

Objectivos:

Com esta sessão pretende-se introduzir os fundamentos do processamento paralelo.

Exercícios:

O “standard” OpenMP permite a especificação do paralelismo explorável num programa através de diretivas. A diretiva mais utilizada é a indicação de que as iterações de um ciclo podem ser executadas em paralelo.

Para utilizar o OpenMP deve adicionar `#include<omp.h>` no início do ficheiro e compilar o programa acrescentando `-fopenmp` na linha de comando da compilação (não se esqueça de alterar `-O3` na `makefile`). Pode gerar uma versão paralela ou sequencial incluindo ou retirando esta opção.

- a) Inclua a diretiva `#pragma omp parallel private(x,y)` na rotina `convolve3x1`. Para verificar a forma como o trabalho, ou seja, as iterações do ciclo são executadas nos núcleos disponíveis inclua a seguinte linha entre os dois ciclos da rotina e execute o programa.

```
printf("Thread %d of %d executing line %d\n", omp_get_thread_num(),
      omp_get_num_threads(), x);
```

As primitivas `omp_get_thread_num` e `omp_get_num_threads` permitem, respetivamente, saber o número da atividade paralela (*threads*) em execução e o número total. Observe e interprete o resultado de várias execuções utilizando uma imagem de 256x256.

```
./convolve AC_images/abe_natsumi256.pgm result.ppm 1
```

- b) Altere a rotina `convolve3x1` para que cada atividade paralela execute metade das iterações do ciclo, introduzindo uma chamada à rotina `omp_get_thread_num` (dividindo assim o trabalho pelas duas atividades). Verifique a semelhança com o resultado obtido pela diretiva `#pragma omp parallel for private(x,y)`. Comente o resultado de várias execuções do programa.
- c) Meça o tempo de execução da rotina `convolve3x1` e calcule o ganho obtido, nesta rotina, com a execução em paralelo. O tempo da rotina pode ser medido com o PAPI (nota: não se esqueça de remover o `printf` da rotina para efetuar a medição dos tempos de execução):

```
#include<papi.h>
....
long long PAPI_start, PAPI_stop;
PAPI_start = PAPI_get_real_usec();
<< segmento de código a medir >>
PAPI_stop = PAPI_get_real_usec();
printf("Time in microseconds: %lld\n", PAPI_stop - PAPI_start);
```

- d) Compare agora o ganho obtido comparando o tempo de execução de **todo** o programa. Comente os resultados.
- e) Troque a ordem dos ciclos “for” na rotina `convolve3x1` e repita a alínea c). Comente os resultados obtidos.
- f) Inclua a diretiva `#pragma omp parallel for` também ciclo interior e comente os resultados.
- g) Acrescente a cláusula `schedule(dynamic)` no final da diretiva `#pragma omp parallel for` e verifique a forma com as iterações do ciclo são mapeadas nas atividades paralelas.