ParCFD’2019
31st International Conference on Parallel Computational Fluid Dynamics
May 14-17 2019, Antalya TURKEY

ACHIEVING LOAD BALANCE ON CPU + MIC HETEROGENEOUS PLATFORM FOR A COMBUSTION SIMULATION APPLICATION

YONGGANG CHE*, CHUANFU XU† AND ZHENGHUA WANG*

* Institute for Quantum Information & State Key Laboratory of High Performance Computing,
College of Computer, National University of Defense Technology
Changsha 410073, China
Email: {ygche, xuchuanfu,zhhwang}@nudt.edu.cn, web page: http://www.nudt.edu.cn

Key words: Combustion simulation, heterogeneous architectures, offloading, load-balance.

Summary. This paper presents our method to achieve load-balance of a combustion simulation application, on the CPU + MIC (Many Integrated Core) heterogeneous platform. For a test case of the turbulent combustion in a cavity-based supersonic combustor with a mesh of 5.65 million cells, the heterogeneous configuration with two KNC coprocessors is maximally 2.59 times faster over two Haswell 12-core CPUs.

1 INTRODUCTION

LESAP (Large Eddy Simulation for Air-breathing Propulsion) is a real-world scramjet combustion simulation application [1,2]. The detailed physical models and numerical methods, including the turbulence models, the numerical schemes, the turbulent inflow conditions and the mass-tracing procedure, were presented in[1]. We port this application to the CPU + MIC (Many Integrated Core) heterogeneous platform (e.g., the Tianhe-2 Supercomputer [3]) using the OpenMP 4.5 offload programming model [4]. Making use of all CPU and MIC cores and achieving load balancing between CPU and MIC coprocessors are crucial for good performance. This paper addresses the two issues and performance evaluation shows good results.

2 METHODOLOGY

To partition the workload between the CPUs and the MIC coprocessors, we design an asymmetric task partition method. There are two types of processes. The offload processes ($P_{offload}$) execute the sequential code segments, control the CPU-MIC data transfer and inter-process communications, and offloads highly parallel code segments to the coprocessor. The compute processes ($P_{compute}$) perform computations by themselves and do not offload tasks to the MIC coprocessors. As most CPU cores are assigned to the compute processes, they can perform their computation in total parallel with the MIC coprocessors.

We allocate the workload to the MPI processes based on the grid block, with each process operates on one grid block. However, the computing power of the CPU and the MIC coprocessor is different. To address this issue, we design a load balancing scheme, which takes into account of the number of MIC coprocessors on one node, the number of total grid blocks, and the number of offloads on each MIC coprocessors. Each compute process utilizes several CPU cores by OpenMP threads. Typically, most CPU cores are used by the compute processes to perform computations. Each offload process can offload task to only one MIC coprocessor. However, multiple offload processes are allowed to offload tasks to a same MIC coprocessor.

For a given platform configuration, the model's hardware parameters are fixed. Whereas, the number of grid block on one node, the number of grid blocks assigned to each MIC coprocessor, and the number of OpenMP threads invoked in each offload are adjustable. The key to achieve load balance is to determine the values of these parameters. We empirically search for the parameters that achieve the highest performance by performing a small number of time-step iterations for different parameter combinations. Then we set the optimal values of these parameters in the production run.

3 RESULTS

Performance evaluation is carried out on a server node consists of two Intel Xeon E5-2692 v3 (Haswell) 12-core CPUs and two Intel KNC coprocessor (one is Intel Xeon Phi 5110P, another is Intel Xeon Phi 7120P). The test case is the turbulent combustion in a cavity-based supersonic combustor. The mesh contains about 5.65 million cells. Figure 1 shows the average runtime per time-step in seconds for varies running configurations. Figure 2 shows the maximum speedup achieved when different computing devices are used. The heterogeneous configuration, with the additional two KNC coprocessors, achieves a maximum speedup of 2.59 times over two Haswell CPUs.

