
OpenMP 2

1 The Mandelbrot Set

The Mandelbrot set is a fractal that can be computed by iteration over points in the complex plane and checking whether the sequence $z_n = z_n^2 + c, z_0 = 0$ remains bounded (where c is the complex number corresponding to the current pixel).

In this task you will practice with data-sharing attributes, race conditions, and synchronization constructs. We provide you with the program `omp-2-mandelbrot.c`¹ which contains at least 4 bugs.

- We ask you to identify and fix these bugs. We also ask you to list the bugs you found and explain which problem they were causing and how you fixed them.
- Test with different number of threads.

2 Fibonacci

In this exercise, we provide you with a program to calculate the Fibonacci sequence. This program is, compared to many example Fibonacci examples, already quite advanced: It runs in linear time instead of being a recursive, exponential formulation. However, that also means that the code has a flow dependency. Luckily, there exists a closed form solution to the fibonacci sequence, in the form of *Binet's formula*. While that formula is relatively expensive to evaluate, it is OK to do so once or twice per thread. The formula is:

$$F_n = \frac{\Phi^n - \Psi^n}{\sqrt{5}}$$

with $\Phi = \frac{1+\sqrt{5}}{2}$ (the golden ratio) and $\Psi = \frac{1-\sqrt{5}}{2}$.

- Parallelize the program using the constructs we have learned about in class.
- Remove the dependency by using the closed form solution.

¹This program was written by Mark Bull and Tim Mattson and is often used in OpenMP tutorials to illustrate a number of common bugs.

3 Dependencies: Anti and Output

In the last exercise, we looked at a program that contained a true/flow dependency (the parallel scan). This time around, we will look at the other two categories of dependencies: Anti and output dependencies.

We provide two files, `omp-2-dep-anti.c` and `omp-2-dep-output.c`.

- Try to parallelize the program using the constructs we have learned about in class.
- Remove the dependency in an appropriate manner.
- Think about ways to reduce the memory footprint of your solution.