

Laboratório 4

Implementação e avaliação de aplicações concorrentes (parte 3)

Computação Concorrente (MAB-117)
Prof. Silvana Rossetto

¹DCC/IM/UFRJ — 22 setembro de 2016

Introdução

O objetivo deste Laboratório é praticar os conceitos estudados para calcular a **soma dos elementos de um vetor de números reais**. Usaremos a linguagem C e a biblioteca *Pthreads*.

Para cada atividade, siga o roteiro proposto e responda às questões colocadas.

Atividade 1

Objetivo: Implementar um algoritmo concorrente para somar todos os elementos de um vetor de floats e imprimir o resultado na saída padrão.

Requisitos do problema:

1. O elementos do vetor deverão ser lidos de um arquivo com o seguinte formato:
 - a primeira linha do vetor contém um valor inteiro que corresponde ao número de elementos dos vetor (N);
 - a linha seguinte contém todos os N elementos do vetor (tipo float) separados por espaço em branco.
2. O nome do **arquivo de entrada** e o **número de threads** deverão ser recebidos como parâmetros de entrada.
3. A solução deve garantir balanceamento de carga entre as threads.

Requisitos do problema:

1. Avalie a corretude do seu programa experimentando-o com vetores de floats onde o resultado da soma é conhecido.
2. Em seguida, experimente seu programa usando como entrada os arquivos que estão na pasta **dados**.

Atividade 2

Objetivo: Implementar uma função no seu programa para somar os elementos do vetor de forma sequencial e comparar a soma calculada com a versão concorrente.

Roteiro:

1. Acrescente no seu código uma função para somar os elementos do vetor de forma sequencial (do primeiro ao último). **Compare os resultados calculados pelas versões concorrente e sequencial** (para os arquivos da pasta “dados”). **Eles são sempre iguais? Explique o que está ocorrendo.**

2. Acrescente no seu código outra função para somar os elementos do vetor de forma sequencial agora **reproduzindo a mesma ordem de associação e soma dos elementos usada na versão concorrente com 2 threads**.
3. Compare o valor calculado pela função sequencial com o valor calculado pela solução concorrente (para cada um dos arquivos de entrada da pasta “dados”). **Os resultados devem ser os mesmos!** (Para essa comparação, você pode considerar apenas o caso de execução concorrente com 2 threads.)