**Instruções:**

1. **Respostas:**
   1. As questões teóricas devem ser respondidas nos arquivos “questão\_*n*.odt” (onde *n* corresponde ao número da questão).
   2. As questões práticas devem ser respondidas de forma isolada na respectiva pasta.
   3. As questões práticas são avaliadas considerando a exatidão e desempenho.
   4. Deve ser entregue uma versão executável de cada questão prática.
   5. Código que não compila não é avaliado.
2. **Quando tiver concluído a prova:** 
   1. Gere um pacote “*aluno*.zip” (substitua “aluno” pelo seu nome) contendo todos os arquivos mais o executável de cada questão prática.
   2. Avise o professor que você está pronto para postar o arquivo na sua conta do moodle.
3. Somente é permitido o uso de código fornecido pelo professor.
4. Não conecte ao computador qualquer tipo de dispositivo!
5. Desligue o celular.
6. Não respeitar as instruções resultará em nota zero.
7. **SALVE CONSTANTEMENTE SEU TRABALHO!**

**Questões Teóricas**

**Questão 1 (25%):** Qual é a lei de Amdah’l? O que ela nos diz?

**Questão 2 (25%)**: Como a sequência de código a seguir pode ser alterada para expor mais paralelismo, mas ainda alcançar o mesmo resultado final (ou seja, no final: x, a, b e c têm o mesmo valor que o código sequencial)? Denote quantas threads e o que cada uma faria.

x++ ;

a = x + 2;

b = a + 3;

c++;

**Questão 3 (20%)**: Diga quais as vantagens e as desvantagens de implementar a questão 4 em OpenMP, MPI e OpenCL. Qual modelo de programação paralela cada uma representa?

**Questões Práticas**

**Questão 4 (30%):** Faça um programa em OpenCL que executa a seguinte operação numa matriz de inteiros:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

Matriz de entrada

|  |  |
| --- | --- |
| 3,5 | 5,5 |
| 11,5 | 13,5 |

Matriz de saída

O programa pega uma matriz de entrada com tamanho m x m (m deve ser lida por scanf, só aceite valores 2n) e reduz ela até uma matriz 2x2. O conteúdo da matriz é sequencia como no exemplo, mas deve funcionar para quaisquer valores. A redução é feita por uma média dos valores em cada cor. Se m for 8x8 então é preciso reduzir para 4x4 e em seguida 2x2. Imprima na tela todas as reduções. O número de cores é igual a m. Cada cor deve ser calculada numa thread. Não gere threads desnecessárias. Deixe um arquivo .txt com os resultados com m=256 na sua resposta.

Dica: O capítulo 7 do livro “Heterogeneous Computing with OpenCL” (leitura solicitada em aula) trata sobre convoluções e apresenta a solução que é muito similar a resposta da questão.