



分布式系统架构7：本地缓存

卷福同学 2025-01-17 132 阅读4分钟 专栏：小白晋级大师之路

这是小卷对分布式系统架构学习的第10篇文章，在开始学习分布式缓存之前，先来学习本地缓存的理论基础，了解为什么需要用缓存

1.引入缓存的影响

我们在开发时，用到缓存的情况，无非就是为了减少客户端对相同资源的重复请求，降低服务器的负载压力。引入缓存后，既有好处也有坏处

引入缓存负面影响:

- 开发角度, 增加了系统复杂度, 需考虑缓存失效、更新、一致性问题
- 运维角度, 缓存会掩盖一些缺陷问题
- 安全角度, 缓存可能泄密某些保密数据

引入缓存的理由：

- 为了缓解CPU压力，将实时计算运行结果存储起来，节省CPU压力
- 为了缓解I/O压力，将原本对网络、磁盘的访问改为对内存的访问

2.缓存的属性

选择缓存时，主要考虑吞吐量、命中率、扩展功能、分布式支持。前3个这篇文章会讲，下一篇再讲分布式缓存

2.1 吞吐量

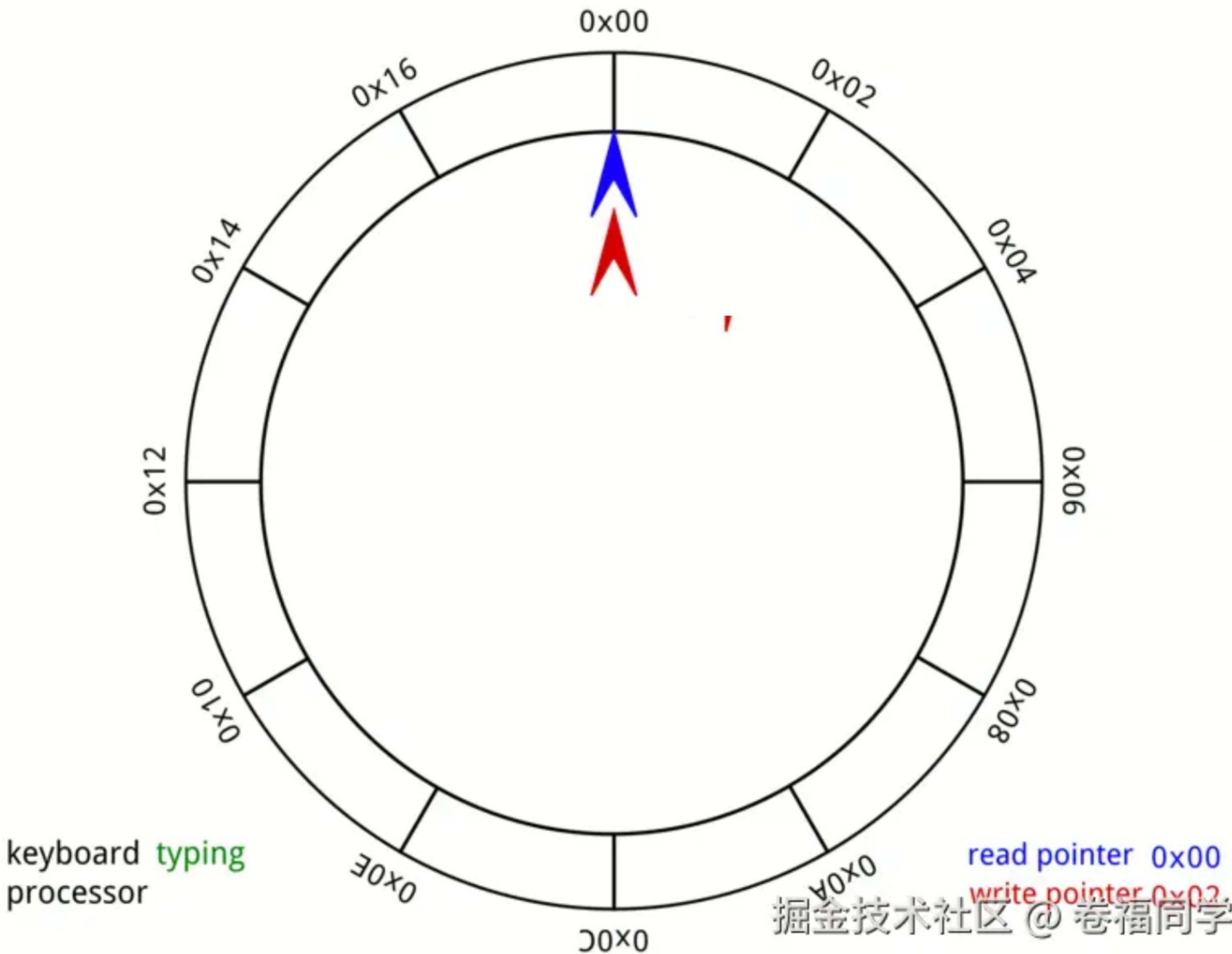
并发场景下，每秒操作数OPS，反映了缓存的工作效率

如Java8并发包的ConcurrentHashMap，线程安全实现原理是CAS+synchronized锁单个元素。但是该类仅有缓存功能，没有命中率、淘汰策略、缓存统计等功能

并发场景下，不可避免的会有读写数据带来的状态竞争问题，当前有2种处理套路：

- 以 **Guava Cache** 为代表的同步处理机制：在访问缓存数据时，一并完成缓存淘汰、统计、失效等状态变更操作，通过分段加锁等优化手段来尽量减少数据竞争。
- 以 **Caffeine** 为代表的异步日志提交机制：参考经典的数据库设计理论，把对数据的读、写过程看作是日志（即对数据的操作指令）的提交过程。

Caffeine使用了环形缓冲区来记录状态变动日志，为进一步减少数据竞争，Caffeine给每个线程都设置了专用的环形缓冲区，如下是Wikipedia上的环形缓冲示意：



