

# CACHE LINE

一个 L1 DATA CACHE 相当于一块小的内存，我们假设它为 16K 大，它会与一般物理内存交互。

它和内存交互一般一次传输 16 个字节(32 个字节),也就是:

CACHE 字节 0-15 一次写到/读取物理内存，字节 16-31 一次写到/读取物理内存.32-47 ... ..

这些一次被传输的字节被称为 cache line。

-----

另外，cache 写到物理内存的位置不是任意的，

我们假定内存为 64K,那么 cache 地址 0 的数值只能和物理内存的地址 0, 16K, 32K 交互；cache 地址 1 的数值只能和物理内存的地址 1, 16K+1, 32K+1 交互

。。。cache 地址 16K-1 的数值只能和物理内存的地址 16K-1, 16K+16K-1, 32K+16K -1 交互

这说明了两点:

(1)假设对象 A 的一个字段长为 16 个字节，如果它放在物理地址 0-15,那么它将和 cache 的第一个 cache line 交互，如果放在物理地址 16-31,那么

如果 CPU 要访问这个字段，必须将第一个和第二个 cache line 都读入，才能获得这个字段的信息，显然这样速度慢，所以一般字段需要 cache line 对齐，在这里就是 16 个字节对齐。

(2)关于 colour

一般一个对象某些字段访问频繁些。

假定一个 cache(这个 cache 指 slab 的 cache,不是上面提到 CPU 的 L1 DATA CACHE)占用 5 个页面也就是 20K。

假定其中对象大小为 32 个字节，前 16 个字节访问频繁许多。

假定对象 A 起始于物理地址 0，对象 C 起始于 31,对象 B 起始于物理地址 16K,那么对象 A，对象 B 的前 16 个字节都和第一个 cache line 交互，后 16 个字节都和第二个 cache line 交互  
对象 C 前 16 个字节与第 3 个 cache 交互。

我们假定内核访问 A 后就访问 B,再访问 A,交错进行,并且前 16 个字节次数都是 50 次,后 16 个为 10 次。C 也是。

这样第一个 cache line 要交互 100 次,第二个 20 次,一共 120 次。

如果让对象 B 向后移动 16 个字节,也就是对象 B 的前 16 个字节与第二个 cache line 交互,后 16 个与第 3 个交互。

那么第一个为 2 次,因为只有开头结尾 2 次要与内存交互,其它每次都在 L1 DATACACHE 中写就可以了。第 2 个 cache line 为 20 次左右(后面的只须在 CACHE 中读写),第 3 个 cache line 为 20 次,

3 个 line 一共才 41 次,你不妨仔细模拟一下。

所以进行错位能降低 CACHE 的交互次数,从而提高 CPU 处理速度能力。

这个错位(也就是上面的 16 个字节)就是 colour.