Department of CSE, The University of Texas at Arlington

CSE5351/CSE4351: Parallel Processing
Spring Semester, 2022
Homework Assignment 6

Due Date May 6, 2022

Write and test the following programs using C and MPI on Stampade. (Email the Word file with the results and c code files to the TA)

1. PROBLEM DESCRIPTION (30 point)

Measure the timing performance of MPI collection communication routines MPI_GATHER and MPI_SCATTER (thus you will write two programs on the Stampede) in the same manner as a collective communication routine's time is measured (see class notes slides and the MPI manual).

HINTS

(1) Your program should work on a dynamically created integer array to be used in these routines. The array size will be the user input.

(2) The routine can also be tested on 2, 4, 8, or 16 processors.

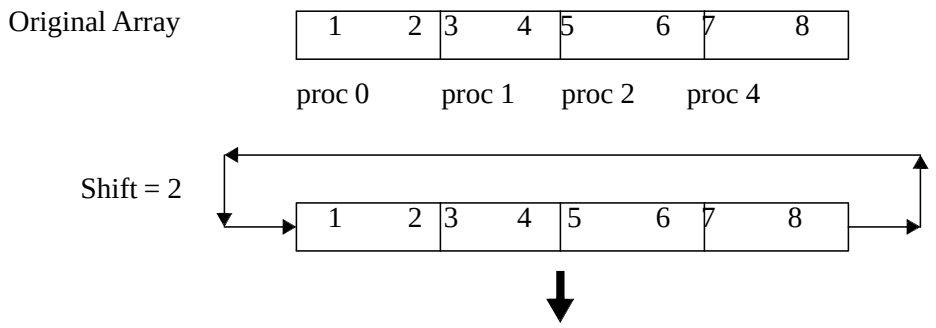
(3) The program should display the original and final array from each processor (along

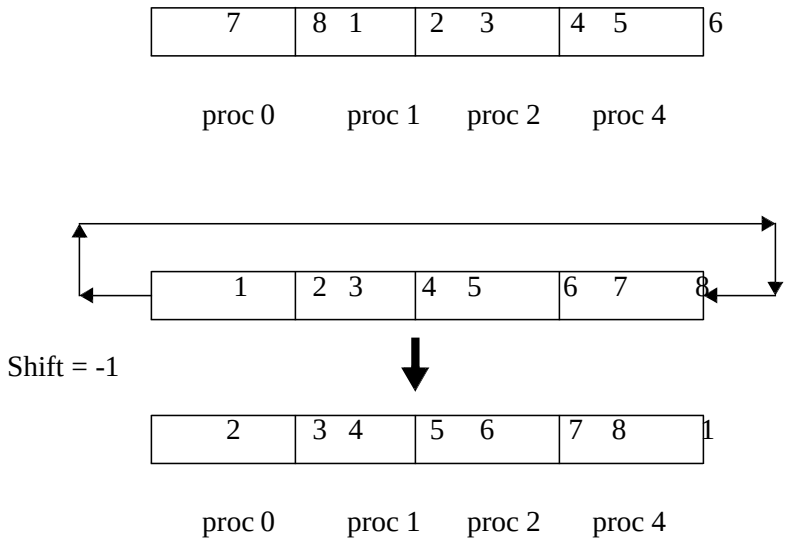
with the processor number)

(5) Draw a curve for MPI_GATHER with the number of processors on the x-axis (2, 4, 6, 16) and the timing in y-axis (one curve for array size 8 in each processors, another for 16 and another for 32). Draw another curve for MPI-SCATTER. Include both curves in a WORD file.

2. PROBLEM DESCRIPTION (30 points)

A procedure named 'SHIFT' is to be implemented using the MPI library (not using the MPI function but writing your own). A one dimensional array is partitioned in equal size across a set of processors. The procedure SHIFT takes a shift factor and performs a circular shift on the array such that the elements shifted out at one end are shifted in at the other end. Two examples of such procedure are shown below.





INPUTS

(1) The number of processors.

(2) The size of the array in each processor; it is assumed to be the same for every processor. You may assume the starting index of the array as either 0 or 1.

(3) Each processor prompts for its input to the array entries (assume that array is of integer type). The order of the input is that proc # 0 prompts for all of its array entries, followed by proc # 1, and so on.

(4) The shift factor (integer).

OUTPUTS

(1) Print out the original array and result array for each processor along with the processor ID.

GUIDELINES

(1) Write a separate routine called SHIFT as a part of your parallel node program.
(2) Assume a one dimensional grid of size *N* (it is the same as a ring).
(3) The routine SHIFT has three arguments: the original array, result array, and shift factor.

HINTS

(1) The shift factor can be both positive or negative.

(2) The routine can also be tested on one processor; other test cases for number of processors include 2, 4, 8, or 16.

(3) The shift factor can be a “reasonable number”.

(4) Assume the data type for the array is integer.

(5) Think about all of the possible cases; design your algorithm and then starting writing the code.

(6) Above all, use your own judgment, make assumptions if necessary and make your own decisions accordingly.

3. PROBLEM DESCRIPTION (40 points)
Write a data-parallel program using distributed non-shared memory model as taught in the class for the sieve of Eratosthenes. Use MPI for message passing and test the program on the Stampede supercomputer.

The program has two inputs: The largest number, *n*, up to which the prime numbers are to be found, and, *p*, the number of processors. The program should run for any number of cores ranging from 1 to 32.

Make the rest of the assumptions yourself but your grade will be based on how good your parallelization and communication scheme is. So feel free to make optimizations. The output of your program is the prime numbers found by your program. Attach a note on your parallelization scheme, and include a time vs processor (include a number of curves, each with a reasonable number *n*, to be chosen by you) plot.

Submissions; Submit your homework with a Word file and text files of the C code by emailing to the TA Addison Clark, addison.clark@mavs.uta.edu

Deadline May 6, 2022.