环形缓冲区结构中，读取和写入是一起进行的，只要读取指针不落后于写指针一圈，这个操作可以永久进行下去，容纳无限多的新字符。

如果不满足，则必须阻塞写指针，等待读取清空缓冲区

2.2 命中率与淘汰策略

缓存的容量是有限的，也因此需要自动地实现淘汰低价值目标，也就是缓存淘汰策略



卷福同学 LV.4

资深开发 @阿里巴巴
 作者榜No.9 >

78 文章

150k 阅读

278 粉丝

关注

私信

目录

收起 ^

- 1.引入缓存的影响
- 2.缓存的属性
 - 2.1吞吐量
 - 2.2 命中率与淘汰策略
- 3.扩展功能
- 4.本地缓存对比

相关推荐

86阅读 · 3点赞

MVCC如何应对MySQL并发问题

RocketMQ 安装使用

ZooKeeper 常见问题与核心机制解析

基于Spring Data Redis的Redis客户端封...
36阅读 · 1点赞

精选内容

Web开发-部署模式概述
天南星 · 24阅读 · 1点赞

OpenHarmony（鸿蒙南向开发）——小...
塞尔维亚大汉 · 23阅读 · 0点赞

掌握线程池在工程实践！（定时线程池...
稻草人2222 · 59阅读 · 2点赞

docker 私有化环境 registry 镜像仓库部...
昵称为空C · 44阅读 · 1点赞

Go框架面试突击！30道高频题解析
王中阳讲编程 · 568阅读 · 2点赞

找对属于你的技术圈子

回复「进群」加入官方微信群



主要实现方案有三种：

第一种：FIFO（First In First Out）

即优先淘汰最早进入被缓存的数据。FIFO 的实现十分简单，但一般来说，越是频繁被用到的数据，往往越会早早地被存入缓存之中。

所以这种淘汰策略，很可能会大幅降低缓存的命中率

第二种：LRU（Least Recent Used）

即优先淘汰最久未被使用访问过的数据。LRU 通常会采用 HashMap 加 LinkedList 的双重结构（如 LinkedHashMap）来实现。每次缓存命中时，将命中对象调整到LinkedList的头部，每次淘汰从链表尾部清理

