Study on GPU-accelerated Extraction of Interconnects Parasitic Using CUDA and MPI

Xiaoyu Xu[1], *IEEE Member*, Guoqiang Liu[1], Hui Qu[1], Wei Xu[2] and Yang Zhang[1]
[1] Institute of Electrical Engineering, Chinese Academy of Sciences
No.6, Beiertiao, Zhongguancun, Haidian, Beijing, 100190, China
[2] School of Electrical, Mechanical and Mechatronic Systems, University of Technology Sydney, Australia
xuxiaoyu@mail.iee.ac.cn

Abstract— Parallel computation is application-oriented, particularly for the GPU (Graphics Processing Unit) with the inherent parallelism. This paper shows the architecture of a GPU cluster based on MPI (Message Passing Interface) and CUDA (Compute Unified Device Architecture). Results show that the acceleration ratio is obviously improved but the acceleration effect seems decelerated in large-scale GPU cluster. The parallel algorithm is mainly focused on task partitioning sparse matrix-vector multiplications (*SpVM*) in GPUs.

I. Introduction

The feature size of integrate circuit shrinking to nano scale and demands for production period always keeping rigorous, the accuracy, consuming time and processing capacity of interconnects parasitic extraction become very critical for EDA industry. And parallel computation is able to reduce computing time, enlarge the scale of problems and increase the physical complexity. Parallel means processing units with multi-core or many-core characteristic or clusters consist of a certain amount of processing units. Comparatively speaking, GPU with inherent parallelism is a cheaper but more efficient option to approach large-scale density scientific computation.

II. System Construction and Implementation

CUDA provides an integrated programming model to facilitate programming on GPUs. Aiming at studying the manifold electromagnetic problems, a high-performance cluster with 27 compute nodes and 1 management node based on GPUs is established as in Fig. 1, which has two levels parallelism: the coarse-grained one between GPUs and the fine-grained on inside each GPU core.

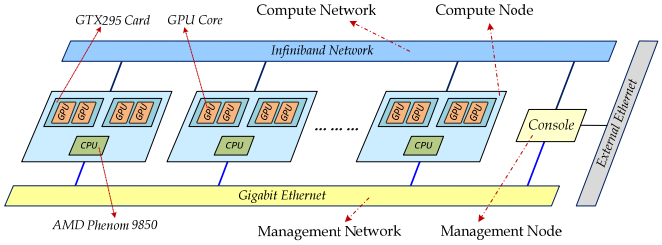

Fig. 1. Structural schematic of parallel system based on CPU+GPU. Each compute node consists of 2 GTX295 cards, i.e. 4 GPUs and 1 four-core CPU.

After meshing the geometrical structure, the coefficient matrix of computing for extraction of capacitance between Metal *1* and *2* is different from that between Metal *3* and *4*. Then the extraction tasks can be partitioned based on the main conductor into different GPUs. The common FEM and nodal discontinuous Galerkin FEM for electrostatic field are used. Parallelization is applied to all aspects of FEM solving process including meshing, coefficient matrix establishing, and sparse matrix solving [1][2][3]. *SpMV* algorithms considering GPU's process and memory architecture are adopted to process sparse matrix [4]. But the patterns of sparse matrix are variable. Concretely, block size should be adjusted for adapting the matrix to obtain higher computing density.

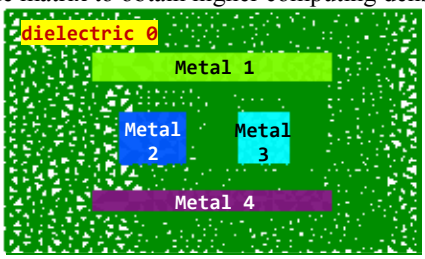

Fig. 2. The discrete mesh result of the sample for extraction of interconnects parasitic under the case that Metal 3 is appointed as the main conductor.

III. Results

Table I. Results Using 2 Computing Nodes with Common FEM

Case	Problem scale	Efficiency
1	Nodes: 8436, Elements: 16614 Matrix dimension: 5639 Non-zero elements: 37961(0.119%)	CPU: 390ms GPU-1: 326ms, *At.* 1.20 GPU-2: 204ms, *At.*1.91 GPU-3: 189ms, *At.*2.06
2	Nodes: 41084, Elements: 81595 Matrix dimension: 28113 Non-zero elements: 193503(0.025%)	CPU: 2218ms GPU-1: 1092ms, *At.*2.03 GPU-2: 693ms, *At.*3.20 GPU-3: 617ms, *At.*3.59

* GPU-1/2 means 1/4 GPU(s) within 1 compute note; GPU-3 means 8 GPUs within 2 compute nodes; All running modes are CUDA+MPI. *At.* means acceleration ratio comparing with the capacity of CPU.

IV. Conclusion

The viability of parallelism based on GPU cluster utilizing CUDA and MPI is approved. In view of electromagnetic field problems, acceleration ratio is easily improved with a small amount of GPUs, but large-scale GPU cluster seems likely suitable for task decomposition level problems as shown in Table I. However, parallel computation is application-oriented, some strategies such as block size, memory allocating pattern should be emphasized for different problems.

V. References

[1] J.S. Hesthaven and T. Warburton, Nodal Discontinuous Galerkin Methods. Springer, 2008.
[2] PETSc. http://www.mcs.anl.gov/petsc/petsc-as/, 2010.
[3] METIS. http://www.cs.umn.edu/~karypis.
[4] N. Bell and M. Garland. "*Efficient Sparse Matrix-Vector Multiplication on CUDA,*" NVIDIA Technical Report NVR-2008-004, December 2008.

978-1-4244-7062-4/10/$26.00 ©2010 IEEE