

4.

(1) PMP 中 X/W/R 作为保护机制, 相当于二次保险: 即便页表中表示有某-权限, 但 PMP 中表示没有该权限, 硬件仍然无法执行该操作。

(2) L: 表示是否锁定了 PMP 和对应的地址寄存器, 锁定后则无法更改。

A: 表示了是否启用此 PMP

$$(1) \frac{4 \times 1000}{8} = 500$$

每项总存储 4 个地址

$$\therefore \frac{2^{64}}{50} \times 4 \div \frac{1}{1000} \div \frac{1}{1000} \div \frac{1}{1000} = 147574 \text{ PB}$$

$$(2) \frac{2^{48}}{50} \times 4 \div \frac{1}{1000} \div \frac{1}{1000} \div \frac{1}{1000} = 2251.8 \text{ GB}$$

所需空间被降到 2251.8 GB

(3) 因为多级页表只将必要的页表加载到内存中, 省去了大量未映射的页表项, 所以大大减少了存储空间。



6.

因为地址的高位通常区别较小,会导致多个地址被分配到组中的同一位置,增大了竞争的可能,不仅无法较好的利用全部缓存空间,还会增大缓存判断数据竞争的开销,降低性能。

7.

这样可以使缓存的地址映射与虚拟内存系统的地址映射保持一致,不仅可以使使用同一套硬件、减少硬件开发开销,还可以使虚拟地址和缓存互通、相互进行地址转换时更加方便。

8.

$$(1) \bar{T} = (1 \times 97\% + 110 \times 3\%) \times T_{\text{cycle}} = 4.27 T_{\text{cycle}}$$

$$(2) \text{命中率} = \frac{64KB}{1GB} = \frac{1}{16384}$$

$$\therefore \bar{T} = \left[\frac{1}{16384} \times 1 + 110 \times \left(1 - \frac{1}{16384} \right) \right] T_{\text{cycle}} \approx 110 T_{\text{cycle}}$$

(3). 是因为内存访问有空间和时间上的局部性,导致了程序并不是随机地访问数据,所以在缓存容量很小的情况下也会有较高的命中率,而不是完全随机时均等于0的命中率。所以在高命中率时即可有效减少访问时间。

$$(4) 1 \times \alpha\% + 110 \times (1 - \alpha\%) < 105$$

$$\text{解得 } \alpha\% > 4.59\%$$



9.

编号	地址位数	缓存大小	块大小	块位数	组数量	组索引位数	标志位数	偏移位数
1	32	4	64	2	32	5	21	6
2	32	4	64	8	8	3	23	6
3	32	4	64	全关联	1	0	26	6
4	32	16	64	1	256	8	18	6
5	32	16	128	2	64	6	19	7.5
6	32	64	64	4	256	8	18	6
7	32	64	64	16	64	6	20	6
8	32	64	128	16	32	5	22	7.5

10.

$$(1) t_A = (1-p_1) \times 0.22 + p_1 \times 100 = 0.97 \times (102 \times 0.11 + 100 \times 0.89) = 79.78$$

$$t_B = (1-p_2) \times 0.52 + p_2 \times 100$$

$$t_A < t_B$$

$$\text{解得 } 99.78 p_1 < 0.3 + 99.38 p_2 \times \frac{1}{100} \Rightarrow p_1 < 0.3 + 0.9938 p_2$$

$$(2) t_A = (1-p_1) \times 0.22 + p_1 \times k \times 0.52$$

$$t_B = (1-p_2) \times 0.52 + p_2 \times k \times 0.52$$

$$t_A < t_B$$

$$\text{解得 } 22 p_1 < \frac{70}{k-1} + 52 p_2$$

$$201 = (100-1) \times 0.11 + 100 \times 0.89$$

$$100 p_2 < 100 \text{ 显然}$$



11.

