CISC 372: Parallel Computing

University of Delaware, Spring 2022

HW 4

This assignment is due Wednesday, March 16, at noon. Topics: block distribution, point-to-point communication, collective communication

Create a directory `hw04` under the `hw` directory of your personal repository.

1. Problem 1: Standard Block Distribution Scheme Exercise

Create a subdirectory of `hw04` named `sbd`. For this problem, you will answer questions in plain text files in `hw04/sbd`.

Compute the complete standard block distribution solution for each of the following values of n (number of elements or tasks) and p (number of processes).

n	p
18	4
18	5
10	4
4	4
2	5
3	5
1	4

Enter your answer in a plain text file named `blocks.txt`. Use the following format:

```
n=10, p=3:  0 1 2 | 3 4 5 | 6 7 8 9
```

Follow this format exactly: `n=` followed by the value of n, a comma, a space, then `p=` followed by the value of p, then a colon, followed by some white space (the exact amount does not matter), followed by the sequence of global indexes owned by process 0 (separated by some white space), a vertical bar, the sequence of global indexes owned by process 1, a vertical bar, . . . , followed by the sequence of global indexes owned by process $p-1$. There should be exactly $p-1$ vertical bars in a solution involving p processes. Your text file should contain exactly one line of the form above for each of the rows in the table.

Suppose $p = 727$ and $n = 100,000,000$. What is the local address of the element with global index $35,456,989$? Enter you answer in a plain text file named `local.txt` using the following format:

```
X
Y
```

where `X` is the rank of the process owning the element, and `Y` is the local index of the element on that process. Again, follow the format exactly. Your solution should contain exactly two lines, each containing a single integer and terminated by a newline.

Suppose again $p = 727$ and $n = 100,000,000$. What is the global index corresponding to the element with local index $10,000$ on the process of rank 656? Enter you answer in a text file named `global.txt` as a single number followed by a newline, i.e.:

```
X
```

where `X` is the global index.

Be sure to commit three plain text files in `sbd`: `blocks.txt`, `local.txt` and `global.txt`.

2. Problem 2: 2d-diffusion striped decomposition

Create directory `diff2d` in `hw04`. You are given a slightly simplied version of the sequential 2d-diffusion code `diff2d.c`. (In this version, the boundary updates are performed in a simpler way, but the result is similar to the original.) Read this program and make sure you understand it.

You are to create an MPI version `diff2d_mpi.c` that uses a striped decomposition. Each process is responsible for some number of columns of the square. See Figure 1 for an example with $n = 8$ and $p = 3$. The figure shows exactly how each cell of u in the sequential program corresponds to a cell of u on one of the processes of the parallel program.

Each process also maintains two columns of ghost cells to mirror the columns on the neighboring processes. The ghost cells are needed in order to update the cells along the process boundary. At each time step, a ghost cell exchange takes place, then the update takes place. Review the video on distribution and nearest neighbor communication if necessary.

You are given a skeleton version of `diff2d_mpi.c` which you may copy and complete. Since you have a correct sequential version, you can always check your results by diffing the `anim` files produced by the sequential and parallel programs. If your parallel version is correct, the output should be identical. The provided `Makefile` will do this automatically by `make test`.

We recommend that you test and debug starting with very small configurations (like the one in Figure 1). Examine the output with `anim2txt`. Try different numbers of processes. Scale up when you feel you have it right, continuing to check by diffing with the sequential version. You may use any machine you like to develop the program, including `bridges2` and `cisc372.cis.udel.edu`.

