

Projeto PCD: Simulação e Análise de Modelos de Difusão de Contaminantes em Água

Profs. Álvaro e Denise (Turmas I e N)

Objetivo: Criar uma simulação que modele a difusão de contaminantes em um corpo d'água (como um lago ou rio), aplicando conceitos de paralelismo para acelerar o cálculo e observar o comportamento de poluentes ao longo do tempo. O projeto investigará o impacto de OpenMP, CUDA e MPI no tempo de execução e na precisão do modelo.

Etapas do Projeto

1. Estudo do Modelo de Difusão

- Estudar a Equação de Difusão/Transporte, representada por:

$$\frac{\partial C}{\partial t} = D \cdot \nabla^2 C$$

onde:

- C é a concentração do contaminante,
 - t é o tempo,
 - D é o coeficiente de difusão,
 - $\nabla^2 C$ representa a taxa de variação da concentração no espaço.
- A equação pode ser aproximada usando diferenças finitas em uma grade bidimensional, onde cada célula da grade atualiza seu valor com base nas células vizinhas em cada iteração.

2. Configuração do Ambiente e Parâmetros da Simulação

- Configurar uma grade 2D onde cada célula representa a concentração de contaminantes em uma região do corpo d'água.
- Definir o coeficiente de difusão (D), as condições de contorno (por exemplo, bordas onde o contaminante não se espalha) e as condições iniciais (como uma área de alta concentração de contaminante).

3. Implementação com OpenMP (Simulação Local em CPU)

- Usar OpenMP para paralelizar o cálculo de difusão entre os núcleos da CPU. Cada núcleo processa uma parte da grade, aplicando as regras de difusão às células sob sua responsabilidade.
- **Entrega 1:** demonstrar o código em OpenMP e apresentar avaliação de desempenho com relação à versão sequencial.

4. Implementação com CUDA (Simulação em GPU)

- Implementar a simulação em CUDA, onde cada célula da grade é processada por uma thread independente na GPU, utilizando um esquema de diferenças finitas para calcular o laplaciano de (C).
- A execução em GPU permite simular uma grade maior e observar o ganho de desempenho com CUDA.
- **Entrega 2:** demonstrar o código em CUDA e apresentar avaliação de desempenho com relação às versões anteriores.

5. Distribuição com MPI (Simulação em Larga Escala)

- Dividir a grade em sub-regiões e distribuir o processamento entre várias máquinas usando MPI.
- Cada máquina processa uma seção do corpo d'água e troca informações nas bordas com as máquinas vizinhas para garantir a continuidade da difusão de contaminantes entre as regiões.
- **Entrega 3:** demonstrar o código em MPI híbrido (pode incluir trechos em OpenMP e CUDA) e apresentar avaliação de desempenho com relação às versões anteriores, porém destacando a escalabilidade possível apenas com MPI.

6. Artigo científico e Discussão dos Resultados

- Criar gráficos que mostrem a evolução da concentração ao longo do tempo e comparar o tempo de execução entre as implementações.
- Discutir as vantagens e limitações de cada abordagem, observando a escalabilidade, precisão e aplicabilidade em simulações ambientais.
- Demonstrar visualmente os resultados que comprovem a corretude da simulação.
- **Entrega Final:** entregar o resultado final no formato de artigo científico (modelo a ser disponibilizado).

Ponto de Partida para a Implementação da Equação

Para aproximar a Equação de Difusão, podemos usar a seguinte fórmula de diferenças finitas central:

$$C_{i,j}^{t+1} = C_{i,j}^t + D \cdot \Delta t \left(\frac{C_{i+1,j}^t + C_{i-1,j}^t + C_{i,j+1}^t + C_{i,j-1}^t - 4 \cdot C_{i,j}^t}{\Delta x^2} \right)$$

Aqui está um trecho de código em C para uma implementação sequencial simples:

```
# include <stdio.h>
# define N 100 // Tamanho da grade
# define T 1000 // Número de iterações
# define D 0.1 // Coeficiente de difusão
# define DELTA_T 0.01
# define DELTA_X 1.0

void diff_eq(double C[N][N], double C_new[N][N]) {
    for (int t = 0; t < T; t++) {
        for (int i = 1; i < N - 1; i++) {
            for (int j = 1; j < N - 1; j++) {
                C_new[i][j] = C[i][j] + D * DELTA_T * (
                    (C[i+1][j] + C[i-1][j] + C[i][j+1] + C[i][j-1] - 4 * C[i][j]) / (DELTA_X * DELTA_X)
                );
            }
        }
        // Atualizar matriz para a próxima iteração
        for (int i = 1; i < N - 1; i++) {
            for (int j = 1; j < N - 1; j++) {
                C[i][j] = C_new[i][j];
            }
        }
    }
}

int main() {
    double C[N][N] = {0}; // Concentração inicial
    double C_new[N][N] = {0}; // Concentração para a próxima iteração

    // Inicializar uma concentração alta no centro
    C[N/2][N/2] = 1.0;

    // Executar a equação de difusão
    diff_eq(C, C_new);

    // Exibir resultado para verificação
    printf("Concentração final no centro: %f\n", C[N/2][N/2]);
}
```

```
    return 0;  
}
```

Esse código calcula a difusão do contaminante em uma grade de 100x100 ao longo de 1000 ciclos. A concentração inicial está configurada no centro da grade, e o coeficiente de difusão (D) pode ser ajustado conforme necessário.

Características do Projeto

- **Caráter Científico:** Esse projeto aplica uma abordagem científica a um problema ambiental relevante.
- **Exploração e Análise de Desempenho:** Permite uma análise detalhada das implementações em OpenMP, CUDA e MPI, com foco em desempenho e escalabilidade.
- **Estrutura de Redação Científica:** O projeto estimula a criação de um artigo científico, semelhante a um trabalho de iniciação científica.