

6. ① 使用中间位值作为索引可以更好地利用缓存空间，增加数据的局部性。

② 有助于减少冲突，较低的位数变化较小。

③ 提供更大的标签空间，提高缓存的质量。

7. ① 可以更好地与虚拟内存系统硬件支持集成。

② 缓存可以直接将虚拟地址的页偏移作为块偏移。

③ 简化缓存系统的管理。

④ 提高访问效率和速度。

$$8. (1) 3\% \times 110 + 97\% \times 1 = 4.27 \text{ 周期}$$

(2) 1GB > 64KB

约为 110 周期。

(3) 对于 (1) 平均缓存缺失率为 3%，有一定局部性。

但是对于 (2)，缓存几乎没有缺失，因此平均延时较高。

$$(4) \cancel{x \times 110 + (1-x)}$$

$$x \times 1 + (1-x) \times 110 < 105$$

$$\text{故 } x > 4.587\%$$

9. 编号 地址位数 缓存大小 块大小 相联度 组数量 组索引位数 标签位数 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 | 1 | 0 | 26 | 6 |
| 4 | 32 | 16 | 64 | 1 | 256 | 8 | 18 | 6 |
| 5 | 32 | 16 | 128 | 2 | 64 | 6 | 19 | 7 |
| 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 | 20 | 7 |

$$10. 11 \quad 0.22 + 100P_1 < 0.52 + 100P_2$$

$$\text{故 } P_1 - P_2 < 0.003 = 0.3\%$$

$$21 \quad 0.22 + k \cdot 0.22 \cdot P_1 < 0.52 + k \cdot 0.52 \cdot P_2$$

$$\text{故 } 11kP_1 < 15 + 26kP_2$$

11. ~~直接映射~~ 索引位为4位

$$0x1001 \bmod 0x000F \text{ 余 } 0x0001 \quad - \text{miss}$$

$$0x1005 \bmod 0x000F \text{ 余 } 0x0005 \quad - \text{miss}$$

$$0x1021 \bmod 0x000F \text{ 余 } 0x0001 \quad - 1$$

$$0x1045 \bmod 0x000F \text{ 余 } 0x0005 \quad - 2$$

$$0x1305 \bmod 0x000F \text{ 余 } 0x0005 \quad - 3$$

$$0x2ee5 \bmod 0x000F \text{ 余 } 0x0005 \quad - 4$$

$$0xff05 \bmod 0x000F \text{ 余 } 0x0005 \quad - 5$$

5次块替换。

2路组相联，索引位为3位。

$$0x1001 \bmod 0x0007 \text{ 余 } 0x0001 \quad - \text{miss}$$

$$0x1005 \bmod 0x0007 \text{ 余 } 0x0005 \quad - \text{miss}$$

$$0x1021 \bmod 0x0007 \text{ 余 } 0x0001 \quad - \text{miss}$$

$$0x1045 \bmod 0x0007 \text{ 余 } 0x0005 \quad - \text{miss}$$

$$0x1305 \bmod 0x0007 \text{ 余 } 0x0005 \quad - 1$$

$$0x2ee5 \bmod 0x0007 \text{ 余 } 0x0005 \quad - 2$$

$$0xff05 \bmod 0x0007 \text{ 余 } 0x0005 \quad - 3$$

3次块替换。

千路组相联，索引位为2位。

Date. Page.

0x1001 余 0x0001 , 0x1005 余 0x0001

~~0x1021 余 0x0001, 0x1045 余 0x0001~~

全部余 0x0001

3次块替换

8 路程相同时没有块替换.

