
 Universidade do Minho	Módulo 4 Medição de desempenho da Hierarquia de Memória	
--	--	---

No final deste módulo os alunos deverão ser capazes de:

- Aplicar as metodologias básicas de medição do desempenho de programas.
- Identificar os principais componentes da hierarquia de memória, os seus parâmetros (nº de níveis e tamanho) e o seu impacto no desempenho global da arquitectura.
- Relacionar o impacto no desempenho da codificação dos programas com a hierarquia de memória da arquitectura subjacente.

1. Conteúdos e Resultados da Aprendizagem relacionados

Conteúdos	7.1 - Tempo de Execução 7.3 - Ciclos por Elemento (CPE) 8.3 - Avaliação do Desempenho
Resultados da Aprendizagem	R7.1 - Identificar e caracterizar as métricas relativas ao desempenho da execução de programas R7.3 - Avaliar diferentes tipos de benchmarks relativamente à qualidade e tipo de informação produzida R8.2 - Quantificar o impacto da hierarquia da memória no desempenho do sistema

Os exercícios deste módulo utilizam a função:

```
double get_counter()
```

que lê o contador de ciclos do relógio da arquitectura Intel x86.

Para medir o número de ciclos decorrido durante a execução de um segmento de código invoca-se a função antes e depois da execução desse segmento. A diferença entre os dois valores corresponderá ao valor pretendido:

```
double c1, c2, ciclos;
c1 = get_counter();
<< Segmento de código >>
c2 = get_counter();
ciclos = c2 - c1;
```

O número de ciclos necessário para processar cada elemento (CPE) pode ser obtido dividindo o número de ciclos obtidos pelo número de elementos processados nesse segmento de código.

Ao fazer uma medição deve ter em consideração de que existem inúmeros factores que fazem com que o tempo efectivamente decorrido varie de execução para execução. Para minimizar este efeito o segmento de código deve ser medido várias vezes e, baseado nos valores destas medições, deve ser calculada uma estatística; para efeitos desta sessão sugere-se que a estatística a apresentar seja a média aritmética.

Exercícios:

1. O programa “sumarray.c” determina o número médio de ciclos (CPE) necessário para calcular cada elemento da matriz nos dois programas seguintes. Compare os valores obtidos para várias dimensões da matriz e explique os resultados obtidos.

```
int sum_array_rows(int a[M][N]) {  
    int i, j, sum = 0;  
    for (i = 0; i < M; i++)  
        for (j = 0; j < N; j++)  
            sum += a[i][j];  
    return sum;  
}
```

```
int sum_array_cols(int a[M][N])  
{  
    int i, j, sum = 0;  
    for (j = 0; j < N; j++)  
        for (i = 0; i < M; i++)  
            sum += a[i][j];  
    return sum;  
}
```

2. O programa “stride.c” determina o número médio de ciclos (CPE) necessário para calcular cada elemento de um vector de elementos. Em cada iteração do ciclo os elementos de vectores de diferentes tamanhos são acedidos avançando 1, 2, 4, 8 e 16 elementos (variável “stride”). Execute o programa o discuta os resultados obtidos.

```
for(int stride=1; stride<32; stride=stride*2) {  
    for(int i=0x1000000; i>128; i>>=2) {  
  
        c1 = get_counter();  
        for(int j=0; j<size; j+=stride) {  
            bm += a[j];  
        }  
        c2 = get_counter();  
  
        << calcular CPE >>  
  
    }  
}
```

3. Altere o programa da alínea 2) para obter também localidade temporal (i.e., acedendo múltiplas vezes ao mesmo elemento).