CFD Solver Case Study

Dated: June 24, 2015

nextremer

Table of Contents

Abstract 3

Background 4

Client Information 5

  Architectural Diagram for CFD solver application 5

  Parallelization Strategy 6

Benefits 9

Technology Highlights 9

Abstract

Making highly accurate and high speed CFD Solver for sound field around a car in short time.
High accurate simulation technique DG –SEM (Discontinuous Galerkin Spectral Element Method)
DG–SEM work on complex shape, accurate of simulation and parallel calculation.
High Speed Simulation technique GPGPU using Computation Fluid Dynamic (CFD)
CFD is lower cost method and easily available without real object.
To simulate within 1% error between experiment and simulation results.

To calculate in 12 hours on GPU cluster computer

Background

Previously, the car used to be designed manually, mechanically, and its prototype was created. After the development of the actual car, it was put to test. If and when the test is okayed, that it used to go for mass production. This process was of course, time consuming and also was a costly affair. The different kinds of test, e.g.

But today's development is fast and efficient in matters of time and cost. Companies resort to the computer software to design a car's prototype and after the design is final, and then the actual production starts. Also, because of the internet, this design can be made anywhere and ported anywhere.

CFD solver application was developed in C, C++ and CUDA on centOS Linux platform.

CFD solver application supports different calculations using various simulation methods of CFD. CFD was used to implement various equations. CFD is the low cost method which works without real objects. CFD solver Application has its own mesh and configuration file as an input parameter to initialize the respective fields. CFD solver application is mainly related with the solving of governing equations and imposing boundary condition.

CFD solver application was mainly concerned with the aero acoustic problem, sound generation around a car. Making highly accurate and high speed CFD solver for a sound field around a car in short time. Developed CFD Solver application which is 20 time faster than the current solver. Highly accurate simulation technique approach was DG-SEM (Discontinous Galerkin Spectral element method) and high speed simulation technique was GPGPU. DG-SEM can be applied to the complex shape and it gave us very accurate result of simulation. It was best solution for the parallel calculation.

CFD solver application was faced with the number of challenges, The key challenges were

1) The application involved numerous mathematical functions. So, the opportunities for performance improvement using CUDA optimization techniques.

2) The application did not cater to running multiple threads; we implemented the MPI and MultiGPU for performance improvement and scalability.

3) CFD solver application source code optimization and reduced each function execution time to an appreciable level.

