Evaluating the Maturity of OpenFOAM Simulations on GPGPU for Bio-fluid Applications

Ahmet Duran
Department of Mathematics
Istanbul Technical University (ITU)
Istanbul, Turkey
e-mail: aduran@itu.edu.tr

Senol Piskin
Department of Mechanical Engineering
Koc University
Istanbul, Turkey

Mehmet Tuncel
Department of Mathematics, ITU and
Informatics Institute, ITU
Istanbul, Turkey

Abstract

It is important to deal with the computational challenges for bio-medical fluid flow simulations and an *OpenFOAM* solver, *icoFoam*, for the large matrices coming from the simulation of blood flow in arteries on different HPC clusters. The flow problem produced various matrices as the time advances in simulation. In this study we examined the behaviour of the solvers for ill-conditioned matrices. We compared the CPU performance of the iterative solver *icoFoam* and the hybrid parallel codes (MPI+OpenMP) of a direct solver *SuperLU_DIST 4.0* (see [2]) at TGCC Curie (a Tier-0 system) thin nodes at CEA, France (see [5]). Moreover, we compared the performance of the hybrid parallel codes of MPI+OpenMP+CUDA versus MPI+OpenMP implementation of *SuperLU_DIST 4.0* at TGCC Curie (a Tier-0 system) hybrid nodes of CPU + GPU at CEA, France (see [5]). We discuss the performance, scalability and robustness of *OpenFOAM* on GPGPU cluster. We present our results regarding the speed-up of the solvers for the large matrices of size up to 20 million x 20 million. The authors thank to PRACE, GENCI and CEA for the opportunity to conduct our research in the frame of the Project 2010PA2505 awarded under the 18th Call for PRACE Preparatory Access.

I. Introduction

We investigated the challenges facing CFD solvers as applied to bio-medical fluid flow simulations and in particular the *OpenFOAM 2.1.1* solver, icoFoam, for the large penta-diagonal matrices coming from the simulation of blood flow in arteries with a structured mesh domain in PRACE-3IP project at TGCC Curie (a modern Tier-0 system) (see [1] and references therein). We generated a structured mesh by using blockMesh as a mesh generator tool. To decompose the generated mesh, we employed the decomposePar tool. After the decomposition, we used icoFoam as a flow simulator/solver tool. We achieved scaled speed-up for large matrices up to 64 million x 64 million matrices and speed-up up to 16384 cores on Curie thin nodes.

In this paper, we examined *OpenFOAM 2.2.2* "*icoFoam*" simulator with an iterative solver such as diagonal incomplete LU preconditioned bi-conjugate gradient in addition to direct solvers such as distributed *SuperLU 4.0* (see [2]). The flow problem produced various matrices as the time advances in simulation. The solution of the matrices obtained after each time step can be more challenging due to the changing structure of the matrices. This change may be caused by mess change or flow variable change. Generally the solution time of the matrices increases as the time advances in simulation.

It is challenging to discuss on the benefits or drawbacks of hybrid nodes. There are tradeoffs using GPU accelerators especially for the software packages or applications where it is not possible to fit the whole part into GPU. While it is expected to obtain a reduced time due to the accelerator, there would be communication over-head between the various processors and the GPU accelerators, as well. Therefore, it is important to obtain a feasible/optimal proportion of the tasks to MPI, OpenMP, and CUDA/OpenCL usages in emerging CPU+GPU systems. For example, it is not possible to do everything only in GPU for a complex algorithm like *SuperLU_DIST*. Therefore hybrid nodes like Curie hybrid nodes at CEA in France provide opportunity.

It would be interesting to discuss about the relative energy requirements for thin nodes versus hybrid nodes. A diversification of hardware solutions based on the application capability may be needed in order to attain a good efficiency (see [6] and [7]). While the compute partition of Curie thin nodes having total of 80,640 cores consumes 2132 kW, the partition of Curie hybrid nodes having total of 288 Intel® + 288 Nvidia processors uses 108.80 kW as the total power (see TOP500 Supercomputing sites [8] and the Green500 List [9]). The partition of Curie hybrid nodes outperforms the Curie thin nodes when the energy efficiency is compared in terms of performance per watt and the rates of computation are 1,010.11 MFLOPS/W and 637.43 MFLOPS/W, respectively.

