Survey on 2D Conjugate Heat Transfer Solver on GPU

[1]Manasi M. Mahamuni,
[1]Department of Computer Engineering,Pune Institute of Computer Technology, Savitribai Phule PuneUniversity,Pune,India.
Email :
mnsmahamuni@gmail.com
[2]R.S. Paswan
[2]Department of Computer Engineering, Pune Institute ofComputer Technology Savitribai Phule Pune University, Pune,India
Email : rspaswan@pict.edu
[3]Vinaya Sivanandan,
[4]Vikas Kumar,
[3,4]CAE Group, CDAC,Pune,India
[3]Email: vinayas@cdac.in
[4]Email:vikask@cdac.in

***Abstract* -Conjugate heat transfer relatesto combination of heat transfer in solids and fluids. Conjugate heat transfer solver in this paper performs fluid flow analysis using finite volume method. This solver uses Semi Implicit Method for Pressure Linked Equation(SIMPLE)algorithm which is pressure based numerical scheme to solve two dimensional steady viscous incompressible flows. The solver will be validated using the open source software OpenFOAM solver.**

Keywords- SIMPLE algorithm; FVM;CUDA;GPU

I. I NTRODUCTION

Conjugate heat transfer is a coupled heat transfer problem which tells how transfer takes place between solids and fluids. In solids, conduction takes place whereas in fluids, convection takes place. Such transfer of heat is observed when the heating or cooling of a solid object takes place due to the flow of air in which it is immersed. Solver for conjugate heat transfer solves different heat equation and finds out temperature variation.Solvers are useful fordesigning coolers,heaters, or heat exchan-gers,refrigerators.

Solver will work more efficiently if it is run on GPU i.e.Graphical Processing Unit.GPU provides co-processing computing model which uses a CPU and GPU together to enhance the speedup. Sequential part is executed on CPU. Parallel tasks are performed on graphics processors. So huge amounts of work can be done with the help of GPU.It consists of many computing cores with high memory bandwidth which are responsible forparallel computations.

NVIDIA has provided CUDA (Compute Unified Device Architecture) which is a programming model for parallel computation on GPU[3].GPUs are actually used to run high definition graphics on machine. By using this programming model, developers can use CUDA-enabled GPUs to perform general purpose processing. CUDA programming model is based on languages like C,C++ and FORTRAN. Proposed solver will be developed using CUDA on GPU.Following sections will introduce with the GPU, CUDA architecture.

A. G PU Architecture

GPU is made up of multiple cores as shown in the fig.1. These cores are Streaming Processors (SPs) which are single instruction multiple data units used for threading during programming using CUDA. Streaming Processors (SPs) together form Streaming Multiprocessors(SMs). These Streaming Multiprocessors are connected to Dynamic Random Access Memory (DRAM) i.e. the global memory or simple GPU memory of the GPU[5].

DRAM of the CPU and GPU are different. Figure shows each block of GPU consists of 2 Stream Multiprocessors.Number of SMs per block in GPU can vary with generation.

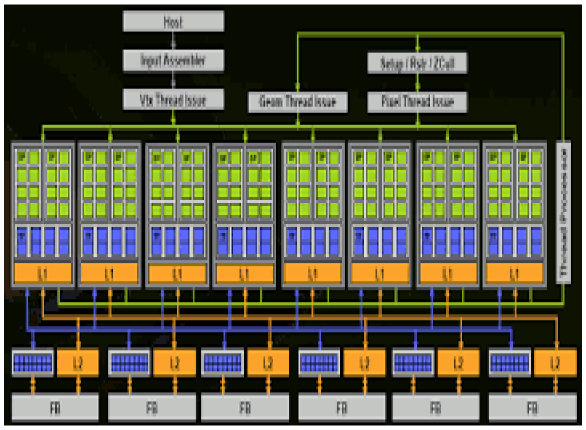

Figure 1. GPU Architecture

B. CUDA Architecture

CUDA divides the GPU into 3 parts grids,blocks,threads[9]. Grids consists of group of blocks. Block is a logical unit which consists of group of threads. Streaming Multiprocessors have blocks assigned to it.Blocks resides their till it finishes its computation.An SM can have several blocks assigned to it at the same time and schedules work between them as it finds suitable. Number of threads per block and number of blocks per grid is decided by developer depending on the requirement.Threads run on stream processors which are coresofstreaming multiprocessors on

