GT HPC Class Default Term Project

Spring 2012

Prof. Jeffrey Vetter

1 Goal

Today's computing systems are very complex systems with large numbers of processors and hierarchical memories. In order to make use of these systems, scientists must program them correctly and efficiently with appropriate programming models.

In this default term project, you will implement a Scalable Universal Matrix Multiply Algorithm using MPI, OpenMP, and, CUDA. The goal of this project is to understand the complex interactions and resulting performance across a range of necessary programming models on a contemporary HPC architecture.

2 Assignment

1) Design and implement a distributed SUMMA as described in the SUMMA paper.
   a) Implement and tune a distributed algorithm (SUMMA) with (only) MPI.
   b) Create two derived versions of the MPI implementation:
      i) Add loop level parallelism with OpenMP to each MPI task
      ii) Add GPU parallelism with CUDA to each MPI task
   c) Optimize each level with the appropriate techniques and tools.
   d) You code should be constructed so that you can evaluate 3 different implementations, all optimized as necessary:
      i) MPI-Only
      ii) MPI and OpenMP
      iii) MPI and CUDA
2) Evaluate these implementations using the scaffold code up to 4 nodes of Jinx (48 cores or 12 GPUs), and for a range of matrix sizes that have runtimes up to 15 minutes. You may want to start quite small, and slowly increase the matrix size until you approach this limit. Depending on your implementation and optimization level, you may hit this limit quickly.
   a) MPI-Only
   b) MPI and OpenMP
   c) MPI and CUDA
3) Submit a written report as a single PDF file that includes answers to the following questions:
   a) Explain your implementation in 2500 words or less.
   b) Answer these questions
      i) Which implementation is fastest in terms of time to solution?
      ii) Which implementation is fastest in terms of floating point rate?
      iii) Which implementation is most scalable?

iv) What are the approximate lines of code required for each implementation?

v) What was the most complex task in implementing your term project?

3 Submission Instructions

- This project report and code must be submitted to the T-Square system by the deadline. A penalty of 10% per day from the final score will be enforced.
- This can be an individual project or teams of size two. Each student or team must complete and submit the assignment. Clearly identify your partner, if any, in your submission subtitle. Outside of this, you are permitted to discuss this assignment with other students, faculty, and the TA, but each submission must be written and tested by each student (or team).
- Honor code. Each student or team should write all the required software. Note that students should not reuse implementations from earlier projects, classes, or students. Your submissions will be compared to other implementations to ensure that they are not copies.
- You are required to provide
  - Report
    - The submission format should be a single PDF file (10 pages maximum) with, 11 point Times New Roman font, 1" margins. Clearly label the answers to the above questions in your submission.
    - You must cite all sources of information that you used to complete this project including books, research papers, websites, user manuals, tutorials, source code, etc.
  - Source code
    - Submit your implementation source files, a makefile, and proper instructions to build the software on jinx.
    - Your solution will be tested with a variant of the driver code, so the interface must be followed exactly.
- Your solutions will be judged on
  - Functionality and correctness
    - MPI implementation (40%)
    - MPI+OpenMP implementation (10%)
    - MPI+CUDA implementation (20%)
  - Performance and scalability of your implementations (10%)
  - Explanation and answers to associated questions (20%)