Redis 基础：为什么要用分布式缓存?

> **相关面试题**：
>
> - 为什么要用缓存?
> - 本地缓存应该怎么做?
> - 为什么要有分布式缓存?/为什么不直接用本地缓存?
> - 多级缓存了解么?

缓存的基本思想

很多同学只知道缓存可以提高系统性能以及减少请求相应时间，但是，不太清楚缓存的本质思想是什么。

缓存的基本思想其实很简单，就是我们非常熟悉的空间换时间。不要把缓存想的太高大上，虽然，它的确对系统的性能提升的性价比非常高。

其实，我们在学习使用缓存的时候，你会发现缓存的思想实际在操作系统或者其他地方都被大量用到。 比如 CPU Cache 缓存的是内存数据用于解决 CPU 处理速度和内存不匹配的问题，内存缓存的是硬盘数据用于解决硬盘访问速度过慢的问题。 再比如操作系统在 页表方案 基础之上引入了 快表 来加速虚拟地址到物理地址的转换。我们可以把快表理解为一种特殊的高速缓冲存储器（Cache）。

我们知道，缓存中的数据通常存储于内存中，因此访问速度非常快。为了避免内存中的数据在重启或者宕机之后丢失，很多缓存中间件会利用磁盘做持久化。

也就是说，缓存相比较于我们常用的关系型数据库（比如 MySQL）来说访问速度要快非常多。为了避免用户请求数据库中的数据速度过于缓慢，我们可以在数据库之上增加一层缓存。

