

6. 这是因为使用地址的中间位作为组索引可以较好地避免缓存抖动。例如, 如果有两个变量, 它们的地址相差一个块大小, 那么如果采用地址高位作索引, 则它们会映射在同一组中。但采用中位, 就会映射在不同组的高速缓存块中。这样就可以同时存放这两个变量, 避免了缓存抖动。

7. 这样做的好处是可以更好地利用虚拟内存系统的页表来管理缓存, 减少缓存系统自己维护页表的开销。

8. 1) 平均延时: $1 \times (1-0.03) + 110 \times 0.03 = 4.27$ 周期

2) 主存足以放下完整数组, 缓存缺失率极高, 因此访问延时接近 105 周期

3) 利用局部性原理时, 缓存命中率高, 从而大大降低平均访问延时, 不利用时, 如 (2), 完全随机时, 平均访问延时接近禁用缓存时直接访问主存的延时, 即 105 周期

4) 禁用时为 105 周期

$$x + 110(1-x) = 110 - 109x < 105 \quad x > \frac{5}{109}, \text{ 才能获得收益}$$

| 9. 组数量 | 组索引位数 Bit | 标签 Bit | 偏移位数 Bit |
|--------|-----------|--------|----------|
| 32 | 5 | 27 | 6 |
| 8 | 3 | 29 | 6 |
| 1 | 0 | 32 | 6 |
| 256 | 8 | 24 | 6 |
| 64 | 6 | 26 | 7 |
| 4256 | 8 | 24 | |
| 16 | 4 | 28 | 6 |
| 8 | 3 | 29 | 7 |

$$10. 1) 0.22(1-P_1) + 100P_1 = 0.22 + 99.78P_1$$

$$< 0.52(1-P_2) + 100P_2 = 0.52 + 99.48P_2$$

$$\Rightarrow P_1 < 0.003 + 0.997P_2 \text{ 时, } A \text{ 优于 } B$$

$$2) 0.22(1-P_1) + 0.22kP_1 < 0.52(1-P_2) + 0.52kP_2 \Rightarrow P_1 < \frac{15}{11(k+1)} + \frac{26}{11}P_2$$

11. 直接映射: $1+1+3=5$ 次

2路组相联: $0+0+2=2$ 次

4路: $0+0+3=3$ 次

8路: $0+0+0=0$ 次

$$12. \text{缓存A: } \frac{3+2 \times 99}{3 \times 100} = 67\% \quad 67\% \times \frac{1}{4} = 16.75\%$$

$$\text{缓存B: } 96/16 \times 96 = \frac{1}{100} = 1\%$$

13. for (int j=0; j<128; ++j)

{ for (int i=0; i<64; ++i)

{ A[j][i] = A[i] + j }

}

14. 1) ~~$\frac{4KB}{32B} = 128$~~ 优化前: $64 \times 128 = 8192$ 次

后: $(64/8) \times 128 = 1024$ 次

2) $\frac{4KB}{32B} = 128$ 优化前后都为 8192 次缺失

3) A: $128 \times 64 = 8192$ 占 32KB $\frac{32KB}{32B} = 1024$ 个块

\therefore 优化前需 $1024 \times 32B = 32KB$

优化后, 数组A每行有64个int类型的元素, 占256字节

$\frac{256}{32} = 8$ 个块, \therefore 优化后 $8 \times 32B = 256B$

| 15. | output | | | | input | | | |
|-----|--------|-----|------|-----|-------|------|------|------|
| | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| 0 | miss | hit | miss | hit | miss | miss | miss | miss |
| 1 | miss | hit | miss | hit | hit | hit | hit | hit |
| 2 | miss | hit | miss | hit | hit | hit | hit | hit |
| 3 | miss | hit | miss | hit | hit | hit | hit | hit |

16.11) $512 \div 16 = 32$ $16 \div 4 = 4$ $32 \times 4 = 128$

$(4-1) \div 4 = 75\%$

12) 可以, 因为增加缓存大小可以增加缓存能够容纳的块数, 从而降低替换发生的概率

13) 不能, 增加块大小并不能降低替换发生的概率, 在此题中每次只会访问两个相邻的地址, 同属于同一块。