

1.

假设已经进行以下测量：

FP 操作频率=25%

FP 操作的平均 CPI=4.0

其他指令的平均 CPI=1.33

FPSQR 的频率=2%

FPSQR 的 CPI=20

假定有两种设计方案，一种方案将 FPSQR 的 CPI 降至 2，一种是把所有 FP 操作的平均 CPI 降至 2.5。请使用处理器性能公式对比这两种设计方案。

解答：FP 指的是浮点。FPSQR 指的是浮点平方根。CPI 为组成与指令集体系结构。

$$\text{CPU时钟周期} = \sum_{i=1}^n \text{IC}_i \times \text{CPI}_i$$

$$\text{CPU时间} = \left(\sum_{i=1}^n \text{IC}_i \times \text{CPI}_i \right) \times \text{时钟周期时间}$$

$$\text{CPI} = \frac{\sum_{i=1}^n \text{IC}_i \times \text{CPI}_i}{\text{指令数}} = \sum_{i=1}^n \frac{\text{IC}_i}{\text{指令数}} \times \text{CPI}_i$$

根据上述公式，首先求出原有 CPI

$$\text{CPI 原} = (4 \times 25\%) + (1.33 \times 75\%) = 2.0$$

从 CPI 中减去节省的周期数求出改进后的 CPI:

$$\text{CPI 后} = 2.0 - 2\%(20 - 2) = 1.64$$

修改后 FPSQR 后加速比为

$$\text{加速比 FPSQR} = \text{CPI 原} / \text{CPI 后} = 2.0 / 1.64 = 1.23$$

求 FR 的 CPI 降至 2.5 时的 CPI

$$\text{CPI 新} = (75\% \times 1.33) + (25\% \times 2.5) = 1.625$$

修改 FR 后的加速比为

$$\text{加速比 FR} = \text{CPI 原} / \text{CPU 新} = 2.0 / 1.625 = 1.23$$

从计算结果我们可以看出，两种方案所得到的加速比相同。

2.

假定有一个计算机，当所有存储器访问都在缓存中命中时，其每条指令的周期数 (CPI) 为 1.0。仅有的数据访问就是载入和存储，占总指令数的 50%。如缺失代价为 25 个时钟周期，缺失率为 2%，当所有指令都在缓存中命中时，计算机可以加快多少？

解答：

计算计算机总是命中时的性能

$$\begin{aligned}\text{CPU 执行时间} &= (\text{CPU 时钟周期} + \text{存储器停顿周期}) * \text{时钟周期} \\ &= (\text{IC} * \text{CPI} + 0) * \text{时间周期} \\ &= \text{IC} * \text{时间周期}\end{aligned}$$

再计算计算机的停顿周期，数据访问占总指令数的 50%

$$\begin{aligned}\text{存储器停顿周期} &= \text{IC} * \text{存储器访问数/指令} * \text{缺失率} * \text{缺失代价} \\ &= \text{IC} * (1 + 0.5) * 0.02 * 25 \\ &= \text{IC} * 0.75\end{aligned}$$

总性能

$$\begin{aligned}\text{CPU 执行时间} &= (0.75 + 1) * \text{IC} * \text{时钟周期} \\ &= 1.75 * \text{IC} * \text{时钟周期}\end{aligned}$$

性能加速比为

$$\begin{aligned}\text{加速比} &= \text{CPU 执行时间} / \text{CPU 总执行时间} \\ &= 1.75 / 1 = 1.75\end{aligned}$$

所以当没有缓存缺失的时候，计算机可以加快 1.75 倍。

3.

两种不同缓存组织方式对处理器性能的影响如何？假定完美缓存的 CPI 为 1.6，时钟周期时间为 0.35ns，每条指令有 1.4 次存储器引用，两个缓存的大小都是 128KB，两者的块大小都是 64 字节。一个缓存为直接映射，另一个为两路组相联。由于处理器的速度直接与缓存命中的速度联系在一起，所以假定必须将处理器时钟周期时间扩展 1.35 倍，才能与组相联缓存的选择多工器相适应。对于一级近似，每一种缓存组织方式的缓存缺失代价都是 65ns。首先，计算存储器平均访问时间，然后再计算处理器性能。假设命中时间为 1 个时钟周期，128KB 直接映射缓存的缺失率为 2.1%，同等大小的两路组相联缓存的缺失率为 1.9%。

