International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 4 154 - 156

Survey Report on Multi GPU Implementation in Open Source CFD Solver

Suraj Salvi,
Department of Computer Engineering
Pune Institute of Computer Technology
Pune, India

Abstract - A CFD (Computational Fluid Dynamics) solver is used for simulation of fluid flow and heat transfer. It helps in analyzing the various parameters in fluid flow of scientific applications. OpenFOAM (Open Field Operation and Manipulation) [14] is an open source Software which can be used to solve Computational Fluid Dynamics (CFD) problems. The OpenFOAM solves various equations e.g. Mass Conservation, Conjugate Heat Transformer. OpenFOAM solves CFD problems serially as well as in parallel. The motivation of the project is to reduce time for solving CFD problems by implementing it on GPU. The OpenFOAM a software contribution which (Open Field Operation and Manipulation) is open source Tool. OpenFOAM has a many of features to get answer to anything from complex liquid (or gas) moves getting mixed in trouble chemical reactions, turbulence and heat give property in law, to solid driving power and electromagnetic. Existing OpenFOAM executes serially, by introducing parallelism, time required for execution of OpenFOAM can be reduced. For introducing parallelism, System requires CUDA (Compute Unified Device Architecture) enabled CPU (Central Processing Unit). To achieve parallelism, openFOAM requires CUDA support and support is achieved by of gpu library. Library is used to introduce parallelism for single GPU and multigpu can improve performance of OpenFOAM.

Keywords – *CUDA, GPU, OpenFOAM*

*****

I. INTRODUCTION

High performance computing becomes increasingly important as advances in electronic and go mixed together with technologies have made it more widely able to be got beyond the future. The computing issues producing an effect, such as poor performance or power loss and limited design scalability in monolithic designs has high-performance processor buildings and structure design lean towards designs that point multiple processing Cores on a single bit broken out.[2] For example, scientific and engineering application domains play a key part in the future; make observations and development activities in colleges, universities, laboratories and companies, especially when the answer of greatly sized and complex problems must do with tight timing forces to limit.[12]

A Graphics Processing Unit (GPU) is a specifically de-signed electronic device, designed for processing mathematical calculations. Basically it is used for Computer Graphics to make gaming faster for graphical rendering. A Central Processing Unit (CPU) is an electronic device used for general purpose computation.[23] It used basically to do memory read write, cache read write, not for specifically for computation.

By making use of GPU computing technology giants were using this technology. The reason for the wide and mainstream acceptance is that the GPU is a computational powerhouse, and its capabilities are growing faster than those of the x86 CPU. So getting the advantage of these graphical processing we are using the GPUs of OpenFOAM.[14] General purpose graphic processor units (GPUs) technology is increasingly being used to increase in rate compute-intensive HPC applications across different disciplines in the HPC town. OpenFOAM CFD simulations can take an important amount of time and are computational getting much out. Making a comparison different those possibly taking place in addition for making able to quicker make observations and discovery using CFD are of key importance. SpeedIT libraries from Vratis make ready GPU-accelerated done again and again solvers that put in place of the done again and again solvers in OpenFOAM. The contribution of this paper is to get the supporting environments for the cuda over the OpenFOAM and introduce parallelism on the OpenFOAM over multiple cores.

II. LITERATURE SURVEY

OpenFOAM is an open starting point CFD package able to solve various fluid dynamics problems, way out of different problems from complex liquid (or gas) moves getting mixed in trouble chemical reactions, turbulence and heat, to solid driving power and electromagnetic. Because of this importance, OpenFOAM has a greatly sized user base in the area of science and physics. [11] And value exploring its power to get an effect in multi-core flat structures. OpenFOAM's core technology is a package consist of c++ unit, which can be used to make several solvers also.[22] The parallel account of OpenFOAM we can cut domain into sub domains to run on separate processors to communicate between processors by using MPI communication protocol. After running in parallel users should remake the field data to redo the complete domain and fields. [1] For this work, we select effective method having an effect equal to the input solvers, which can be used in many scientific applications. The direct logic behind making out the cuda support is to parallelize the OpenFOAM code structure for that ends for this purpose are, getting cuda support for OpenFOAM and implementing more than two or more GPU code for the same. To do first work the supporting libraries are used for having the ground for implementation. Following part has a discussion about different libraries and their good outcome[3].

