Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-218-RC1, 2018

Geoscientific Model Development Discussions

Interactive comment on "HOMMEXX 1.0: A Performance Portable Atmospheric Dynamical Core for the Energy Exascale Earth System Model" *by* Luca Bertagna et al.

Anonymous Referee #1

Received and published: 23 November 2018

Summary The authors shared their experiences and results in porting the dynamical core model (HOMME) of an Earth System Model (E3SM) from optimized Fortran implementation to C++ with the Kokkos library, with the aim to achieve performance portability on conventional CPUs, KNL and GPUs. The experimental results show that the new implementation (HOMMEXX) matches the performance and strong-scaling characteristics of the Fortran code on Haswell CPU cluster, and improves the performance on KNL. The abstraction of parallel loops in Kokkos opens the possibility of porting to GPUs, where the results show better performance on V100 compared with a Haswell node.

Comments - I think the introduction and problem description is clear for someone from other fields in numerical methods to follow without too much difficulty. For completeness, the authors could consider adding mathematical equations for the differential operators. - I feel Section 3.3 needs some improvement. I found the part describing how paralle_for loops map to different execution policies slightly unclear. I suggest describing the kernel with pseudo-code of nested loops, decorated with execution policy choice. - Have the authors verified that vectorization on CPU is effective, potentially by looking at the generated assembly code? - I'm curious that if the authors encountered any limitations for the vector data types, e.g. for maths function calls, conditionals etc. - Could the authors elaborate more on "reuse of subviews is important to minimize index arithmetic" on CPU (page 11 line 9)? I don't quite follow what is "... the number of connections per elements..." (page 11 line 12). - In Section 4, HOMME and HOMMEXX have very similar performance on Haswell, but using different (if I understand correctly) strategy, could the authors explore a bit more on the reason behind it? - Subtle point: is Turbo-Boost a potential source of randomness in the experiments? - One thing I feel the paper is missing is that we do not know if the achieved performance is "good enough". The paper could be improved (by a lot) by e.g. showing the percentage of peak performance achieved and/or roofline model of the hardware. This is especially helpful because the experiments are carried on a large range of hardware with very different characteristics, and finding some common metrics to compare and contrast between them would help the authors in organizing the presentation of experimental results.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-218, 2018.