Lab 07: OpenMP and Scalability

Assigned: 2019-03-05 10:00:00

Due: 2019-03-07 23:59:00

Instructions:

- Written portions of this assignment are submitted via Canvas. Unless specified otherwise, the written portion of the assignment is to be completed using LaTeX. All derivations, images, graphs, and tables are to be included in this document. Handwritten solutions will receive zero credit.
- Code portions of this assignment are submitted via `code.vt.edu`. Source code must be in the private repository, to which the CMDA 3634 instructors must have access.

Deliverables: For this assignment, you are to submit the following:

1. (Canvas) `<pid>_Lab_07.pdf`: A PDF file, renderd by `pdflatex` (the file generated by Overleaf is sufficient) containing the answers to the questions requiring written answers. Use the template provided in the project repository.
2. (`code.vt.edu`) The source files required to compile and run your solutions to the lab and the tex and image files for your report, in the appropriate directories.

Collaboration: This assignment is to be completed by yourself, however, you make seek assistance from your classmates. In your submission you must indicate from whom you received assistance.

Honor Code: By submitting this assignment, you acknowledge that you have adhered to the Virginia Tech Honor Code and attest to the following:

> I have neither given nor received unauthorized assistance on this assignment. The work I am presenting is ultimately my own.

Resources

- OpenMP:
  - Reference Book https://mitpress.mit.edu/books/using-openmp
  - General tutorial https://computing.llnl.gov/tutorials/openMP/
  - Timing https://gcc.gnu.org/onlinedocs/gcc-4.5.0/libgomp/omp_005fget_005fwtime.html

Task

In this lab you will use OpenMP to parallelize the basic linear algebra tools we have developed in the labs so far. Once you have parallelized the routines, you will perform strong and weak scalability studies and evaluate if you are obtaining the expected speedups.

You will perform your studies on Cascades, but you should perform the initial developments in your VM.

1. **Setup** your coding environment.
   (a) Pull the lab materials from the upstream repository.
2. **Implement** the following requirements in C. Be sure to use git to commit your code regularly. Push early, push often!
   (a) First look at the `readme` to understand what files and scripts are available. Read the entire assignment.
   (b) Use OpenMP to parallelize the following routines in `vector.c`:
   - `norm()`
   - `normalize()`
   - `axpy()`
   - `inner_product()`

   You must specify all private or shared variables. A simple parallel for is insufficient and will not receive full credit.

   (c) Use OpenMP to parallelize the following routines in `matrix.c`:
   - `matvec()`
   - `matvec_triangular()`
   - `matvec_triangular_guided()`

   You must specify all private or shared variables. A simple parallel for is insufficient and will not receive full credit.

   The routine `matvec_triangular()` performs a matrix-vector product with an matrix $\boldsymbol{A}$ that is *lower triangular*. In this routine, use OpenMP as you usually would. You will see that you observe poor scalability. In `matvec_triangular_guided()`, use guided scheduling to improve the scalability.

   (d) Use the provided experiment script `time_vector.sh` to conduct strong and weak scalability studies for these vector operations for 1-8 OpenMP threads. Use the provided experiment script `time_matrix.sh` to conduct strong and weak scalability studies for these matrix operations for 1-8 OpenMP threads.
   (e) An sbatch submission script has been provided to assist in running your experiments on Cascades.
   (f) Push your work to `code.vt.edu`. `ssh` into Cascades and clone your code. Submit the bash script as a job using `sbatch`. Then use `scp` to copy the results back to your laptop and use the data to answer the questions.
3. **Answer** the questions listed below. You may use Overleaf, but your tex source must be committed to the `reports/` directory.
4. **Submit** your results.

(a) After you have completed this lab (which we'll continue in class on Thursday), upload a PDF of your report to Canvas.

(b) Push your source code and latex files to `code.vt.edu`.

(c) Examine your assignment repository on `code.vt.edu` to be sure that all of your materials have been correctly submitted.

Questions

Answer the following questions. No template is provided (you may copy one from a previous week if you like). Place a copy of your report in the `reports/` directory.

1. For each routine you parallelized, give the complete OpenMP pragma you used. Use a listings block to typeset the code.
   - `norm()`
   - `normalize()`
   - `axpy()`
   - `inner_product()`
   - `matvec()`
   - `matvec_triangular()`
   - `matvec_triangular_guided()`
2. For the vector operations, create one plot showing the speedup in the strong scalability study and one plot showing the speedup in the weak scalability study. Which operations show good strong scalability? Which operations show good weak scalability? Why do you think this is the case?

   You plots must be correctly labeled and have a legend. The y-axis should be speedup and the x-axis should be the number of processors.
3. For the matrix operations, create one plot showing the speedup in the strong scalability study and one plot showing the speedup in the weak scalability study. Which operations show good strong scalability? Which operations show good weak scalability? Why do you think this is the case?

   You plots must be correctly labeled and have a legend. The y-axis should be speedup and the x-axis should be the number of processors.
4. Why did you observe poor scalability for `matvec_triangular()`? Why did you get better scalability with `matvec_triangular_guided()`?
5. Other than the instructor or TAs, who did to assist or receive assistance from on this assignment?