HPC Lab 2 – Matrix Multiply with MPI and OpenMP
CSC462/GCSC 562
Spring 2025

Comments:

The entirety of this lab can be developed and tested on a laptop or desktop PC. However, for the scaled tests, Titan will be used. It is recommended that the lab be developed on a personal device; this gives the greatest flexibility in access and ease of use. Once it has been developed and tested, then the source file can be copied to Titan and the "official" test runs can be performed.

In this lab, you will start with the MPI version of Matrix Multiply and add to it the OpenMP directives to gain parallelism on each node in addition to the MPI parallelism across nodes. The stages of this lab are as follows.

- Lab exercises

Compiler Environment

Use what you set up on your personal device as well as what you have done on Titan for HPC Lab 1.

Lab Exercises

This lab is designed to modify the MM-MPI program from HPC Lab 1.

Part 1: Local Parallelism

- Copy your mmmpi.c to mmmpiOMP.c. Here you will add OpenMP directives.
  - Add appropriate #pragma omp directives within the dotProduct. Run this with matrix sizes of 100, 1000, 4000, and 6000, on both the PC and on Titan using only one node. Capture the timings in the appropriate spreadsheet tabs.
  - On the PC, run these with the following thread counts, 1, 2, 4, 8, 16 (up to and including the maximum real core count of your laptop, but no more. If the laptop has 4 threads, then 1, 2, 4. As for Titan, use 1, 2, 4, 8, and 16.
- Copy your mmmpiOMP.c to mmmpiOMP2.c.
  - Remove the prior #pragma statements.
  - Add appropriate #pragma omp directives for a for-loop to parallelize the second, inner loop of the mm algorithm. Run this with the same matrix sizes, on both the PC and on Titan. Capture the timings in the appropriate spreadsheet tabs.
- Copy your mmmpiOMP2.c to mmmpiOMP3.c.
  - Remove the prior #pragma statements.
  - Add appropriate #pragma omp directives for a for-loop to parallelize the first/outer loop of the mm algorithm. Run this with the same matrix sizes, on both the PC and on Titan. Capture the timings in the appropriate spreadsheet tabs.
- Pick any one of the configurations above and add the scheduling directive to use static scheduling. Capture run performance with matrix size of 2000 with chunksize 1 and 32.

Part 2: Combining local parallelism with distributed parallelism.

- Of the two fastest mmmpiOMP versions, run each on Titan with a matrix size of 20,000 across four nodes. Capture the timings in your log.
- Working with the fastest of these two versions, run these sizes on the following node counts on Titan, capturing the runtimes in your log.
    a. For core counts of 8 and 16 per node, do the following combinations:
        i. Matrix size 40,000 on node counts of 1, 2, 4, 8, 16, and 20.
        ii. Matrix size 60,000 on node counts of 32, 48, and 64.

    This will require 16 distinct runs.

Part 3: Data analysis and lab report. For this, you will write a paper and a PowerPoint presentation. The PowerPoint should be considered your 3-minute presentation of the key findings. The paper should be in greater detail, giving deeper observations and interpretations.

1. Write a report describing your observations of the performance of each of the runs, comparing the scalability. Additionally, identify what you observe as being the most scalable configuration (that which runs fastest overall) and give your rationale for why it is better than the next fastest configuration.

   For each matrix size, use charts with plotted curves to visualize the performance of each configuration over the different core counts and node counts. The Y axis will be time, and the X axis will be core counts.

   Consider the following questions in your analysis.

   1. What is the CPU frequency of the processors being used?
   2. What is the memory frequency of the computers being used?
   3. How big is the largest cache on each processor for each computer (your laptop and the SB node on Titan).
   4. Specifically, answer whether there are any configurations or matrix sizes where the laptop runs faster than the SB node. And, if so, explain why that may be.

   Write a 4-slide presentation capturing the salient conclusions, giving details as necessary for each.

2. Graduate students: Write a 4-slide PowerPoint presentation capturing the performance of the two fastest configurations you can configure for matrix sizes of 4,000 and 6,000 on eight and 16 nodes with 8 and 16 cores per node (eight runs total). It should show the following:
    a. The Configuration (matrix size, node information, and the compile line used to compile the code.
    b. The charts show the performance over time.
    c. A bullet list of your observations and conclusions about the best compile-line configuration. This list should include your rationale for why the fastest of the two is the fastest.