MPI: Practical session 5

1 Mandelbrot

Download all Mandelbrot examples from `http://www.tcm.phy.cam.ac.uk/~mjr/courses/MPI/index.html` In particular, try the `mandel_mpi` and `mandel_mpi_ms` examples.

Compare and contrast their performance, by running them on 1, 2, 4, 8, 16 processors. Plot a graph of

$$\frac{\text{Run-time on 1 processor}}{\text{Run-time on } P \text{ processors}} \tag{1.1}$$

against P.

1.1 Area calculation

The area of the Mandelbrot Set is unknown there are merely numeric approximations to it, and some analytic upper and lower bounds. Can you modify the code so that each rank calculates the area of Mandelbrot set that it has been asked to calculate, and then sum these with an `MPI_Reduce`?

In order to provide better performance, consider removing the code that is used to output the data, so that (virtually) all of the run-time is spent doing useful computations, not waiting for file I/O.

(The accepted answer is in the region of 1.506592. We would expect a slight bias to overestimating, for the mathematical definition of z_∞ being finite is stricter than our definition of $|z|$ not exceeding two after 320 iterations.)

For the master-slave version, consider letting each slave keep count of the number of rows it has calculated, and print this, together with its rank, immediately before exiting. Are they all the same? Are they all the same if the resolution is not divisible by the number of slaves? Does the algorithm adapt well when the resolution is not divisible by the number of slaves?

Can you modify the Mandelbrot set generator to make it easier to specify a region of interest without recompiling? If exploring the set, one may find that setting the bottom left corner to $-0.75 + 0.15i$ with a range of about 0.015, or the bottom left at $-0.42 + 0.57i$ with a range of about 0.12, produces something pretty. One may well wish to improve on the colour map used. As one zooms in further, it can be useful to increase the maximum number of iterations. However, eventually the precision of double precision arithmetic becomes an issue, and the set stops appearing fractal but becomes smoothed out. (If tempted to print something, try to avoid printing large areas of dark colour, as toner is not free.)

2 Laplace

Download all Laplace examples from `http://www.tcm.phy.cam.ac.uk/~mjr/courses/MPI/index.html`

Compare and contrast their performance on varying numbers of processors. Try changing the `SIZE` parameter and see whether the relative performance improves.