

Interactive comment on “Advanced parallel implementation of the coupled ocean-ice model FEMA0 with load balancing” by Pavel Perezhogin et al.

Pavel Perezhogin et al.

pperezhogin@gmail.com

Received and published: 12 November 2020

We greatly thank the reviewer for the careful reading of this manuscript and given suggestions.

1. The two step procedure of first dividing the model domain into small blocks and then redistributing those blocks between cores was not really clear to me at first and should be better communicated. It would be helpful for uninitiated readers if you can mention earlier on that the requirement is to preserve the structured nature of the code. So your partitions can't be of arbitrary shape, like in unstructured mesh models, but should be constructed out of small rectan-

Printer-friendly version

Discussion paper



gles. I would suggest creating a schematic that shows all the steps of the procedure - splitting into so called blocks, fitting the Hilbert curve, distributing the blocks among CPU cores and finally allocating "shared" arrays. Of course it's not possible to demonstrate with 128x128 blocks you use for a realistic model, but something like a 10x10 schematic representation would do the job.

A bit more details on how the partitioning handled in the model setup would be appreciated. Does the partitioning created by the library and then read by the model? Or it's computed each time. If the latter is the case - do you guarantee that the partitioning will be the same each time the model is run?

We do not think that there is a need to additionally explain algorithm of distribution of the blocks over the cores, because it doesn't meet the main objective of the paper, have been shown many times by Dennis and there is a general-purpose solution (METIS). "Shared" arrays are clarified in figures 2 and 3. There is no need for another figure.

The introduction is changed: P2 L34 "In numerical ocean model. . ." is moved to new paragraph; P2 L39 "We give preference . . ." is removed; P2 L41 "Note that some modern. . ." is moved to previous paragraph.

We add the last paragraph in the introduction: "In sections 2-4 we provide model configuration and organization of the calculations in the non-parallel code on structured rectangular grid. In section 5 we describe parallelization approach, which preserves original structure of the loops. Domain decomposition is carried out in two steps: first the model domain is divided into small blocks and then these blocks are distributed between CPU cores. For all blocks belonging to a given core a "shared" array is introduced, and mask of computational points restricts calculations. Partition could be of arbitrary shape, but blocks allow us to reach the following benefits: simple balancing algorithm (Hilbert curves) can be applied as the number of blocks along a given direction is chosen to be a power of 2; boundary exchanges can be easily constructed for arbitrary halo width, but smaller than the block size. In section 6 we report parallel

[Printer-friendly version](#)[Discussion paper](#)

acceleration on different partitions for particular 2D and 3D subroutines and the whole model.”

Section name “Organization of the calculations” is changed to “Organization of the calculations in non-parallel code”.

We add the first paragraph to the section “Modifications of the non-parallel code”: “In this section we describe the partitioning algorithm of the model domain into subdomains, each corresponding to a CPU core, and subsequent modifications of the single-core calculations, which require only minor changes of algorithms 2 and 3. Grid partition is performed in two steps: model domain is decomposed into small blocks and then these blocks are distributed over CPU cores in such a way that computational load imbalance is minimized. We utilize common grid partition for both sea-ice and ocean submodels, and provide theoretical estimates of the load imbalances resulting from the application of different weight functions in the balancing problem. Partition is calculated during the model initialization stage, as our balancing algorithm (Hilbert curves) is computationally unexpensive. Also, we guarantee that the partition is the same each time the model is run, if parameters of the partitioner were not modified.”

2. Minor comments will be taken into account in the revised version of the manuscript.

Interactive comment on Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2020-182>, 2020.

Printer-friendly version

Discussion paper

