

INE5645 – Programação Paralela e Distribuída

Tarefa Unidade 3

Exemplos de orientação OpenCL

a) Exemplo Raiz Quadrada no link [Introdução ao OpenCL - Douglas Adriano Augusto - LNCC – 2012.](#)

b) Exemplo 5.1 no link [Introdução em OpenCL - Uma introdução prática \(Sua Primeira Experiência – Exemplo 5.1\)](#)

c) Exemplo do problema no slide 80 dos slides apresentados em aula. Ver os códigos nos links List 4.8, List 4.9 (modelo de paralelismo de dados) e List 4.10 e List 4.11 (modelo de paralelismo de tarefas). O mesmo problema é resolvido usando paralelismo de dados e por paralelismo de tarefas.

d) Ou qualquer outro exemplo que você queira.

Depois de estudar exemplos, passe aos seguintes exercícios.

Exercício de implementação OpenCL

O objetivo deste exercício é familiarizá-lo com o ambiente de programação OpenCL.

Para tanto, você deve estudar os exemplos mencionados nos slides e links na página da disciplina, na Unidade 3.

www.inf.ufsc.br/~bosco.sobral

Ensino -> INE5645 > (Unidade 3)

Você deve instalar o driver do seu *dispositivo OpenC (GPU)*, seguindo as instruções dos respectivos fabricantes, dependendo do que você tiver disponível:

- Especificação OpenCL Khronos: <https://www.khronos.org/opencl/>
- NVIDIA: <https://developer.nvidia.com/opencl>

- Intel: <https://software.intel.com/en-us/intel-openc>
- AMD/ATI: <http://developer.amd.com/tools-and-sdks/openc-zone/>

1. Em seguida você deve executar e entender o programa que detecta as plataformas e *devices OpenCL* de um computador (podem existir mais de 1 GPU), imprimindo suas características. Veja o exemplo básico seguinte. Você pode adaptar o que quiser, baseando-se nos exemplos que foram mencionados em aula, que estão na página da disciplina.

```
#include <stdio.h>
#include <stdlib.h>

// Para uso de qualquer GPU
#ifdef _Apple_
#include <OpenCL/openc.h>
#else
#include <CL/cl.h>
#endif

// Para uso de C++
#include <iostream>
#include <vector>
#include <cstdlib>
#define __CL_ENABLE_EXCEPTIONS
#include <CL/cl.hpp>
int main(int argc, char *argv[]) {
    try {
        // Obter plataformas
        std::vector<cl::Platform> platforms;
        // [...]

        // Interagir sobre as plataformas disponíveis
        for(/*...*/) {
```

```

        // Obter devices
        std::vector<cl::Device> devices;

        // Interagir sobre o device (GPU) disponível
        for(/*...*/) {
            // criar o contexto, etc, ...
            ...
            ...
        }
    }

} catch(cl::Error& err) {
    std::cerr << "OpenCL error: " << err.what() << "(" << err.err()
        << ")" << std::endl;

    return EXIT_FAILURE;
}

return EXIT_SUCCESS;
}

```

Para facilitar o acesso às informações, recomendo a utilização de:

- www.inf.ufsc.br/~bosco.sobral/ensino/ine5645 - **Unidade 3**
- Manuais e Guias de Referência:
<https://www.khronos.org/registry/cl/>

2. Finalmente, você deve escrever um programa OpenCL que aloque dois vetores A e B, de números inteiros de tamanho pré-definido, preencha-os com o valor do índice em cada posição ($v[i]=i$), e em seguida calcule o produto escalar entre eles.

Por exemplo, se $A = [1, 2, 3, 4, 5]$ e $B = [6, 7, 8, 9, 10]$, o produto escalar $\langle A, B \rangle = (1 \times 6 + 2 \times 7 + 3 \times 8 + 4 \times 9 + 5 \times 10) = (6 + 14 + 24 + 36 + 50) = 130$

Crie um kernel (ou kernels) necessário(s) para paralelizar no *device OpenCL (GPU)*. Procure utilizar (CPU + GPU), por exemplo. O uso de CPU (o boot remoto em Ubuntu 16.04 da UFSC tem OpenCL, mas deve ser sem a GPU, até porque é uma VM, e OpenCL somente na CPU não tem muita graça!). Veja os ambientes que você pode utilizar na página da disciplina.