The remainder of this work is organised as follows: In Section 2, the test environment and the flow of approach are described. In Section 3, thin nodes results of the CPU performance for the iterative solver *icoFoam* and the hybrid parallel codes (MPI+OpenMP) of a direct solver SuperLU_DIST 4.0 are compared. Moreover, simulation test results of hybrid node using MPI+OpenMP+CUDA versus MPI+OpenMP with *SuperLU_DIST 4.0* solvers are presented. Section 4 concludes this work.

II. Test Environment and Flow of Approach

OpenFOAM (see [10]) is an open source Computational Fluid Dynamics (CFD) toolbox. It is a software package with many tools for several main tasks of the simulation such as pre-processing (meshing), decomposition and solution. Here, the solver refers to not only linear system solver but also Navier Stokes solver and simulator.

The first four matrices in Table 1 are obtained at time 0.00005 (s) of the simulation where the time step size is 0.00005 (s), as in [1]. Unlike [1], the last six matrices in Table 1 are encountered at the third time step, at time 0.012 (s) of the simulation where the time step size is 0.004 (s). This is a relatively large time step size for such a very small mesh size. Thus, we obtained challenging ill-conditioned matrices. Almost 5 or 7 banded sparse matrix occurs at each time step and the matrices are described in Table 1.

The flowchart in Figure 1 shows the flow of approach in the paper.

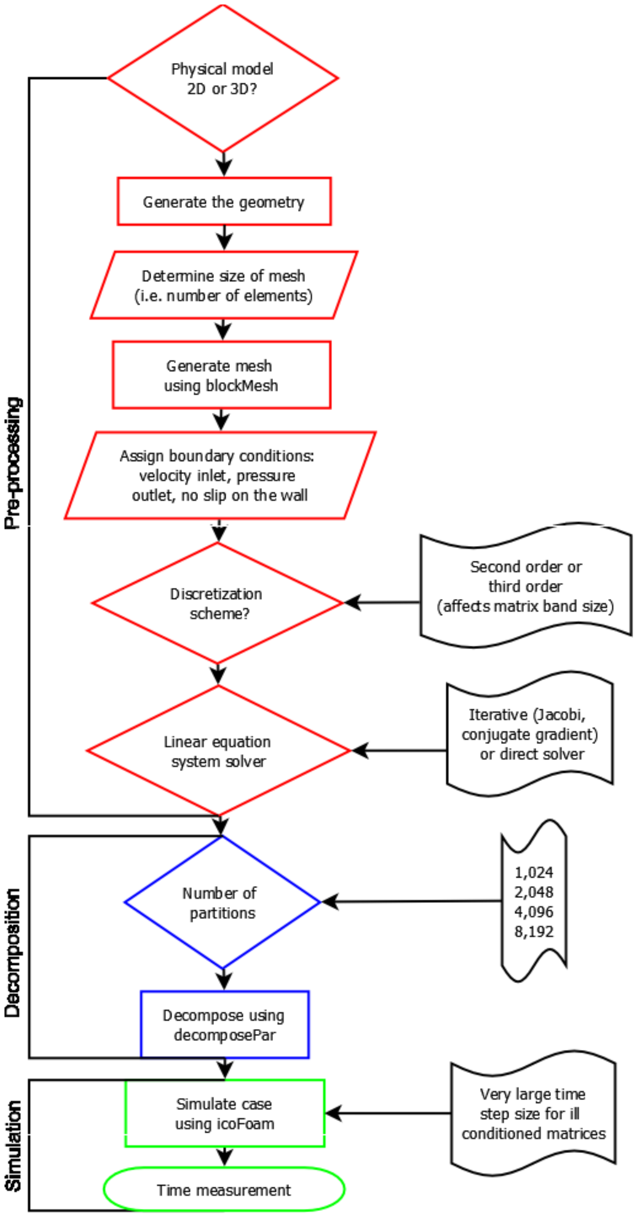
Physical model 2D or 3D?
Generate the geometry
Determine size of mesh (i.e. number of elements)
Generate mesh using blockMesh
Assign boundary conditions: velocity inlet, pressure outlet, no slip on the wall
Discretization scheme?
Second order or third order (affects matrix band size)
Linear equation system solver
Iterative (Jacobi, conjugate gradient) or direct solver
Number of partitions
1,024
2,048
4,096
8,192
Decompose using decomposePar
Simulate case using icoFoam
Very large time step size for ill conditioned matrices
Time measurement
Pre-processing
Decomposition
Simulation

