

Parallel Design Patterns: Exercise 2b

Lawrence Mitchell

January 13, 2012

Domain decomposition

In this exercise, you will extend your code from exercise 2a to carry out the domain decomposed calculations in parallel using MPI. For this practical, you can either start with your finished code from exercise 2a, or use the template code for exercise 2b. It may be instructive to look at the template code to see if you've made similar design decisions.

Template code

The template code is available in WebCT as `exercise2b.tar`. Unpack this in the same place that you unpacked exercise 1. You should now have an `exercise2b` directory inside the `mandelbrot` directory. Again, you have the option of carrying out the exercise in either C or Fortran: choose the appropriate subdirectory.

The template code takes care of initialising MPI and provides utility functions to send (`send_image_slice`) and receive (`recv_image_slice`) an image slice. These assume that the domain has been sliced in the Y direction. To ease computation of domain boundaries, a further utility function `calc_slice_bounds` is also provided.

As before `compute_mandelbrot_set` divides up the domain into slices, computes each slice and then copies it into the global image. This time, however, it is carried out in parallel.

You need to implement the functions `compute_mandelbrot_slice` and `copy_slice_to_image`. The former should be very similar (if not identical) from the version you wrote for exercise 2a. The latter will require some modification: you may wish to use the provided utility functions for sending and receiving image slices, or you can do it yourself.

Debugging problems

As for exercise 2a, the likelist source of problems is incorrect copying of image slices into the global image. Again, a good check is to just set each slice's values to the slice number. You can then reasonably easily check if slice values are ending up in the correct place.