Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-280-RC2, 2018

Geoscientific Model Development Discussions

Interactive comment on "Improving climate model coupling through a complete mesh representation: a case study with E3SM (v1) and MOAB (v5.x)" *by* Vijay S. Mahadevan et al.

Anonymous Referee #2

Received and published: 19 December 2018

This paper describes some new algorithms and implementation in E3SM with regard to generation of interpolation weights on diverse and complex grids. It includes a fairly comprehensive set of results and description. It is well organized, well thought-out, and well written. These results seem to leapfrog prior efforts.

General Comments:

The main body text where references are given needs to be reformatted. The references and text are not clearly separated and make them difficult to read. There is also a reference to "Section. (1)" on Page 6, line 6, that I believe suffers from the same formatting problem?

There are clearly trade-offs in generating weights online vs offline. This is highlighted in the paper a number of times with emphasis on the benefits in workflow associated with online capability. High performing online weights generation also has the ability to support non-static grids. Both of these are great benefits of online weights generation. The outstanding questions that are not answered in the paper are (1) can the weights generated online be counted on to produce error free interpolation (conservation, monotonicity, etc) without first being reviewed and validated offline? (2) is the weights generation capability robust and reliable enough to run on different platforms and expect the same results to at least roundoff? (3) is it faster to generate weights online vs reading them in? (4) Is there some benefit to generating the weights online and then being able to reuse them as compared to regenerating them each time the model is run with regard to performance or reproducibility? It would be helpful if the paper addressed these issues if possible. These issues are partly raised in a few places in the paper, at least Page 6, Lines 25-27 and Page 18, Lines 9-10. Some addiitonal discussion/results might be interesting.

Page 8, line 1, what does mesh aware entail? You discuss the potential all-to-all nature of weights generation in the prior paragraph. What information does MOAB carry around that help this problem and how much memory does it require? Does each task have access to the global grid information without requiring communication? Or is there just neighbor connectivity stored? If the grid description is not global on all tasks, how much is communication reduced vs having only local information? The MCT gsmap has global information on each task related to ID, and pe. I assume the MOAB mesh has the same plus coordinate information? The MCT gsmap is generally compressed significantly because the information can be defined via a single start and end ID for certain kinds of decompositions. Since the MOAB mesh carries more info, I assume that compression is not possible and that the mesh consists of "n" fields of data for each gridpoint/corner/edge/etc? Is that a lot of data? Does the memory scale at all at higher resolutions and higher pe counts? I'm sure much of this is documented in MOAB papers, but it would be nice to add a sentence or two about it in this paper.

How the new capabilities are implemented in E3SM is somewhat unclear. In Figure 1, it looks like there is no longer a coupler. Where are the non-coupling non-mapping coupler operations (merging, atm/ocn flux, diagnostics, etc) being computed? In text, it sounds like the coupler component still exists but that the underlying MCT datatypes were swapped for MOAB datatypes, an additional set of calls were added in the component coupling layer to more fully describe the meshes, the online weights generation was added, and the online sparse matrix multiply was converted from MCT calls to MOAB calls. But then at page 17, line 12-15, it sounds like the coupling is between pairs of components excluding a coupler. It would be good if this were clarified.

Specific Comments:

Page 6, line 20 “oas (2018)” ?

Page 7, line 29 fix “a in-line”, should be “an in-line”

Page 8, line 7 Alg. 1 -> Algorithm 1

Page 11, Fig 5b. It seems unlikely that the trivial decompostion would be someone’s first guess for best performing decomposition with knowledge of how the coupling/mapping work. Having said that, I’m surprised it performs as well as it does in Figure 14. There are lots of other resonable decompositions, why were Trival and Zoltan chosen to be highlighted in this paper? And why does the trivial decomposition perform so well in Figure 14.

Page 12 line 23, remove “is” in “is results in”

Page 16, line 28, bit-for-bit capability is sometimes important to achieve, certainly for identical runs, also for runs on different pe counts (sometimes with a performance penalty via an optional flag). This sentence left me asking what the bit-for-bit capabilities are and what risks are introduced when computing online versus reusing.

Page 19, Fig 9. it would be nice if the scale were not so ad-hoc and instead something more like (0,80,4). Scales like the one shown just make the figure more difficult to

digest and in this case, there is no benefit to have the breaks defined as they are relative to something simpler and easier to read. Also, I’m not sure color adds anything, I think the same could be shown via a contour plot, possibly clearer and simpler still.

Page 18, line 29 “serial runs” vs page 20, line 1 “better performance in MBTempest . . . offers avenues to incorporate task level parallelism . . .”. Are these serial runs or something else? Serial in MPI but using shared memory parallelism? Is that still serial?

Page 20, lines 18-19. For the 1024ˆ3 test case, are weights being generated in 2d or 3d? If 3d, is this test case an order of magitude (or more) larger than the largest climate model grids? Might be worth clairfying in text.

Page 20, lines 24-26. I agree that the initialization cost is amortized for long production climate runs. But in your example, that init cost is order (hundreds) of seconds (see fig 10c). That is for a single set (pair?) of weights. In coupled climate model, there are often order (10) of these to be done. Now we’re talking 1000s of seconds which starts to sound expensive in production but is certainly very expensive for short development test runs. Would it be cheaper to store the weights in a file and read them in the next time? (see general comments). Having said that, please confirm that weights are generated on each of the 1 billion gridcells (1024ˆ3) in 3d. And if so, that’s a lot of gridcells.

Page 21, Figure 10, I am struggling to read the axes and other text on the plots

Page 21, Figure 10b shows scaling to 512k pes for a problem size of 1024ˆ3. The final point has 2000 gridcells per process which is still relatively big. What if you chose a problem size of 128ˆ3 or 256ˆ3 and tried to scale to 512k cores?

Page 25, figure 13. Is there benefit to showing the three results (colocated plus two disjoint). The results are very similar for the three cases, at least as presented. And there is no discussion of the differences/similarities in text.

Page 26, figure 14b. I am surprised there is so little scaling of the send/recv at NE120

and the core counts presented. I understand the claim that the absolute cost is small in all cases. I guess you are only redistributing 86k (NE120) elements, maybe that's expected then? The jump between 64 and 128 must be a machine thing, going off-node or something? At 128 cores, you should be transferring over 500 elements per core. Do you expect no scaling beyond that given the message size? Do you want to mention any of this in the paper?

Please confirm that you describe which machine the tests are run on in text and it might be beneficial to include that information in the figure captions. For page 27, figure 15, maybe remind us that it's case B of Table 1 (I think that's correct) in text.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-280, 2018.