

Parallel Programming Assignment 3 W

Daniel Crankshaw

1)

I would characterize this as a data distribution problem. The data is run on each process, and the division of labor is such that each process is responsible for processing. This is a data distribution problem that works well when the program needs to coordinate many processes. There are both data sharing and ordering dependencies between processes. Some of the local data of its (spatial) neighbors is needed directly below it in the global grid, which is then passed to the next process (in the circular message passing scheme). Also, there are ordering dependencies between processes. Local data before they can share it with their neighbors. Once calculated the entire new iteration's coordinates are passed to the next iteration. In other words, all processes must wait for the data from the previous iteration to be passed to the next one. This ordering dependency is a data distribution problem.

This is a geometric decomposition problem. It is a linear decomposition problem, and is a SPMD (single process multiple data) problem. The use of geometric decomposition.

2)

- c. I am going to exclude process 0 from the discussion. In both decomposition schemes, process 0 needs to print the results for each iteration so it can print the total. Process 0 needs two copies of its local data to be able to keep track of the old iteration until the new iteration is ready. It needs $2 * \text{ROWS}/N * \text{COLUMNS}$ (or $2 * \text{C} * \text{R}/N$) above and just below it. This adds $2 * \text{ROWS}/N * \text{COLUMNS} + 2 * \text{C}$ to the total. Process 0 needs 2 copies of the local data to be able to do the geometric decomposition. But it also needs to keep adjacent rows and columns, plus the total of $2 * \text{C} * \text{R}/N + 4 * \text{R}/N^{.5} + 4$.
- d. The geometric decomposition is more efficient for rectangular (e.g. no longer square) grids. The message passing scheme is more efficient for square (being passed). Also, I believe process 0 being flexible is because the grid is arbitrary. Doing it about it recursively is difficult and the geometric decomposition being much more efficient would first deal with transmitting the data.