Universidade de Aveiro

Mestrado Integrado em Engenharia Computacional

Computação Paralela

Lesson 3: OpenMP

Academic year: 2019/2020 Nuno Lau/Manuel Barroso

OpenMP is an API that allows the development of explicit multi-threaded shared memory parallelism. In this Lesson we will perform a few exercises using OpenMP to enhance the performance of serial programs. The base code for this lesson is provided in the package **`cp_openmp.zip`**.

1. Unpack cp_**`openmp.zip`**.
2. Read **`hello.c`** carefully.
   2.1. Modify **`hello.c`** to create an OpenMP parallel region. Compile and execute.
   2.2. Change the value of the environment variable **`OMP_NUM_THREADS`** to different values and execute **`hello`** again.
   2.3. Use the clause **`num_threads`** to control the number of threads in the parallel region.
   2.4. Use the function omp_set_num_threads () to control the number of threads in the parallel region
3. The program **`pi.c`** determines the value of π by integrating numerically:

$$\int_0^2 \frac{4}{1+x^2} dx$$

   3.1. Compile **`pi.c`** and execute it.
   3.2. Use **`#pragma omp parallel`** (without using **`omp for`**) to create a parallel version of this algorithm. Determine the speedup.
   3.3. Use **`#pragma omp for`** with mutual exclusion constructs to create a parallel version of this algorithm. Determine the speedup.
   3.4. Use **`omp for`** and **`reduction`** construct to create a parallel version of this algorithm. Determine the speedup.
4. The program **`mandel.c`** determines (incorrectly) the area of the Mandelbrot set. The Mandelbrot set is defined by the set of complex numbers *c* for which the iterating:

$$z_{n+1} = z_n + c \quad \text{with} \quad z_0 = 0$$

   keeps $|z_n| < \infty$ as *n* tends to ∞. The shape of the Mandelbrot set is a fractal.
   4.1. Debug and correct the program, so that the area is correctly determined.
5. The program **`linked.c`** uses a linked list in which each node determines a number in the Fibonacci sequence.
   5.1. Create a parallel version of this program using OpenMP without **`task.`** Determine the speedup.

5.2. Create a parallel version of this program using OpenMP using **`task`.** Determine the speedup.

Bibliography

[1] "A 'Hands-on' Introduction to OpenMP", Tim Mattson, Intel Corp., timothy.g.mattson@intel.com