

Projeto Final (Individual)

O objetivo deste projeto é paralelizar a simulação da propagação de uma onda acústica em um domínio 2D. A simulação é feita por equações diferenciais parciais do tipo:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right)$$

Para executar essa simulação em um computador, precisaremos utilizar a forma discretizada dessa EDP:

$$p_{i,j}^{n+1} = 2p_{i,j}^n - p_{i,j}^{n-1} + \Delta t^2 \times v^2 \left(\frac{p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n}{\Delta x^2} + \frac{p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n}{\Delta y^2} \right)$$

onde representa a pressão (p) na posição ij do domínio 2D no passo de tempo $n+1$. A solução computacional é feita pelo loop que aparece no trecho de código abaixo:

```
// wavefield modeling
for(size_t n = 0; n < iterations; n++) {
    for(size_t i = 1; i < rows - HALF_LENGTH; i++) {
        for(size_t j = 1; j < cols - HALF_LENGTH; j++) {
            // index of the current point in the grid
            current = i * cols + j;

            // stencil code to update grid
            value = 0.0;
            //neighbors in the horizontal direction
            value += (prev_base[current + 1] - 2.0 * prev_base[current] +
                prev_base[current - 1]) / dxSquared;
            //neighbors in the vertical direction
            value += (prev_base[current + cols] - 2.0 * prev_base[current] +
                prev_base[current - cols]) / dySquared;
            value += dtSquared * vel_base[current] * vel_base[current];
            next_base[current] = 2.0 * prev_base[current] - next_base[current]
                + value;
        }
    }

    //swap arrays for next iteration
    swap = next_base;
    next_base = prev_base;
    prev_base = swap;
}
```

Durante a aula de 16/06/21 foram discutidos diversos aspectos de como pode ser implementado em MPI e CUDA (o problema é muito parecido com a equação de Laplace de propagação do calor) . Faça também experimentos de escalabilidade e gere um relatório. Para isso:

- 1) Faça a paralelização do código sequencial disponível em:

https://github.com/HPCSys-Lab/HPC-101/blob/main/examples/acoustic-wave/wave_seq.c

A paralelização pode ser feita para CPU ou GPU, utilizando qualquer um dos

ambientes de programação estudados na disciplina. Pode ser feito individualmente ou em dupla.

- 2) Calcule o speedup obtido tomando por base o tempo de execução do programa sequencial executando em CPU convencional.
- 3) Gerar um relatório com:
 - a) Identificação do(s) aluno(s).
 - b) Características da máquina onde você executou (tipo de processador e da GPU utilizados).
 - c) Se for GPU descreva também:
 - Número de blocos por grid
 - Número de threads por bloco
 - d) Breve discussão sobre os resultados (ex: procure comentar se a escalabilidade foi boa ou ruim, e os prováveis motivos).
- 4) Prepare uma apresentação (slides) de até 15 minutos. Vocês deverão fazer uma apresentação no dia **04 de Maio** no horário da aula.
- 5) Apresentação intermediária: no dia **27 de Abril** vocês devem fazer uma apresentação intermediária contendo:
 - a) Qual será a arquitetura alvo (CPU ou GPU) e ambiente de programação paralela (Pthreads, OpenMP, MPI ou CUDA).
 - b) Discuta em linhas gerais como pretende decompor o problema, estratégia de paralelismo que pretende utilizar e outros detalhes que achar importante. Faça alguns slides para isso.

Usando o próprio formulário da atividade no Google classroom, vocês devem enviar uma pasta compactada (exemplo Projeto_Fulano_Cicrano.zip), contendo os seguintes arquivos: 1) Relatório (.pdf)

- 2) Os slides da apresentação
- 3) Quaisquer artefatos utilizados no experimento, tais como:
 - a) Código fonte em C/CUDA (extensão .cu)
 - b) Arquivo de definição do container (.def), se houver
 - c) Script do job (.sh), se houver