Project 1, due on 02/17.

Problem 1. Parallel Numerical Integration for Undergraduate Students.

We can calculate π by approximating $\int_0^1 \frac{4}{1+x^2}dx$. Implement a parallel code using composite Trapezoidal rule to approximate this definite integral. Use the framework implemented in parallel_trapezoidal.c. In general, each process is assigned with a subinterval $[local\_a, local\_b]$. We apply composite Trapezoidal rule on this subinterval to compute a partial sum. Modify the script "HPCC_1.sh" to submit your runs.

1. Use point-to-point communication to compute the sum of partial sums from each composite Trapezoidal rule. Specifically, use non-blocking send and blocking receive to transfer the partial sums to process 0 and let process 0 compute the sum.

2. Use $n = 10000, 20000$ and 40000 numbers of subintervals to do the calculation respectively. For each computation, use 4, 8 and 16 processors respectively. Find the overall the wall clock time spent by the whole program. Make a table to list the results.

Hand-In. Turn in the hardcopy of all your source code, and the report which contains results and a description of your implementation on point-to-point communication.

Problem 2. Parallel Explicit Finite Difference Scheme for Solving 1D Heat Equation for Graduate Students.

Consider to solve $\begin{cases} u_t(x,t) = u_{xx}(x,t), & 0 \le x \le \pi,\ \ t > 0 \\ u(x,0) = \sin(x) & 0 \le x \le \pi \end{cases}$ with periodic boundary condition by the explicit finite difference scheme. Compute the solution for $t = 2.0$.

The exact solution is given by $u(x,t) = e^{-t}\sin(x)$.

Assume we use $M+1$ grid points. The grid space then is $\Delta x = \frac{2\pi}{M}$. The grid points are $x_k = k\Delta x,\ k = 0, \dots, M$. Let Δt be the time step size. For stability, we should satisfy $\frac{\Delta t}{\Delta x^2} \le 0.5$.

Let $v_k^n \approx u(k\Delta x, n\Delta t)$ be the approximate solution. The explicit scheme is

$v_k^{n+1} = v_k^n + \frac{\Delta t}{\Delta x^2}(v_{k+1}^n - 2v_k^n + v_{k-1}^n)$ for $k = 0, \dots, M$.

Implement a parallel version of the scheme to solve the above problem by arbitrary number of grid points $M+1$ using P processors. Assume $M + 1 \gg P$. Use non-blocking send and receive for message passing. Use $M = 1000, 2000, 4000, 8000$ respectively to do the mesh refinement study. Compute $L_{2,\Delta x}$ error with respect to the mesh refinement. Do each of these calculations with 4, 8, 16 processors respectively. Make a table to list the wall clock time space on computation and communication respectively. Hint: the code to compute work load assignment is work_division.c.

Hand-In. Turn in the hardcopy of all your source code, and the report which contains results and algorithmic notes on both computation and communication.