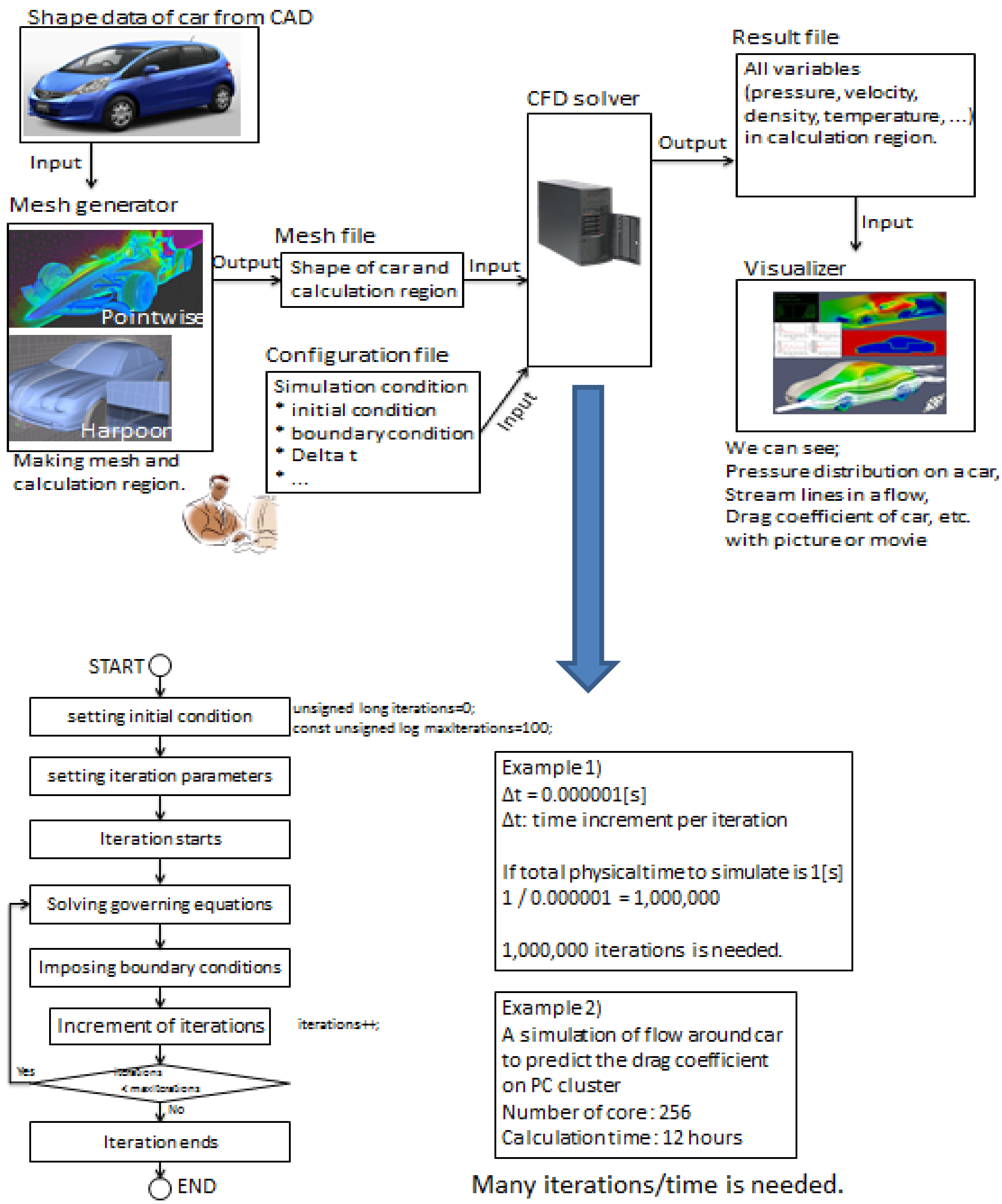
4) Physical Boundary condition implementation of CFD solver application including Jacobian.

5) External force term implementation.

Client Information

The client is a Japan Automobile Giants R&D division that is focused on intelligence science, conducting research on interface intelligence, and nano science, participating in genome research.

Architectural Diagram for CFD solver application



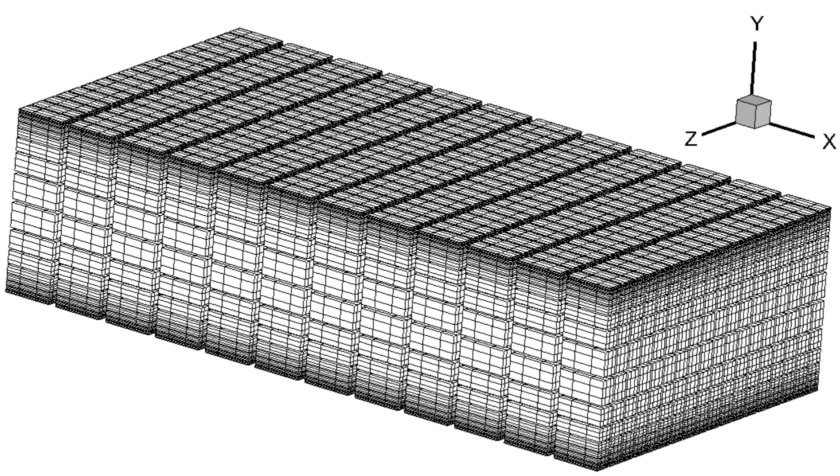
The above figure shows the architecture of CFD solver.

