些

Write Behind Caching 数据的读写直接通过缓存，性能最好，在高并发场景下可以降低数据库的压力，通过定时或阈值的方式刷新到数据库中，但是存在宕机丢失数据的风险。

👍

0人点赞 >

👎

📖

日记本

...

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



半分舒适 寻觅那原滋原味的半份舒适!
总资产1 (约0.05元) 共写了1159字 获得0个赞 共0个粉丝

关注



写下你的评论...

全部评论 0 只看作者

按时间倒序 按时间正序

被以下专题收入，发现更多相似内容

+ 收入我的专题

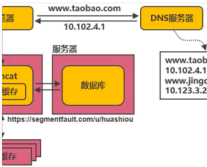
推荐阅读

更多精彩内容 >

阿里巴巴为什么能抗住90秒100亿？看完这篇你就明白了！

作者：huashiou链接：https://segmentfault.com/a/1190000018626163...

夜空_2cd3 阅读 120,195 评论 58 赞 997



基于 Flink SQL CDC 的实时数据同步方案

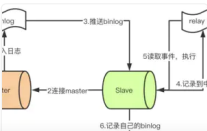
简介：Flink 1.11 引入了 Flink SQL CDC，CDC 能给我们数据和业务间能带来什么变化？本文由...

阿里云云栖号 阅读 366 评论 0 赞 6



Mysql的几个灵魂拷问（五）

这一篇继续讲SQL的优化问题，在常规应用开发中，Mysql的单表性能都是够用的，从量级来看，一般以整型值为主的表在...



写下你的评论...

评论0 赞 ...



简介：供应链物流场景下的业务复杂度高，业务链路长，节点多，实体多，实时数仓建设难度高。菜鸟跨境进口业务场景更是如此...

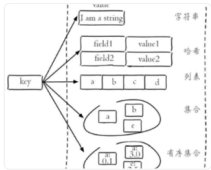
阿里云栖号 阅读 3,741 评论 1 赞 71



看完这20道Redis面试题后，蚂蚁金服面试可以约起来了？

1、什么是Redis，Redis有哪些特点？ Redis全称为：Remote Dictionary Server (...)

程序员追风 阅读 2,007 评论 1 赞 32



写下你的评论...

评论0 赞 ...