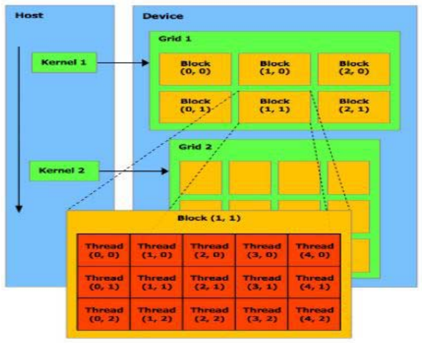

Figure 2. CUDA Architecture

the GPU.

CUDA program has two parts kernel and sequential. Kernel contains instructions to be executed on GPU. It resembles a function in sequential programming.When a kernel starts its execution on GPU, each thread executes the statements in that kernel.Each thread maps to a different element of data. Thus, the architecture can be classified as SIMD (single-instruction, multiple-data)[5].

C.CUDA Memory

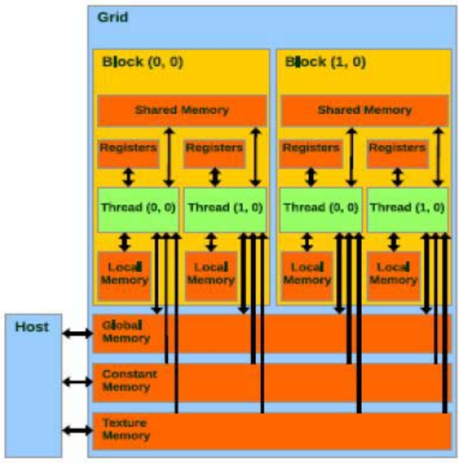

Figure 3. CUDA Memory Types

CUDA refers to 5 types of memories,Global Memory,Texture Memory,Constant Memory,Shared Memory,Local Memory[9]. Global Memory is a read-write memory which is accessible by all blocks and threads.Texture memory is a read only memory.All constants and kernel arguments are stored on in constant memory.Shared memory is associated with a single block which is shared by threads in it for read-write operations.Every thread has a local memory associated to it for storing its local data.

II.RELATED WORK

Till now we have seen brief information about GPUs and CUDA. Due to multi-core architecture of GPUs the computing power of desktop computers is increased by an order of magnitude. For many years, utilization of this increasing computational power was exclusively for visualization of graphics applications. Nowadays, things are starting to get different, cutting edge engineering and scientific applications are using the extra processing power from the graphics processor to gain speedup in orders of magnitude. Fastest supercomputers in the world are designed with Graphics Processing Units (GPUs) to increase computational power and to reduce power consumption.The speedup is due to cores present on GPU. Specific programming platforms and models, such as CUDA and OpenCL, have emerged to allow for general purpose GPU programming.

Graphics Processing Units (GPUs) can be used as highly capable computational accelerators for different scientific and engineering applications.With the availab- ility of small clusters of GPUs, performances of several tens of teraflops can be achieved on low footprint and low energy consuming supercomputers. The use of GPUs for CFD applications like solvers is rapidly getting popular, and a number of researchers have found their usage to be beneficial. Researchers have developed solvers using different techniques on GPUs and have compared its performance with CPU. After study, they observed large improvement in performance of solver. Below there is overview of work of some of such researchers.

Tadeusz Tomczak, Katarzyna Zadarnowska implemented pressure-implicit with splitting of operators (PISO) and semi-implicit method for pressure-linked equations (SIMPLE) solvers on Graphics Processing Units using CUDA technology.They tested the validity of the implementation for many standard, steady and unsteady problems.They found that a GPU (Tesla C2070) outperformed a server-class 6-core, 12-thread CPU (Intel Xeon X5670) by a factor of 4.2[7].

