

### Trabalho 03

Este trabalho consiste na adição de instruções de paralelismo para sistemas de memória compartilhada (utilizando o OpenMP) no código `laplace.cxx` (o mesmo utilizado no segundo trabalho da disciplina).

O código deverá ser compilado com o compilador da família GNU (g++) apenas com a adição da flag `-fopenmp` para habilitação das diretivas do OpenMP a serem adicionadas ao código. Deverá ainda ser incluída a biblioteca `<omp.h>`.

Após as modificações o código deverá ser executado para os casos em que  $nx = 512, 1024, 2048$ ,  $n_{iter} = 1000$  e  $eps = 10^{-16}$ .

As adições das diretivas de OpenMP deverão ser feitas com base nos resultados da perfilagem feita para o Trabalho 02, que permitem a identificação dos loops computacionalmente mais intensivos.

O relatório (em formato **.pdf**) deverá ser redigido contendo os seguintes itens:

- Identificação e listagem dos loops mais intensivos onde deverão ser feitas as alterações.
- Justificativa das modificações/instruções feitas utilizando o OpenMP. Ex.: escolha das variáveis privadas, compartilhadas, variáveis que sofrerão instruções de redução e etc.
- Para cada dimensão de grid ( $nx$ ) deverão ser feitas as comparações do desempenho com relação ao código base (sem utilização do OpenMP), sendo necessária a apresentação dos devidos comentários com relação à escalabilidade do código.
- Por fim, os códigos deverão ser recompilados com a adição da flag `-O3` com e sem a presença da flag `-fopenmp`. Deste modo, teremos as versões do código:
  - puramente serial (sem nenhuma flag).
  - código com vetorização (`-O3`)
  - código com paralelismo de threads (`-fopenmp`)
  - código com paralelismo de threads + vetorização (`-fopenmp -O3`)

Além disso, passos e modificações adicionais devem ser adicionados ao relatório.

#### Observações:

- Deverá ser utilizada a função `omp_get_wtime()` para medição do tempo de execução de códigos instrumentados com OpenMP.
- Para este código em particular, o erro obtido deverá ser o mesmo para todas as versões do código (serial, vetorizada e com paralelismo por threads (OpenMP)).
- O número de threads deve ser variado de 1 até o máximo disponível no sistema utilizado. Isto pode ser controlado por meio da variável de ambiente `OMP_NUM_THREADS`.