列出 = 地址

0x001: 0001 0000 0000 0001

0x003: 0001 0000 0000 0101

0x005: 0001 0000 0000 1001

0x007: 0001 0000 0000 1101

0x009: 0001 0000 0001 0101

0x00B: 0001 0000 0010 0101

0x00D: 0001 0000 0011 0101

① 直接映射: $\text{index} = 4$ 位

\therefore 替换 5 次

② 2 路: $\text{index} = 3$ 位, 有 2 个选择

\therefore 替换 3 次

③ 4 路: $\text{index} = 2$ 位, 有 4 个选择

\therefore 替换 1 次

④ 8 路: $\text{index} = 1$ 位

\therefore 替换 0 次



12.

$$\text{块数} = \frac{256}{16} = 16$$

分析程序：会进行 96×100 次循环，每次都会读取 array 首地址，且根据 fisc-v 编译原则，每次都会将 i, j 存入栈中，所以也需要读取 i, j

① B：直接映射

每次 j 都不同，并且 $96 > 16$ ，所以每次 j 都会 miss

首地址永远相同， i 在 96 个循环内相同

→ 第一次：3 个 miss

第二次：1 个 miss

:

第四次：1 个 miss，缓存已满

第五次：1 个 miss，array 被替换

第六次：array miss， i 被替换为 array， i 也 miss， j miss，三次

第七次：1 个 miss

:

第九次：与十次相同，三次 miss， $i=0$

第十次：array hit，但 i miss，二次 miss

:

第十次：一个 miss，array 被替换

第十一次：array miss，但 array 放在了 $i=0$ 处， $i=1$ 仍 hit，二次 miss

第十二次：array hit， i hit，但读新 j 替换，一次 miss

第十三次：array hit， i miss， j miss，二次 miss

:

第十四次：与第十三次相同

第十五次：array hit， i miss，读入 $i=2$ ，两次 miss

:



所以：整个过程可看作 50 次循环。

第一次 miss 2,9 次 (包含初始化 array)。

剩下 49 次 miss 2,8 次

∴ 共 miss 109 次。

共访问缓存 $3 \times 96 \times 100$ 次

∴ B 的缺失率为 37.5%

地址范围：0-95

同时内存中也有 2 个 100 字节的数据

访问 1 后，从 1 开始

访问 2 后，从 2 开始

访问 3 后，从 3 开始

访问 4 后，从 4 开始

访问 5 后，从 5 开始

访问 6 后，从 6 开始

访问 7 后，从 7 开始

访问 8 后，从 8 开始

访问 9 后，从 9 开始

访问 10 后，从 10 开始

访问 11 后，从 11 开始

访问 12 后，从 12 开始

访问 13 后，从 13 开始

访问 14 后，从 14 开始



13.

修改为

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

14. 块数 = $\frac{4 \times 128}{32} = 128$

一个 int 4 byte

一个块中可以存 4 个 int

(1) 优化前. 在 j 的 128 次循环中. 每 4 次会 miss 1 次. 共 $32 \times 64 = 2048$ 次.

优化后. 在 j 的 128 次循环中. 每次 miss 1 次. 共 miss 32 次.

(2) 优化前. 在 i 的 64 次循环中. 每 4 次有一次 128 个 j 会 miss. 共 $128 \times 16 = 2048$ 次.

优化后. 在 j 的 128 次循环中. i 每 4 次 miss 1 次. 共 $128 \times 16 = 2048$ 次.

(3). 优化前优化后都需要. $128 \times 64 \times 4 = 1024 = 32 \text{ KB}$



15.

	input				out put			
	3 0	3 1	3 2	3 3	3 0	3 1	3 2	3 3
ij0	miss	hit	hit	hit	miss	miss	miss	miss
ij1	hit	hit	hit	hit	hit	hit	hit	hit
ij2	miss	hit	hit	hit	miss	miss	miss	miss
ij3	hit	hit	hit	hit	hit	hit	hit	hit

16. 块数 = $\frac{512}{16} = 32$, 一个块 4个 int

(1) 在 128 x 2 次访问中, 只会 miss 2次

∴ 命中率为 99.22%

(2) 可以, 因为会增加每一路中的块数

当一路中的块数多过 32 块时, 则命中率为 100%

(3) 不会, 因为总大小不变时, 每一路同样只能存 64 个 int,

所以无法改变 2 次 miss.