Niklas Karlsson implemented CG, GMRES and BiCGStab iterative solvers on GPU using CUDA, evaluated them together Polynomial preconditioners. He also implemented double precision Navier-Stokes solver using CUDA, SIMPLEC pressure-velocity coupling scheme, and implicit time discretization. For the iterative solvers,he found speedups of between six and thirteen against the MKL CPU library[11].For the full Navier-Stokes solver, speedups obtained was up to a factor twelve compared to an equivalent commercial CPU code when equivalent iterative solvers were used.

Cohen and Molemaker [4] accelerated a solver for the Boussinesq approximation of the Navier-Stokes equations on a regular three-dimensional (3D) grid, and found an 8-fold speedup versus the corresponding multithreaded FORTRAN code developed on an 8-core dual-socket Intel Xeon processor. TÂ¨olke and Krafczyk [6],implemented 3D Lattice-Boltzmann method for flows through porous media on GPU.They found speedup of up to two orders of magnitude faster than a corresponding single-core CPU code.S. Pratap Vanka,Aaron F. Shinn,Kirti C. Sahu developed five different CFD algorithms on GPU and have found speed-ups over a CPU of factors between 10-25[10].

From survey we can see that GPUs are becoming very attractive for computing industrial fluid flows.So it becomes necessary to study parallelization on GPUs.Accelerating engineering computations using parallelization is what brings the GPU into IT field for development of different applications.

III.PROPOSED SYSTEM

Proposed system consists of a solver for detecting conjugate heat transfer due to two dimensional steady incom- pressible flows.Such flows are governed by Navier-Stokes equations.A pressure based numerical scheme will be used to solve incompressible fluid flow equations in this solver. These equations are made up of convection and diffusion terms.It is known that the convective terms are non-linear and influenced by the flow direction. However, diffusion terms are linear, isometric and is not influenced by the flow direction. In the numerical approximation of these governing equations, the combined effects of the convection and diffusion processes must be taken together.That makes the development of numerical schemes to these equations very challenging.These schemes are divided into two categories: density based and pressure based.

In the pressure based schemes (equations written in primitive variables) the pressure will be introduced artificially into the continuity equation and solved from the same. In this approach, it is required to enforce a coupling between the velocity components and the pressure. In general, the convergence rate depends on the coupling of these field variables.There are various schemes reported in the literature to enforce a good coupling between the unknown variables, out of which the first popular scheme may be the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) which will be used for developing this solver.

This scheme is based on finite volume method for fluid flow analysis. In this method region of interest is divided into number of small control volumes which are rectangular in shape[2]. A single node at the centre of the control volume is used for evaluation of each control volume.Equation is solved for each nodes to find out the heat flow in x and y direction parallely using GPU.

As said previouly Semi Implicit Method for Pressure Linked Equation (SIMPLE) algorithm is pressure based numerical scheme to solve incompressible fluid flow equation.In this scheme the coupling between velocity components and the pressure has been achieved by introducing pressure correction , artificially, into the continuity equation[1].Guessed velocity components are used for evaluation of the convective fluxes per unit mass through cell faces.Then a guessed pressure field is used to solve the momentum equations and pressure correction equations.This pressure correction field is used to update the velocity and pressure fields. The algebraic equations will besolved parallely with a guessed pressure field or with the pressure field obtained from some initial velocity field for different velocity components. The computation of velocity field using momentum equations, computation of pressure correction and then correcting pressure and velocity fields using, respectively, has to be repeated iteratively until the obtained velocity field satisfies the mass conservation i.e. convergence of the velocity and pressure fields.

The solver will be validated using the open source software OpenFOAM solver. This is an open source program suite capable of tackling a wide range of both fluid and structural problems.

IV.CONCLUSION

From survey it can be seen that GPUs have been used for designing solvers due to their high processing capability for parallelization. Proposed Solver will be developed on GPU to achieve good speedup.Solver uses SIMPLE algorithm to solve heat equations, which gives us changing heat values due to flow of a fluid.With the help this solver, effect of fluid flow on solid can be studied which will be helpful to design various parts of machines, automobile, heating and cooling systems etc.