存在问题：如果热点数据一段时间没被访问，就会被淘汰；

第三种：LFU（Least Frequently Used）

优先淘汰最不经常使用的数据。LFU 会给每个数据添加一个访问计数器，每访问一次就加 1，当需要淘汰数据的时候，就清理计数器数值最小的那批数据。

缺点：每个数据都需要维护计数器，不便于处理随时间变化的热点数据

以上只是列了三种基础的淘汰策略，实际Caffeine 官方还制定了两种高级淘汰策略：ARC（Adaptive Replacement Cache）和LIRS（Low Inter-Reference Recency Set），更复杂的淘汰策略都是为了提高命中率的。

3.扩展功能

缓存不是只实现一个Map接口就可以的，还需要一些额外的功能，下面列出缓存的扩展功能：

- **加载器**：从被动放入，变为主动通过加载器去加载指定 Key 值的数据
- **淘汰策略**：支持用户根据需要自行选择淘汰策略
- **失效策略**：缓存数据在超过一定时间内自动失效，Redis的策略是定时删除、定期删除、惰性删除
- **事件通知**：提供事件监听器，在数据状态变动时进行一些额外操作
- **并发级别**：如Guava Cache通过分段加锁来实现缓存的并发设置
- **容量控制**：设置初始容量和最大容量
- **统计信息**：命中率、平均加载时间、自动回收计数等信息
- **持久化**：将缓存数据存储到数据库或者磁盘

4.本地缓存对比

| | ConcurrentHashMa p | Ehcache | Guava Cache | Caffeine |
|------|-----------------------|------------------|-------------|-------------------------|
| 访问控制 | 最高 | 一般 | 良好 | 优秀（接近ConcurrentHashMap） |
| 淘汰策略 | 无 | 多种：FIFO、LRU、LFU等 | LRU | W-TinyLFU |
| 扩展功能 | 无，只有基础访问接口 | 并发控制、失效策略、容量控制等 | 同左 | 同左 |

标签：

后端

Java

话题：

金石计划征文活动

本文收录于以下专栏



小白晋级大师之路

专栏目录

有深度的技术文稿

7 订阅 · 27 篇文章

订阅


上一篇

架构学习：7种负载均衡算法策略

下一篇

分布式系统架构8：分布式缓存

评论 0



登录 / 注册

即可发布评论！



暂无评论数据

为你推荐

盘点一下面试中涉及到的缓存内容！！

小丶丶琦哥 | 3年前 | 863 | 4 | 评论

面试

缓存八股文

一枚空指针 | 3年前 | 1.2k | 3 | 评论

后端数据库CDN

解析分布式系统的缓存设计

vivo互联网技术 | 2年前 | 1.7k | 14 | 3

服务器分布式后端

项目中多级缓存设计实践总结

CoderV的进阶笔记 | 1年前 | 4.4k | 47 | 4

后端架构Redis

分布式系统架构8：分布式缓存

卷福同学 | 25天前 | 52 | 点赞 | 评论

后端架构设计

10分钟搞定分布式应用缓存

俞凡 | 8月前 | 524 | 1 | 评论

架构

如果不知道这4种缓存模式，敢说懂缓存吗？

程序新视界 | 2年前 | 2.6k | 17 | 2

后端Redis

操作系统、Java及数据库缓存详解| 小册免费学

CVNot | 3年前 | 766 | 6 | 评论

操作系统Java

缓存技术：加速应用，提高用户体验

MobotStone | 1年前 | 956 | 4 | 评论

后端架构

在做分布式缓存的时候如何选择，如何解决遇到的坑？

码农开花 | 4年前 | 231 | 1 | 评论

分布式

MyBatis中的多级缓存机制（一级缓存和二级缓存）

写bug写bug | 4月前 | 192 | 2 | 评论

后端JavaMyBatis

缓存、异步、集群和分布式等架构模式的实践

编程学习网 | 4年前 | 503 | 点赞 | 评论

分布式

《终章》后端分布式多级缓存架构，也许你一直考虑的太简单了

莲花童子哪吒 | 4年前 | 1.3k | 20 | 4

架构

什么是分布式缓存？

猿java | 4月前 | 369 | 19 | 评论

后端架构面试

跟着 Guava 学 Java 之缓存

xiaohezi | 2年前 | 1.2k | 2 | 评论

Java