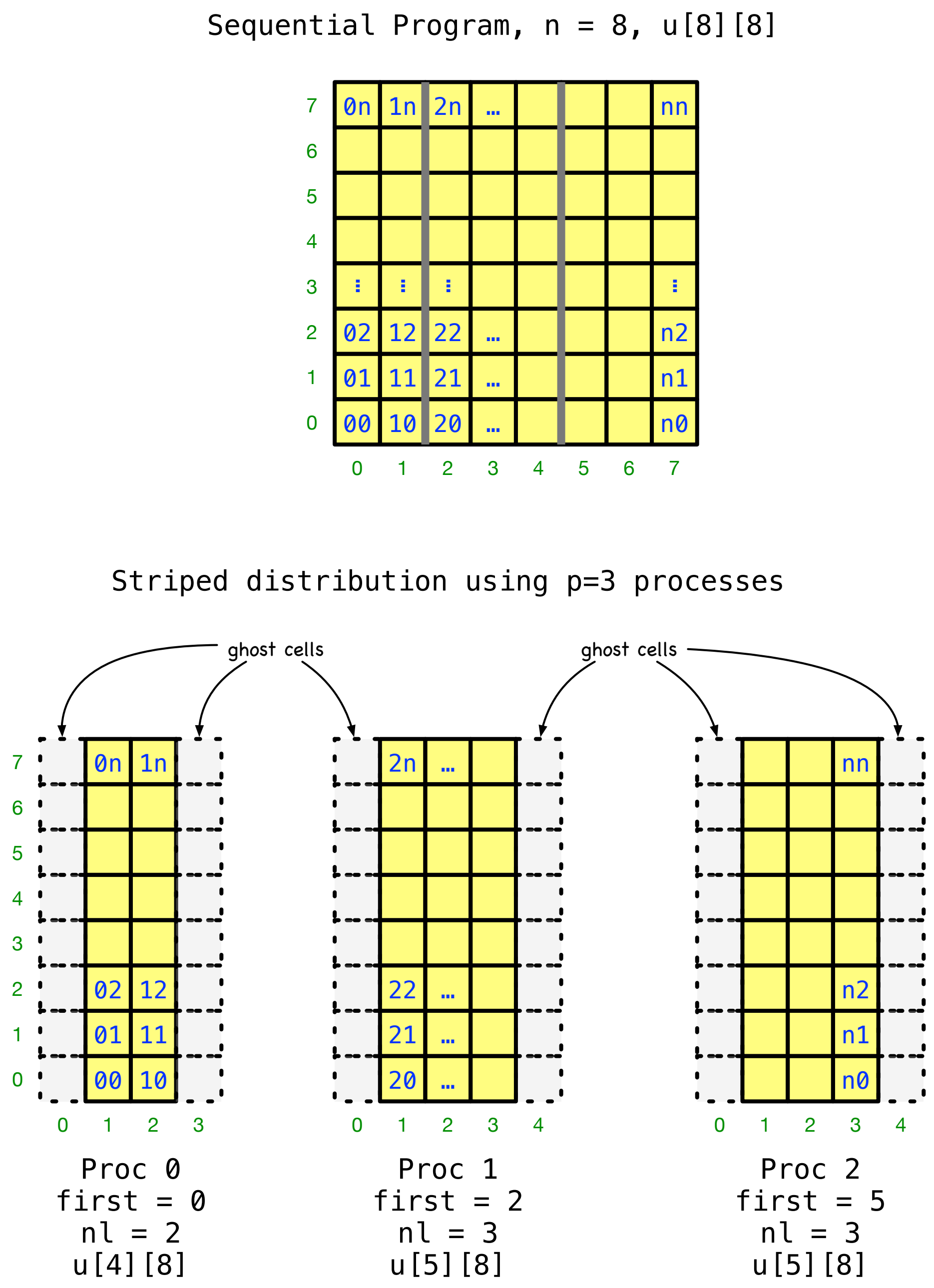

FIGURE 1. Striped distribution of a square 2d-array u. In the sequential program, u has dimensions 8×8. In the parallel program executed with 3 processes, each process owns `nl` columns. The value of `nl` on process 0 is 2; on processes 1 and 2 the value of `nl` is 3. Each process also has 2 columns of ghost cells. (The left ghost column on process 0 and the right ghost column on process 2 are not used.)

3. Problem 3: Collectives

Now modify your `diff2d_mpi.c` so that two additional pieces of information are printed at the end: the final mean temperature over all n^2 cells, and the final minimum temperature. The output should look like this:

```
Time (s) = 177.857859, Mean = XX.XXXXXX, Min = XX.XXXXXX
```

Feel free to modify the sequential version to do the same so that you can check your results. Commit your work.

Note: for full credit, you must do these computations efficiently. Gathering all n^2 values onto one process is not considered efficient.

Finally, run the program on `bridges2` with $n = 10^3$, `nstep` $= 2 \times 10^6$, and `wstep` $= 5 \times 10^3$. You choose how many nodes and cores to use but for full credit your run should complete in less than 3 minutes. Commit your bash script file `diff2d_mpi.sh` and the slurm output file as `diff2d_mpi.out` under `diff2d`.