AM 250, Spring 2022
Homework 6

Miles Miller
Applied Math

Question 1. Latency

For the latency study I ran a series of tests with different message sizes and timed them. The time per message is given by,

$$T_{\text{msg}} = \frac{T_{\text{tot}}}{2N_{\text{loop}}}, \tag{1}$$

where N_{loop} is the number of times that the ping-pong routine is looped over and the two is included in the denominator because each run of the pin-pong routine has two message send and receives. Furthermore, T_{msg} then can be written as a linear function of the startup time t_s, the time cost per word t_w, and the message length N as follows,

$$T_{\text{msg}} = t_s + t_w N \tag{2}$$

Thus the linear relationship can be deduced from a linear regression of a set of different message runs and times. The resulting linear regression is given by 1.

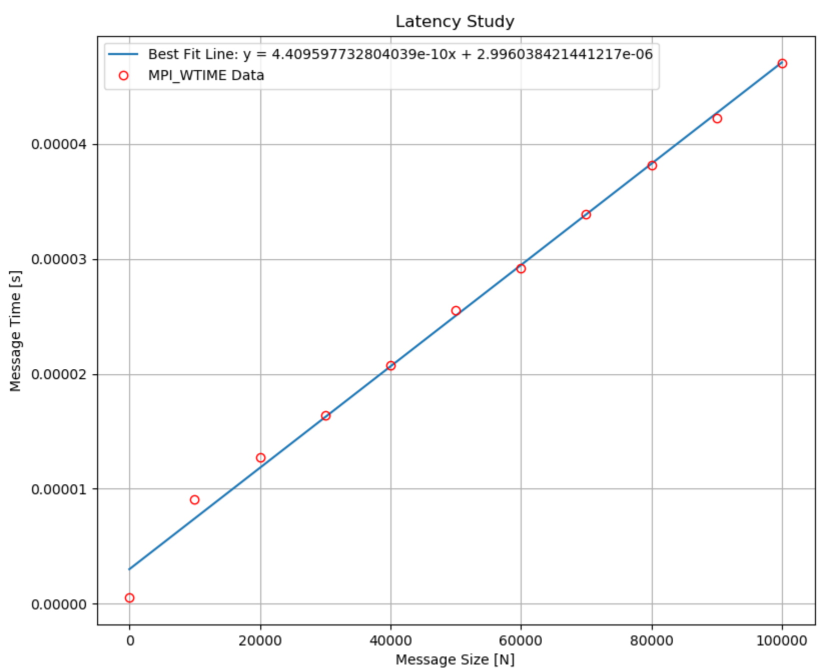

Figure 1. The resulting linear regression for the latency time of MPI send and receive.

The resulting linear fit for the latency of the MPI message send and receive is the following,

$$T_{\mathrm{msg}} = t_s + t_w N = 2.996 \times 10^{-6} + 4.410 \times 10^{-10}. \tag{3}$$

Thus the startup time is 2.996×10^{-6} and the cost per word is 4.410×10^{-10}.

Question 2. Finite Difference Performance Model (2-D Decomposition)

The execution time is given by the following,

$$T_{\mathrm{2D\_FD}} = \frac{T_{\mathrm{comp}} + T_{\mathrm{comm}}}{P}, \tag{4}$$

where T_{comp} is the computation time, T_{comm} is the communication time, and P is the number of processors. If we decompose the domain in two dimensions for parallelization, then each processor will be performing computation on a column in the vertical z direction. Thus for communication, given the finite difference 9-point stencil, each processor will exchange $2N_z$ points with each of its four neighbours. Thus the communication time will be given by,

$$T_{\mathrm{comm}} = 4P(t_s + 2t_w N_z). \tag{5}$$

The computation requirement of each processor is given by time per node and the total number of nodes requiring computation as follows, assuming $N_x = N_y = N$,

$$T_{\mathrm{comp}} = t_c N^2 N_z. \tag{6}$$

The total execution time is then given by the following,

$$\begin{aligned} T_{\mathrm{2D\_FD}} &= \frac{T_{\mathrm{comp}} + T_{\mathrm{comm}}}{P} & (7)\\ &= \frac{t_c N^2 N_z}{P} + 4t_s + 8t_w N_z & (8)\end{aligned}$$

The efficiency is then,

$$E_{2D\_FD} = \frac{t_C N^2 N_z}{t_C N^2 N_z + 4Pt_s + 8Pt_w N_z}. \tag{9}$$

The isoefficiency is then given by the relationship between the size of the problem and the number of processors such that the efficiency remains constant. Rewriting 9 gives the following,

$$t_c N^2 N_z = E(t_c N^2 N_z + 4Pt_s + 8Pt_w N_z), \tag{10}$$

Thus from the above it is apparent that horizontal domain of the problem, N^2, scales with the number of processors for E to remain constant such that N^2 P, which means that the 2D decomposition results in an isoefficiency of P.