CUDA Notes

Eric Andrews

April 6, 2018

CUDA Implementation

- General Structure
  - Send matrices to be multiplied to the GPU
  - Pass pointers to the starts of sub-matrices to be multiplied to the GPU
  - Have each thread compute the element of C corresponding to its id—replace the outermost two loops with threadIdx.x-based indexing
  - Accumulate these results into a blank C matrix stored on the GPU
  - Pull the C matrix from the GPU to the processor
- Issues
  - Testing a 31 × 31 matrix of all 1's yields accurate results (a 31 × 31 of 31's); however, when using random number distributions the benchmarking program throws a componentwise bounds error.
- Speedup
  - There is no noticeable speedup (only around 1%, which could very well be due to factors independent of the program).

OpenACC Implementation

- General Structure
  - Basic code structure is identical to dgemm-blocked; however, the do-block code is (theoretically) parallelized with openACC, using two pragma acc parallel loop independents as per the manual
  - Each sub-matrix is sent to the GPU using copyin, then calculated and the resulting C matrix sent back with copyout
- Issues
  - There are no known issues

- Speedup
    - Speedup is negligible, if existent.