

姓名: 李雅帆

学号: 2213041

5.8

5.8.1 地址以字地址形式给出, 每个32位块包含4个字, 每4次访问为一次缺失, 即缺失率为 $\frac{1}{4}$ 。

缓存大小和工作集的大小会影响缺失率, 增大缓存大小会使更多的数据能够被缓存, 从而减小缺失率; 增大工作集大小, 更多的数据需要在缓存中存储, 可能会增加缺失率。

根据3C模型分类为: 容量缺失、冲突缺失和替换缺失

5.8.2 16字节: $\frac{1}{2}$

64字节: $\frac{1}{8}$

128字节: $\frac{1}{16}$

5.8.3 0%

5.18

5.18.1 $\frac{2^{43}}{2^{12}} = 2^{31}$ $2^{31} \times 4 = 2^{33} = 8 \text{ GB}$

5.18.2 至少2级; n级页表访问n次

5.18.3 是足够大的

每个表项的有效到达量为 $2^{31} \times 4 \text{ KB} = 8 \text{ TB}$, 足以覆盖机器的物理地址空间

5.18.4 每个页表项包含 $\frac{4 \text{ KB}}{4 \text{ B}} = 1024$ $\log_2(1024) = 10$
使用13位虚拟地址和4KB页面

$\frac{43-12}{10} = 4$ (级). 需要4级页表

5.18.5 在反向页表中, PTE的数量可以减少到哈希表的大小加上冲突的代价; TLB缺失时, 需要额外的引用来比较哈希表中存储的一个或多个标签。

5.23. 被模拟的系统通常比在本地 ISA 上运行得更慢。
 仿真过程包括指令重编译和额外的性能开销, 会导致执行效率下降。然而这种虚拟化可能是必要的, 特别是在需要在不同的 ISA 之间进行迁移或者兼容性支持的情况下。对于需要对非本地 ISA 进行仿真的系统, 可通过优化指令重编译和增强硬件支持等方式来尽量减少性能开销, 以提高仿真系统的性能。

5.27

5.27.1 每项 $4 + 128 + 16 + 4 + 64 = 216$ 字节

$$\frac{216}{64} = 3.375 \approx 4 \text{ 平均触发 4 次}$$

5.27.2 struct entry {
 int srcIP;
 long long refTime;
 char URL [128];
 int status;
 char browser [64];
 } long [NUM_ENTRIES];

5.27.3 peak_hour (int status);
 将 srcIP, refTime, status 放在相邻位置

6.4

6.4.1 每个迭代的总周期是 $1 + 1 + 6 + 4 + 2 + 2 = 16$ 个周期
 循环从 $j=2$ 到 $j < 1000$ 执行, 共 998 次
 总周期数 $998 \times 16 = 15968$ 周期

6.4.2

```
li $s0, 8000
add $s1, $a0, $s0
addi $s2, $a0, 16
loop = l.d $f0, -16($s2)
      l.d $f2, -8($s2)
      add.d $f4, $f0, $f2
      s.d $f4, 8($s2) // 改变偏增量
      addi $s0, $s0, -8
```

l.d \$f6, 0(\$s2) //将下次迭代的指令提取出来,进行预加载

addi \$s2, \$s2, 16 //更新 \$s2

bne \$s2, \$s1, loop

6.4.3 数组元素 $D[j]$ 和 $D[j-1]$ 将具有携带循环的依赖关系;

它们在当前迭代中为 \$f4, 在下一次的迭代中为 \$f0

6.4.4 li \$s1, 2

li \$s0, 1000

addi \$a0, 16

l.d \$f2, \$8(a0)

l.d \$f4, \$0(a0)

loop =

add.d \$f6, \$f2, \$f4.

mov.d \$f4, \$f2

mov.d \$f2, \$f6

s.d \$f6, \$8(a0)

addi \$a0, \$a0, 8

addi \$s1, \$s1, 1

bne \$s1, \$s0, loop

6.4.5 li \$s1, 2

li \$s0, 1000

addi \$a0, 16

loop:

l.d \$f0, \$0(a0)

l.d \$f4, \$-8(a0)

add.d \$f6, \$f0, \$f4

s.d \$f6, \$8(a0)

l.d \$f0, \$8(a0)

add.d \$f8, \$f6, \$f0

s.d \$f8, \$8(a0)

l.d \$(f4), \$16(a0)

add.d \$f10, \$f8, \$f4

s.d \$f10, \$24(a0)

```
addi $a0, $a0, 24
addi $s1, $s1, 3
bne $s1, $s0, loop
```

每次处理3个迭代总周期数: $3 + 18 + 12 + 2 + 2 = 37$ 周期

$998 \div 3 = 333$ 次

$333 \times 37 = 12321$ 周期

6.4.6 如果迭代次数不是循环展开次数的整数倍。
则展开处理大部分迭代, 剩余处理部分迭代

6.4.7 假设分布式存储器系统有2个节点。
send 延迟1周期, receive 阻塞多个周期
如果阻塞周期多于指令执行周期, 加速效果不明显。

6.6.

6.6.1 加速比接近4

6.6.2 减少3倍

6.6.3 通过跨列而不是行遍历矩阵来计算C中的元素, 这些元素将被映射到不同的缓存行, 我们只需要确保我们在同一个核心上处理计算 (i, j) 和 $(i+1, j)$ 的矩阵索引。