

## 1、考察点: **CPI、Amdahl**

**例题** 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** Amdahl 定律:  $\text{加速比} = \frac{\text{原执行时间}}{\text{新执行时间}} = \frac{IC \times \text{Cycles} \times CPI_{\text{原}}}{IC \times \text{Cycles} \times CPI_{\text{新}}} = \frac{CPI_{\text{原}}}{CPI_{\text{新}}}$

$$CPI \text{ 计算公式: } CPI = \frac{\sum_{i=1}^n IC_i \times CPI_i}{IC}$$

对于改进前,  $CPI_{\text{原}} = 25\% \times 4 + 75\% \times 1.33 = 2.0$ ;

(1) 采用第一种方法, 计算改进后的 CPI:

设 FP 操作中, 非 FPSQR 指令的 CPI 为  $x$ ; 那么:

$$CPI_{FP} = \frac{2\% \times 20 + 23\% \times x}{25\%} = 4.0 \Rightarrow 23\% \times x = 0.6$$

$$\text{所以, 改进后的 } CPI_{FP} = \frac{2\% \times 2 + 23\% \times x}{25\%} = \frac{2\% \times 2 + 0.6}{25\%} = 2.56$$

得到改进后的  $CPI_{\text{新}} = 25\% \times 2.56 + 75\% \times 1.33 = 1.6375$

$$\text{加速比} = \frac{CPI_{\text{原}}}{CPI_{\text{新}}} = \frac{2.0}{1.6375} = 1.22$$

(2) 采用第二种方法, 计算改进后的 CPI:

改进后,  $CPI_{\text{新}} = 25\% \times 2.5 + 75\% \times 1.33 = 1.6225$

$$\text{加速比} = \frac{CPI_{\text{原}}}{CPI_{\text{新}}} = \frac{2.0}{1.6225} = 1.23$$

最终发现两种方案性能提高的程度几乎相同, 第一种方案稍微好一点。

## 2、考察点: **存储器平均访问时间**

**例题** 利用图 2-3 和附录 B 中表 B-4 中的数据, 判断 32 KB 四路组相联 L1 缓存的存储器访问时间是否快于 32 KB 两路组相联 L1 缓存器。假定 L2 的缺失代价是快速 L1 缓存访问时间的 15 倍。忽略超出 L2 之外的缺失。哪种缓存的存储器平均缓存时间较短?

**解答** 存储器平均访问时间为:

$$\text{命中时间}_{L1} + \text{缺失率}_{L1} \times (\text{命中时间}_{L2} + \text{缺失率}_{L2} \times \text{缺失代价}_{L2})$$

由于题目中说明忽略超出 L2 之外的缺失, 所以

存储器平均访问时间为:  $\text{命中时间}_{L1} + \text{缺失率}_{L2} \times \text{缺失代价}_{L2}$

从图 2-3 可以得到, 对于 32KBcache, 两路组相联访问时间约为 380us, 四路组相联访问时间约为 530us, 为了简化起见, 这里将两路组相联的访问时间设置为 1, 那么四路组相联的访问时间即为  $1 \times (530/380) = 1.4$ 。

另外由于 L2 的缺失代价是快速 L1 缓存访问时间的 15 倍, 所以对于两路缓存, 缺失代价为 15; 由于四路缓存的速度是它的 1.4 倍, 所以四路缓存的缺失代价是  $15/1.4=10$ 。

对于两路缓存:

$$\text{存储器平均访问时间} = 1 + 0.038 \times 15 = 1.57$$

对于四路缓存:

$$\text{存储器平均访问时间} = 1.4 + 0.037 \times 10 = 1.77$$

可以得出结论, 两路缓存的访问时间更短, 具有较好的性能。

### 3、考察点: 缺失代价、缓存CPI、指令执行速度

**例题** 考虑以下 3 种假设的非典型处理器, 我们将在上面运行 SPEC gcc 基准测试。

- (1) 一个简单的 MIPS 双发射静态流水线, 其时钟速率为 4 GHz, 所实现的流水线 CPI 为 0.8。这一处理器的缓存系统每条指令发生 0.005 次缺失。
- (2) 一个双发射 MIPS 处理器的深度流水线版本, 缓存稍小一些, 时钟速率为 5 GHz。处理器的流水线 CPI 为 1.0, 缓存较小, 平均每条指令生成 0.0055 次缺失。
- (3) 一个推测超标量处理器, 具有一个 64 项窗口。它能实现的发射率为这一窗口大小理想发射率的一半。(使用图 3-12 中的数据) 这一处理器的缓存最小, 每条指令产生 0.01 次缺失, 但通过动态调度可以隐藏每次缺失 25% 的缺失代价。这个处理器的时钟为 2.5 GHz。

