ABSTRACT

Matrices are one of the most used data representation form from real-world problems. Lot of matrix were formed very big but sparse, hence information inside the matrix is relatively small compared to its size. This caused into heavy computational resources needed to process those matrices within short time.

One of the solution to do an efficient process to the matrix is to form it into a specialized form of sparse matrix, such as Sliced Coordinate List (SCOO). SCOO format for sparse matrix has been developed and combined within an implementation using Compute Unified Device Architecture (CUDA)-architecture Graphics Processing Unit (GPU). In this research, performance of SCOO implementation performansi using GPU – CUDA will be compared to the other sparse matrix format, Coordinate List (COO) based on its memory usage and execution time when running Sparse Matrix-Vector Multiplication (SpMV) operation.

Results obtained from this research show that although SCOO implementation for sparse matrix use memory 1.000529 larger than COO format, its serial performance is 3,18 faster than serial COO, beside that, if SCOO implementation is conducted parallel using GPU – CUDA then its performance can be achieved around 123,8 faster than parallel COO or 77 times faster than parallel COO using one of the available library for CUDA, named CUSP.

***Keywords**: sparse matrix*, COO, SCOO, CUSP, SpMV.