Figure 1. Flowchart for the flow of the approach including the main tasks

III. Test Results

The tests were done for only a few time steps due to time limitations, while the real case runs are conducted for more than thousands of time steps. No single CPU solution was possible because of long waiting times, so, information regarding the pre-processing (meshing), partitioning etc. are given for parallel processing. The most time consuming part of the simulation was the decomposing of the mesh. For 8192 partitions, it took over 3 hours. The “Simple” decomposition method was preferred since the running cases were for a structured mesh. This technique simply splits the geometry into pieces by direction, such as 32 pieces in x direction and 32 pieces in y direction. Since the mesh is structured, mC_20M_n matrix means 20M of cells in the fluid domain.

TABLE I. Description of Matrices

	N	NNZ	NNZ/N	Origin
mC_8M	8,000,000	39,988,000	4.999	ITU Mathematics
mC_16M	16,000,000	79,984,000	4.999	ITU Mathematics
mC_6M_D	6,000,000	41,800,000	6.967	ITU Mathematics
mC_8M_D	8,000,000	55,760,000	6.970	ITU Mathematics
mC_8M_n	8,000,000	39,988,000	4.999	ITU Mathematics
mC_16M_n	16,000,000	79,984,000	4.999	ITU Mathematics
mC_20M_n	20,000,000	99,982,000	4.999	ITU Mathematics
mC_6M_n_D	6,000,000	41,780,000	6.963	ITU Mathematics
mC_8M_n_D	8,000,000	55,760,000	6.970	ITU Mathematics
mC_10M_n_D	10,000,000	69,660,000	6.966	ITU Mathematics

A. Thin Node Results

We compared the CPU performance of the iterative solver *icoFoam* and the hybrid parallel codes (MPI+OpenMP) of a direct solver SuperLU_DIST 4.0 (see [2]) at TGCC Curie (a Tier-0 system) thin nodes at CEA, France (see [5]). Fig. 2 and Fig. 3 show the wall-clock time comparisons of the solvers, excluding the refinement time, for mC_16M_n and mC_20M_n on Curie thin nodes, respectively. The iterative solver with a diagonal incomplete LU preconditioned bi-conjugate gradient outperforms the direct solver *SuperLU_DIST 4.0* for the simulation matrices.

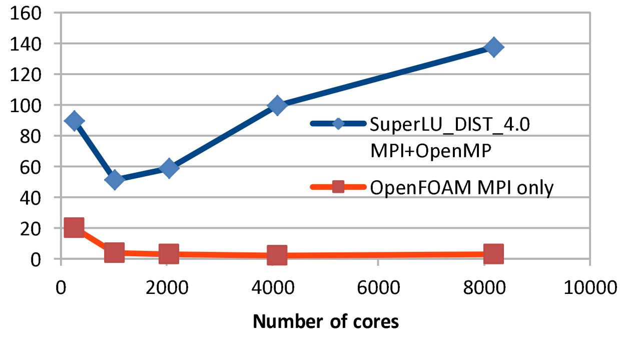

Figure 2. Wall-clock time comparison of the solvers for mC_16M_n on Curie thin nodes

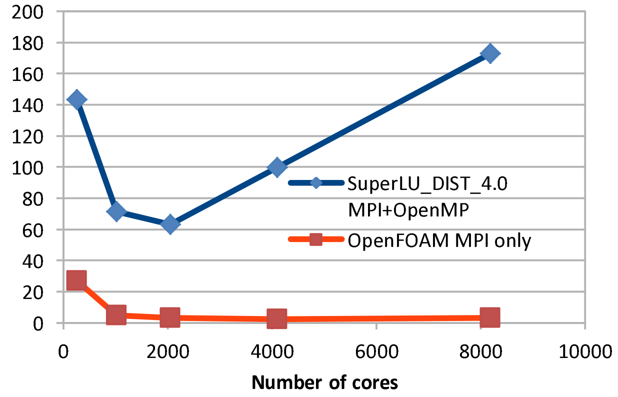

Figure 3. Wall-clock time comparison of the solvers for mC_20M_n on Curie thin nodes