Here, car is designed using CAD and generated in Mesh file.This mesh file is provided to CFD solver.In addition configuration file is also used as a input for the CFD solver. Configuration file contains initial conditions, boundary condition, time step, maximum number of iterations, and number of degree of freedom of each element in the mesh and fluid properties.

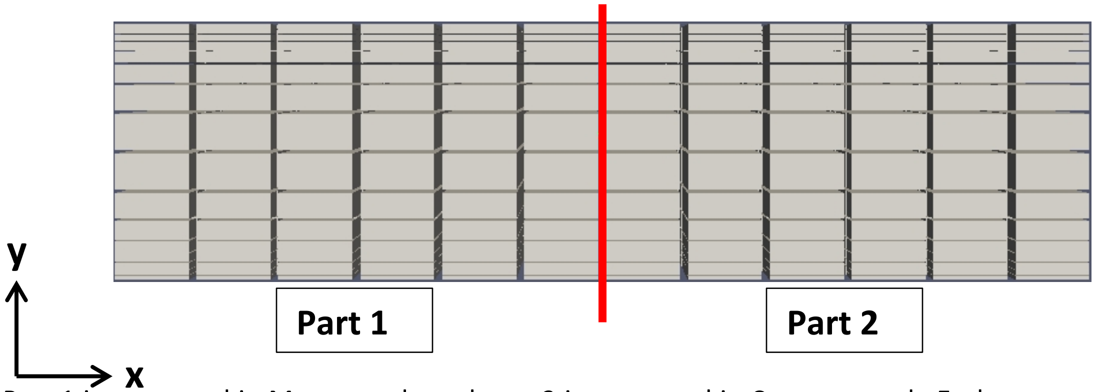
Using this mesh and configuration file, the solver runs for number of iterations mentioned within the configuration file.Ouputs of this solver is collected in file at every specific iterations. These files contains Momentum in X, Y and Z directions and energy density for each and every element in the mesh. Using these files, animation of flow around a car was drawn with the help of Paraview tool.This animation helps in the study of sound generated by flow vortex casued by turbulence.

Parallelization Strategy

Below figure shows the domain in 3D which is used for turbulent channel flow.
Number of degree of freedom used for each element is 5 in all directions.

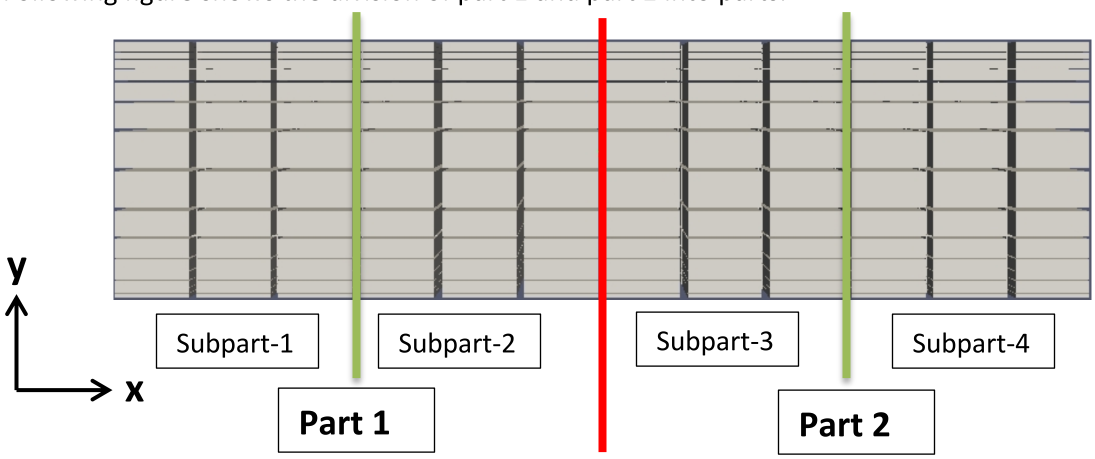


The above domain is divided into two equal parts. Each part is processed in two separate compute nodes. Following figure shows the division of the whole domain.



