

第13周 5.16

(第5题见上次练习)

6. 因为使用高位作索引位时, 相邻地址的内存块会被映射到缓存的同一组。若按顺序访问内存中的块, 则缓存将长时间只存一个块的大小, 使用效率很低。

7. ① 简化地址转换: 对虚拟内存系统, 将地址空间分成了大小相同的页面, 因此可以使用相同方式处理缓存地址, 使得缓存地址的计算更加直观、简单。② 减少地址冲突: ~~将地址的组索引~~ 使地址空间的划分更加均匀, 降低地址冲突的概率。③ 提高缓存的性能: 使缓存的访问更加高效。实现缓存时, 可以利用虚拟内存系统的页表来优化缓存的访问, 从而提高缓存的性能。④ 简化硬件实现: 可以~~采用~~<sup>共用</sup>相同的硬件模块来处理虚拟内存和缓存, 从而降低了硬件实现的复杂度。

$$8. (1) T_1 = \frac{(1-m)t}{M \times P} = 4.27 \text{ 周期}$$

$$(2) R = \frac{64}{1024 \times 1024} = 0.00006 \quad T_2 = Rt + (1-R) \times P \approx 110 \text{ 周期}$$

(3) 由于局部性原理的存在, 即~~处理器~~空间局部性和时间局部性, ~~因此~~尽管缓存 L1 大小远小于内存, 但其所存储的内容命中率较高 (C1) 题结果; 但在处理完全数据随机访问性质的程序时, 局部性原理失效, 命中率从题 (1) 中的 97% 下降至不到 1%, 缓存的作用也大幅下降, 也就是说, 当局部性原理失效时, 缓存的命中率相应提高, 处理器访存性能也提高。

$$(4) T_{CR} = Rt + (1-R)P = 110 - 109R < 105 \quad \therefore R > \frac{5}{109} \approx 0.04587$$

$\therefore$  当平均缓存命中率大于 4.587% 时, 才能有性能收益。



9.	Bit 地址位数	kB 缓存大小	Byte 块大小	相联度	组数量	Bit 组索引位数	Bit 相联位数	Bit 偏移位数
1	32	4	64	2	$(2^{11})2048$	11	15	6
2	32	4	64	8	$(2^9)512$	9	17	6
3	32	4	64	全	1	0	26	6
4	32	16	64	1	$(2^{14})16384$	14	12	6
5	32	16	128	2	$(2^{13})8192$	13	12	7
6	32	64	64	4	$(2^{14})16384$	14	12	6
7	32	64	64	16	$(2^{12})4096$	12	14	6
8	32	64	128	16	$(2^{12})4096$	12	13	7

$$10. (1) T_A = t_A(1-P_1) + t_P P_1 = 0.22 + 99.78 P_1$$

$$T_B = t_B(1-P_2) + t_P P_2 = 0.52 + 99.48 P_2$$

$$\because T_A < T_B \quad \therefore 99.78 P_1 < 0.3 + 99.48 P_2 \quad \text{即为 } (P_1 - P_2) < 0.003$$

$$(2) T_A' = t_A(1-P_1) + K t_A P_1 = 0.22 [1 + (K-1)P_1]$$

$$T_B' = t_B(1-P_2) + K t_B P_2 = 0.52 [1 + (K-1)P_2]$$

$$\because T_A' < T_B' \quad \therefore 11 [1 + (K-1)P_1] < 26 [1 + (K-1)P_2]$$

11. 直接映射:  $N=7 < 16$   $\therefore$  替换0次

~~这7个数~~ 这7个数换算为10进制分别为 4097, 4101, 4129, 4165, 4869, 12005, 65285

~~模1024分别为 1, 5, 33, 69, 773, 741, 773~~

2路组相联:  $\frac{16}{2}=8$  故这7个数模8分别为 1, 5, 1, 5, 5, 5, 5 (5-2) 故替换3次

4路组相联:  $\frac{16}{4}=4$  4分别为 1, 1, 1, 1, 1, 1, 1 (7-4) 故替换3次

8路组相联:  $\frac{16}{8}=2$  2分别为 1, 1, 1, 1, 1, 1, 1 (7-8<0) 故替换0次





12. A. 共  $\frac{256}{16 \times 2} = 8$  组 每个块存储  $\frac{16}{4} = 4$  个整型

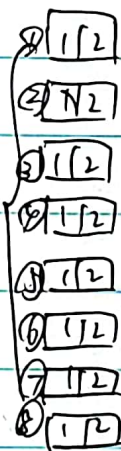
$\therefore$  共需  $\frac{96}{4} = 24$  个块  $24 \div 8 = 3 \dots 0$

$\therefore$  分为前32 中32 后32 考虑

对每个块 缺失率为  $\frac{1}{4} = 25\%$

而每个块数据存储逻辑相同

$\therefore$  缺失率为  $25\%$



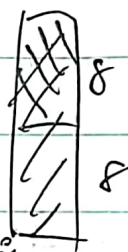
B. 共  $\frac{256}{16} = 16$  组 每个块4个整型

$24 \div 16 = 1 \dots 8$

每一次  $i$  改变时缓存中剩下

块能命中

$\therefore \frac{2}{3} \times 25\% + \frac{1}{3} \times \frac{100\%}{100} = 16.75\%$   
缺失率



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

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

$A[j][i] = A[j][i] + 1;$

}

}

+

14. (1) 共  $\frac{6 \times 1024}{32} = 128$  组

优化前每次  $i, j$  改变均缺失  $\therefore$  缺失  $64 \times 128 = 8192$  次

一个块存储  $\frac{32}{4} = 8$  个整型

优化后  $i, j$  改变时 每缺失1次命中7次  $\therefore$  缺失  $\frac{64 \times 128}{8} = 1024$  次

(2) 共1组128块 共需  $\frac{6 \times 128}{8} = 1024$  块

对优化后无影响 仍为1024次, 优化前在缺失64次后命中  $64 \times 7$  次, 故缺失  $\frac{8192}{8} = 1024$  次

(3) 优化后需  $8 \times 128$  个块 故缓存大小为  $8 \times 128 \times 32^{bit} = 32$  KB

优化前需 优化后为128倍 共 9MB



15.

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

16. (1) 一个块共  $\frac{16}{4} = 4$  个整型 共  $\frac{512}{64} = 8$  个组 需  $\frac{2 \times 128}{4} = 64$  个块

∴ 命中率为  $1 - \frac{1}{4} = 75\%$

(2) 增加缓存的总大小 不可以改善命中率。因为每个块都存储 4 个 int 变量的值，意味着在每次缓存缺失的下三次都将命中，而缓存总大小时此没有影响。

(3) 增加缓存的块大小可以改善命中率。因为增加块大小使每个块中的 int 变量增加，缓存缺失后的命中数增加，命中率增加。

