Laboratório 4

Este laboratório é composto de um exercício que deve ser realizado usando os arquivos fornecidos. O código deve estar identado, organizado e comentado. A entrega do laboratório deverá ser feita até o dia 13/12 às 23:59, através de um arquivo zip na tarefa do Google Classroom.

ATENÇÃO

Deve ser entregue apenas os arquivos cpp, compactado em um ZIP, sem nenhum arquivo de projeto ou executável. O arquivo DEVE estar com o nome pedido.

Exercício 1: Multiplicação Paralela de Matrizes usando Pool de Threads (5 pontos)

Objetivo: Implementar um algoritmo de multiplicação de matrizes em C++, utilizando um pool de threads para paralelizar a operação e melhorar o desempenho.

Desenvolvimento do Programa: No arquivo `ex1_exemplo.cpp` é fornecida uma implementação do pool de threads, com o accumulate paralelo visto na aula, como um exemplo de uso. O código serial da multiplicação de matrizes está no arquivo `ex1.cpp`, e vocês devem modificá-lo, implementando a multiplicação de matrizes usando esse pool de threads.

Análise do Código Fornecido:

- Estude o código serial da multiplicação de matrizes para entender sua lógica e funcionamento.
- Examine a implementação do pool de threads para compreender como as tarefas são enfileiradas e executadas de forma concorrente.

Desenvolvimento da Multiplicação Paralela de Matrizes:

- Modifique a multiplicação de matrizes para que utilize o pool de threads.
- Assegure-se de que a distribuição das tarefas entre as threads seja eficiente e balanceada.
- Não imprima a matriz. Porém é necessário realizar a multiplicação da matriz serial e da matriz paralela, medindo o tempo gasto em cada uma. Ao final do programa, devem ser impressos os tempos e também o speedup. Além disso, imprima uma mensagem informando se as matrizes são iguais (ou seja, se sua multiplicação paralela produziu o mesmo valor que a multiplicação serial, pode usar um intervalo de erro, na parte decimal, para considerar problemas de precisão com o float).

Entrega: O arquivo `ex1.cpp` deve ser compactado junto com o arquivo do exercício 2 em um arquivo zip, nomeado conforme instruções da tarefa, e entregue no Google Classroom.

Exercício 2: Análise de Dados Meteorológicos (5 pontos)

Objetivo: Utilizar `std::transform_reduce` para calcular estatísticas a partir de um conjunto de dados meteorológicos.

Preparação de Dados:

- Crie uma estrutura `WeatherData` que contém informações como temperatura, umidade e velocidade do vento.
- Gere um grande conjunto de dados (pelo menos 10.000 registros) de `WeatherData`. Você pode criar dados fictícios ou usar um conjunto de dados reais disponível publicamente.

Tarefa de Análise:

- Escreva um programa em C++ que lê o conjunto de dados de WeatherData.
- Use `std::transform_reduce` para calcular a média da temperatura.
- Calcule a umidade média usando `std::transform_reduce`.
- Calcule a velocidade média do vento, também usando `std::transform_reduce`.

Implementação de std::transform_reduce

- A função `std::transform_reduce` deve ser usada de forma paralela (se possível, usando executores paralelos).
- Trate a soma e a divisão para calcular as médias de forma eficiente.

Entrega: O arquivo `ex2.cpp` deve ser compactado junto com o arquivo do exercício 1 em um arquivo zip, nomeado conforme instruções da tarefa, e entregue no Google Classroom.