A Study of CUDA/MPI Parallel Computations for CADMAS-SURF/3D

Hiroshi Okumura
Information Technology Center, University of Toyama.
Toyama City, Toyama, Japan

Taro Arikawa
Maritime Structures Group, Coastal and Ocean Engineering Field, Port and Airport Research Institute.
Yokosuka City, Kanagawa, Japan

ABSTRACT

We present a new faster GPU/MPI computation and algorithm for CADMAS-SURF/3D by a CUDA parallelization technique on a GPU accelerator “Tesla”. In this paper, we examine the CUDA porting tool for GPU computation and suggest a macro code by C preprocessor for CADMAS-SURF/3D originally coded in FORTRAN to be adapted into CUDA C language. Furthermore, we optimize the data access sequence pattern from a profiling result to eliminate the calculation bottleneck. We inspect the legitimacy of the solution by examining the rounding error, and then examine the program tuning a data sequence access pattern optimization.

KEY WORDS:
CADMAS-SURF/3D, GPU, CUDA parallelization, array data access optimization

INTRODUCTION

For designing of seawall coastal/maritime structures, a computational two/three dimensional free surface flow simulation, in which CADMAS-SURF/3D is a defactostandard representative in the coastal engineering of Japan, could be used in actual occasionally as an effective to a conventional hydraulic model experiment and its based design formulas. Recently the CADMAS-SURF/3D is a kind of open source code from the cite of the Coastal Development Institute of Technology (CDIT) and becomes a powerful tool for the actual cases in Japan widely used by scientist and engineers in the field of coastal engineering to design a seawall coastal/marine structure. The CADMAS-SURF/3D enables us to compute both of velocity and pressure in three dimension by a SMAC (Simplified Marker and Cell) based fractional step method which is one of a finite element method to solve Navier-Stokes equations. The most part of a computational cost in CADMAS-SURF/3D appears to solve pressure of Poisson equation, whose matrix is sparse and asymmetric, by ILU-BiCG method as algorithm. In the assembler level of CADMAS-SURF/3D, it’s processing and routines are almost calculations of double floating point numbers. Therefore, there would be left a problem in a sense of computational cost and cpu time within CADMAS-SURF/3D although the accuracy of computation by CADMAS-SURF/3D achieves a level requested to design accuracy seawall coastal/marine structures. More speeding up of CADMAS-SURF/3D itself on a single computer machine would be eagerly anticipated because a high specified parallel computer is usually expensive for ordinary institutions and scientists, although the conventional CADMAS-SURF/3D has tried with the implementation of distributed memory parallel computation (MPI) to resolve this computational problem in CADMAS-SURF/3D.

In this paper, we therefore inspect its applicability and the effectiveness of a new faster computation of CADMAS-SURF/3D by CUDA (Compute Unified Device Architecture) parallelization technique on a GPU (Graphics Processing Unit) accelerator which would be focused widely for a next generation of computations. In CUDA which is the platform or the integrated development environment proposed by NVIDIA Corporation recently, we focus massive data parallel calculations of GPU in which a plenty of simple processing units is on-board. A GPU computation could expect faster calculations and these proceedings than those of CPU computation which consist of a little of processing units and a complicated system. The code name ‘Kepler’ for the CUDA integrated development environment has been introduced by NVIDIA cooperation in 2012. At this time, the Kepler is in widespread use, whose core number in GPU achieves a thousand and more. It is safe to say that GPU devices are going to trend towards increased number of core in GPU in the future. In this paper, we examine the CUDA porting tool for GPU computation and suggest a macro code by C preprocessor for CADMAS-SURF/3D originally coded in FORTRAN to be adopted into CUDA C language in this research. Consequently the source code ported and parallelized by the CUDA obtains an ideal linear speeding up with replacing a newer GPU device without any additional porting.

In this paper, in order to speed up the original CADMAS-SURF/3D program (which currently runs on x86 system) on-board the GPU accelerator ‘Tesla’, we would like to summarize a porting the original codes on the GPU and examining an effectiveness. Furthermore we optimize the data access sequence pattern from a profiling result to eliminate the calculation bottleneck. Finally we would like to conclude the performance measurement resulting from the CUDA parallelization.

EXTRACT FOR CUDA PARALLELIZATION

In order to run a program of CADMAS-SURF/3D on-board of the GPU accelerator, it is necessary that the CUDA porting as a GPU code is