

Computação Paralela e Distribuída

Aula de Laboratório Nº 3

Coordenação de Engenharia Informática
Departamento de Engenharias e Tecnologias
Instituto Superior Politécnico de Tecnologias e Ciências

Objectivo

- Introduzir o OpenMP

Introdução

Ao longo desta aula o estudante é encorajado a consultar a documentação em <http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf> sempre que tiver dúvidas sobre as directivas ou APIs do OpenMP.

Problema 1

Este problema tenta familiarizar os estudante com o ambiente de trabalho e a compilação do código OpenMP. Considere o seguinte programa (omp1.c):

```
1. int main(int argc, char *argv[]) {
2.
3.     int i, tid;
4.
5.     #pragma omp parallel private(i,tid)
6.     {
7.         tid = omp_get_thread_num();
8.
9.         #pragma omp for
10.        for(i = 0; i < NUMITER; i++)
11.            printf("Thread: %d\titer=%d\n", tid, i); fflush(stdout);
12.
13.        printf("Thread %d, almost...\n", tid); fflush(stdout);
14.        printf("Thread %d, done!\n", tid); fflush(stdout);
15.    }
16.
17.    printf("All threads have finished!\n");
18. }
```

- a) Comece por compilar o código acima, não esqueça de adicionar a flag de compilação “-fopenmp”. Execute este programa com um número diferente de threads (definindo OMP_NUM_THREADS).

- b) Adicione a cláusula “`nowait`” à directiva `for` na linha 9. **A saída foi alterada? Porquê?**
- c) Adicione uma barreira (`#pragma omp barrier`) entre as linhas 13 e 14. Compare a saída com a) e b).

Problema 2

No segundo problema, o estudante é solicitado a paralelizar um loop interno com dependências de dados. O código básico que deve ser executado em paralelo é o seguinte:

```
for(iter = 0; iter < numiter; iter++)
{
    for (i=0; i<size-1; i++)
    {
        V[i] = f(V[i], V[i+1]);
    }
}
```

onde definimos uma função simples para manipular os dados.

```
#define f(x,y) ((x+y)/2.0)
```

- a) Compile e execute a versão serial do código (omp2.c).
- b) Crie uma versão paralela simples (e incorreta) adicionando uma directiva paralela ao loop interno. Tente encontrar uma execução com uma saída diferente da versão serial. **Porquê isso pode acontecer?**
- c) Faça a correcção da versão paralela copiando o vector `V` da iteração `i-1` antes de executar a iteração `i` e usando os valores copiados como argumentos para a função `f` na iteração `i`.
- d) Escreva uma modificação na versão paralela que evite copiar o vector `V` em todas as iterações.