Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-186-RC1, 2017

Geoscientific Model Development Discussions

Interactive comment on "Crossing the Chasm: How to develop weather and climate models for next generation computers?" *by* Bryan N. Lawrence et al.

Anonymous Referee #1

Received and published: 6 October 2017

This paper provide a good (referenced) summary of the pressures that are growing the complexity of Earth system models and consequently limiting our ability to exploit emerging high-performance computing platforms. Most of these pressures should be familiar to practitioners and the topics have been discussed in varying detail in other articles.

However, some of the proposed approaches for solving these issues will be less familiar to many in the community: domain specific languages, "dwarfs", data models, etc. In this respect, the paper is an excellent review article that provides ample references for researchers to become familiar with these potential solutions.

Ultimately, though, what the authors are proposing is a large community-wide coordinated effort to (further) distribute the burdens of software development. By reducing duplication and emphasizing separation-of-concerns, the hope is that models can leverage a more powerful suite of infrastructure tools, libraries, and frameworks that can more readily address performance challenges. The challenge then becomes one of coordination within the community in the face of daunting complexity.

Some specific comments and concerns:

The discussion of DSL's could be extended by including KPP/KPPA as examples. These are tools used my atmospheric chemists to translate reaction mechanisms into executable code. The tools can generate optimized code for various architectures and are generally treated as black boxes by the end-user. While most Earth system modelers have probably not directly used these, they are likely a more familiar example and could better serve as an introduction to DSL's.

The statements on page 7 regarding PGAS languages and mixed-mode programming are perhaps a bit too strong. I can only speak for Co-Array Fortran (CAF), but there are not any obvious mechanism in that language that correspond to mixed-mode support. CAF images generally correspond to MPI processes and can be implemented to run on threads (just as MPI can) or can leverage threading under-the-hood (just as MPI can). The main point of PGAS languages, at least to my understanding, is that they provide a simpler/clearer mechanism for expressing locality than say, MPI. But they do not have a tiered approach in this respect. Things are either local (in the image) or remote (off the image).

I found that the beginning section 2.1 on Hardware Complexity to be a bit of exaggeration on the whole. Too many quoted words and exclamation points. If there is another revision, I would hope that the general tenor could be altered to match the remainder of the paper.

2017.

C3