缓存中数据按照缓存行（一般一个缓存行64B）进行读取。

1、[cpu无论单核心还是多核心都有二级缓存吗\_百度知道 (baidu.com)](https://zhidao.baidu.com/question/142988795.html)

2、[(2条消息) CPU中的多级缓存\_风物长宜放眼量-CSDN博客\_cpu多级缓存](https://blog.csdn.net/gcoder_/article/details/104928512)

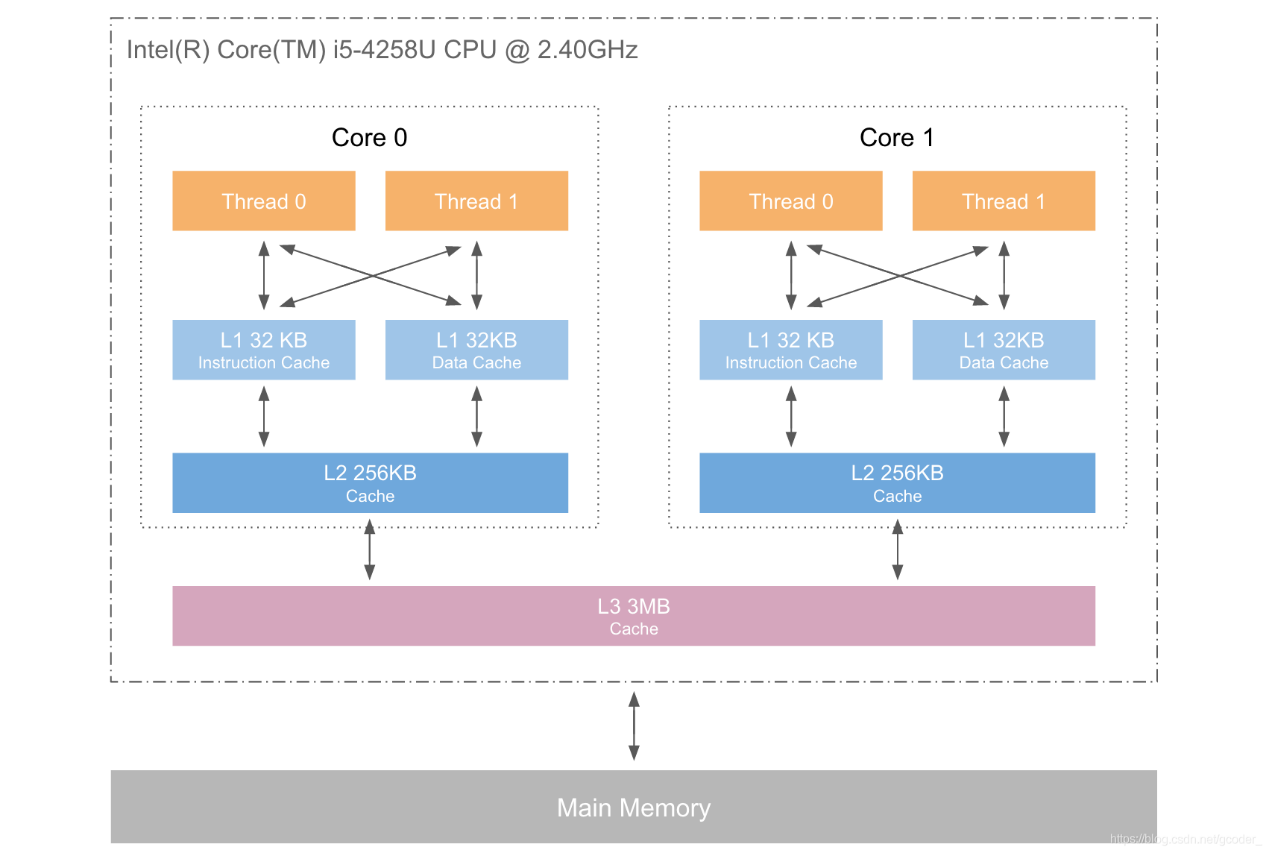
3、**[既然CPU有缓存一致性协议（MESI），为什么JMM还需要volatile关键字？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/296949412)**

4、[(2条消息) java怎么避免重排序,Java volatile 怎么保证不被指令重排序优化\_我要WhatYouNeed的博客-CSDN博客](https://blog.csdn.net/weixin_35782148/article/details/115504528)（happens-before准则）

5、[Java synchronized 能防止指令重排序吗？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/337265532)

6、[谁能用通俗的语言解释一下什么是 RPC 框架？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/25536695)

