

Programação Concorrente e Paralela

Projeto 1 - openMP

1 Definição do problema

Sistemas de equações lineares, do tipo $(A.x = b)$, podem ser resolvidos de diversas formas. Uma delas é por meio do método de Jacobi, que usa uma abordagem iterativa, em que a cada iteração k se calcula uma nova aproximação da solução.

A representação algébrica do método de Jacobi é dada pela composição entre uma matriz diagonal e as matrizes inferior e superior da matriz A . Usando essa decomposição se pode calcular cada variável a partir da equação:

$$x_i^{k+1} = \frac{1}{a_{ii}} \cdot (b_i - \sum_{j \neq i} a_{ij} \cdot x_j^k) \quad (1)$$

2 O que deve ser feito

Implemente uma versão paralela do método de Jacobi, usando openMP. Implemente também uma versão sequencial, para servir de comparação com a sua implementação paralela.

Seus programas (dois códigos separados) devem resolver sistemas de equações lineares, devendo ser testados com um sistema de 2000 equações (com solução factível).

O critério de parada a ser adotado é que a maior diferença entre X_i^{k+1} e X_i^k deve ser menor que 10^{-5} .

2.1 Entrada de dados

A entrada consiste em um único arquivo, contendo 2001 linhas.

As primeiras 2000 linhas contém os coeficientes das 2000 variáveis (mesmo se igual a zero).

A última linha contém as 2000 constantes do vetor b .

Todos os valores são reais, com até 4 casas decimais, dentro do intervalo entre -500 e +500.

2.2 Saída de dados

Seus programas devem produzir um arquivo com uma única linha, contendo os valores solução do sistema.

Esses valores devem ter 4 casas decimais.

2.3 Condições de teste

Os tempos de execução devem ser medidos para cada condição de teste a seguir, sendo que para cada uma delas deve ser apresentada a média de três execuções:

1. Versão sequencial pura;
2. Versão openMP com 2 threads, schedule estático simples
3. Versão openMP com 4 threads, schedule estático simples
4. Versão openMP com 8 threads, schedule estático simples
5. Versão openMP com 2 threads, schedule estático, chunk de 25
6. Versão openMP com 4 threads, schedule estático, chunk de 25
7. Versão openMP com 8 threads, schedule estático, chunk de 25
8. Versão openMP com 2 threads, schedule dinâmico, chunk de 25
9. Versão openMP com 4 threads, schedule dinâmico, chunk de 25
10. Versão openMP com 8 threads, schedule dinâmico, chunk de 25
11. Versão openMP com 2 threads, schedule guided, chunk de 25
12. Versão openMP com 4 threads, schedule guided, chunk de 25
13. Versão openMP com 8 threads, schedule guided, chunk de 25

3 O que deve ser entregue

Devem ser entregues os códigos dos dois programas (sequencial e openMP) e um relatório sobre as comparações feitas a partir dos testes indicados.

4 Data de entrega

19/09/2025