Performance Optimization of OpenFOAM® for Clusters of Intel® Xeon Phi™ Coprocessors

Nishant Agrawal¹, Ambuj Pandey¹, Ravi Ojha¹, Rihab Abdulrazak¹, Gopal Shinde², Kishor Nikam², Michael Klemm³, Paul Edwards⁴

¹ Tata Consultancy Services Ltd, Gateway Park, MIDC Road No#13, Andheri East, Mumbai, Maharashtra 400093, India
{nishant.agrawal, ambuj.pandey, rihab.abdulrazak, ravi.ojha}@tcs.com
² Tata Consultancy Services Ltd, Sahyadri Park, Plot No. 2 & 3, Rajiv Gandhi Infotech Park, Phase-III, Hinjewadi, Pune, Maharashtra 411057, India
{kishor.nikam, shinde.gopal}@tcs.com
³ Intel GmbH, Dornacher Str. 1, 85622 Feldkirchen, Germany
michael.klemm@intel.com
⁴ Intel Corporation UK Ltd, 650 Wharfedale Road, Winnersh Triangle, RG41 5TP, UK
paul.m.edwards@intel.com

OpenFOAM® is a well-known and popular software package for solving partial differential equations (PDE) and is used by industry, researchers and academia to solve variety of computational fluid dynamics (CFD) problems. It comes with several different kinds of solvers for different PDEs, but also ships with a framework to implement third-party solvers for custom PDEs. OpenFOAM uses a finite volume method (FVM) and employs the Message Passing Interface (MPI) for communication; as of today it does not yet support multi-threading.

Recent advances in parallel computing have enabled solving time-consuming problems in a reasonable time frame or to tackle larger problems, that is, more detailed analysis of existing structures or simulation of more complex and larger structures. Technologies include multicore CPUs with few but powerful individual cores that strive to balance single-threaded and multi-threaded performance, and GPUs that have a higher number of cores although less powerful and designed to work in lock-step with all other cores executing the same instruction on different elements of the data sets.

A more recent hardware introduced by Intel is the Intel® Xeon Phi™ coprocessor based on the Intel® Many Integrated Core (MIC) architecture. The coprocessor is a multicore device and has more cores than the traditional multi-core Intel® Xeon processor albeit operating at lower frequency and being less powerful compared to the Xeon counterpart. The Intel Xeon Phi coprocessor offers the convenience of porting an application that already runs on an Intel Xeon processor, because it supports well-known (parallel) programming models. To get good performance one needs to consider various optimization techniques to improve performance and to tune the code towards the specific requirements of the coprocessor’s cores. As the problem size to be solved increases, it becomes necessary to use clusters with multiple nodes that are equipped with both Intel Xeon processors and Intel Xeon Phi coprocessors.
This paper describes the work done for optimizing the performance of the OpenFOAM for both Intel architectures, but with focus on the coprocessor. Our objective is to optimize its performance with good scalability across nodes in a High Performance Computing (HPC) cluster. The tuning techniques applied include improvements to vectorize the solver code and software prefetching to reduce memory latency. A key contribution is a novel decomposition method that takes the heterogeneous hardware layout into account and that strives to reduce communication between the different cluster nodes and their respective coprocessor cards.

The optimization techniques that have been applied on the coprocessor improve the performance by a factor of three in the native mode. We also propose a new decomposition algorithm which handles load imbalance and minimizes the communication between the processors and coprocessors. Our work demonstrates that OpenFOAM can run on heterogeneous clusters with good scalability of the Intel® Xeon® E5-2697v3 processor and Intel® Xeon Phi™ 7120P coprocessors. The benchmarks show up to a 1.44 times better performance of the heterogeneous cluster with 2400 hybrid cores over the Xeon-only cluster.

Below graph shows per-iteration time comparison between Xeon only(448 cores) vs Hybrid(2368 cores) cluster on 16 nodes for the 240-million car case.