除了能够提高访问速度之外，缓存支持的并发量也要更大，有了缓存之后，数据库的压力也会随之变小。

缓存的分类

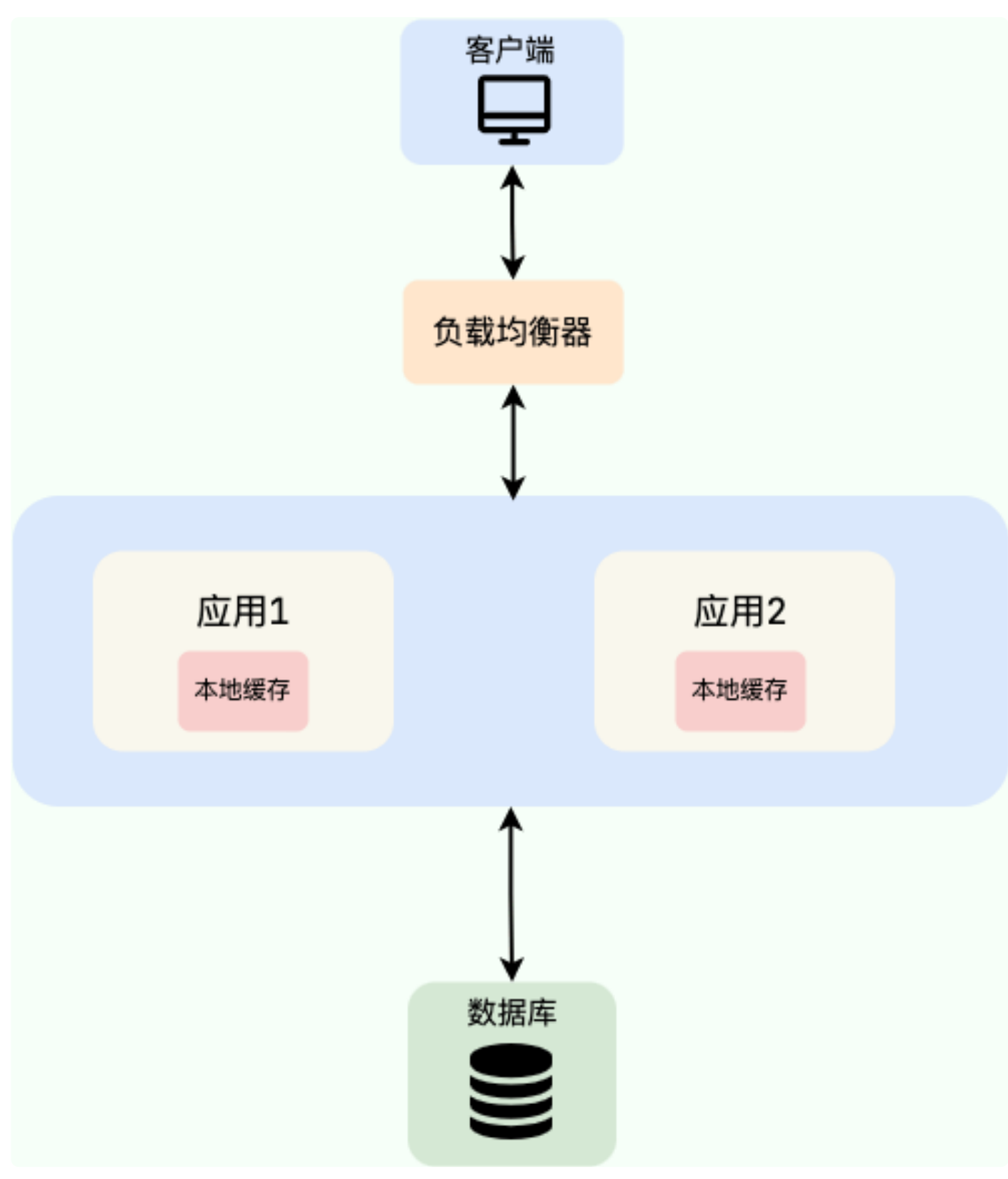
本地缓存

什么是本地缓存?

这个实际在很多项目中用的蛮多，特别是单体架构的时候。数据量不大，并且没有分布式要求的话，使用本地缓存还是可以的。

本地缓存位于应用内部，其最大的优点是应用存在于同一个进程内部，请求本地缓存的速度非常快，不存在额外的网络开销。

常见的单体架构图如下，我们使用 Nginx 来做**负载均衡**，部署两个相同的应用到服务器，两个服务使用同一个数据库，并且使用的是本地缓存。



本地缓存的方案有哪些?

1、JDK 自带的 `HashMap` 和 `ConcurrentHashMap` 了。

`ConcurrentHashMap` 可以看作是线程安全版本的 `HashMap` ，两者都是存放 key/value 形式的键值对。但是，大部分场景来说不会使用这两者当做缓存，因为只提供了缓存的功能，并没有提供其他诸如过期时间之类的功能。一个稍微完善一点的缓存框架至少要提供：**过期时间**、**淘汰机制**、**命中率统计**这三点。

2、 `Ehcache` 、 `Guava Cache` 、 `Spring Cache` 这三者是使用的比较多的本地缓存框架。

- `Ehcache` 的话相比于其他两者更加重量。不过，相比于 `Guava Cache` 、 `Spring Cache` 来说， `Ehcache` 支持可以嵌入到 hibernate 和 mybatis 作为多级缓存，并且可以将缓存的数据持久化到本地磁盘中、同时也提供了集群方案（比较鸡肋，可忽略）。
- `Guava Cache` 和 `Spring Cache` 两者的话比较像。 `Guava` 相比于 `Spring Cache` 的话使用的更多一点，它提供了 API 非常方便我们使用，同时也提供了设置缓存有效时间等功能。它的内部实现也比较干净，很多地方都和 `ConcurrentHashMap` 的思想有异曲同工之妙。
- 使用 `Spring Cache` 的注解实现缓存的话，代码会看着很干净和优雅，但是很容易出现问题比如缓存穿透、内存溢出。

3、后起之秀 Caffeine。

相比于 `Guava` 来说 `Caffeine` 在各个方面比如性能要更加优秀，一般建议使用其来替代 `Guava` 。并且， `Guava` 和 `Caffeine` 的使用方式很像！

本地缓存有什么痛点?

本地的缓存的优势非常明显：**低依赖**、**轻量**、**简单**、**成本低**。

但是，本地缓存存在下面这些缺陷：

- **本地缓存应用耦合，对分布式架构支持不友好**，比如同一个相同的服务部署在多台机器上的时候，各个服务之间的缓存是无法共享的，因为本地缓存只在当前机器上有。
- **本地缓存容量受服务部署所在的机器限制明显。** 如果当前系统服务所耗费的内存多，那么本地缓存可用的容量就很少。

分布式缓存

什么是分布式缓存?

