

## AValiação Substitutiva – Supercomputação

1. Escreva um programa em C++ que simula uma partida de Ping Pong entre dois processos (3,0 pontos):
  - Rank 0 e Rank 1 são os processos
  - Rank 0 inicia com a bola
  - O rank que está com a bola envia a bola para o outro rank
  - Ambos os ranks contabilizam quantas vezes estiveram com a posse da bola (incrementos de 1 em 1, a cada posse da bola)
  - Ao receber a bola, além de contar quantas vezes esteve com a posse da bola, devolve a bola para o outro rank
  - Após ter uma posse de bola no valor de 1 milhão, o nó tem 70% de chance de desistir e finalizar a sua execução, terminando toda a comunicação MPI. Enquanto ele não finaliza, o ping pong vai sendo executado.

A submissão da questão será via blackboard, na área respectiva da prova substitutiva, e deve conter: (i) código-fonte, (ii) arquivo de submissão do slurm e (iii) arquivo de output.

2. Nesta questão vamos desenvolver um programa baseado na estratégia de dividir para conquistar em OpenMP. Uma única thread deve gerar um grande vetor (+ de 500 mil elementos) e dividir esse vetor em pequenos pedaços, de modo que as threads busquem um valor específico na porção de dados que receberam e que, ao final, cada thread exiba em tela a quantidade de vezes em que encontrou o elemento procurado (3,0 pontos).

**Obrigatoriamente você deve fazer uso de tasks em OpenMP.**

Caso necessite de inspiração para gerar os números aleatórios em um vetor, use o código abaixo.

Seu programa deve ser submetido via SLURM.

A submissão da questão será via blackboard, na área respectiva da prova substitutiva, e deve conter: (i) código-fonte, (ii) arquivo de submissão do slurm e (iii) arquivo de output.

```
#include <iostream>
#include <vector>
#include <random>

int main() {
    std::vector<int> numerosAleatorios;
    numerosAleatorios.reserve(500000);

    // Motor de geração de números aleatórios
    std::random_device rd;
    std::mt19937 gen(rd()); // Mersenne Twister

    // Define o intervalo dos números aleatórios, por exemplo, de 0 a 99999
    std::uniform_int_distribution<> distrib(0, 99999);

    for (int i = 0; i < 500000; ++i) {
        numerosAleatorios.push_back(distrib(gen)); // Gera e adiciona um número aleatório ao vetor
    }

    return 0;
}
```

3. Implemente em C++, fazendo uso da biblioteca Thrust, um programa para redução paralela. Você deve gerar um grande vetor (pelo menos 200 mil elementos), preenchido com números aleatórios, transferi-lo para a GPU e realizar uma operação de redução, fazendo a soma desses elementos. A soma deve ser exibida na tela. Para a submissão dessa questão no blackboard, você deve submeter código-fonte, e output da sua execução. (3,0 pontos)
4. Considerando o conteúdo visto no semestre, comente sobre a importância da computação de alto desempenho nas aplicações tecnológicas atuais. (0,5 ponto)
5. Considerando as estratégias e ferramentas abordadas no semestre, comente sobre suas diferenças e complementações para resolução de problemas computacionais complexos. (0,5 ponto)

BOA PROVA.