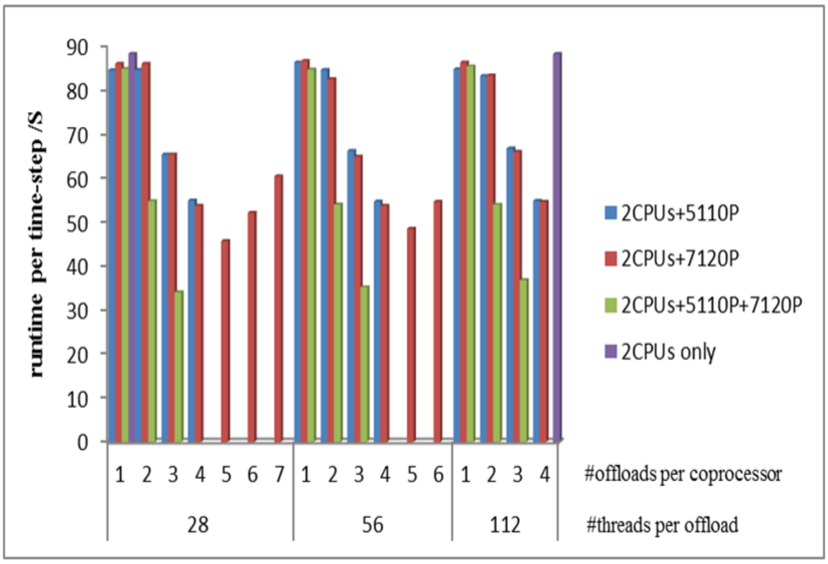

Figure 1. Average runtime per time-step in seconds

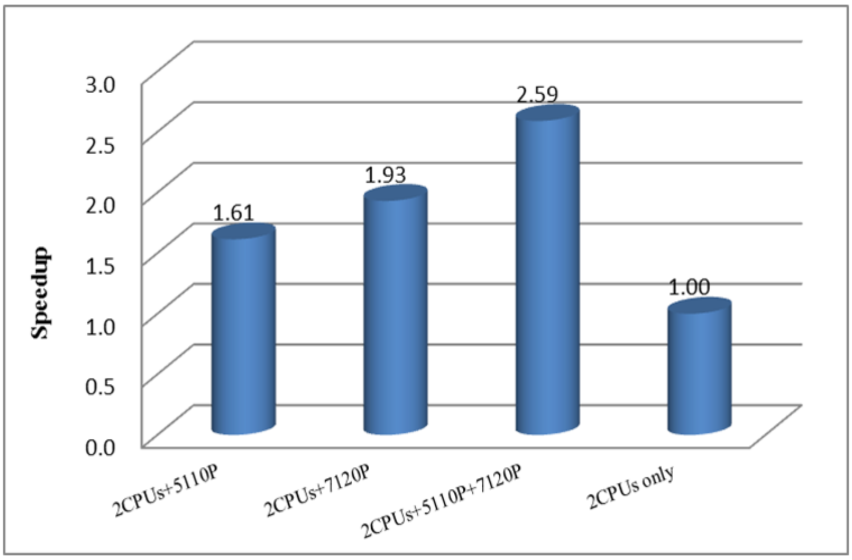

Figure 2. Maximum speedup achieved

4 CONCLUSIONS

We have optimized the load-balance of a combustion simulation application with a model and experiment combined method on heterogeneous platform. Good performance is achieved when using heterogeneous hardware, as compared to the pure CPU configuration.

Acknowledgements. This work was supported by the National Key Research and Development Program of China under Grant No. 2017YFB0202403, the Joint NSFC-ISF Research Program under Grant No. 61561146395 and the NSFC project under grant Nos. 61772542 and 11502296.

REFERENCES

[1] Hongbo Wang, Zhenguo Wang, Mingbo Sun and Ning Qin. Large eddy simulation based studies of jet-cavity interactions in a supersonic flow. Acta Astronautica, 93, 182-192 (2014).
[2] Yixin Yang, Hongbo Wang, Mingbo Sun, Zhenguo Wang, Yanan Wang. Numerical investigation of transverse jet in supersonic crossflow using a high-order nonlinear filter scheme. Acta Astronautica 154, 74-81 (2019).
[3] Xiangke Liao, Liquan Xiao, Canqun Yang, Yutong Lu. MilkyWay-2 supercomputer: System and application. Frontiers of Computer Science, 8(3), 345-356 (2014).
[4] OpenMP Architecture Review Board. OpenMP Application Programming Interface, Version 4.5 November 2015.