12. 缓存由 8 个组，每个组有 2 路。

簇数B为16⁴组，每⁴组有1路。

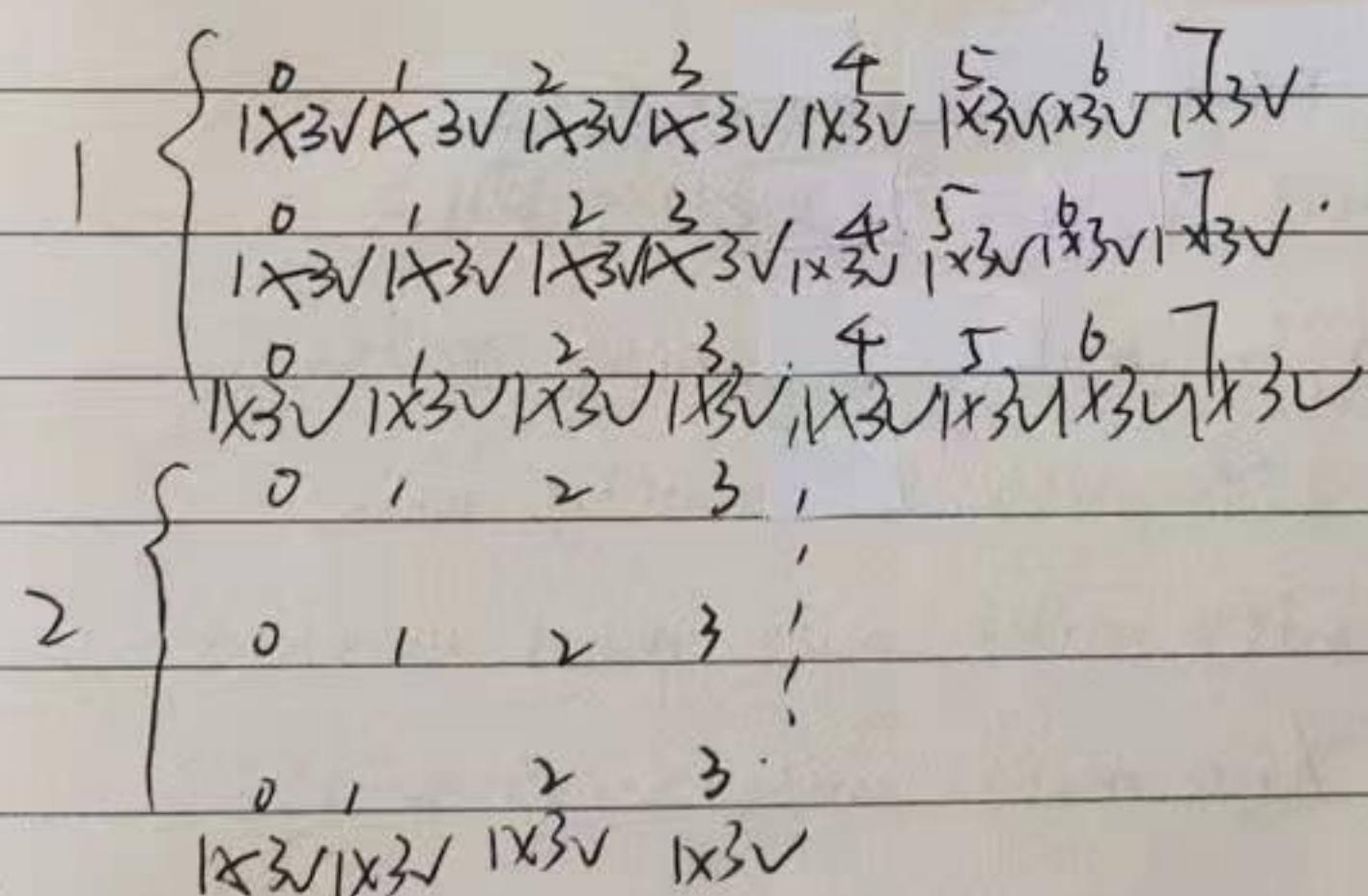
一其網因存倉共 $100 \times 96 = 9600$ 人

对于~~修改稿~~，循环调用了 100 次完整数组。

数组 array 可分为²⁴ 个块。

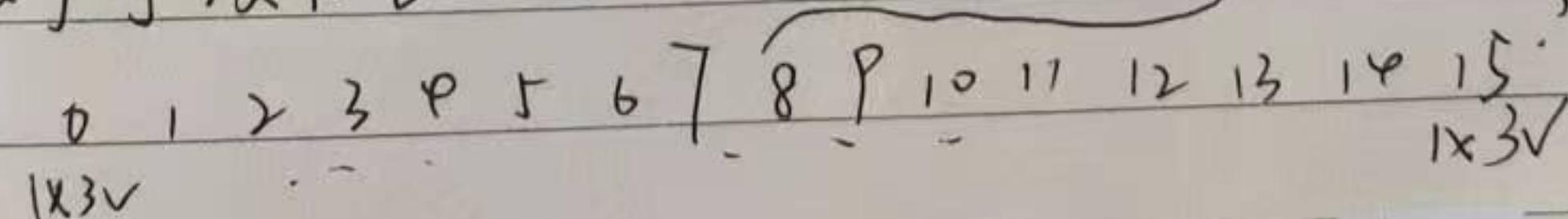
对于缓存 A.