A. cufflink-library

Cuda For FOAM Link (cufflink) is an open source library for connecting number methods based on Nvidia's work out joined apparatus buildings and structure design (CUDA) C/C++ listing of knowledge processing machine orders language and OpenFOAM. currently, the library puts to use the sparse having an effect equal to the input solvers of cusp and methods from force to get answer to the having an effect equal to the input tree cutting tool= b system formed from OpenFOAM's lduMatrix teaching room and come back the answer vector.[4] Cufflink is designed to put to use the course- grained parallelism of OpenFOAM (through domain break- down) to let multi-GPU parallelism allows application to have more speed up and more parallelization using the cufflink. [8] Cufflink has support over the earlier version and having single file configuration. But, the cufflink is able to exist together to the OpenFOAM-extend fork of the OpenFOAM code only. So to use it to one that makes point of comparison that is OpenFOAM latest account 2.4.x is not useful.[13]

B. PyFR

PyFR is an open-source computational liquid solver for unstructured networks. Doing a play of our putting into effect is benchmarked using clear hexahedral and mixed prismatic- tetrahedral nets of the nothing space around a going round in circles body in roller form. specifically, for each net operation is value put on at different orders of accuracy on three different hardware flat structures;[21] a NVIDIA Tesla K40c GPU, a Intel Xeon E5-2697 v2 CPU, And a AMD FirePro W9100 GPU.[17] Doing a play is then value put on a heterogeneous multi- node system made from a mix of the named before hardware. Outcomes put examples on view that PyFR gets done doing a play able to be taken about across different hardware flat structures. In particular, the power of PyFR to Target person flat structures with their low level of development language leads to importantly gave greater value to doing a play cf. marking for attack each flat structure with OpenCL by oneself. PyFR is also found to be performant on the heterogeneous multi-node system, doing an important fraction of the ready (to be used) FLOP/s. [11]

As stated above PyFR is a having methods to parallelize the code and complete work code of OpenFOAM is written in c++. The PyFR provides the cuPython based solver point which will not possibly moved to the OpenFOAM code structure and the data moves are also different in both the Case so, PyFR is not able to exist together with OpenFOAM.[5]

A. AmgX

AmgX provides a simple foot-way to increase in rate core solver technology on NVIDIA GPUs. AmgX provides up to 10x acceleration to the computationally very strong (great) having an effect equal to the input solver part of simulations, and is especially well was good, right for if true, then some other is necessarily true unstructured ways of doing. It is a high operation, state-of-the-art library and includes a flexible solver thing in place of natural one system that lets a user to easily make complex put together solvers and preconditioners. [9]

The AmgX can possible able to exist together with the OpenFOAM because the solver structure provided by AmgX is similar as the OpenFOAM it operates on Un-smoothed aggregation algebraic multigrids and Krylov methods like PCG, GMRES, BiCGStab, and flexible things changed are also able to exist together. in company with smoothers like Block-Jacobi, Gauss-Seidel, not complete Lu, more than one math part, thick Lu are gave effect to in AmgX.[18]

B. HYPRE

Hypre, a software library of high operation preconditioners and solvers for the answer of greatly sized, sparse having an effect equal to the input systems of equations on massively parallel computers. The Hypre library was made come into existence with the first end, purpose of making ready users with increased parallel preconditioners. The library features parallel multigrid solvers for both structured and unstructured network questions. For take in of use, these solvers are made way in from the application code via Hypre with the idea having an effect equal to the input system connections, which let a range of natural hard question descriptions. [10]

