School of Computing, University of Leeds, 2013–2014

Parallel Scientific Computing COMP3920

Embarrassingly Parallel Problems: Linear Algebra

Introduction

This worksheet is designed to help you practice developing simple parallel programs to solve problems which can easily be split into independent processes. The supplied codes calculate the sum and the scalar product of two vectors, *i.e.*

$$\mathbf{c} = \mathbf{a} + \mathbf{b} \qquad \text{and} \qquad d = \mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i \,,$$

where each vector contains n elements and a_i, b_i represent the i^{th} elements of the vectors $\mathbf{a}$, $\mathbf{b}$ respectively.

Two data files, `vecs_small.dat` and `vecs_large.dat`, are provided with this worksheet to allow you to test the programs. In each file, the first line contains the size of the two input vectors, respectively 12 and 27720, while the remaining lines contain the pairs of entries of the two vectors, $\mathbf{a}$ and $\mathbf{b}$. Note that, in the larger of the two, all of the entries are 1.0, so in each case it is easy to calculate the answers you *should* get at each stage of the following exercises.

Addition of Two Vectors

The file `vector_addition.c` contains a parallel code for reading in two vectors from a file and then adding them. It assumes that the length of the vectors is divisible by the number of processes: note that 12 is divisible by the integers 1 to 4 inclusive, while 27720 is divisible by the integers 1 to 12 inclusive. The code finishes by writing the first 12 elements of the resulting vector to the screen.

1. Test the code with the two data files and confirm that it works.

2. The code is again supplied without comments so a good exercise is to go through the code line by line and insert appropriate comments to explain what the code does. In particular you should understand how the array of data has been divided up between the processes.

3. Describe how you would modify the code to remove the artificial constraint that the value of n has to be an exact multiple of the number of processes.

Scalar Product of Two Vectors

The file `serial_scalar_product.c` contains a serial code for reading in two vectors from a file and then calculating their scalar product. The following exercises should be carried out with reference to the parallel code provided in `vector_addition.c`, and using the data files `vecs_small.dat` and `vecs_large.dat` to test your program as you proceed.

1. Using the files `serial_scalar_product.c` and `vector_addition.c` construct a code for calculating the scalar product of two vectors which runs in parallel using MPI.

   (a) The vectors **a** and **b** should be read in by process 0.

   (b) The computation required to calculate the scalar product should be divided evenly between all of the processes (including process 0) using block decomposition. Each process must be sent the size of the vectors (n) and the elements of the vectors corresponding to its block of calculation.

   (c) Once the partial sums have been calculated by each process they should be added together, with the result sent to process 0, using the `MPI_Reduce` function.

   (d) The scalar product should be printed to the screen by process 0.

   You should confirm that each process is carrying out the correct calculations by printing out their partial sums to the screen and checking that they are correct.

2. Supplementary Exercise: It is likely that you have used `MPI_Send` and `MPI_Recv`:

   (a) modify your code so that it only uses global communication functions, such as `MPI_Scatter`;

   (b) modify `vector_addition.c` so that it also only uses global communication functions.

All of your results for this worksheet should be obtained using multiple processes on a single processor (simply to avoid running processes on other machines).