CS 4380 / CS 5351
Parallel Programming / Parallel Processing
Fall 2016

Final Project (Part 1)

Due: *Thursday, December 8 at 3:30 PM*

All project files including the report have to be submitted using TRACS. Please follow the instructions at http://tracsfacts.its.txstate.edu/trainingvideos/submitassignment/submitassignment.htm. Note that files are only submitted if TRACS indicates a successful submission. See below for what files to submit. The project report has to list and discuss the results of your measurements and provide the answers to the questions. The answer to each question is limited to 50 words. This project has to be done *individually for graduate students* and can be done *in pairs for undergraduates*. You have to be able to explain all of the code that you submit, including code written by your partner in case of a pair project.

Compiling hybrid parallel programs on Stampede

Since some versions of nvcc do not support OpenMP and icc does not support CUDA, the source code of programs that combine these parallelization strategies may need to be distributed over multiple files, which have to be compiled and linked in the following way to generate a combined executable:

```
module load cuda
icc -xhost -openmp -O3 -c fractal_hyb1.cpp -o Cfractal.o
nvcc -O3 -arch=sm_35 -c fractal_hyb1.cu -o CUfractal.o
icc -xhost -openmp -O3 Cfractal.o CUfractal.o -lcudart \
  -L$TACC_CUDA_LIB -o fractal_hyb1
```

Note that the above commands will not work if copied verbatim into a makefile.

Compiling hybrid parallel programs on the lab computers

Use the following commands to generate a combined executable on the lab machines:

```
export LD_LIBRARY_PATH=/usr/local/cuda-5.5/lib64/
g++ -march=native -O3 -fopenmp -c fractal_hyb1.cpp -o Cfractal.o
nvcc -O3 -arch=sm_20 -c fractal_hyb1.cu -o CUfractal.o
g++ -march=native -O3 -fopenmp Cfractal.o CUfractal.o -lcudart \
    -L /usr/local/cuda/lib64/ -o fractal_hyb1
```

6.1 OpenMP + CUDA Fractal [50u/35g points]

Make a copy of the provided skeleton files (and submission script) at /home1/00976/burtsche/Parallel/fractal_hyb1.* and study them carefully (prototypes, comments, etc.).

Complete the hybrid OpenMP and CUDA program by writing and inserting the missing code sections following the comments in the code. In addition to the frame width, there are now two command-line parameters (both integers) to specify the number of frames to compute on the CPU and the number of frames to compute on the GPU. Use 16 threads on the CPU and 512 threads per block on the GPU.

Then compile the files as outlined above. Run a small test first, check the resulting bitmaps, and make sure they are correct, especially at the transition point between the CPU and GPU computation. Once everything works, run the code on Stampede using the provided submission script. Do not modify this script. Present the runtimes in a single chart. Explain and discuss the results, especially the cases where the CPU or GPU gets all the frames and the general behavior of the runtime as a function of the CPU frames. Which CPU/GPU distribution of the frames results in the fastest speed? How much faster is the best hybrid execution relative to just using the GPU? Submit the completed .cpp and .cu files for this part of the project.

Code Requirements

- Make sure your code is well commented.
- Make sure your code does not produce unwanted output such as debugging messages.
- Make sure your code's runtime does not exceed the specified maximum.
- Make sure your code is correctly indented and uses a consistent coding style.
- Make sure your code does not include unused variables, unreachable code, etc.

Code Submission

- Delete all files that you do not need anymore such as *.o and *.bmp files.
- Make sure your code complies with the above requirements before you submit it.
- Any special instructions or comments to the grader should be included in a "README" file.
- Upload all the files you need to submit onto TRACS. The report has to be in PDF. All other files, including source code, have to be plain text files (e.g., *.cpp files).
- For group projects, include a comment at the beginning of the source code and the report that lists both group members. The two group members have to submit identical files on TRACS.
- Upload each file separately and do not compress them.
- Do not submit any unnecessary files (e.g., provided or generated files).

You can submit your file(s) as many times as you want before the deadline. Only the last submission will be graded. Be sure to submit at least once before the deadline.

December 1, 2016