

Tools of high performance computing 2024

Exercise 7

Return by Wednesday 11.3.2024
Exercise session: Friday 12.3.2024

Note the exceptional times
due to exam week.
No lecture on 4.3.2024.
No exercise session on 5.3.2024.

Problem 1. (6 points)

Write a program to test how fair the message passing implementation is. To do this, have all processes except process 0 send 100 messages to process 0. Have process 0 print out the messages as it receives them, using `MPI_ANY_SOURCE` and `MPI_ANY_TAG` in `mpi_recv`. Is the MPI implementation fair? Unfair means that a certain sender gets its messages through before other senders. If the implementation is fair the order of senders getting their messages through is more or less random.

Problem 2. (6 points)

Calculate the value of π using the Monte Carlo (MC) integration of function $y = \sqrt{1 - x^2}$ over interval $x \in [0, 1]$ using MPI. As a random number generator (RNG) you may use either the one provided by the language or some external one. Remember to have for all processes a different RNG seed number.

MC integration in this case means generating random points in unit square $x, y \in [0, 1]$ and calculating the fraction of points within unit distance from origin.

Problem 3. (6 points)

Write an MPI program where a set of floating point numbers is read from a file by one process. Then the program calculates the average and variance of the data by dividing the data using `MPI_Scatterv` among all processes. You should take into account the fact that the amount of numbers is not necessarily divisible by the number of processes. The data set is in the gzipped file `ex7p3.dat.gz`. You may give the number of data points as a command line argument if you wish.

Problem 4. (6 points)

Write a program where the values of a single integer variable on all processes is summed and printed. Use the process id as the variable. Use the tree-like communication pattern shown on lecture notes (Parallel computations, part 1, page 16, lowest figure, also reproduced below). Note that this pattern only works if the number of processes is a power of two. Write your program for a general case of power of two (not just for 8 processes as shown below).

