

Laboratório 2

Produto interno de dois vetores

Programação Concorrente (ICP-361) 2024-2
Prof. Silvana Rossetto

¹Instituto de Computação/UFRJ

Introdução

Vamos começar nosso segundo laboratório! O objetivo é continuar nossa introdução à programação concorrente abordando **problemas com paralelismo de dados**. Usaremos a linguagem C e a biblioteca *Pthreads*.

Teremos novamente vetores como estrutura de dados de entrada, mas dessa vez precisaremos gerar como saída um único valor. Para isso, o fluxo principal de execução terá que receber os resultados parciais do processamento de cada fluxo secundário, e computar o valor final.

Acompanhe a explanação da professora mostrando: (i) como retornar um valor da função que as *threads* executam; e como gerar e receber vetores de entrada para/de um arquivo binário separado.

Atividade 1

Objetivo: Mostrar exemplos de programas concorrentes em C que retornam o resultado do processamento das threads para o fluxo principal. Abra os arquivos `retorna1.c` e `retorna2.c`. [Acompanhe a explanação da professora.](#)

Atividade 2

Objetivo: Avaliar uma solução concorrente para o problema de somar todos os elementos de um vetor de números reais.

Roteiro:

1. Abra o arquivo `gera_vet_rand.c` que implementa um programa auxiliar para gerar os **vetores de entrada** para os testes da aplicação e o **resultado esperado** (soma de todos os elementos do vetor). [Acompanhe a explanação da professora.](#)
2. Execute esse programa e guarde os resultados em arquivos separados.
3. Abra o arquivo `soma_vetor_conc.c` que implementa um programa concorrente para somar os elementos de um vetor de floats. [Acompanhe a explanação da professora.](#)
4. Experimente o programa usando os arquivos de teste gerados. **As diferentes formas de somar todos os números deram resultados iguais? Por que?**

Atividade 3

Objetivo: Projetar, implementar e avaliar uma solução concorrente para o problema de calcular o **produto interno** de dois vetores de números reais.

Descrição: Sejam $(a_1, a_2, \dots, a_N)^T$ e $(b_1, b_2, \dots, b_N)^T$ vetores em um espaço de dimensão N expressos em termos de um sistema ortogonal de coordenadas cartesianas. O produto interno desses dois vetores é um valor real dado pela equação:

$$a_1b_1 + a_2b_2 + \dots + a_Nb_N$$

.

Roteiro:

1. Comece gerando os casos de teste (vetores de entrada e resultado esperado). Escreva um **programa sequencial em C** que gere dois vetores de entrada (tipo: float) de dimensão N , com valores randômicos (gere números fracionários). Escreva em um **arquivo binário** o valor de N (tipo: inteiro) e os dois vetores ($N * sizeof(float) + N * sizeof(float)$). Depois calcule o produto interno desses dois vetores e escreva o resultado encontrado no mesmo arquivo binário (tipo: double).
2. Certifique-se da corretude desse programa. Execute-o gerando arquivos de teste com diferentes valores de N e os armazene.
3. Escreva um **programa concorrente em C** que receba como entrada o número de threads T e um nome de arquivo, carregue desse arquivo a dimensão N e dois vetores de entrada, execute o cálculo do produto interno desses dois vetores dividindo a tarefa entre as T threads de forma balanceada, e ao final compare o valor calculado com o valor registrado no arquivo de entrada.
4. Calcule a variação relativa considerando como valor de referência o resultado do cálculo sequencial, ou seja:

$$e = \left| \frac{v_s - v_c}{v_s} \right|$$

(onde, v_c : valor do programa concorrente e v_s : valor do programa sequencial).

5. Avalie os resultados encontrados.

Entrega do laboratório:

Disponibilize os códigos implementados na **Atividade 3** em um ambiente de acesso remoto (GitHub ou GitLab). Use o formulário de entrega desse laboratório para enviar o link do repositório do código implementado e responder às questões propostas.