

**Universidade de São Paulo  
Instituto de Ciências Matemáticas e de Computação  
Departamento de Sistemas de Computação  
Laboratório de Sistemas Distribuídos e de Programação  
Concorrente**

***Caderno de Desafio para Programação Paralela***

***Caderno Desafio C/MPI/OpenMP***

**Quantos Elementos Há nas Colunas da Matriz que São Menores que a Média da Matriz**

por

*Paulo Sérgio Lopes de Souza*

Baseado em desafio anterior feito por Guilherme Martins e Paulo S. L. de Souza

Este caderno de desafio representa um Recurso Educacional Aberto para ser usado por alunos e professores, como uma introdução aos estudos de programação paralela com C/MPI/OpenMP. Este material pode ser utilizado e modificado desde que os direitos autorais sejam explicitamente mencionados e referenciados. Utilizar considerando a licença *GPLv2 (GNU General Public License version 2)* ou *posterior*.

São Carlos/BR, dezembro de 2020

# 1. Desafio

O objetivo deste desafio é desenvolver um algoritmo paralelo em C/MPI/OpenMP que determine quantos elementos das colunas da matriz são menores que a média da matriz. Para sua solução, considere uma matriz quadrada **MAT** de valores inteiros positivos e uma média aritmética simples **MED** dos elementos de **MAT**. Determine quantos elementos existem em cada coluna de **MAT** que são menores que **MED**.

A aplicação concorrente executará com **P** processos **MPI** e cada processo com **T** threads **OpenMP**, onde considera-se que a dimensão **N** de **MAT** é bem maior que **P** e **T**. O valor de **P** pode ser determinado pelo **mpirun** com o parâmetro **-np <P>**. O valor de **T** pode ser dinâmico, i.e., em função do número de núcleos do computador onde o processo criador das threads vai executar (em outras palavras, não precisa especificar o número **T** de threads, deixe o **OpenMP** escolhê-lo em função do processador usado).

Os valores de **N** e **MAT** serão obtidos a partir de um arquivo de entrada pelo processo **MPI** de *rank* 0. Os valores de **MAT** variam entre 0 e 99, inclusive.

O exemplo a seguir ilustra a especificação descrita para uma matriz **MAT** com **N = 5**:

MAT					MAT espalhada nos processos MPI (média 39,88)								vet_menor[ ]		
em p0					p0		p1		p2		p3		p4	em p0	
30	07	90	88	18	<b>30</b>		<b>07</b>		90		88		<b>18</b>	[0]	3
05	90	54	51	90	<b>05</b>		90		54		51		90	[1]	4
90	05	05	05	27	90		<b>05</b>		<b>05</b>		<b>05</b>		<b>27</b>	[2]	1
47	22	44	17	05	47		<b>22</b>		44		<b>17</b>		<b>05</b>	[3]	2
12	33	49	90	23	<b>12</b>		<b>33</b>		49		90		<b>23</b>	[4]	4

Os dados lidos do arquivo de entrada para este exemplo acima são:

```

5
30 07 90 88 18
05 90 54 51 90
90 05 05 05 27
47 22 44 17 05
12 33 49 90 23

```

A saída da aplicação impressa em **stdout** deve exibir **vet\_menor[ ]**. A saída esperada para os dados acima é:

```

3 4 1 2 4

```

Há um espaço em branco entre cada número e também após o último. Há uma quebra de linha (“\n”) ao final da linha.

Para executar no *bash*, por exemplo, utilize este padrão:

```
mpirun -np 4 menorescol entrada.txt
```

Obs: na linha de comando acima, considera-se que o programa foi inserido em *menorescol.c* e o executável chama-se *menorescol* e está no diretório atual. O número de processos (*-np 4*) é um exemplo, não um requisito; o algoritmo deve funcionar para um número variado de processos. Os dados da matriz estão em *entrada.txt*, seguindo o padrão já explicado.

## 2. Um ponto de partida para a solução do desafio

A implementação sequencial do desafio e um arquivo de entrada estão em anexo.