~~缺失你數為~~

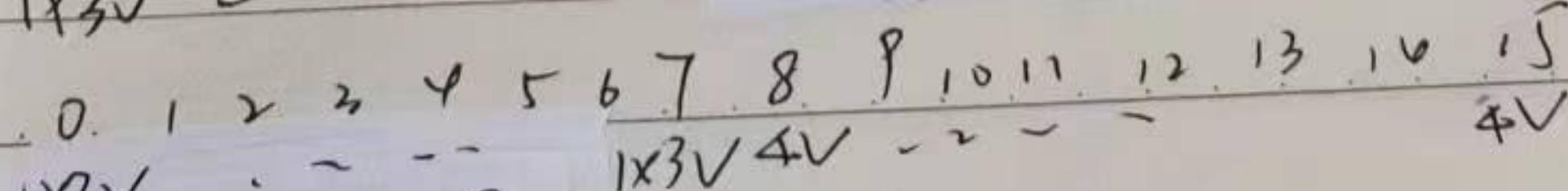


∴ $\frac{1}{2} \times 25\%$

对于缓存 B



0 1 2 3 4 5
1x3v - - - -



$$\therefore \text{Ans} \frac{24 + 16 \times 99}{9600} = 16.75\%$$

```

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

```

14. 11 一个块可装 8 个数。

缓存可装 128 块，128 组

访问共 64×128 次。

缺失次数 { 优化前: $64 \times 128 = 8192$ 次。 }

优化后: $64 \times 128 \div 8 = 1024$ 次。

2) 缺失次数 { 优化前: $64 \times 128 \div 8 = 1024$ 次 }

优化后: $64 \times 128 \div 8 = 1024$ 次。

3) 优化前需要 32KB

优化后需要 4KB。
input 数组 output 数组。

15. $3n_0 \quad 3n_1 \quad 3n_2 \quad 3n_3 \quad 3n_0 \quad 3n_1 \quad 3n_2 \quad 3n_3$

行 0 miss miss hit miss miss miss miss

行 1 miss hit miss hit miss miss miss miss

行 2 miss miss hit miss miss miss miss miss

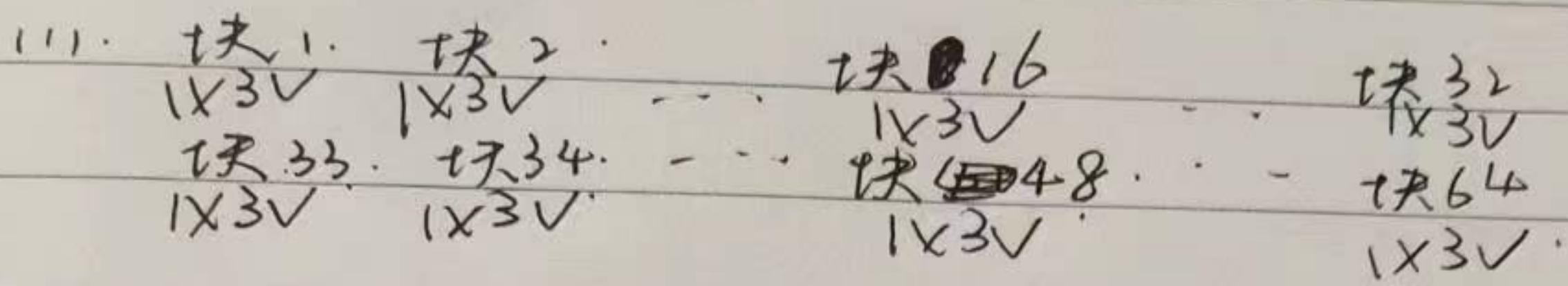
行 3 miss hit miss hit miss miss miss miss

一个缓存可放 2 块，2 组，一组一路。

一个块可以放 4 个数据。

16. 硬盘有 16 组，每组 2 路。

数组 input 2 行，每行 128 个数，分为 32 个块



命中率为 75%。

(2) 增加硬盘大小，增加组数，不能

命中率依旧为 75%。

(3) 增加块大小，增加了每个块容纳的数据量
可以增加命中率。