B. Hybrid Node Results Using MPI+OpenMP+CUDA

TABLE II. THE CONFIGURATION OF MPI+OPENMP AND MPI+OPENMP+CUDA FOR THE DIRECT SOLVER

Testbed:CURIE/	hybrid	hybrid	hybrid	hybrid
SuperLU_DIST version	4	4	4	4
# of cores	64	256	512	1024
# of processes	16	64	128	256
# of threads per process	4	4	4	4
# of GPUs per process	1	1	1	1

We compared the performance of the hybrid parallel codes of MPI+OpenMP+CUDA (see [4]) versus MPI+OpenMP implementation of SuperLU_DIST 4.0 at TGCC Curie (a Tier-0 system) hybrid nodes of CPU + GPU at CEA, France (see [5]). Table 2 describes the corresponding configurations while we run the direct solver.

Table 3 shows the performance results for the ten simulation matrices described in Table 1. For example, Fig. 4 shows the comparison for the performances of MPI+OpenMP+CUDA and MPI+OpenMP implementations for mC_20M_n on Curie hybrid nodes. In Fig. 5, we observe a linear speed-up of the direct solver up to 512 cores for both implementations for mC_20M_n on Curie hybrid nodes.

Generally, we see that MPI+OpenMP implementation outperforms the hybrid of MPI+OpenMP+CUDA for this set of simulation matrices when we consider the wall clock times for the optimal number of cores because of several overheads coming from CUDA implementation for the direct solver algorithm. It is not possible to put everything only in GPU for *SuperLU_DIST*. Therefore, the tasks should be proportioned to MPI, OpenMP, and CUDA/OpenCL. In *SuperLU_DIST 4.0* (see [4]), cuBLAS library execution is one of the most time consuming tasks performed in GPU in order to gain from explicit parallelization. On the other hand, there are overheads such as data transfer on PCIe between host and device memory (CPU and GPU) and new data structure changes related to data packing and scattering. Moreover, *SuperLU* is a complex algorithm and it is challenging to select the right combination for better intra-node communications and inter-node communications within CPU+GPU heterogeneous systems, under current technology limitations (see [3]).

The last eight matrices in Table 3 are challenging large matrices because they are relatively denser or ill-conditioned. The error labelled Error 1 occurs for small number of cores. We meet with an error message labelled Error 2 related to buffer size during the factorization subroutine pdgstrf, for the hepta-diagonal matrices. Error 3 is a CUDA stream error related to setting cuBLAS library execution stream.

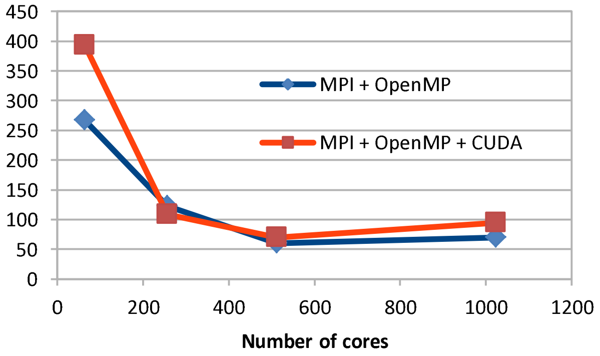

Figure 4. Wall-clock time of direct solver for mC_20M_n on Curie hybrid nodes

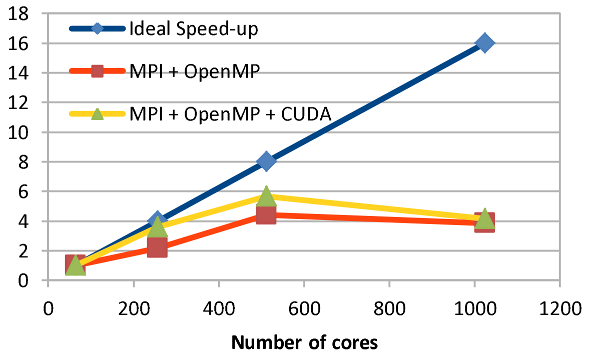

Figure 5. Speed-up of direct solver for mC_20M_n on Curie hybrid nodes

TABLE III. WALL CLOCK TIMES (S) OF SUPERLU_DIST 4.0 FOR THE LARGE PENTA-DIAGONAL MATRICES FOR 2D PROBLEMS AND HEPTA-DIAGONAL MATRICES FOR 3D PROBLEMS, DESCRIBED IN TABLE I, ON MPI+OPENMP VERSUS MPI+OPENMP+CUDA IMPLEMENTATIONS

