

## 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 pudéssemos 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

Tendo em vista que a ideia é observar como as politicas de afinidade de *thread* afetam a escalabilidade do nosso programa, faremos uma tabela de escalabilidade para cada politica o metodo para criação dessa tabela já foi visto anteriormente na tarefa 12.

### 4.1 Default

# cores	20x20x20	20x20x40	20x20x80	20x20x160	20x20x320	20x20x640
1	1.00	1.00	1.00	1.00	1.00	1.00
2	0.84	0.86	0.90	0.91	0.92	0.96
4	0.67	0.71	0.76	0.78	0.81	0.83
8	0.45	0.51	0.56	0.59	0.66	0.65
16	0.25	0.32	0.39	0.44	0.35	0.50
32	0.11	0.15	0.19	0.22	0.25	0.27

Table 1: Tabela de escalabilidade da política default.

### 4.2 Spread

# cores	20x20x20	20x20x40	20x20x80	20x20x160	20x20x320	20x20x640
1	1,00	1,00	1,00	1,00	1,00	1,00
2	0,85	0,87	0,90	0,91	0,93	0,96
4	0,67	0,72	0,76	0,78	0,80	0,83
8	0,46	0,52	0,57	0,59	0,66	0,65
16	0,26	0,33	0,39	0,44	0,49	0,50
32	0,11	0,16	0,19	0,22	0,25	0,27

Table 2: tabela de Escalabilidade da Política Spread

### 4.3 Close

# cores	20x20x20	20x20x40	20x20x80	20x20x160	20x20x320	20x20x640
1	1,00	1,00	1,00	1,00	1,00	1,00
2	0,85	0,87	0,90	0,91	0,93	0,96
4	0,67	0,72	0,76	0,78	0,81	0,83
8	0,45	0,52	0,57	0,59	0,66	0,65
16	0,25	0,33	0,39	0,44	0,35	0,50
32	0,11	0,16	0,19	0,22	0,26	0,27

Table 3: tabela de Escalabilidade da Política Close

#### 4.4 Master

# cores	20x20x20	20x20x40	20x20x80	20x20x160	20x20x320	20x20x640
<b>1</b>	1,00	1,00	1,00	1,00	1,00	1,00
<b>2</b>	0,85	0,87	0,90	0,91	0,93	0,96
<b>4</b>	0,67	0,72	0,76	0,78	0,80	0,83
<b>8</b>	0,45	0,52	0,57	0,59	0,65	0,65
<b>16</b>	0,25	0,33	0,39	0,44	0,35	0,50
<b>32</b>	0,11	0,15	0,19	0,22	0,26	0,27

Table 4: tabela de Escalabilidade da Política Master