解答：

$$\text{存储器平均访问时间} = \text{命中时间} + \text{缺失率} * \text{缺失代价}$$

所以每种组织方式的时间为

$$\text{一路存储器平均访问时间} = 0.35 + (2.1\% * 65) = 1.72\text{ns}$$

$$\text{两路存储器平均访问时间} = 0.35 * 1.35 + (1.9\% * 65) = 1.71\text{ns}$$

从存储器平均访问时间来看，两路组相联的性能优于一路组相联。

$$\begin{aligned}\text{CPU 时间} &= \text{IC} * (\text{CPI} + \text{缺失数/指令数} * \text{缺失代价}) * \text{时钟周期时间} \\ &= \text{IC} * [(\text{CPI} * \text{时钟周期时间}) \\ &\quad + (\text{缺失率} * \text{存储器访问次数/指令数} * \text{缺失代价} * \text{时钟周期时间})]\end{aligned}$$

性能计算为

$$\text{一路 CPU 时间} = IC * [1.6 * 0.35 + (2.1\% * 1.4 * 65)] = 2.47 * IC$$

$$\text{两路 CPU 时间} = IC * [1.6 * 0.35 * 1.35 + (1.9\% * 1.4 * 65)] = 2.49 * IC$$

所以从存储器平均访问时间上看，两路组相联优于直接映射，但是其平均性能次于直接映射，考虑到直接映射的结构更加简单，所以选择直接映射比较好。

4.

假定在 1000 次存储器引用中，第一级缓存中有 40 次缺失，第二级缓存中有 20 次缺失。各缺失率等于多少？假定 L2 缓存到存储器的缺失代价为 200 个时钟周期，L2 缓存的命中时间为 10 个时钟周期，L1 的命中时间为 1 个时钟周期，每条指令共有 1.5 次存储器引用。每条指令的存储器平均访问时间和平均停顿周期为多少？忽略写入操作的影响。

解答：

$$\text{一级缓存缺失率} = 40/1000 = 4\%$$

$$\text{二级缓存全局缺失率} = 20/1000 = 2\%$$

$$\text{二级缓存局部缺失率} = 20/40 = 50\%$$

$$\begin{aligned} \text{存储器平均访问时间} &= \text{命中时间 1} + \text{缺失率 1} * (\text{命中时间 2} + \text{缺失率 2} * \text{缺失代价 2}) \\ &= 1 + 4\% * (10 + 50\% * 200) = 5.4 \end{aligned}$$

所以存储器平均访问时间为 5.4 个时钟周期。

$$\text{每千条指令 L1 缺失数} = 40 * 1.5 = 60 \text{ 次}$$

$$\text{每千条指令 L2 缺失数} = 20 * 1.5 = 30 \text{ 次}$$

$$\begin{aligned} \text{每条指令的平均存储器停顿} &= \text{每条缺失的缺失数 1} * \text{命中时间 1} \\ &\quad + \text{每条指令的缺失数 2} * \text{缺失代价 2} \\ &= (60/1000) * 10 + (30/1000) * 200 \\ &= 6.6 \end{aligned}$$

所以每条指令的存储器平均停顿时间为 6.6 个时钟周期。

5.

设磁盘子系统的组件及 MTTF 如下：

10 个磁盘，各自的等级为 1000000 小时 MTTF

1 个 ATA 控制器，500000 小时 MTTF

1 个电源，200000 小时 MTTF

1 个风扇，200000 小时 MTTF

1 根 ATA 电缆，1000000 小时 MTTF

采用简化假设：寿命符合指数分布，各故障相互独立，试计算整个系统的 MTTF。

解答：

故障率之和：

$$\begin{aligned}\text{故障率} &= 10 \times 1/1000000 + 1/500000 + 1/200000 + 1/200000 + 1/100000 \\ &= 23/1000000\end{aligned}$$

系统的 MTTF 就是故障率的倒数

$$\text{MTTF} = 1/\text{故障率} = 1000000/23 = 43500 \text{ 小时}$$

所以寿命为 43500 小时。

6.

