Goethe-University
Frankfurt am Main

Lab Parallelization · Summer Semester 2017

Prof. Dr. Ulrich Meyer
Dipl. Inf. Gerhard Leuck

Assignment 3

Hand out: 23.05.2017
Hand in the tasks at ppva-tut@informatik.uni-frankfurt.de
Task 1 and Task 2: 5.06.2017

Task 1

Solving a linear system of equations with the Jacobi Method

In numerical linear algebra, the Jacobi method is an algorithm for determining the solutions of a diagonally dominant system of linear equations. We seek the solution to a set of linear equations, written in matrix terms as $Ax = b$ with $A = (a_{ij})$, b=(b_i), i=1...n, j=1...n and $|a_{ii}| > \sum_{j=1,j\neq i}^{n} |a_{ij}|$.

Let $A = D + (L + U)$ where D, L, and U represent the diagonal, lower triangular, and upper triangular parts of the coefficient matrix A. Then the equation can be rephrased as:

$$Ax = Dx + (L + U)x = b.$$

Moreover,

$$x = D^{-1}\left[b - (L + U)x\right],$$

if $a_{i,i} \neq 0$ for all i. By iterative rule, the definition of the Jacobi method can be expressed as:

$$x^{(k+1)} = D^{-1}\left[b - (L + U)x^{(k)}\right]$$

where k is the iteration count. Often an element-based approach is used:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{(k)} - \sum_{j=i+1}^{n} a_{i,j}x_j^{(k)}\right), i = 1, 2, ..., n.$$

Write an MPI program that solves a set of linear equations $Ax = b$ with the Jacobi method that converges if the distance between the vectors $x^{(k)}$ and $x^{(k+1)}$ is small enough.

$$\sum_{i=1}^{n} \left|x_i^{(k+1)} - x_i^{(k)}\right| < \epsilon$$

You can assume that the number of rows of the matrix can be devided by the number of processes. The root process reads matrix A and all processes read the vector b from files. The file names and

ϵ have to be specified by the user as parameters. After calculation the root process writes the result vector to a file. Use MPI_IO functions for read and write files. You can find example files in:
/home/lab/2017/src/3

For distributing the specific raws of matrix A to the other processes use a colletive communication operation. Tranfer only necessary values (not broadcasting all values). For updating the iteration vector x use also a colletive communication operation.

Task 2

Modify the program from Task 1:
Each process reads its part of the matrix A as a block by columns from a file using split collective file operations with shared file pointers. For reading the vector b use MPI_File_read_ordered. Use collective communication and global reduce operations.