Department of CSE, The University of Texas at Arlington

CSE5351/CSE4351: Parallel Processing

Homework Assignment 4

Assigned on April 9, 2019
Due on April 22, 2019

1. PROBLEM DESCRIPTION

You are familiar with the numerical integration for calculating using the rectangle rule. Implement the algorithm taught in class. Write another program using the Simpson’s Rule. Suppose we want to compute \( \int_{a}^{b} f(x) \, dx \). We divide the interval \([a, b]\) into \(n\) subintervals where \(n\) is even. Let \(x_i\) denote the end of the \(i\)th interval, for \(1 \leq i \leq n\), and let \(x_0\) denote the beginning of the first interval. According to Simpson’s rule:

\[
\int_{a}^{b} f(x) \, dx \approx \frac{1}{3n} \left[ f(x_0) - f(x_n) + \sum_{i=1}^{n-2} (4f(x_{2i-1}) + 2f(x_{2i})) \right]
\]

In the case of \(\pi\) calculation problem, \(f(x) = \frac{4}{(1 + x^2)}, a = 0, b = 1, \) and \(n\) is an input parameter.

(1) Partition the data using contiguous blocks instead of columns (the whole data should be automatically divided by the number of processors to be determined by run time). Write a parallel program using MPI (blocking communication routines) to compute \(\pi\) 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 2.

(2) Write the same program using Non-blocking communication and check the correctness of the program by comparing it with the actual \(\pi\) value, using the number of intervals to be from 10,000 to 100,000 with step size of 10,000.

(3) Draw two curves: one compares the timing of the two algorithms and the other the accuracy (the number of intervals should be on the \(x\)-axis).

INPUTS

The number of intervals, \(n\), and the number of processors, \(p\).

Work to be reported.

(a) Prints out the value of \(\pi\) (from Processor 0 only) for part 1 and compare it (the error) with the simple integration equation shown in class.

(b) Print out the value of \(\pi\) for part 2.

(c) Analyze the scalability of the problem by varying both the number of processors and \(n\) \((P = 4, 8, 16, 32, n = 100,000, 200,000, 400,000, 800,000)\). Draw a plot of this scalability (efficiency vs number of processors and 4 curves, one for each data size).
3. PROBLEM DESCRIPTION

Measure the timing performance of MPI collection communication routines broadcast and also the timing performance of individual send and receive from one processor to a group of processors, using an integer array. Compare their performance

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 broadcast 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 the same communication using send and receive. Include both curves in a WORD file.

SUBMISSION: WHAT, WHEN & HOW

(1) This assignment is due on or before April 22, 2019

(2) Send your source programs, including any comments, and the WORD file to the TA (Khalid, Saifullah <saifullah.khalid@mavs.uta.edu>).