（指令并行执行，指的对硬件资源的操作并行）



一些个人猜想：

1、static变量对应到物理内存中就只有一个地址。所以多个线程都是从这个地址中去取。那么多线程的在执行时，单核cpu中如果线程1的写操作还没有完成就切换线程2读，那么就会出现数据的错误（所以我们需要用锁，保证单核线程上的代码原子性）。多核cpu且多线程运行在不同的cpu上（运行在同一上就和单核上一致），那么线程1还没有写线程2就读，是不是，线程2再回（同样产生的问题）。那么如果我们把这个数据块用锁锁起来是不是行了呢，确实是这样。如果这样锁起来，那么是不是就和单cpu并发加锁一致呢？那么性能的利用是可想而知的。接着就出现一系列的解决方式。比如调整锁粒度。知识有限。

上述并没有加上缓存情况，缓存的主要作用就是解决了cpu与内存频率不匹配问题。但是紧跟着就导致了共享数据的在更多地方可能存在不一致性，那么以前的锁功能能够完全解决数据的不一致性吗？明显不能，所以出现MESI、内存屏障等等，用来增强锁功能，提供底层支持。

2、其实内存中数据一致性问题和数据库中数据一致性极其类似。（事务原子性、隔离性、持久性就是为了保证数据的一致性，因为数据库需要将数据长期存起来加以复用，所以有持久性，而内存中虽然不存在长期保存，但是只要需要暂时保存而且需要复用那差别不大。所以数据中的脏读、幻读、不可重复读的现象也应当相对存在。）

3、可见性是对于ｃｐｕ缓存之间。Java中的IO模型主要是针对于单体服务上的。

４、要彻底理解锁、volatile（挥发性、可见性）、内存屏障、MESI等协议、最好进行一步步了解多线程编程、以及计算机硬件性能、设计等各种发展历史。

５、[(2条消息) 锁（Lock）、内存屏障（Memory barrier）与 缓存一致性（ Cache coherence）\_Jeff\_的博客-CSDN博客](https://blog.csdn.net/weixin_40539125/article/details/104215766)

６、Java 中解决缓存不一致的方法有两种，第一种就是 synchronized 使用的总线锁方式，也就是在总线上声言 LOCK# 信号；第二种就是著名的 MESI 协议。这两种都是硬件层面提供的解决方式。

７、首先volatile是作用于共享数据。

８、volatile保证可见性和有序性、synchronized保证原子性、有序性以及可见性。

９、[为什么synchronized无法禁止指令重排，却能保证有序性 - 会飞的鱼喵喵 - 博客园 (cnblogs.com)](https://www.cnblogs.com/bcl88/p/12400747.html)

１０、**[既然synchronized是"万能"的，为什么还需要volatile呢？ - HollisChuang - 博客园 (cnblogs.com)](https://www.cnblogs.com/hollischuang/p/11386988.html)**

１１、禁止指令重排序（通过禁止指令重排序解决有序性问题）：Java层面volatile－＞JVM层内存屏障－＞操作系统层使用汇编ｌｏｃｋ指令

１２、[(2条消息) CPU多级缓存、MESI缓存同步协议、指令重排、内存屏障解决指令重排和多级缓存带来的问题\_留夕的博客-CSDN博客](https://blog.csdn.net/u013546788/article/details/105829283)

１３、[volatile如何保证有序性？内存屏障都不知道怎么拿offer呀 (baidu.com)](https://baijiahao.baidu.com/s?id=1666283243240041852&wfr=spider&for=pc)

１４、MESI协议是cpu硬件层面（MESI保证CPU的L1与L3之间可见性，也就是说该CPU通过遵循该协议能够保证各个CPU之间的缓存知道共享数据在各自CPU缓存中所处的状态－如何知道的可以采取的机制与很多，比如嗅探机制。而CPU仅仅只是知道了数据的状态，如果要进行数据的更新还是需要重新重主内存中进行统一的更新，而不是说我直接可以重别CPU缓存中直接拿，想想也不安全。而修改的CPU就需要在其他cpu读取之前讲值写回主存）

１５、[(2条消息) MESI cpu\_大曹的博客-CSDN博客](https://blog.csdn.net/qq_36057860/article/details/93386882)

１６**、**[**(2条消息) mesi协议怎么实现\_MESI一致性协议与JMM内存模型的实现\_weixin\_39575054的博客-CSDN博客**](https://blog.csdn.net/weixin_39575054/article/details/113452092)（简练）

17、java中的volatile或者加锁会生成lock前缀的指令（[缓存一致性协议mesi，是自动触发的吗，还是必须使用volatile或者同步锁等才会触发？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/474960496/answer/2023908581)）虽然 mesi 会自动触发，但一般要确认触发了 mesi 还是需要 volatile 的，否则无法避免编译器各种优化和乱序。

18、[volatile的原理分析 - Yrion - 博客园 (cnblogs.com)](https://www.cnblogs.com/wyq178/p/8552950.html)（使用场景、工作内存、限制）