Abstract

Recent year, large scale numerical simulations on On-site environment is widely exploited in many fields as General-purpose computing on graphics processing units (GPGPU) technology advances. GPGPU is a technology that apply GPU for general application in order to acceleration, not for image processing. Since GPU has huge calculation cores compared to CPU, applications can be calculated by high parallelism on GPU. Solving a large linear system is necessary for calculating Finite Different Method (FDM) and Finite Element Method (FEM) which are often used for numerical simulation. Therefore, the acceleration of solving a large linear system by using GPGPU leads to reducing the calculation time of the numerical simulation. Huge memory space is required to store a sparse matrix which represent a linear system during the solving linear system. The sparse matrix must be transferred to GPU memory from CPU memory before calculation in order to solve the linear system on GPGPU. However, if a sparse matrix does not fit GPU memory, GPU spends much time on the frequency of data transfer between CPU memory and GPU memory. To solve this problem, we propose two memory-saving sparse matrix storage formats, Pattern Compression (PatComp) and Row Block Packing (RBP), that take into account the regularity of sparse matrices generated by FDM and FEM. The PatComp method focuses on the fact that the order of non-zero elements in several rows of a sparse matrix is the same. By using patterns to represent the positions of non-zero elements, memory usage can be reduced. By patterning the sequence of non-zero elements in each row of a sparse matrix and registering the pattern in a table, it is possible to decode the positions of all the non-zero elements in a row from a few patterns. Therefore, the memory usage of PatComp is much lower than that of existing methods. In order to solve the problem that the usable problems of PatComp are limited due to its long conversion time, the RBP method is designed to speed up the conversion process. Assuming that some parts of the column indices in the sparse matrix are consecutive and such parts can be described with its minimum and maximum column number. RBP method is able to conversion at high speed because finding minimum and maximum columns number is easy. In our experiments, the PatComp method reduces the memory usage of the general sparse matrix up to 31.1% in 13 out of 15 matrices compared with conventional storage format. Also, the RBP method reduces the memory usage of the general sparse matrix up to 28.0% in 13 out of 15 matrices. in the evaluation of Sparse Matrix-Vector multiplication (SpMV) execution time, the PatComp method and RBP method reduce the execution time of SpMV in several sparse matrices. In the evaluation of the execution time of GMRES which is a kind of iterative method, we confirmed that conversion time of RBP method is shorter than PatComp method. From this result, the RBP method is able to adopt a lot of problems. In conclusion, the PatComp method and RBP method are able to drastically reduce the memory usage of the sparse matrix. Also, the execution time of SpMV using PatComp and RBP are comparable to conventional sparse matrix formats. We are able to execute larger numerical simulations because of reducing memory usage of the sparse matrix by the PatComp method and RBP method.

Keywords: GPGPU, SpMV, Sparse matrix format, Data compression, FEM