

1.

1. 在包含式层次结构中，L2缓存包含L1缓存的所有内容。

先检查L2缓存是否有请求的数据块。

如果L2有，将该数据块加载到L1中。如果要从L1逐出一个数据块，如果逐出的数据块是干净的，可以直接逐出，如果是脏的，将该脏块更新到L2中，因为L2包含了L1所有内容，所以L2不会发生逐出情况。

如果L2没有，从内存中将请求的数据块加载到L2中。如果要从L2逐出一个数据块，如果逐出的数据块是干净的，可以直接逐出，如果是脏的，将脏块写回内存。同时为了满足L2包含L1，如果逐出的数据块也存在L1中，那就将L1中该数据块丢弃或标为无效，仍可满足L2包含L1。再将请求的数据块加载到L1中，如果要从L1逐出一个数据块，如果逐出的数据块是干净的，可以直接逐出，如果是脏的，将脏块写回L2，因为L2包含了L1所有内容，所以L2不会发生逐出情况。

2. 在互斥式层次结构中，L1和L2缓存的内容互不重叠。

先检查L2缓存是否有请求的数据块。

如果L2有，将该数据块加载到L1中，同时移除L2中该数据块。如果要从L1逐出一个数据块，将逐出的数据块加载到L2中，如果是脏的，在L2中标记该数据块为脏。如果L2也需要逐出数据块。如果该数据块为干净的，直接丢弃。如果该数据块为脏的，将其写回内存。

如果L2没有，从内存加载请求的数据块到L1，如果要从L1逐出一个数据块，将逐出的数据块加载到L2中，如果是脏的，在L2中标记该数据块为脏。如果L2也需要逐出数据块。如果该数据块为干净的，直接丢弃。如果该数据块为脏的，将其写回内存。

2.

1. $0.22 + m1 \times 100 < 0.52 + m2 \times 100 \Rightarrow m1 - m2 < (0.52 - 0.22)/100 \Rightarrow m1 - m2 < 0.003$

即当 $m1 - m2 < 0.003$ 时，使用较小的8KB缓存更好。

2. a. 缺失惩罚=10ns时：

条件变为： $m1 - m2 < (0.52 - 0.22)/10 = 0.03$ (3%)

- b. 缺失惩罚=1000ns时：

条件变为： $m1 - m2 < (0.52 - 0.22)/1000 = 0.0003$ (0.03%)

结论：

- 当缺失惩罚较低(如10ns)时，较大的缓存需要提供相对较大的缺失率优势(3%)才能证明其更长的命中时间是合理的。
- 当缺失惩罚很高(如1000ns)时，即使很小的缺失率差异(0.03%)也足以使较大缓存更优。
- 因此，在缺失惩罚很高的情况下，通常倾向于使用较大的缓存以减少缺失率；而在缺失惩罚较低时，较小的缓存可能更优，特别是当较大缓存的缺失率优势不明显时。