CS3014 – Sparse Parallel Multi-Channel and Multi-Kernel Convolution

Student name: Davy Nolan

Student no: 17330208

Function Modifications

- Improved new_empty_3d_matrix() function
  - Given function was not functioning correctly due to it using a 4-D matrix to create a 3-D matrix.
- Added function for copy_3d_matrix() function
  - Given code only contained copy_4d_matrix() method.
- Improved gen_random_3d_matrix() function
  - Modified to work similarly to gen_random_4d_matrix() function.

OpenMp

- #pragma omp parallel
  - This allowed parts of the program to run using multiple threads concurrently.
  - Improved runtime.
- Parallel For
  - Divided up the iterations of a for loop between the threads.
- Collapse()
  - Merged several for loops into an iteration space and divided accordingly to the schedule clause.

OpenMp

- The sequential execution of the iteration in all associated loops determines the order of the iterations in the collapsed iteration space.
- This increased the total number of iterations that will be partitioned across the threads by reducing the granularity of work to be done by each thread.

Runtimes

- Small
  - 16 16 1 32 32 5 => 0.000893 secs
- Medium
  - 128 128 1 512 512 5 => 9.47 secs
- Large
  - 256 256 3 1024 1024 5 => 43.56 secs

What I Learned…

- I learned that optimisation of code and algorithms is key is so many areas of computer science.
- However, if the optimisation of code requires so much effort and little reward, it is oftentimes redundant, especially if execution time is not critical to the functionality of the application.