

# LAB 3 – OpenACC

Computação Heterogénea de Alto Desempenho (2019/2020)

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

1. SAXPY (Single-Precision A.X plus Y) é uma operação muito comum em computação e é definida da seguinte forma:  $y = a \cdot x + y$ , onde  $a$  é um valor escalar,  $x$  e  $y$  são vetores com  $N$  elementos cada. Implemente, usando directivas OpenACC, uma rotina que execute esta operação. Compare a performance com a equivalente versão sequencial.

2. a) Escreva um programa em OpenACC para calcular a aproximação do logaritmo natural (de base  $e$ ) usando os primeiros 10 000 000 de termos da expansão de Maclaurin:

$$\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \dots, \text{ para } -1 < x \leq 1.$$

b) Desenhe a curva de speedup em função do número de iterações.

3. Escreva o código OpenACC de um programa que calcule o valor médio dos elementos que ocupam a parte triangular inferior (i.e., todos os elementos em e por baixo da diagonal) de uma matriz quadrada de grandes dimensões. É possível otimizar o programa de modo a que:

a) A eficiência das transferências de dados seja melhorada (e.g., evitando transferir dados que não são usados nos cálculos efetuados no device)?

b) O trabalho realizado em cada iteração seja equilibrado pelas diferentes threads da GPU? Implemente as otimizações possíveis. Como afetam o desempenho? Teste várias dimensões da matriz.

4. Para fazer debugging de um programa OpenACC, as partes irrelevantes do código foram removidas, restando o seguinte:

---

```
#include <stdio.h>
const int N=100, M=200;
int main() {
int m[N][M];
for(int i=0; i<N; i++)
    for(int j=0; j<M; j++)
        m[i][j]=1;
#pragma acc kernels
    for(int i=0; i<N; i++)
        for(int j=M-i; j<M; j++)
            m[i][j]=i+j+1;
// verify result
int errcnt=0;
for(int i=0; i<N; i++)
    for(int j=0; j<M; j++) {
        int expect=(j>=M-i)? i+j+1: 1;
        if(m[i][j]!=expect) errcnt++;
    }
printf("Encountered %d errors\n", errcnt);
return errcnt != 0;
}
```

---

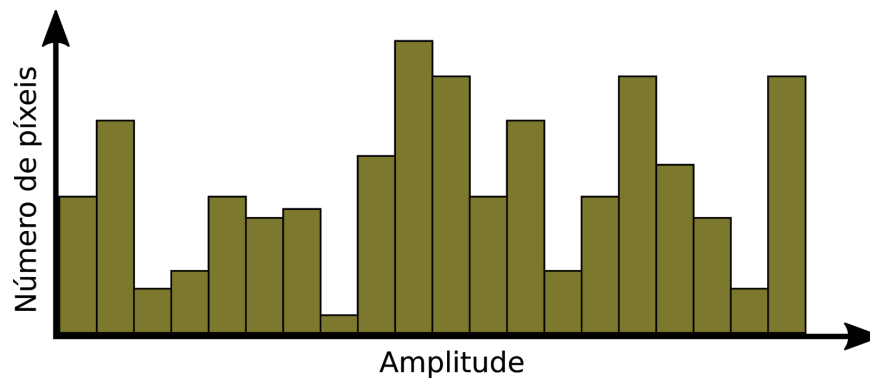
O código falha (produz um valor diferente de zero para *error count*) se compilado com alguns compiladores OpenACC. Qual poderá ser a causa? Como pode o erro ser prevenido?

5. A equação de Laplace (  $\nabla^2 T = 0$  ) pode ser usada para descrever a variação de temperatura numa placa metálica. Na prática, esta equação calcula o valor de temperatura de um dado ponto como sendo a média aritmética dos valores de temperatura da vizinhança:

$$T(i,j) = [T_{old}(i+1,j) + T_{old}(i-1,j) + T_{old}(i,j+1) + T_{old}(i,j-1)]/4.$$

Neste exercício, a placa metálica é representada por uma grelha 2D e o estado inicial pode ser aleatoriamente gerado (valores entre 0 e 100 graus). O programa deve iterar sobre a placa metálica e a condição de paragem é  $T(i,j) - T_{old}(i,j) < 0.01$ .

6. O histograma de uma imagem mostra a distribuição da intensidade dos píxeis nessa imagem. Na figura seguinte está ilustrado um histograma, com a amplitude no eixo horizontal e o número de píxeis, em função da amplitude, no eixo vertical.



Implemente um programa em OpenACC que obtenha o histograma da distribuição das intensidades de uma imagem com N píxeis. Implemente também a versão sequencial que permita comparar *throughput performance* com a correspondente versão paralela em OpenACC. Para representar a imagem, gere uma matriz de números aleatórios (intensidade dos píxeis entre 0 e 255).