假定主存储器时间 (这一时间决定了缺失代价) 为 50 ns。判断这 3 种处理器的相对性能。

**解答**  $\text{缺失代价} = \frac{\text{存储器访问时间}}{\text{时钟周期}}$

$$\text{缓存CPI} = \text{每条指令缺失数} \times \text{缺失代价}$$

$$\text{指令执行速度} = \frac{CR}{CPI}$$

三种处理器的时钟周期分别是 250ps、200ps、400ps, 可算得缺失代价:

$$\text{缺失代价}_1 = \frac{50ns}{250ps} = 200$$

$$\text{缺失代价}_2 = \frac{50ns}{200ps} = 250$$

$$\text{缺失代价}_3 = \frac{0.75 \times 50ns}{400ps} = 94$$

可以得到缓存 CPI:

$$\text{缓存}CPI_1 = 0.005 \times 200 = 1.0$$

$$\text{缓存}CPI_2 = 0.0055 \times 250 = 1.4$$

$$\text{缓存}CPI_3 = 0.01 \times 94 = 0.94$$

计算处理器 3 的流水线 CPI:

$$\text{流水线}CPI_3 = \frac{1}{\text{发射速率}} = \frac{1}{9 \times 0.5} = 0.22$$

得出每个处理器的 CPI:

$$CPI_1 = 0.8 + 1.0 = 1.8$$

$$CPI_2 = 1.0 + 1.4 = 2.4$$

$$CPI_3 = 0.22 + 0.94 = 1.16$$

最后得出相对性能:

$$\text{指令执行速度}_1 = \frac{4000MHz}{1.8} = 2222MIPS$$

$$\text{指令执行速度}_2 = \frac{5000MHz}{2.4} = 2083MIPS$$

$$\text{指令执行速度}_3 = \frac{2500MHz}{1.16} = 2155MIPS$$

最终可以看出, 简单的双发射静态超标量看起来是最好的。

#### 4、考察点: 分支预测器

**例题** 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** 一个(m,n)预测器的位数为:  $2^m \times n \times \text{由分支地址选中的预测项数目}$

具有 4K 项的预测器拥有:

$$2^0 \times 2 \times 4K = 8K \text{位}$$

在预测缓冲区中共有 8K 位的(2,2)预测器中:

$$2^2 \times 2 \times \text{由分支地址选中的预测项数目} = 8K$$

因此, 由分支选中的预测项数=1K。

#### 5、考察点: 钟鸣、VMIPS 时钟周期

**例题** 给出以下代码序列在护航指令组中是如何排列的，假定每个向量功能单元只有一个副本：

LV	V1,Rx	;载入向量 X
MULVS.D	V2,V1,F0	;向量-标量乘
LV	V3,Ry	;载入向量 Y
ADDVV.D	V4,V2,V3	;两个向量相加
SV	V4,Ry	;存储所得之和

这个向量序列将花费多少次钟鸣？每个 FLOP（浮点运算）需要多少个时钟周期（忽略向量指令发射开销）？

**解答** 钟鸣：执行护航指令组所花费的时间单位。

VMIPS 时钟周期：执行由  $m$  个护航指令组构成的向量序列，需要  $m$  次钟鸣，当向量长度为  $n$  时，对 VMIPS 来说，大约为  $m \times n$  个时钟周期。

护航指令组：一组可以一直执行的向量指令。

第一个护航指令组从第一个 LV 指令处开始。MULVS.D 依赖于第一个 LV，但链接操作允许它位于同一组护航指令中。

第二个 LV 指令必须放在另一个护航指令组中，因为它与上一个 LV 指令的载入/存储单元存在结构性冒险。ADDVV.D 与第二个 LV 相关，但它也可以通过链接操作位于同一组护航指令组中。

SV 指令与第二个护航指令组中的 LV 存在结构冒险，所以必须把它放在第三个护航指令组中。

因此有三个护航指令组：

- LV MULVS.D
- LV ADDVV.D
- SV

这个序列需要 3 次钟鸣，每个结果有 2 个浮点运算，所以每个 FLOP 的时钟周期数为 1.5。