ACKNOWLEDGMENT

I take this opportunity to express my deep sense of gratitude to Principal Dr. P.T. Kulkarni and HOD Prof. G.P.Potdar for encouragement, providing with best facilities indispensablesupport and priceless suggestions as well for most valuabletime lent as and when required.I would like to heartly thankCDAC,Pune for providing me excellent opportunity to workwith the organization as an intern and contribute to real timeongoing project. I also thank my family for their continuousencouragement and support. I thank my friends for their helpin collecting information.

REFERENCES

[1] S.V. Patankar, Numerical Heat Transferand Fluid Flow, Hemisphere PublishingCorporation,1980.

[2] H.K.Versteeg and W. Malalasekera, An Introduction to computational FluidDynamics, The Finite Volume Method, Longman Scientific & Technical,1995.

[3] NVIDIA, 2010, CUDA C programming guide, version 3.2.

[4] J. M. Cohen and M. J. Molemaker, “Parallel Computational Fluid Dynamics. Recent Advances and Future Directions, chapter A Fast Double Precision CFD Code using CUDA”,pages 414 - 429. DEStech Publications, 2010.

[5] Jayshree Ghorpade, Jitendra Parande, Madhura Kulkarni, Amit Bawaskar, “GPGPU Processing In CUDA Architecture”, Advanced Computing: An International Journal (ACIJ), Vol.3, No.1, January 2012.

[6] J. TÂ¨olke, M. Krafczyk, “TeraFLOP computing on a desktop PC with GPUs for 3D CFD”, International Journal of Computational Fluid Dynamics, 22(7):443-456, 2008.

[7] Tadeusz Tomczak, Katarzyna Zadarnowska, “Complete PISO and SIMPLE solvers on Graphics Processing Units”, Computers Fluids,2012.

[8] William D. Henshaw, Kyle K. Chand, “A Composite Grid Solver for Conjugate Heat Transfer in Fluid-Structure Systems”, Elsevier,2009.

[9] Er.Paramjeet kaur and Er.Nishi, “A Survey on CUDA”, International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014.

[10] Aaron F. Shinn,S. Pratap Vanka,KirtiC. Sahu,"Computational Fluid Dynamics Using Graphics Processing Units: Challenges And Opportunities", International Mechanical Engineering Congress & Exposition,November 11-17, 2011.

[11] Niklas Karlsson, An Incompressible Navier-Stokes Equations Solver on the GPU Using CUDA, Chalmers University of Technology University of Gothenburg, Sweden, August 2013.

AUTHORS PROFILE

Manasi M. Mahamuni has completedher B.E. in computer engineering fromMKSSS´s Cummins College of Engineeringfor Women,Pune.She is pursuing her M.E.in computer engineering from Pune Instituteof Computer Technology,Pune. HerResearch area is High Performance Computing.

Ratnamala S. Paswan is presently working as assistantprofessor in the Department of Computer Engineering at PuneInstitute of Computer Technology, Pune. She receivedM.E. in Computer Engineering from Pune Institute of ComputerTechnology,Pune. Her research area includes Big Data Analytics and Machine Learning.

Vinaya Sivanandan is presently working as Technical officer of HPC Scientific and Engineering group at CDAC, Pune. She received M.Tech in Industrial Mathematics and Scientific Computing from IIT, Madras. Her research area includes High performance Computing,Computational Fluid Dynamics.

Vikas Kumar has been leading CFD groupof CDAC since 2002. He worked as a Research Engineer inComputer Aided Engineering Department at Thapar Centre forIndustrial Research Development, Patiala from 1996 to 2002.He has got his Ph.D. in 2005 in Mech. Engg with specializationin Heat Transfer from Thapar University, Patiala. He has carriedout his Post Doctoral research work at University of Nebraska, Lincoln, Nebraska, USA during 20072009. His areas of interestis Computational Fluid Dynamics Simulation in Heat TransferFluid Flow, and applications of High Performance Computing