

6.缓存使用地址的中间位作为组索引、高位作为标签的主要原因是为了避免冲突。如果使用高位作为组索引,由于高位的取值范围较小,容易导致多个不同的地址被映射到同一个组中,造成冲突。

而使用中间位作为组索引,则可以保证相邻的内存地址映射到不同的组中,从而减少冲突的概率。同时,由于组索引使用的是中间位,因此标签需要使用高位,才能保证标签的唯一性。

因此,缓存使用地址的中间位作为组索引、高位作为标签的设计方案,可以平衡冲突和标签唯一性的问题,从而获得较高的缓存命中率。

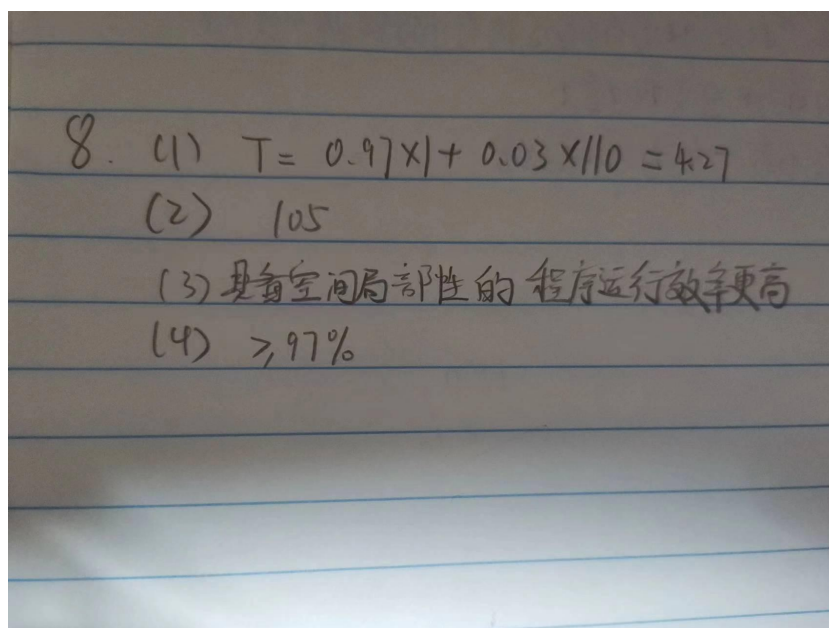
7.将缓存系统的组索引和快内偏移的总位数与虚拟内存系统的页偏移数相同,可以使缓存系统的管理与虚拟内存管理保持一致,从而提高系统的整体性能和稳定性。具体好处包括:

1. 相同的位数可以方便地进行地址转换。由于虚拟内存系统和缓存系统都需要进行地址转换,如果它们的位数不同,就需要进行复杂的位数转换,增加了复杂性和开销。而将它们的位数设置相同,可以使内存地址在缓存系统和虚拟内存系统之间进行转换时更加简单高效。

2. 可以避免缓存与页面的碎片化问题。由于虚拟内存系统和缓存系统的页大小一般是相同的,将组索引和快内偏移的总位数与虚拟内存系统的页偏移数相同,可以保证在虚拟内存系统中相邻的页在缓存系统中也是相邻的,从而避免了缓存与页面的碎片化问题,提高了缓存的利用率和性能。

3. 可以统一管理虚拟内存和缓存的数据。虚拟内存系统和缓存系统都是为了提高数据访问的效率和速度而设计的,将它们的管理进行统一可以更好地控制数据的存储和访问,避免了数据的重复存储和管理,提高了系统的整体性能和稳定性。

因此,将缓存系统的组索引和快内偏移的总位数与虚拟内存系统的页偏移数相同,是一种优良的设计策略,能够提高系统的整体性能和稳定性。



9.

组数 组中值 标准差 偏度

1	32	5	21	6
2	8	3	23	6
3	1	0	26	6
4	256	8	18	6
5	64	6	19	7
6	256	18	18	6
7	64	6	20	6
8	32	5	20	7

$$(0. (1) T_1 = 0.22(1-p_1) + 100p_1 = 0.22 + 99.78p_1$$

$$(2) T_2 = 0.52(1-p_1) + 100p_2 = 0.52 + 99.48p_2$$

$$T_1 < T_2 \Rightarrow p_1 < 0.003 + 0.997p_2$$

A 平均访问时间优于系统 B

$$(2) AMAT_1 = 0.22(1-p_1) + 0.22kp_1$$

$$AMAT_2 = 0.52(1-p_2) + 0.52kp_2$$

$$AMAT_1 < AMAT_2 \Rightarrow p_1 < \frac{15}{11(k+1)} + \frac{26}{11}p_2$$

11. 直接映射: 0x1001 0x1021 替换1次

0x1005, 0x1045, ~~0x1035~~ 0x1305, 0x2005, 0x1045

替换4次, 共5次

2路组相联: 前2个不替换

后5个替换3次, 共3次

4路组相联: 共4组, 替换3次

8路组相联: 共2组, 不替换

12. A: array[0]~[3], [4]~[7] ... [92]~[95]

每个数组元素存入一个缓存块中

其中 array[0]~[3], ... [64]~[67]

array[28]~[31], ... [92]~[95] - 映射到同一块

存入 [64]~[95] 时, [0]~[31] 会被刷新

[32]~[63] 不变

∴ ① [0]~[31] 缺失, ② [32]~[63] 命中 ③ [64]~[95]

缺失率  $\frac{2}{3}$

缺失

B: [0]~[95] 共 8 组

[0]~[3], [32]~[35], [64]~[67] 映射到同一个缓存块

j 循环: ① 有入 0~3 ② 有入 32~35 ~~③ 有入~~

③ 刷掉 0~3, 有入 64~67

i 循环: ① 未命中 0~3, 有入 0~3 刷掉 32~35

② 未命中 32~35, 同上

缺失率为 1

13. 优化: 
$$\begin{aligned} & \text{for}(\text{int } i=0; i < \overset{64}{80}; ++i) \\ & \{ \text{for}(\text{int } j=0; j < \overset{128}{80}; ++j) \\ & \quad \{ A[i][j] = A[i][j] + 1; \\ & \quad \} \\ & \} \end{aligned}$$

缓存共 128 块

14. (1) 优化前:  $A[j][i]$  与  $A[j+1][i]$  并不相邻

每次访问时: 将从内存中索要数据.  $\therefore$  j 循环均是如此

共  $127 \times 63 = 8001$  次

优化后:  $A[i][j]$  与  $A[i][j+1]$  相邻

j 循环不缺失, i 循环开始时将缺失一次 共 63 次

(2) 优化前: 127 次

优化后: 63 次

(3) 优化前:  $4 \times 64 = 256 \text{ KB}$

优化后: 4KB

15.

	input				Output			
	0	1	2	3	0	1	2	3
0	miss	中	中	中	miss	miss	miss	miss
1	miss	中	中	中	中	中	中	中
2	miss	中	中	中	中	中	中	中
3	miss	中	中	中	中	中	中	中

缓存共有2路16组

16 (1) 每访问16次, 将会缺失一次 命中率  $\frac{15}{16} = 93.75\%$ (2) 可以。缓存总大小增大后, 块内有存储的内存中的数据个数增多。  
每两次缺失之间的访问次数增大

(3) 不可以。每块只存储一个内存数据。