Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-186-RC2, 2018

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 #2

Received and published: 9 February 2018

This paper does a good job of describing the challenges associated with development of software for climate and earth system models. While these challenges are well known within the climate community they are not always well known even in the very related but different numerical weather prediction (NWP) community.

The paper uses hyperbole in the first several sections of the paper to described several issues. In particular they use the term "an impending cliff" to describe the need to modernize climate applications to take advantage of upcoming architectures. Honestly this is overstated as the current challenges are no more ominous then the challenges that were faced in the conversion from long vector machines to scalar machines in the

late 80's and early 90's. Then as today codes evolved or became obsolete. There is no mentioned that I can see that the climate community has been here before and scientific progress was not devastated.

A similar excessive hyperbole is seen in the Hardware Complexity section 2.1 Yes, there are a number of different architectures that exist today due to greater specialization. However only a limited number are actually viable for climate and NWP. The greater diversity in architectures should really be thought of as a boon for certain classes of computing problems that were not well served by existing architectures instead of a burden to the climate and NWP community.

I found that Figure 3 does not really add much to the software complexity discussion. What they really need to emphasize in this section is not that the IFS model has a bunch of blue boxes connected together in a serial fashion, but rather that the correct data-structures and computational patterns of which different models are composed are quite varied. There is some discussion of climate dwarfs in Section 3.2, and brief mentions that IFS is a spectral transform model while HiRAM is a finite volume however. But I think this concept of different computational patterns and its impact on complexity should really be fleshed out better in the software complexity section.

One issue that I feel is missing in the discussion in Section 4 is how to manage support funding across different countries outside the European Union. For example, if I am using a I/O library developed in Germany in my US climate model, how would I get the German developers to fix a critical bug that only occurs in my US model? Or is the sharing of common software infrastructure only viable across institutions within a common economic market? I think this question becomes very critical with respect to the use of a Domain Specific Language. If the shared software component is simply a library then it is relatively easy to substitute in an alternative. However the commitment to using a Domain Specific Language is much larger and bugs that were not addressed in a timely fashion could have catastrophic impact on scientific productivity. It is for this reason that I don't see a widely adopted domain specific language, other than Fortran of course, as a viable option.

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