

## 1 Introdução

**Programação Paralela.** O objetivo deste exercício é familiarizar os alunos com um ambiente para a programação distribuída. Veremos, como criar, para um problema real, um programa paralelo.

**Ferramentas.** Neste EP deverá ser usada a biblioteca de comunicação MPI-LAM, em conjunto com a linguagem C.

**Problema.** Em algumas áreas da computação, como a biologia computacional, é importante encontrar combinações de elementos que obedecem a um critério específico. Para este EP queremos encontrar a quantidade de combinações de inteiros que satisfaçam uma dada propriedade.

**Entrada.** Os dados de entrada do programa serão dados em um arquivo com a seguinte forma:

```
4 2
10 5 8 1
```

Onde na primeira linha são dados o número total de elementos, e os tamanhos das combinações, e na segunda linha são dados os elementos a serem combinados.

Como saída o programa deverá imprimir a quantidade de combinações cuja soma seja um número primo, a quantidade de combinações que seja potência de 2 e a quantidade de combinações onde a média dos valores for igual a sua mediana<sup>1</sup>

**Requisitos.** O programa deve ser capaz de ser executado, em até pelo menos quatro máquinas. A leitura dos dados e a impressão dos resultados deve ser feita pela mesma máquina, a partir de um nome de arquivo dado na linha de comando, a qual também será responsável por imprimir o tempo total de processamento (usando a primitiva `MPI_Wtime`) como abaixo:

```
double starttime, endtime;
MPI_Init( &argc, &argv );
starttime = MPI_Wtime(); // O tempo começa logo após o Init
.... stuff to be timed ...
endtime = MPI_Wtime(); // O tempo termina logo antes do Finalize
MPI_Finalize();
printf("That took %f seconds\n",endtime-starttime);
```

**Desafio.** O grupo que apresentar o melhor EP vai receber como prêmio a média final 10.0. Supõe-se que o melhor produza resultados corretos para o conjunto de testes de avaliação. Para a escolha do melhor serão realizados os seguintes testes: Dados três conjuntos de dados, os programas que obtiverem o melhor desempenho em quatro máquinas homogêneas serão classificados para a fase final.

Na fase final, os programas serão executados três vezes para cada conjunto de testes. Será considerado o tempo de execução médio. O programa que obtiver o maior número de melhores tempos médios será considerado vencedor. Os dados de desempenho serão divulgados na página da disciplina. Para eventuais desempates, no caso de desempenhos muito semelhantes, serão usados cenários com mais máquinas, ou mesmo com máquinas não homogêneas.

---

<sup>1</sup>Se o número de elementos for ímpar corresponde ao valor central dos mesmos ordenados. Se for par é a média dos valores centrais dos valores ordenados.