Hypre is a package of thing making adjustment code that lets AmgX to easily say yes data already got ready for HYPRE. This should be got facts together to a separate .a or .so text record then connected with the application to make ready the HYPRE-AmgX thing making adjustment. It is on condition that with the HYPRE LGPL 2.1 License[19], since it depends on the HYPRE.h header records. The HYPRE restructures the input of OpenFOAM which cannot be said totally able to exist together with the OpenFOAM. Also HYRRE is built with BoomerAmgx and gave another in place of with AmgX can increase the complexity of part out.[15]

C. OFGPU

OFGPU v1.1 is the latest account of our free GPL library that provides GPU (sometimes said something about to as GPGPU) having an effect equal to the input solvers for Open- FOAM v2.2.x. The library persons marked NVIDIA CUDA apparatuses on Windows, Linux, and (untested) Mac 3 Os X.[16] GPU acceleration holds the undertaking of making ready important rate of motion up at relatively low price and with low power using up made a comparison to other those possibly taking place in addition. If you need to do ones hardest effort our OFGPU library with OpenFOAM then we suggest that you use either a made with a written offering (i.e. not putting on view giving clear, full picture) high-performance NVIDIA giving clear, full picture card. [6]

OFGPU supports the single GPU putting into effect but although more than one or 2 GPU is untested. So, it does not surely work on more than one or 2 GPU. It is able to exist together with the older account of the OpenFOAM and has need of cusp and qt dependent relations separately.

D. ISPM

Library Multi-GPU able preconditioned word with root as another degree of slope (PCG) solver for Extend-project OpenFOAM. It is chiefly of an out-of-tree GPU library and an in-tree solver (the 2 GPL-licensed like OpenFOAM).[20] It is a CUDA sparse matrix-vector multiplication for a Custom matrix form and size based on cut thin, bits ELLPACK and a few connected fixed (regular) order for building a GPU done again and again having an effect equal to the input solver. [7] ISPM library is providing support for the underlining GPU shared libraries and it can be built with the cuda. So use of ISPM can support the GPU environment. It allows creating a GPU solver over general solver.

III. Conclusion

Getting rightly the parallel behavior of OpenFOAM is an overcoming and complex work needed for getting high throughput. The speed up is solely achieved by making use of CUDA derivatives. OpenFOAM is ported on multi-core flat structure on NVIDIA GPU, the GPU putting into effect make ready get done the best performances in debt to having greatly sized number of threads and Cores resulting into the faster execution of OpenFOAM.

Acknowledgment

I would like to express my gratitude to Principal Technical Officer for the useful comments, remarks and engagement through the learning process of this part of master thesis. Furthermore I would like to thank for introducing me to the topic as well for the support on the way. This research is sponsored by Centre for Development of Advanced Computing, Pune, India and the contents of this paper belong to Centre for Development of Advanced Computing, Pune, India.

References

[1] Zahra Jamshidi and Farshad Khunjush "Optimization of OpenFOAMs Linear Solvers on Emerging Multi-core Platforms" *School of Electrical and Computer Engineering Shiraz University, Shiraz, Iran*, Vol 6, No 5, 2011, p824-829

[2] Shuai Che, Jiayuan Meng "A Performance Study of General Purpose Applications on Graphics Processors" *The University of Virginia, De- partment of Computer Science* Volume 1, 5 October 2015,P1-p10

[3] Vratis "Multi-GPU simulations in OpenFOAM with SpeedIT technology" *GPU-based library of iterative solvers*. p1-16

[4] mani AlOnazi, David E. Keyes, Alexey Lastovetsky, Vladimir Rychkov "Design and Optimization of OpenFOAM-based CFD Applications for Hybrid and Heterogeneous HPC Platforms" *A. AlOnazi et al. / Computers and Fluids* 00 (2015) 112

[5] OpenFOAM "http://www.openfoam.com/documentation/user-guide/" *The Open source CFD toolbox* 27-01-2016