Matrices / Number of cores		64	256	512	1024
mC_8M	MPI + OpenMP	99.96	34.70	28.78	37.89
	MPI + OpenMP + CUDA	94.70	39.10	43.70	60.72
mC_16M	MPI + OpenMP	230.30	83.19	47.73	59.02
	MPI + OpenMP + CUDA	236.83	87.23	60.00	81.41
mC_6M_D	MPI + OpenMP	Error 1	260.38	296.74	239.52
	MPI + OpenMP + CUDA	Error 1	Error 2	254.44	257.15
mC_8M_D	MPI + OpenMP	Error 1	1005.96	516.86	387.20
	MPI + OpenMP + CUDA	Error 1	680.25	Error 2	353.40
mC_8M_n	MPI + OpenMP	94.70	31.00	32.79	35.83
	MPI + OpenMP + CUDA	70.94	38.27	Error 3	61.34
mC_16M_n	MPI + OpenMP	181.53	75.93	49.53	58.61
	MPI + OpenMP + CUDA	233.22	75.58	61.42	83.61
mC_20M_n	MPI + OpenMP	266.82	122.59	60.30	69.49
	MPI + OpenMP + CUDA	393.49	108.90	69.60	94.99
mC_6M_n_D	MPI + OpenMP	1178.51	409.15	248.84	211.70
	MPI + OpenMP + CUDA	782.22	294.14	Error 2	222.04
mC_8M_n_D	MPI + OpenMP	Error 1	948.03	533.78	386.72
	MPI + OpenMP + CUDA	Error 1	682.02	Error 2	349.16
mC_10M_n_D	MPI + OpenMP	Error 1	877.92	465.60	373.09
	MPI + OpenMP + CUDA	Error 1	752.78	Error 2	Error 3

IV. CONCLUSION

We performed bio-medical fluid flow simulations for the large matrices coming from the simulation of blood flow in arteries in emerging CPU+GPU systems. The flow problem produced various challenging matrices during the simulation. We compared the CPU performance of the iterative solver icoFoam and the hybrid parallel codes (MPI+OpenMP) of a direct solver SuperLU_DIST 4.0 (see [2]) at TGCC Curie (a Tier-0 system) thin nodes at CEA, France (see [5]). We observe that the iterative solver with a diagonal incomplete LU preconditioned bi-conjugate gradient outperforms the direct solver *SuperLU_DIST 4.0* for the simulation matrices. Moreover, we compared the performance of the hybrid parallel codes of MPI+OpenMP+CUDA versus MPI+OpenMP implementation of SuperLU_DIST 4.0 at TGCC Curie (a Tier-0 system) hybrid nodes of CPU + GPU at CEA, France (see [5]). Generally, we notice that MPI+OpenMP implementation outperforms the hybrid of MPI+OpenMP+CUDA for the set of simulation matrices when we consider the wall clock times for the optimal number of cores because of several overheads coming from CUDA implementation for the complex direct solver algorithm. Furthermore, we met with several errors for the challenging simulation matrices. We believe that the technology developments in emerging CPU+GPU systems will increase the scalability of related complex algorithms by eliminating the bottlenecks coming from communication and right matching of system components required for special applications.

ACKNOWLEDGEMENT

This research was supported by the Project 2010PA2505 awarded under the 18th Call for PRACE Preparatory Access and we acknowledge that the results of this research have been achieved using the PRACE Research Infrastructure resource TGCC Curie (a modern Tier-0 system) based at CEA in France.

REFERENCES

[1] A. Duran, M.S. Celebi, S. Piskin, and M. Tuncel, “Scalability of OpenFOAM for bio-medical flow simulations,” Journal of Supercomputing, 71(3), 2015, pp. 938-951.