写出在进行调度与不进行调度的情况下，这个循环在 MIPS 上的执行过程，包括所有停顿或空闲时钟周期。调度时要考虑浮点运算产生的延迟，但忽略延迟分支。

```
Loop:  L.D      F0,0(R1)      ;F0=数组元素
        ADD.D   F4,F0,F2      ;加上 F2 中的标量
        S.D     F4,0(R1)      ;存储结果
        DADDUI  R1,R1,#-8      ;使指针递减 8 个字节
                                   ;( 每个 DW )
        BNE     R1,R2,Loop     ;R1!=R2 时跳转
```

解答：

不进行任何调度

```
Loop: L.D      F0,0(R1)
      停顿
      ADD.D   F4,F0,F2
      停顿
      停顿
      S.D     F4,0(R1)
      DADDUI  R1,R2,#-8
      停顿
      BNE     R1,R2,Loop
```

这个循环花费 9 个周期。

调度这个循环

```
Loop: L.D      F0,0(R1)
      DADDUI  R1,R2,#-8
      ADD.D   F4,F0,F2
      停顿
      停顿
      S.D     F4,0(R1)
      BNE     R1,R2,Loop
```

调度之后，只花费 7 个周期。

7.

在具有 4K 项的(0,2)分支预测器中有多少位？在具有同样位数的(2,2)预测器中有多少项？

解答：

具有 4K 项的预测器拥有

$$2^{0+2} \times 4K = 8K \text{ 位}$$

所以具有 4K 项的(0,2)分支预测器中有 8K 位。

同样位数的(2,2)预测器

$$2^{2+2} \times X = 8K$$

$$2^4 \times X = 8K$$

$$X = 1K$$

所以，预测器中有 1K 项。

8.

考虑下面这样一个循环

```
for (i=0; i<100; i=i+1) {  
    A[i] = A[i] + B[i]; /* S1 */  
    B[i+1] = C[i] + D[i]; /* S2 */  
}
```

S1 和 S2 之间是什么样的相关？这一循环是否为并行的？如果不是，说明如何使之成为并行循环。

解答：

(1)不存在从 S1 到 S2 的相关。如果存在这种相关，那就可能存在环式相关，循环就不可能是并行的。因为不相关，两个语句之间的互换不会影响执行。(这里打一个？)

(2)在循环的第一次迭代中，语句 S2 依赖于 B[0]值，我们用以下代码序列来代替上面的循环。

A[0] = A[0] + B[0];

for(int i = 0 ; i < 99 ; i = i+1)

{

 B[i+1] = C[i] + D[i];

 A[i+1] = A[i+1] + B[i+1]

}

B[100] = C[99] + D[99];

修改之后两个语句之间的相关不再是循环相关，所以循环的各次迭代可以重叠，只要每次迭代中的语句保持相对顺序即可(打个问好？)。

9.

假定希望用 100 个处理器获得 80 倍的加速比。原计算中的串行部分可以占多大比例？

解答：

Amdahl 定律：

$$\text{加速比} = 1 / (\text{增强比例} / \text{增强加速比} + (1 - \text{增强比例}))$$

假定此程序仅以两种模式运行，一种模式是并行方式，所有处理器都得到充分应用。

另一种是串行方式，仅使用一个处理器。可以得到下式：

$$80 = 1 / (\text{并行部分所占比例} / 100 + (1 - \text{并行部分所占比例}))$$

化简所得：

$$0.8 * \text{并行部分所占比例} + 80 * (1 - \text{并行部分所占比例}) = 1$$

$$80 - 79.2 * \text{并行部分所占比例} = 1$$

$$\text{并行部分所占比例} = 79 / 79.2$$

$$= 0.9975$$

所以为了以 100 个处理器实现 80 倍的加速比，原计算中只有 0.25% 是可以串行的。

10.

假定有一个应用程序运行在包含 32 个处理器的多处理器上，它在引用远程存储器时需要的时间为 200ns。对于这一应用程序，假定除涉及通信的引用之外，其他所有引用都会在本本地存储器层次结构中命中，这一假定稍微有些乐观了。处理器会在远程请求时停顿，处理器时钟频率为 3.3GHz。如果基础 CPI(假定所有引用都在缓存中命中)为 0.5，请对比在没有通信，0.2% 的指令涉及远程通信引用这两种情况下，多处理器会快多少？

解答：

当涉及 0.2% 的远程引用时，多处理实际 CPI 为

$$\text{CPI} = \text{基础 CPI} + \text{远程请求率} \times \text{远程请求成本}$$

$$= 0.5 + 0.2\% * \text{远程请求成本}$$

远程请求成本为

$$\text{远程访问成本} / \text{周期时间} = 200\text{ns} / 0.3\text{ns} = 666 \text{ 个周期}$$

所以我们可以得出 CPI:

$$\text{CPI} = 0.5 + 1.2 = 1.7$$

多处理性能提高:

$$1.7 / 0.5 = 3.4$$

所以引用均在本地时，多处理器要快出 3.4 倍。