[6] ofgpu v1.1 symscape "http://www.symscape.com/gpu-1-1-openfoam"*GPU v1.1 Linear solver library for OpenFoam*27-01-2016

[7] Alexander Monakov "https://github.com/amonakov/ispm-sparse-lib "*ispm-sparse-lib*27-01-2016

[8] cufflink-library "https://code.google.com/p/cufflink-library/" *a library for linking numerical methods based on CUDA C/C++ with OpenFOAM*.27- 01-2016

[9] AmgX Nvidia Developer "https://developer.nvidia.com/amgx " *Nvidia Accelerated Computing*22-01-2016

[10] Hypre "http://acts.nersc.gov/hypre/" *Hypre is a library for solving large, sparse linear systems of equations*20-01-2016

[11] F.D. Witherden, B.C. Vermeire, P.E. Vincent "Heterogeneous computing on mixed unstructured grids with PyFR " *Computers and Fluids*Volume 120, 5 October 2015, Pages 173186

[12] NVIDIA CUDA " Compute Unified Device Architecture" *Programming Guide* Version 2.0 ,6/7/2008 , p1-107

[13] Open MPI: "Open source high performance computing. The Open MPI Project. [Online]" *Available: http://www.open-mpi.org/* 27-01-2016

[14] MVAPICH2: "High performance MPI over InfiniBand, 10GigE/iWARP and RoCE" OSU. [Online]. *Available: http://mvapich.cse.ohio- state.edu/*27-01-2016

[15] J. R. Edwards and M.-S. Liou, "Low-diffusion flux-splitting methods for flows at all speeds" *AIAA journal*, vol. 36, no. 9, pp. 16101617, 1998.

[16] Alexander Monakov, amonakov@ispras.ru "Efficient Multi-GPU CUDA Linear Solvers for OpenFOAM" *Institute for System Programming of Russian Academy of Sciences*March 20, 2013

[17] Liu, W., Schmidt, B., Voss, G. and Muller-Wittig, W., "Molecular Dynamics Simulations on Commodity GPUs withCUDA," *Lecture Notes in Computer Science, High Performance Computing – HiPC 2007*, Vol. 4873, Springer, New York, 2007, pp.185-196.

[18] Ufimtsev, I. and Martinez, T., "Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-electron Integral Evaluation," *Journal of Chemical Theory and Computation*, Vol. 4, No. 2, 2008, pp. 222-231.

[19] Schatz, M. C. and Trapnell, C. Delcher, A. L. and Varshney, A., "High-throughput Sequence Alignment using GraphicsProcessing Units," *BMC Bioinformatics*, BioMed Central, 2007.

[20] Barrachina, S., Castillo, M., Igual, F. D., Mayo, R. and Quintana-Orti, E. S., "Solving Dense Linear Systems on Graphics Processors," Technical Report ICC 02-02-2008, Universidad Jaume I, Depto. de Ingenieria y Ciencia de Computadores, February 2008.

[21] Castillo, M., Chan, E., Igual, F. D., Mayo, R., Quintana-Orti, E.S., Quintana-Orti, G., van de Geijn, R. and Van Zee, F.G."Making Programming Synonymous with Programming for Linear Algebra Libraries", Technical Report, University of Texas at Austin, Department of Computer Science, Vol. 31, April 17, 2008, pp. 8-20

[22] Michalakes, J. and Vachharajani, M., "GPU Acceleration of Numerical Weather Prediction," *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing*, IEEE Computer Society, Washington, DC, 2008.

[23] Bleiweiss, A., "GPU Accelerated Pathfinding," *Proceedings of the 23rd ACM SIGGRAPH/Eurographics Symposium on Graphics Hardware*, Eurographics Association, Aire-la-Ville, Switzerland, 2008, pp. 65-74.

vi Working at Centre for Development of Advanced Graduated in B.Tech Information Technology from Dr. mbedkar Technological University, Lonere. Currently r in from Pune Institute of Computer Technology.