

在物理结构上，内存是硬盘和 cache 之间的缓存，cache 是内存和 core 之间的缓存，它们都是为了了一次载入超量信息数据供前级使用而存在的。

从逻辑上讲，计算机只需要有处理器和对应数据输入输出即可，即 cpu 对硬盘数据进行处理即可。但是硬盘无法满足高速访问的需求，所以用内存作为中间层。

cpu和内存通信快了，但内存不能持久保存数据，所以还是必须从硬盘加载，这是一个串行任务，cpu必须等内存加载到硬盘内数据后才能从内存中访问该内容，那么若要一点取一点，最后总的交流速度还是被硬盘限制了。所以，用一定策略，将可能要访问的数据提前加载到内存中，就能减少与硬盘

交通的等待, 从而提升处理速度。

这个策略一般有最近访问内容保留在内存, 以及其相关数据块提前(一同)载入内存, 命中率是内存有效性的关键。

Cache 对于内存的意义也一样。

有内存加速的同时, 也有一定的数据冲突风险。

在内存层面, 对于内存中已改变

但对应硬盘内容未改变时,另一个线程对硬盘同一数据块作出改变,两者都要保存改变到硬盘及文件,就可能产生冲突,若是块同作业,这样可能就破坏了他人的工作了.

Cache 缓存也一样,因有不同cpu 线程可以都操作同一内存数据,若不采取策略,可能

... 内存互斥锁

Cache 中已更改的数据，
有时另一线程也要这个数据，两者
就冲突了。

任何处理器上有缓存功能的
中间层都会有这个问题，解决
策略也都不外乎进行一定权限
限制以及进行不同等级的沟通
协商。