

1 Introdução

Este relatório tem como objetivo apresentar os conhecimentos adquiridos durante a realização da Tarefa 13 da disciplina de **Computação Paralela**. A atividade consistiu em avaliar a escalabilidade do programa desenvolvido na Tarefa 11 — um simulador da velocidade de um fluido utilizando a equação de Navier-Stokes — aplicando diferentes políticas de afinidade de *threads*.

2 Enunciado

Avalie como a escalabilidade do seu código de Navier-Stokes muda ao utilizar os diversos tipos de afinidades de *threads* suportados pelo sistema operacional e pelo OpenMP, no mesmo nó de computação do NPAD utilizado para a Tarefa 12.

3 Desenvolvimento

Na Tarefa 11, desenvolvemos duas versões de um programa para simular a velocidade de um fluido: uma versão sequencial (serial) e outra paralelizada com OpenMP. Para a análise requerida nesta tarefa, utilizamos a versão paralelizada do código.

Nesta tarefa, analisamos os impactos da cláusula `proc_bind()` com as seguintes políticas de afinidade:

- **spread** — distribui as *threads* de forma espalhada pelos processadores, maximizando a distância entre elas. O objetivo é utilizar o máximo de recursos de hardware possível, como diferentes *sockets*, nós NUMA ou núcleos físicos.
- **close** — agrupa as *threads* próximas umas das outras, preferencialmente no mesmo *socket*, nó NUMA ou núcleos adjacentes.
- **master** — todas as *threads* são alocadas no mesmo local onde a *thread* principal (*master/primary*) está executando.
- **true** — herda a política de afinidade da região paralela pai. Se não houver região pai, comporta-se de acordo com a política padrão definida pela implementação.
- **false** — não estabelece nenhuma política de afinidade específica; as *threads* podem migrar livremente entre processadores durante a execução.

Reorganizamos o código de forma a possibilitar o teste de todas as políticas em uma única execução. Assim como realizado na Tarefa 12, o código foi executado com 1, 2, 4, 8, 16 e 32 *threads*, para que ao final pudessemos analisar se houve influência dessas políticas de afinidade na eficiência do código.

O código foi executado no super computador da universidade utilizando o nó com o processador intel-128, cada teste foi executado 6 vezes para termos a certeza da constância dos resultados.

4 Resultados