CS/ECE 566 Parallel Processing

Programming Assignment 3

1 Goals

In this assignment, you will study various aspects of parallel programming using MPI. The well-known matrix multiplication (MM) problem which is fundamental to many high-performance applications will be considered. You will be using the EXTREME parallel cluster provided by ACCC as the experimetal testbed. Your goal is to compute $|X^k|$ in parallel, where

- X is a $n \times n$ matrix
- k is an input parameter

First compute the product of matrix multiplication. Then compute the determinant of the product. Do not use shortcuts; although determinant of products is the product of the individual determinants, you cannot use this approach. The computation of determinant is just to check for correctness.

2 Experiment

1. Identify a set of different mappings (parallel formulations) of the MM problem, such as those that vary how to decompose the input data or the final product data, how to decompose the computation workload, the choice of MPI calls used, etc.. Matrix multiplication may be parallelizable, can associativity be leveraged?

   Experiment with the traditional $\theta(n^3)$ algorithm, Cannon's method, as well as others, and those of lower complexity (e.g., Strassen's - refer Cormen, Leiserson, and Rivest text on Algorithms), if possible. Let $\mathcal{S}$ denote the set containing each of these parallel formulations.

2. You need to design the code for each of the parallel formulations you choose.

3. Experiment the *timing*, *speedup*, and *efficiency* variations by varying the different parameters along the lines suggested here. Note that these are only suggestions: the actual parameter ranges you vary will depend on various factors such as the workload on and the availability of the EXTREME nodes, OS limits etc..

   n, size of matrix: $n = 2^6, 2^8, 2^{10}, 2^{12}, 2^{14}, \ldots$

   p, number of processors: $p = 1, 2, 4, 8, 16, 32, 64$

   k, problem input parameter: $k = 1, 2, 3, 4, \ldots$

3 Input and Testing

- While you debug your code, please work with small data sets, few processes, and few processors i.e., small n, small k, small p.
- The matrix X has numbers taken from the range $\{-1, 0, 1\}$.

- **Input Method 1:** The input is 2 numbers: the probability of -1, and the probability of $+1$. Create the matrices at the root process using the given probabilities. *This matrix is created at the logical root process - this is n̲ot the master node on EXTREME. You cannot use this formula to create the matrix at the worker processors.*

- **Input Method 2:** The input is taken at the keyboard as follows. Take in a sequence of 4 numbers from the range $\{-1, 0, 1\}$. For example, the sequence $[-1, 0, +1, 0]$.

  - Let A be this sequence repeated $n/4$ times.
  - Row i of matrix X is A after a cyclic-right-f-shift of A, where $f = \sum_{k=1}^{i} k$.

  This matrix is created at the logical root process - this is n̲ot the master node on EXTREME. You cannot use this formula to create the matrix at the worker processors.

4 Output

1. Submit a hard copy of all the files including the code file.

2. Submit a hard-copy report describing your experiment. The report must have the following sections/information.

   (a) **Formulations:** Describe each of the formulations with in the set $\mathcal{S}$ in about 2+ pages *each.* Justify your selection of this formulation. Include the list of MPI calls you used in the implementation of each formulation.

   Give theoretical analysis for the bounds on the (i) local memory usage M_p, (ii) parallel computation time for local computations, T_{comp}, (iii) parallel communication time T_{comm}, and (iv) parallel run-time T_p, like we studied in class for sorting.

   Formulate expressions for speedup and efficiency. Analyze conditions for cost-optimality and isoefficiency functions, wherever applicable.

   (b) **Parameter ranges:** For each of the formulations describe the range of parameters you experimented with.

   (c) **Results:** Give the tables and plot the graphs showing the *timing*, *speedup*, and *efficiency* variations for each of the formulations, for the range of parameters you used. Remember to use appropriate scales for the graphs.

   (d) **Analysis:** Analyse the results. Give your interpretation and reading of the results for each formulation in $\mathcal{S}$, and across formulations in $\mathcal{S}$, i.e., also compare the results of the different formulations. You should address questions such as the following sample list:

   - Which formulation is better? under what (or all) circumstances? Why?
   - How and why does efficiency vary the way it does?
   - How and why does speedup vary the way it does?
   - Can you measure the computation time versus communication time? How do the formulations compare with each other with respect to this breakup of the time overhead?
   - Can the communication overhead be overlapped with your computation overhead? Do your formulations/algorithms do so?

   Explain why you observe what you observe. In particular, any anomalous observations should be explained.

   (e) **Lessons:** What insights you learned from this assignment.

5 Grading

The problem is reasonably well-formulated but the experimental approach is open-ended. Your goal is to get the most parallelism in solving the MM problem (and the determinant calculation). The depth of your choice of parallel formulations you choose to experiment with will be judged. Your assignment will also be judged on how *comprehensively* and *methodically* you have designed and run the experiment, and reported on the results. Your insights into the analysis of the results will also be considered in judging the assignment. Your grade will be based primarily on your report; (assuming, of course, that you have managed to run your programs).

6 Reference Material/Chapters

Chapters 3, 6, and 8 from the text. The sample programs of the ACCC web site.