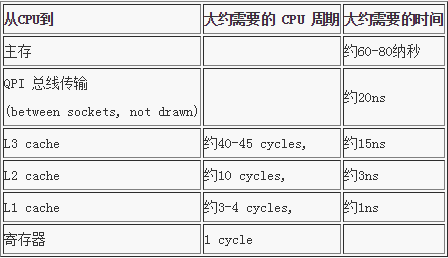
# 1：伪共享

定义：缓存系统中是以缓存行为单位进行储存的，当多线程修改相互独立的变量时，如果这些变量共享一个缓存行，就会无意消耗彼此的性能。



M（修改，Modified）：本地处理器已经修改缓存行，即是脏行，它的内容与内存中的内容不一样，并且此 cache 只有本地一个拷贝(专有)；E（专有，Exclusive）：缓存行内容和内存中的一样，而且其它处理器都没有这行数据；S（共享，Shared）：缓存行内容和内存中的一样, 有可能其它处理器也存在此缓存行的拷贝；I（无效，Invalid）：缓存行失效, 不能使用。

初始：缓存行没有任何数据， I 状态。

本地写（Local Write）：如果 I 状态的缓存行，则缓存行的状态变成 M。

本地读（Local Read）：如果本地处理器读取处于 I 状态的缓存行，很明显此缓存没有数据给它。此时分两种情况：(1)其它处理器的缓存里也没有此行数据，则从内存加载数据到此缓存行后，再将它设成 E 状态，表示只有我一家有这条数据，其它处理器都没有；(2)其它处理器的缓存有此行数据，则将此缓存行的状态设为 S 状态。（备注：如果处于M状态的缓存行，再由本地处理器写入/读出，状态是不会改变的）

远程读（Remote Read）：假设我们有两个处理器 c1 和 c2，如果 c2 需要读另外一个处理器 c1 的缓存行内容，c1 需要把它缓存行的内容通过内存控制器 (Memory Controller) 发送给 c2，c2 接到后将相应的缓存行状态设为 S。在设置之前，内存也得从总线上得到这份数据并保存。

远程写（Remote Write）：其实确切地说不是远程写，而是 c2 得到 c1 的数据后，不是为了读，而是为了写。也算是本地写，只是 c1 也拥有这份数据的拷贝，这该怎么办呢？c2 将发出一个 RFO (Request For Owner) 请求，它需要拥有这行数据的权限，其它处理器的相应缓存行设为 I，除了它自已，谁不能动这行数据。这保证了数据的安全，同时处理 RFO 请求以及设置I的过程将给写操作带来很大的性能消耗。

# 2：指针压缩