Department of CSE, The University of Texas at Arlington

CSE5351/CSE4351: Parallel Processing

Homework Assignment 3

Assigned on April 12, 2017
Due on April, 24, 2017

1. PROBLEM DESCRIPTION

You are familiar with the numerical integration for calculating using the rectangle rule. Simpson's Rule is a better integration algorithm than the rectangle rule because it converges quickly. Suppose we want to compute $\int_a^b f(x)\,dx$. We divide the interval [*a*, *b*] into *n* sub intervals where *n* is even. Let x_i denote the end of the *i*th interval, for $1 \leq i \leq n$, and let *x0* denote the beginning of the first interval. According to Simpson's rule:

$$\int_a^b f(x)dx \approx \frac{1}{3n}\left[fx_o - fx_n + \sum_{i=1}^{\frac{n}{2}} \left(4f(x_{2i-1}) + 2f(x_{2i})\right) \right]$$

In the case of π calculation problem, $f(x) = \frac{4}{(1+x^2)}, a = 0, b = 1$, and *n* is an input parameter. Write a parallel program using MPI to compute π using Simpson's Rule. The program should be able to run on any number of processors (to be specified at run-time). Run and test your program on the Stampede.

INPUTS

The number of intervals, *n,* and the number of processors, *p*.

OUTPUTS

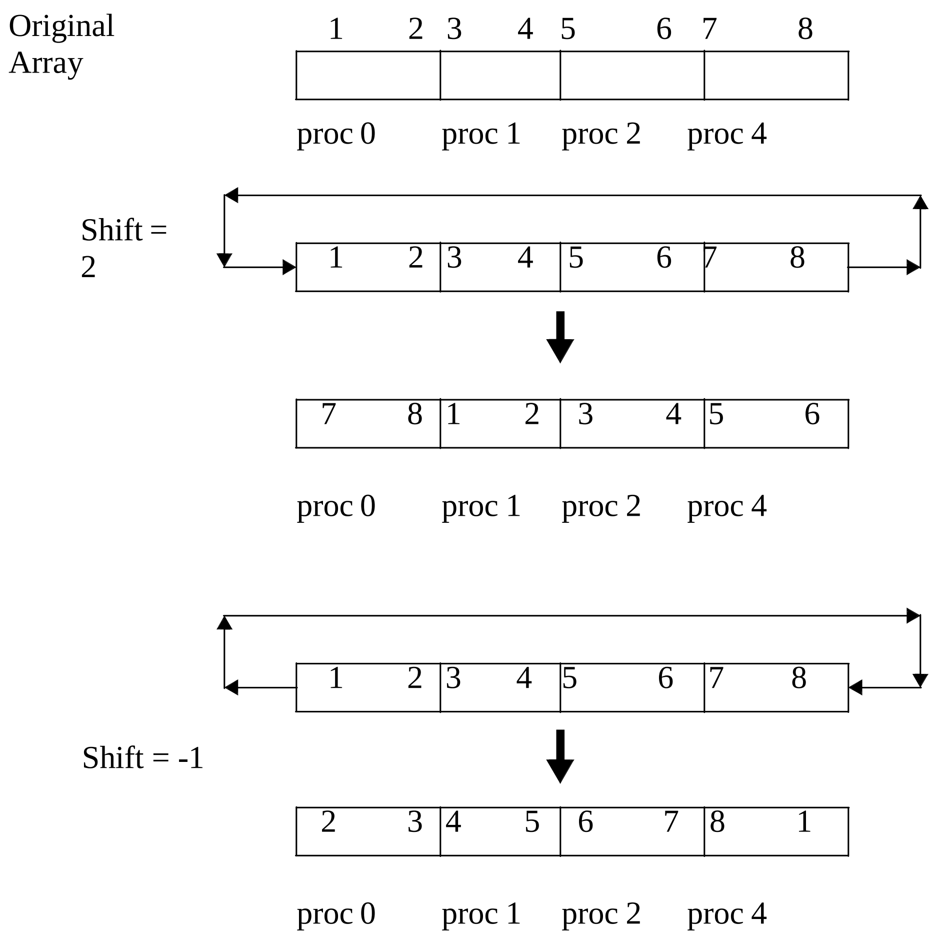
Prints out the value of π (from Processor 0 only).

HINTS

You must come up with your own data allocation strategy and make decisions about solving the problem. The program should work for any number of processors. For testing, you should try a very large value of *n* (e.g. 400, 000).

2. PROBLEM DESCRIPTION

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

Measure the timing performance of MPI collection communication routines MPI_GATHER and MPI_SCATTER (thus you will write two programs) in the same manner as a collective communication routine’s time is measured (see Part 3 of lecture notes, page 19).

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.

SUBMISSION: WHAT, WHEN & HOW

(1) This assignment is due on or before April 24, 2017

(2) Send your source programs, including any comments, and the WORD file to the TA (**zahra.anvari@mavs.uta.edu)**.