6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11-15 June 2018, Glasgow, UK

Energy consumption of algorithms for solving the compressible Navier-Stokes equations on CPUs, GPUs and KNLs

Satya P. Jammy[1], Christian T. Jacobs[2], David J. Lusher[3] and Neil D. Sandham[4]

[1] University of Southampton, SO17 1BJ, United Kingdom. S.P.Jammy@soton.ac.uk
[2] University of Southampton, SO17 1BJ, United Kingdom. C.T.Jacobs@soton.ac.uk
[3] University of Southampton, SO17 1BJ, United Kingdom. D.Lusher@soton.ac.uk
[4] University of Southampton, SO17 1BJ, United Kingdom. N.Sandham@soton.ac.uk

Keywords: *Energy efficiency, Finite difference methods, Graphics Processing Unit (GPU), Knights Landing (KNL)*

In addition to hardware wall-time restrictions commonly seen in high-performance computing systems, it is likely that future systems will also be constrained by energy budgets. In the present work, six finite difference algorithms to solve the compressible Navier-Stokes equations with varying computational and memory intensities are evaluated with respect to both energy efficiency and runtime on an Intel Ivy Bridge CPU node, an Intel Xeon Phi Knights Landing processor, and an NVIDIA Tesla K40c GPU using the OpenSBLI framework[1]. The conventional way of storing the discretised derivatives to global arrays for solution advancement [2] is found to be inefficient in terms of energy consumption and runtime for the current set of equations. In contrast, a class of algorithms in which the discretised derivatives are evaluated on-the-fly (yielding high compute intensity) or stored as thread-/process-local variables (replacing global arrays with local variables) is optimal both with respect to energy consumption and runtime.

On all three hardware architectures considered, a speed-up of ~ 2 and an energy saving of ~ 2 are observed for the high compute intensive algorithms compared to the memory intensive algorithm on that architecture. The energy consumption is found to be proportional to runtime, irrespective of the power consumed and the GPU has an energy saving of ~ 5 compared to the same algorithm on a CPU node. For optimising the algorithms on GPU care should be taken to improve data locality.

REFERENCES

[1] Christian T. Jacobs, Satya P. Jammy, and Neil D. Sandham. OpenSBLI: A framework for the automated derivation and parallel execution of finite difference solvers on a range of computer architectures. Journal of Computational Science, 18:12 23, 2017.

[2] Satya P. Jammy, Christian T. Jacobs, and Neil D. Sandham. (In Press) Performance evaluation of explicit finite difference algorithms with varying amounts of computational and memory intensity. Journal of Computational Science DOI:10.1016/j.jocs.2016.10.015