CIS625 – Concurrent Software Systems

Homework 5: Hybrid distributed programming

Assigned: 11/6/17
Due: Sunday, 11/19/17 by midnight
Individual or group: Group
Points: 15

Description

The purpose of this project is to get you started doing more sophisticated hybrid programming on a machine like Beocat.

Task(s)

Write a Hybrid MPI/OpenMP code to complete problems 1-3. Run each program on Beocat using a significant amount of compute time. For example, you could use about 30s of compute time per problem. Compare the runtimes as you increase the number of threads per MPI program (1, 2, 4, 8). Perform the tests on up to 16 cores. How do the results compare to each other? Are the hybrid versions faster?

1. Write an MPI program to perform numerical integration on Easom's function -

$$f(x, y) = -\cos(x)*\sin(y)*\exp(-((x - PI)*(x - PI) + (y - PI)*(y - PI)))$$

Assume that the starting and ending points for both x and y are provided as command line arguments. Your program must make use of the MPI_Reduce function.

2. Assume that you wish to test the performance of several numerical optimization methods on a particular function

$$f(x) = \cos(x)+(pow(fabs(7.0-x), 2.0/15.0))+2*(pow(fabs(5.0-x), 4.0/35.0))$$

Write an MPI program that will broadcast the starting and ending locations for x that will be used to analyze this function. You may assume that one program will be used. This program will use a Monte-Carlo method to analyze the function (i.e generate a whole bunch of points within the range given at the command line and evaluate them to find the minimum). Be sure to print the resulting minimum.

3. Write an MPI program that will compute the standard deviation of a set of randomly generated numbers. Assume that processes will generate these numbers independently and send their results (sums and sums of squares) to process rank 0 for computation of the standard deviation.

Plot the CPU time expended by the various versions of the program running with 1, 2, 4, 8, and 16 threads total (feel free to combine these into one graph). Use the MPI-scatter environment to spread

your MPI processes across multiple machines. Note that the x axis of your plot should be threads used, and the y-axis of your plot should be time expended in seconds. Discuss the results. Are there any race conditions? How much communication (in terms of # of messages and bytes/message) is there per variant?

Resources

MPI - https://computing.llnl.gov/tutorials/mpi/

For help using the Beocat scheduler and its command lines, see the general help page
https://support.beocat.ksu.edu/BeocatDocs/index.php/Main_Page
and the scheduler help page
https://support.beocat.ksu.edu/BeocatDocs/index.php/SGEBasics
Compute node specifications are at
https://support.beocat.ksu.edu/BeocatDocs/index.php/Compute_Nodes

You will probably need to install a shell program on your Windows machine – I use PuTTY, and program to transfer files back and forth (I use WinSCP, but other options are fine). Linux and Mac OS have terminal and file transfer programs installed by default.

Submission instructions

Submit a PDF which contains your analysis and a copy of the two versions of your code and your various controlling shell scripts as appendices. Submit this via Canvas. Grading will be split – 75% for the correctness and performance of your implementation (including use of shell scripts), 25% for your performance analysis (probably 1-2 pages, including graphs but excluding appendices, assuming a reasonably compact formatting scheme).