Part 1 is processed in Master node and part 2 is processed in Compute node.Each compute node is having two “Tesla K20X” GPUs of same configuration. Again each part of the domain is divided into two equal parts. Here, part 1 is divided into 2 equal parts, subpart-1 and subpart-2. Part 2 is also divided into two equal subpart, subpart-3 and subpart-4.

Following figure shows the division of part 1 and part 2 into parts.

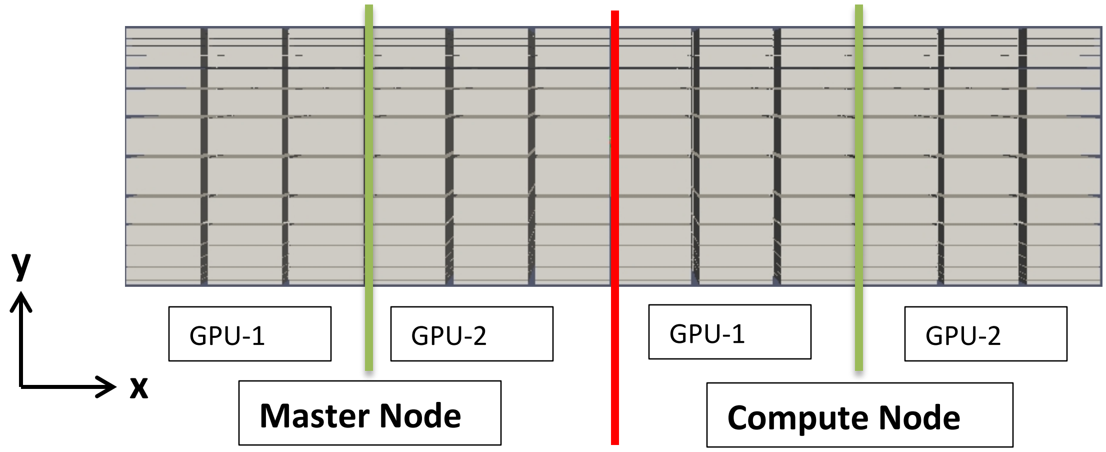


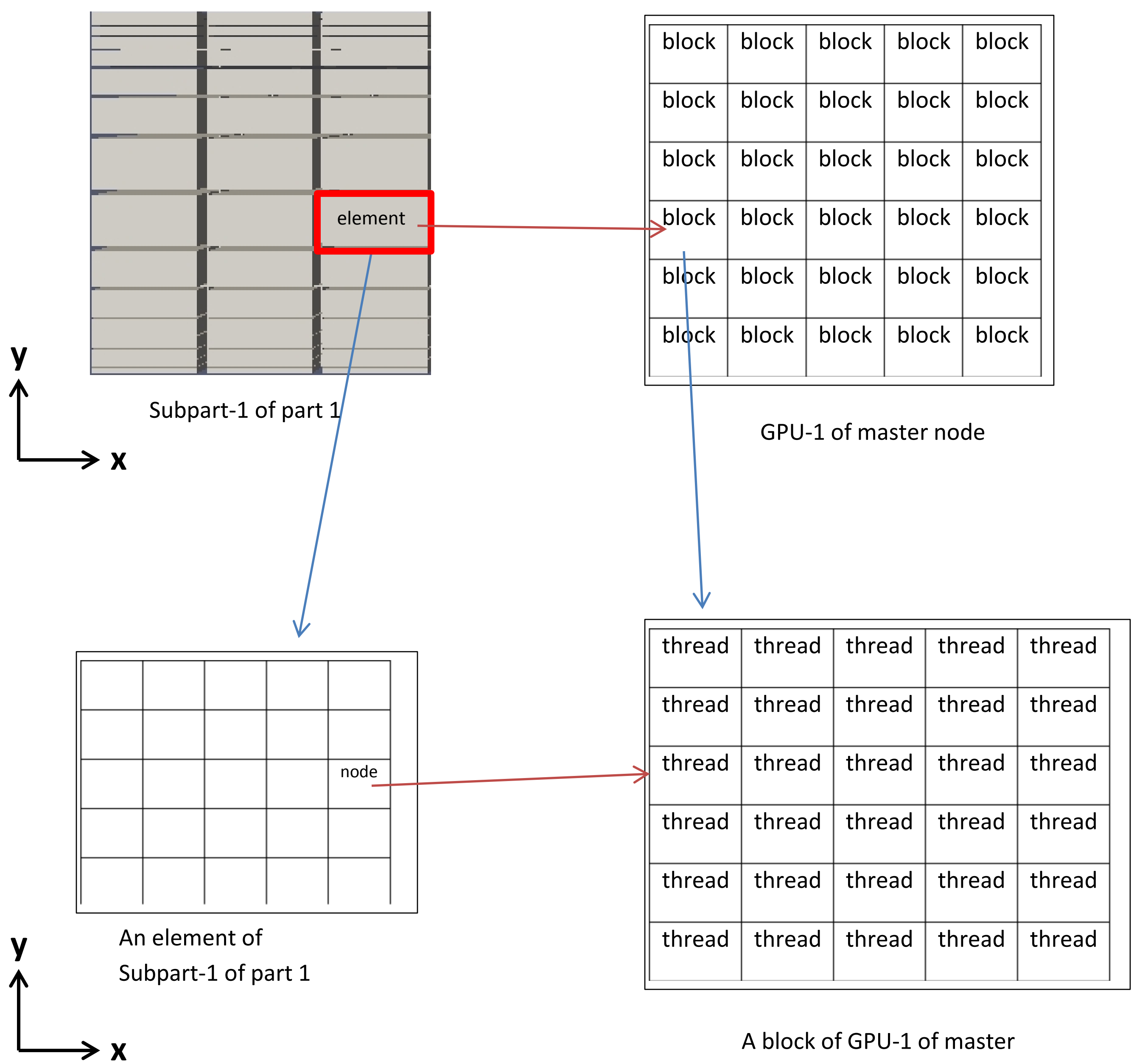
Subpart-1 and subpart-2 are processed on GPU-1 and GPU-2 respectively of master node. Similarly subpart-3 and subpart-4 are processed on GPU-1 and GPU-2 respectively of compute node.

In current turbulent channel flow simulation, there are 3456 total elements. As per the above division, 864 elements are processed on each GPU of each node. A kernel function is launched on GPU with grid and block size. Grid is made up of number of blocks and a block is made up of number of threads. Each thread in a block runs parallel to each other. Each block can have maximum 1024 threads. Here, each element is having number of degree of freedom as 5 in all directions. Therefore each element consists of 25 internal nodes. Each element is processed in each separate block and each node of an element is processed by one thread of a block. That means the number of blocks to be launched on a GPU is equal to the number of elements being processed on the same GPU. And the number of threads in each block is equal to the number of internal nodes of elements.

Note: In case of number of internal nodes is greater than 1024, one element is processed in multiple blocks instead of one.

Following figure shows the number of blocks and threads launched on GPUs.





Benefits

Initially CFD solver application was taken too much time for the execution, after applying the concepts of CUDA optimization and MPI with MultiGPU, we reduced the execution time of all the functions.

Detailed of how the CFD solver was made to run faster

1) We had developed CFD solver application which is 20 times faster than the current CPU source code.
2) Impact to manufacturing flow – If sound simulation is available then we can do physical simulation with mass production in factory. There is no redesigned of the parts of a car.
3) Having the simulation technique we can avoid Mechanical design as well as prototype test directly.

Technology Highlights

- C / C++
- Linux-centOS
- CUDA
- MPI
- MULTI-GPU
- CFD
- DNS (Direct Numerical Simulation)