

6. 试简要分析为什么缓存一般使用地址中间位作为组索引, 高位作为标签。

解: 使用中间位作为索引的目的是使缓存块更可能占据不同的组, 而采用高位作为索引则可以减少冲突现象, 保证缓存的有效性, 若以相反的方式进行读取, 则同一程序的缓存可能仅占据某些项。

7. 一些真实的缓存系统使组索引和块内偏移总位数与虚拟内存的页偏移相同, 有什么好处。

解: 可以减少计算页内偏移的额外开销, 使用组索引可以直接定位并取出块, 随后可以直接索引块得到数据, 减少了译码的时间。

8. Cache hit, 1 周期, 缓存命中 1 周期, 缓存失 110 周期, 直接访问主存 105 周期的二级存储结构。

解: 1) 使用 L1 缓存, 平均缺失率为 3%, 求平均延时,

$$\text{有 } 1 \times 0.97 + 110 \times 0.03 = 4.27 \text{ 周期。}$$

2) 若继续访问一个 1GB 数组, L1 大小为 64KB, 求平均访问时间,

$$\text{由题意, 不考虑数据局部性, 则有 } \frac{64}{1 \times 10^6} \times 1 + \left(1 - \frac{64}{10^6}\right) \times 110 = 110 \text{ 周期。}$$

3) 局部性对处理器性能,

于局部性较好的程序中, 缓存机制拥有较高的命中率以提供高速而有效的数据,

但于没有有效存储的状态下, 缓存不能有效地提供数据, 因而必须于内存中进行取数, 降低了处理能力。

4) 收益条件,

$$\text{应有 } \alpha \times 1 + 110 \times (1 - \alpha) \leq 105, \text{ 即 } \alpha \geq 4.6\%, \text{ 命中率大于 } 4.6\% \text{ 时使用 L1 更有效。}$$

7.

地址位数	缓存大小	块大小	相联度	组数量	组索引位数	标签位数	偏移位数
32	4	64	2	32	5	20 21	6
32	4	64	8	8	3	23	6
32	4	64	全相联	1	0	26	6
32	16	64	1	256	8	18	6
32	16	128	2	128 64	6	18 19	7
32	64	64	4	64	6	20	6
32	64	64	16	16 64	4 6	20	6
32	64	128	16	8 32	5 5	20	7

0. 较大的缓存有低缺失率及较大的命中延时, 设A使用I1, 8KB直接映射, ~~命中延时0.22ns, 缺失率~~ 命中延时0.22ns, 缺失率 p_1
 B使用64KB四路组相联, 命中延时0.52ns, 缓存缺失为 p_2 .

1) 缺失代价为100ns, 则 p_1, p_2 在什么条件下, A的平均时间优于B.

解. $0.22 \cdot p_1 + 100(1-p_1) \leq 0.52 \cdot p_2 + 100(1-p_2)$

$1.003(1-p_1) \geq (1-p_2)$, 此时A的平均时间优于B.

$p_1 \leq p_2 + 0.003$ 时, A的平均时间优于B.

2) 若命中及缺失代价为命中延时 k 倍, 什么时候A优于B.

解. $0.22 \cdot p_1 + k \cdot 0.22(1-p_1) \leq 0.52 \cdot p_2 + k \cdot 0.52(1-p_2)$.

$p_1 \leq \frac{0.3k}{k-1} \leq p_2$ 时, A优于B.

11. 缓存有16个块,块大小为64字节,访问 $0x1001, 0x1005, 0x1021, 0x1045, 0x1305, 0x2ee5, 0x9f05$.

解 1) 直接映射.

替换 $1+4=5$ 次.

2) 2路组相联.

替换 $0+3=3$ 次.

3) 4路组相联.

替换 $0+3=3$ 次.

4) 8路组相联.

替换 0 次.

12. 增大缓存相联度可降低缺失率,但对特殊程序可能不一样,设块大小16字节,总容量为256字节缓存, A 为2路相联

B 为直接映射,均使用 LRU 替换.

解 1) 2路相联.

访问模式为 $96/4=24$ ~~mod~~, 索引为 $j \bmod 8$

由于块数量为 $256/16=16$ 个,在2路相联下为模8,则当进行第二次循环时, $j=0 \sim 24$ 对应的块已被写出,命中率为 $\frac{3}{4}=75\%$.

2) 直接映射.

块数量为16个,索引式为模十六,则有命中率为 $\frac{12+8}{16}=83\%$.

13. 从利用空间局部性的角度进行优化.

解 $\text{for} (\text{int } j=0; j<128; ++j).$

$\text{for} (\text{int } i=0; i<64; ++i)$

$A[j][i] = A[j][i] + i;$

由于可使程序访问连续的位置,降低提升缓存性能.

14. 假设系统仅在访问A时读写, 初始缓存为空.

1) 4KB的直接映射缓存, 块大小32字节, 优化前后的缓存缺失数为

其将发生 128×64 次缺失.

在优化后, 仅128项缺失, 8×128 次缺失.

2) 4KB全相联结构

其将发生 128×8 次缺失.

在优化后, 亦仅 128×8 次缺失.

3) 若希望除必要缺失外不出现其他缺失, 则优化前需要多大容量.

优化前为不发生缺失, 应使每个缓存块都放置到缓存中, 有 $4 \times 64 \times 128 = 32KB$.

优化后完成计算的块不再使用, 则仅需要256B容量

15. 矢矩阵转置的C语言代码片段, 设系统int为4B, input起始为0x00, output为0x40, 使用32B的L1缓存.

块大小为16B, 直接映射, 写直达, 写分配.

input					output			
行	列1	列2	列3	列4	列0	列1	列2	列3
行0	miss	miss	hit	miss	miss	miss	miss	miss
行1	miss	hit	miss	hit	miss	miss	miss	miss
行2	miss	miss	hit	miss	miss	miss	miss	miss
行3	miss	hit	miss	hit	miss	miss	miss	miss

16. 考虑一特定代码段 input 起初地址为 0x0, 仅对 input 读写时访问缓存.

1) 若存在一个 512 字节大小的缓存, 俩路组相联, 块大小为 16 字节, LRU 策略,

系统采用 6 位进行索引, 由于 2 路组相联, 则每 4 次访问产生一次 miss, 命中率为 75% .

2) 增加缓存大小对命中率的影响.

不影响, 缓存替换并没有对命中率产生影响, 因此增大缓存不改善命中率.

3) 增加缓存块大小.

影响命中率, 在组织方式不改变下, 缓存的命中率为 $\frac{\text{块大小}-1}{\text{块大小}/4}$, 故更大的块大小有命中率.