CSE 457 Assignment I
Due: 10/05/06

This assignment concerns writing efficient "single program multiple data" parallel programs using C/C++ and the MPI library for message passing. You should submit your report and any program outputs that were used for the data and analysis in your report. This assignment will be graded out of 80 points and is worth 8% of the total score for this class.

In your report, you must cite all reference materials (other than the text book and lecture notes for the class) used in preparing your solution.

You are encouraged to work with a partner and submit a single joint project report.

Part I. Determining Communication Costs (40 points)

Your goal in this part of the project is to empirically determine the values of t_s and t_w, the start-up time and the per-word time for communicating integers between processor pairs. These are the values used in modeling the communication cost of a message of length m as $t_c(m) = t_s + m \times t_w$.

Design your program to run correctly on an even number of processors. Let p be the number of processors in a given run; pair each processor of rank i with its partner of rank $i \pm p/2$. For each pair of processors use the `MPI_Sendrecv` call to exchange the contents of integer buffers of length m. For a given value of m measure the total time exchanging 100 messages and compute the average as the time to communicate a message of length m. Next, use a suitable `Reduce` operation to sum the average over all processor pairs; divide the sum by the number of processor pairs to compute the final average time $t_c(m)$ for communicating a message of length m integers. Add a loop to perform the experiment for values of m equal to $1, 2, 4, 8, 16, 32, 64, 128, 256$.

- Using an 8-processor run, provide a table of your observed values of $t_c(m)$ for the specified values of m. (10 points)
- You should now use the values in the item above to determine the best values of t_s and t_w. Note that these can be obtained by fitting a line to your observations; i.e., fit a line to points in the plane with the X-coordinate given by the value of m and the Y-coordinate given by the corresponding value of $t_c(m)$.
  Describe briefly your method for determining t_s and t_w (in seconds) and provide values of t_s and t_w; if necessary, justify the values you obtained. (20 points)
- What is the scientifically correct method for minimizing the error in fitting a line to observed data points? Describe briefly. (10 points)

Part II. Parallelizing Circuit Satisfiability (40 points).

Consider the second program for circuit satisfiability from chapter 4 of our textbook.

- Show benchmarking results for 1, 2, 4 and 8 processors (with printing disabled). Report mean execution times for 5 runs. Summarize and interpret the results you observed, taking into account the communication times (previous question) relative to the time for one operation (compute time). (15 points)
- Change your code to add four more inputs (i.e., 20 inputs as opposed to the original 16). Construct a new expression with 20 variables by adding a suitable subexpression to the original one. Show your new expression. (10 points).
- Repeat the benchmarking for this new function of 20 inputs; summarize and interpret the results. Are speedups better or worse than those for the function with 16 inputs? Explain briefly. (15 points).