COMP-150-06 Instructor: Leopold Grinberg

Home assignment No. 3

Create a parallel program to:

Create a parallel code which will execute the following:

1. Read 2D symmetric array **A** from a disk, such that each MPI rank stores only a particular block of the array. For example, rank 0 stores A(0:9,0:9) rank 1 stores A(0:9,10:19) and so on. Use N processors aligned in NprxNpc two-dimensional grid where Npr is the number of rows, Npc is the number of processors column and Npr*Npc = N.

2. Redistribute the 2D array **A** stored block wise to obtain two-dimensional block-cyclic distributions. Here the data must be communicated using MPI library. It is up to you what MPI routine to use.

You may get additional information on 2D-block cycling from the following web-page:
http://www.netlib.org/scalapack/slug/node75.html#SECTION04431000000000000000
and
http://www.netlib.org/scalapack/slug/node76.html

You may start from writing a program for 1D row-cyclic distribution.
Then you may continue with writing a code to perform 1D column-cyclic distribution.
And then write a code to store **A** using the 2D block-cycling.
You can use the basic blocks of a size 2, 4 and large. Small size basic blocks allows easier testing for the correctness of the results

You can test the software on any computer and using any compiler.
The goal here is not to achieve the best performance but to learn how to distribute a 2D array among processors using the 2D block-cycling.

Submit a report presenting your strategy for performing the 2D cycling distribution (not more then 1.5 pages. Submit your code as well.

FYI:
Your next assignment will include implementation of the ScaLapack library to perform LU decomposition and matrix inversion in parallel.