我们可以把分布式缓存（Distributed Cache） 看作是一种内存数据库的服务，它的最终作用就是提供缓存数据的服务。

分布式缓存脱离于应用独立存在，多个应用可直接的共同使用同一个分布式缓存服务。

如下图所示，就是一个简单的使用分布式缓存的架构图。我们使用 Nginx 来做负载均衡，部署两个相同的应用到服务器，两个服务使用同一个数据库和缓存。

使用分布式缓存之后，缓存服务可以部署在一台单独的服务器上，即使同一个相同的服务部署在多台机器上，也是使用的同一份缓存。 并且，单独的分布式缓存服务的性能、容量和提供的功能都要更加强大。

软件系统设计中没有银弹，往往任何技术的引入都像是把双刃剑。 你使用的方式得当，就能为系统带来很大的收益。否则，只是费了精力不讨好。

简单来说，为系统引入分布式缓存之后往往会带来下面这些问题：

- **系统复杂性增加** ：引入缓存之后，你要维护缓存和数据库的数据一致性、维护热点缓存、保证缓存服务的高可用等等。
- **系统开发成本往往会增加** ：引入缓存意味着系统需要一个单独的缓存服务，这是需要花费相应的成本的，并且这个成本还是很贵的，毕竟耗费的是宝贵的内存。

分布式缓存的方案有哪些?

分布式缓存的话，比较老牌同时也是使用的比较多的还是 **Memcached** 和 **Redis**。不过，现在基本没有看过还有项目使用 **Memcached** 来做缓存，都是直接用 **Redis**。

Memcached 是分布式缓存最开始兴起的那会，比较常用的。后来，随着 Redis 的发展，大家慢慢都转而使用更加强大的 Redis 了。

另外，腾讯也开源了一款类似于 Redis 的分布式高性能 KV 存储数据库，基于知名的开源项目 RocksDB 作为存储引擎 ，100% 兼容 Redis 协议和 Redis4.0 所有数据模型，名为 Tendis。

关于 Redis 和 Tendis 的对比，腾讯官方曾经发过一篇文章：Redis vs Tendis：冷热混合存储版架构揭秘 ，可以简单参考一下。

从这个项目的 Github 提交记录可以看出，Tendis 开源版几乎已经没有被维护更新了，加上其关注度并不高，使用的公司也比较少。因此，不建议你使用 Tendis 来实现分布式缓存。

多级缓存

我们这里只来简单聊聊 **本地缓存 + 分布式缓存** 的多级缓存方案。

这个时候估计有很多小伙伴就会问了：**既然用了分布式缓存，为什么还要用本地缓存呢？** 。

的确，一般情况下，我们也是不建议使用多级缓存的，这会增加维护负担（比如你需要保证一级缓存和二级缓存的数据一致性），并且，实际带来的提升效果对于绝大部分项目来说其实并不是很大。

多级缓存方案中，第一级缓存（L1）使用本地内存（比如 Caffeine)），第二级缓存（L2）使用分布式缓存（比如 Redis）。读取缓存数据的时候，我们先从 L1 中读取，读取不到的时候再去 L2 读取。这样可以降低 L2 的压力，减少 L2 的读次数。并且，本地内存的访问速度是最快的，不存在什么网络开销。

J2Cache 就是一个基于本地内存和分布式缓存的两级 Java 缓存框架，感兴趣的同学可以研究一下。

25 人点赞

陈龙 fxz

SnailClimb 10-11 11:54 15048 7 IP 属地湖北 | 举报

上一篇
MySQL 索引：索引为什么使用 B+树?

下一篇
Redis 基础：常见的缓存更新策略有哪几种?

所有评论（7）

Asphyxia 03-08 16:08

> 引用原文：[图片]

图是不是错了，cache存在，先更新cache

3

SnailClimb 03-09 10:29

嗯嗯 笔误了，已经修改。

王超群 03-20 17:32

> 引用原文：但是，两个又有很大的不同：Read/Write Through 是同步更新 cac...

这里的 Write Behind **Caching** 是 **Pattern** 的笔误吗?

SnailClimb 03-21 10:00

嗯嗯，笔误了，已修改。

波哥咩咩 07-27 10:40

> 引用原文：请求 1 从 DB 读数据 A->请求 2 写更新数据 A 到数据库并把删除 c...

这块儿没看明白，请求1为什么会直接从DB读数据呢。

Lian 07-27 13:19

我也没看懂，是不是请求1进来发现缓存没有数据A尼

SnailClimb 07-29 23:19 IP 属地湖北

简单完善了一下描述，突出了前提条件，出现这种问题的概率非常小。

25
7