

CSC 4760 and 5760 MPI Homework (#1)

Anthony Skjellum, Instructor

Assigned: February 11, 2026 (Revised Never)

Due (in class): February 25, 2025

We will configure Github Classroom beforehand to support submission.

Note: (X/Ypts) – X is the points value for CSC 4760, and Y is the points value for CSC 5760.

1 Warp-1 Info

Please compare this with notes from the ITS visitors presented last week (see iLearn for the lecture PDF). (We will update the info below if needed too.)

1.1 Accessing nodes

Find what accounts you can access, run

```
sacctmgr show assoc format=account%20,user%20
```

This should show that you can access an account named `csc4760-001-2026s`

To enter a node with resources,

```
hpcshell --cpus-per-task=2 -- account=csc4760-001-2026s
```

1.2 Running MPI

Load the MPI spack module.

Use MPICC/MPIC++ and MPIEXEC (or MPIRUN) along with required parameters.

2 Problems (CSC4760/5760)

Problem 1 (15pts/12pts):

Write an MPI program (by hand, no LLMs) that passes a message of one integer around in a logical ring of processes with `MPI_COMM_WORLD`. The integer should start at 0 in process 0 and be incremented each time it passes around the ring, and you should be able to have the message go around the ring N times, where N is specified at compile time.

Problem 2 (15pts/12pts):

Using your favorite LLM, vibe code Problem 1, and compare and contrast with what you developed by hand for correctness/functionality.

Show the sequence of prompts you devised and applied, together the sequence of code produced, and the final product. Report what version and details of your LLM environment too. If it is incorrect after several prompt rounds, please explain what the LLM seems to be getting wrong.

Problem 3 (15pts/12pts):

Write a parallel vector-vector element by element multiplication program (by hand, no LLM) followed by a tree reduction to compute the dot-product of two vectors of length N with P processes using MPI. For instance, show with $N = 1024, 2048, 4096$ and $P = 1, 2, 4$.

Use only point-to-point operations. Code your own tree reduction using sends and receives.

Divide the data as equally as you can between your processes. We will discuss how to compute these partitions in class early next week (week of February 16).

Use `MPI_Wtime()` to measure the cost of the operation as you vary P and N . We will explain how to use this operation in lecture.

Use easy-to-evaluate data for testing in the vector elements for correctness testing (but not all zeroes).

Problem 4 (15pts/12pts):

Using your favorite LLM, vibe code Problem 3, and compare and contrast with what you developed by hand for correctness/functionality.

Show the sequence of prompts you devised and applied, together the sequence of code produced, and the final product. If it is incorrect after several prompt rounds, please explain what the LLM seems to be getting wrong.

Report what version and details of your LLM environment too.

Problem 5 (20pts/16pts):

Write a program that uses `MPI_Comm_split` to create two sub-communicators for each process in the world of processes given initially in `MPI_COMM_WORLD`. The first split should put processes together that have the same color when their ranks are divided by an integer Q (ranks in `MPI_COMM_WORLD`). The second split should put processes together that have the same color when your compute the color as their rank mod Q . In this situation, your world size must be at exactly $P \times Q$, $P, Q \geq 1$. You get to pick P, Q but $P \times Q$ has to be the size of the process group of `MPI_COMM_WORLD`.

Problem 6 (20pts/16pts):

1. Demonstrate the use of `MPI_Bcast` and `MPI_Reduce` to achieve the same result as `MPI_Allreduce`.
2. Demonstrate the use of `MPI_Gather` and `MPI_Bcast` to achieve the same result as `MPI_Allgather`.
3. Demonstrate the use of `MPI_Alltoall` to send a personalized communication between each process in `MPI_COMM_WORLD`.

3 CSC5760—Graduate Student Problem(s)

Problem G1 (20pts):

Your goal is to develop a two-dimensional decomposition of the Game of Life (a cellular automaton created by Prof. John Conway in 1970) using MPI for inter-process communication (and sequential processing in each process for the algorithm):

- a 2D decomposition of an $N \times N$ world of cells (let space be a dead space and '*' is a live space). Represent cells as 'char'.
- Divide the world as evenly you can among a logical array of $P \times Q$ processes.
- For each cell on iteration i , it is alive in iteration $i + 1$ if it is alive in iteration i , and has two or three live neighbors.

- For each cell on iteration i , if it is dead on iteration i , then it is alive $i + 1$ if it has three live neighbors.
- The boundaries of your world wrap, forming a torus. That is, if you go off the left side of the domain, you come back on the right (and vice versa). If you go off the top, you come back on the bottom (and vice versa).

Note: You will need two copies of your domain, one for even iterations and one for odd.

The way Life works, each cell has eight neighbors. These eight neighbors define the so-called stencil. This stencil induces communication between your processes. Around your local domain, you will need the “halo” (or ghost cells) that come from the eight neighbors. To a process (p, q) , there are North, South, East, West, NW, NE, SW, SE neighbor communications.

Test this with at least eight MPI processes and at least with $N = 512$. Make sure it works correctly even when $P = 1$ and/or $Q = 1$ (which means there is no communication in one dimension or the other, just local wrapping of boundaries).

Notes:

- We will go over lots of the details in class of how to manage the local data structures and halos to help.
- Compute your local updates sequentially for this problem set. We will explore parallelizing on-node in the next problem set.
- There are many famous patterns: blinkers, gliders, glider guns (they generate gliders), etc. The emerging properties of the Game of Life are amazing; for instance, it has been proven to be a Turing complete system, meaning any computer program can be coded in a sufficiently large Life world.

Reference: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life .

Hint: Do problem #5 first before this problem.

Note: CSC4760 students (aka undergrads) may do grad problem(s) for extra credit.