[2] X. S. Li, J. W. Demmel, J. R. Gilbert, L. Grigori, M. Shao, and I. Yamazaki, SuperLU Users' Guide, Tech. Report UCB, Computer Science Division, University of California, Berkeley, CA, 1999, update: 2011

[3] M.S. Celebi, A. Duran, M. Tuncel and B. Akaydın, Scalable and improved SuperLU on GPU for heterogeneous systems, *PRACE (Partnership for Advanced Computing in Europe),* PRACE PN: *283493, PRACE-2IP white paper, Libraries,* WP 44, July 13, 2012.

[4] P. Sao, R. Vuduc, and X.S. Li, “A distributed CPU-GPU sparse direct solver,” Euro-Par 2014 Parallel Processing, Lecture Notes in Computer Science vol. 8632, 2014, pp. 487-498.

[5] http://www-hpc.cea.fr/en/complexe/tgcc-curie.htm.

[6] N. Meyer, M. Lawenda, et al., Best Practices for HPC Procurement and Infrastructure, PRACE-2IP project, under Grant agreement No. RI-283493, Aug. 2013.

[7] J. David, JN Richet, E. Boyer, N. Anastopoulos, G Collet, GC Verdiere, et al., Best Practice Guide - Curie v1.17, PRACE, Nov. 2013.

[8] TOP500 Supercomputing sites, http://top500.org/

[9] The Green500 List, http://www.green500.org

[10] OpenFOAM main site. http://www.openfoam.com

Proceedings of the

EMerging Technology (EMiT) Conference 2016

EMiT
Emerging Technology Conference
20
16
MANCHESTER 1824
The University of Manchester
BSC
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

2-3 June 2016
Barcelona Supercomputing Center, Spain

Edited by B.D. Rogers, D. Topping, F. Mantovani, M.K. Bane

http://emit.tech

June 2016

Foreword to the 2016 Emerging Technology (EMiT) Conference

Dear Delegate,

The Emerging Tech conference, EMiT, is now firmly established. In its third year we are delighted to be hosted by Barcelona Supercomputing Center and the Mont-Blanc project. BSC has long been held in very high regard, both for its innovative HPC and also its renowned work on advancing new tools for performance analysis and programming models. The presence of the Mont-Blanc project on the EMiT Organising Committee emphasises the growing importance of emerging technologies & techniques to improve the cost and energy efficiency of next generation HPC platforms.

This third EMiT conference follows the philosophy of those preceding by seeking out the challenges of hardware, software, tools and algorithms that we are expecting from over the horizon or are helping create ourselves.

The Organising Committee has overcome many challenges to bring EMiT2016 to fruition. Please join me in extending your appreciation to each of them.

I would also like to thank each keynote, everybody submitting papers (accepted or not), and each sponsor and stall, for showing support to the EMiT series.

We are looking towards EMiT 2017. If you are inspired by this year's conference to host or join the Organising Committee next year then please speak to us.

Michael Bane
Chair, EMiT2016 Organising Committee
http://highendcompute.co.uk
@highendcompute

Committee

Organising Committee

Dr Michael K. Bane	High End Compute
Dr Filippo Mantovani	Barcelona Supercomputing Center
Renata Gimenez	Barcelona Supercomputing Center
Dr David Topping	University of Manchester, School of Earth, Atmospheric & Environmental Sciences
Dr Stephen Longshaw	Science & Technology Facilities Council, STFC
Dr Benedict D. Rogers	University of Manchester, School of Mechanical, Aerospace & Civil Engineering (MACE)
Irfan Alibay	University of Manchester, School of Pharmacy
Ignas Daugalas	University of Manchester, IT Services
Shih-Chen Chao	University of Manchester, IT Services
Prof. Dave Emerson	Science & Technology Facilities Council, STFC

Printing

"The Organising Committee of the 2016 EMiT Conference" shall not be held responsible for any statement or opinion advanced in papers or printed in this volume. The authors' papers have been prepared for final reproduction and printing without any reduction, correction, etc. The authors are fully responsible for all the information contained in their papers.

Copyright © Published by The Emerging Technology (EMiT) Conference
Manchester, U.K.
ISBN 978-0-9933426-3-9

CONTENTS

DAY 1: THURSDAY, JUNE 1ST, 2016 6