Laboratório 3
Programação concorrente com paralelismo de dados

Programação Concorrente (ICP-361) 2025-2
Profa. Silvana Rossetto

[1]Instituto de Computação/UFRJ

Introdução

O objetivo deste laboratório é continuar nossa introdução à programação concorrente abordando problemas com paralelismo de dados total. Usaremos a linguagem C e a biblioteca *Pthreads*.

Atividade 1

Objetivo: Mostrar exemplos de programas concorrentes em C que retornam o resultado do processamento das threads para o fluxo principal.

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 (thread) e computar o valor final.

Veremos primeiro como **retornar um valor da função que as *threads* executam**.

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: Levantar métricas de desempenho da solução concorrente para o problema de somar todos os elementos de um vetor de números reais.

Roteiro:

1. Gere um conjunto de vetores de entrada, com dimensões distintas.
2. Inclua a tomada de tempo no programa, centrada na parte de processamento da soma.
3. Execute o programa variando o arquivo de entrada. Para cada arquivo de entrada execute com 1, 2, 4 e 8 threads.
4. Registre todos os tempos de execução coletados em uma tabela.
5. Calcule a **aceleração** e **eficiência** e avalie os resultados. Acompanhe a explanação da professora.

Atividade 4 (Exercício 1)

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

Descrição: Sejam $(a_1, a_2, \ldots, a_N)^T$ e $(b_1, b_2, \ldots, b_N)^T$ 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:

$$a_1 b_1 + a_2 b_2 + \ldots + a_N b_N$$

.

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.
2. Escreva em um **arquivo binário** o valor de N (tipo: long inteiro) e os dois vetores.
3. Depois calcule o produto interno desses dois vetores e escreva o resultado encontrado no mesmo arquivo binário.
4. Certifique-se da corretude desse programa. Execute-o gerando arquivos de teste com diferentes valores de N e os armazene.
5. Escreva um **programa concorrente em C** que recebe 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. Para isso, 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).
6. Faça a tomada do tempo de execução do produto interno.
7. Experimente o programa variando os parâmetros de entrada.
8. Avalie os resultados encontrados, **respondendo as questões propostas no formulário de entrega**.

Entrega do laboratório:

Disponibilize os códigos implementados na **Atividade 4** 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 as questões propostas.