

第13周作业

6 1 标签的位数取决于缓存的大小和容量，作为标签的高位较长，可以增加标签的比特数，提高缓存的准确度 2 地址的中间位作为组索引可以保证内存中的数据可以被尽可能地分散存储在缓存的各个组中，避免不同的内存地址映射到同一个组中的情况，减少缓存命中率的下降，使地址的组索引部分与块偏移部分相距，提高访问效率

7 1 实现简单效率高：可以使缓存系统中的地址映射能够直接使用或者与页的映射相对应
2 可以共享地址索引：减少缓存内部开销，减少缓存的复杂性并提高缓存性能
3 转换效率高：如果管理物理块的数据结构使用了哈希表或其他复杂的数据结构，



扫描全能王 创建

会降低缓存访问耗时，提高执行性能，4更好的操作系统支持

8 (1) $97\% \times 1 + 3\% \times 110 = 4.27$ 周期

(2) $1GB = 2^{30}$ 字节 访问 4个字节 $64KB = 2^{16}$ 字节 可放 2^{14} 个缓存块
访(1) 次命中率 $= \frac{2^{14}}{2^{30}} = 2^{-16}$ $T = 2^{-16} \times 1 + (1 - 2^{-16}) \times 110 \approx 110$ 周期

(3) 在(GB随机访问数据情况下，缓存命中率极低，平均延时大幅增加，而在局部性较好的情况下，处理器可以利用访问过的数据和指令的空间局部性来预先缓存数据和指令

(4) 设命中率为 $a\%$ $a\% \times 1 + (1 - a\%) \times 110 = 105$ $a\% = 4.59\%$

平均命中率至少达到4.59%

$1GB = 2^{30}$ Byte

	Bit	kB	Byte	组数	Bit							
1	32	4	64	2	32	5	21	6				64KB
2	32	4	64	8	8	3	23	6				1
3	1	4	64	全相联	1	0	21	6				1
4	1	16	64	(直接映射)	256	8	18	6				256KB
5	1	16	128	2	64	6	19	7				128KB
6	1	64	64	4	256	8	18	6				1024KB
7	1	64	64	16	64	6	20	6				!
8	1	64	128	16	32	5	23	7				512KB

(0) (1) $(1-p_1) \times 0.22 + p_1 \times (0.22 + 100) < 0.52 \times (-p_2) + p_2 \times (0.52 + 100)$

$p_1 - p_2 < 0.3\%$

(2) $(1-p_1) \times 0.22 + p_1 \times 0.22 \times 100 < (1-p_2) \times 0.52 + p_2 \times 0.52 \times 100$



$$0.22p_1 - 0.52p_2 < \frac{0.3}{k-1}$$

- 11 $N=16$ 直接相互联时地址 mod 16 相同会发生替换 $n=7-2=5$ 次
 2路组相联 mod 8 相同且组内均填满会替换 $n=7-2=3$ 次
 4路组相联 $n=5-4=1$ 次 8路组相联 不发生替换

```
13 for (int j=0; j<128; ++j) {
    for (int i=0; i<64; ++i) {
        A[j][i] = A[j][i]+1;
    }
}
```

14 (1) $4KB = 2^{12}$ 字节 每个缓存块缓存一列元素 $n = \frac{3^2}{4} = 8$ 个
 内存访问 $128 \times 64 = 2^{13}$ 次 缓存块数 $N = \frac{2^{13}}{32} = 2^7 = 128$ 个
 优化前 $64 \times 128 = 8192$ 次
 优化后 $64 \times 128 \div 8 = 1024$ 次

(2) 优化前 $P = \frac{1}{8} \cdot N = (28 \times 64 \times \frac{1}{8}) = 1024$ 次
 优化后 $P = \frac{1}{8} \cdot N = (28 \times 64 \times \frac{1}{8}) = 1024$ 次

(3) 优化后 直接映射缓存每次均为强制缺失一块即 32 字节
 优化前 要将 $A_{10} \sim A_{127}$ 块放在缓存中 要 $8 \times 128 = 1024$ 块
 $1024 \times 32 = 32768$ 字节



12 块数 = 16 $n = \frac{16}{4} = 4$ 个 内存共 24 块 缓存 16 块

2 级组块映射时 每轮替换 2 次 $P = \frac{2^4}{16} = 25\%$

直接相联时 除了第一次，之后的 array [0 ~ 31] 均重新覆盖 缓存 64~95 位
置，成功率为 75%，综合成功率 $\frac{1}{3} \times 75\% + \frac{1}{3} \times 100\% + \frac{1}{3} \times 75\% = 83\%$
 $\frac{31}{32} \times (1 - 0.83) + \frac{1}{32} \times 0.25 = 0.169 \approx 16.9\% < 25\%$

15 列 1 | 列 2 | 列 3 | 列 4 | 列 5 | 列 6 | 列 7 | 列 8 | 列 9 | 列 10 | 列 11 | 列 12 | 列 13 | 列 14 | 列 15 | 列 16 | 列 17 | 列 18 | 列 19 | 列 20 | 列 21 | 列 22 | 列 23 | 列 24 | 列 25 | 列 26 | 列 27 | 列 28 | 列 29 | 列 30 | 列 31

行 0 miss miss

行 1 ✓ ✓ ✓ ✓ miss miss miss miss miss miss

行 2 ✓ ✓ ✓ ✓ miss miss miss miss miss miss

行 3 ✓ ✓ ✓ ✓ miss miss miss miss miss miss

一块有 4 个字，共有 2 个块

16 (1) 一块 4 个字，共 32 块， $P = \frac{4-1}{4} = 75\%$

(2) 不行，增加总大小只会改变缓存总块数，对于单次命中是否不产生影响

(3) 可以，增大块大小可以增加一次命中后替换缓存的量，又由于访问数组的地址是相连的，因此可以增加命中率



扫描全能王 创建