Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-28-RC2, 2019

Geoscientific Model Development Discussions

Interactive comment on "OpenArray v1.0: A Simple Operator Library for the Decoupling of Ocean Modelling and Parallel Computing" *by* Xiaomeng Huang et al.

David Webb (Referee)

djw@noc.ac.uk

Received and published: 25 March 2019

This is a well written paper concerned with generating an ocean model from a set of equations closer in form to the underlying differential equations than is usual. It is an interesting computational exercise but the resulting code has some important deficiencies and for that reason I think it would be more suited for a computational journal than the present one.

My main concerns all involve computational efficiency. As the authors state, ocean models are memory bandwidth limited and for that reason the code is usually written in such a way that once a variable is in one of the processor registers or in the highest

speed cache it is used as much as possible before being replaced. In fact the aim should be never to move variables more than once each timestep. In the present code the authors spend a major section reporting on one small step in this direction, but really this should only be the first of many steps.

I am also concerned about the way the code deals with multi-processor architectures. Unfortunately although I was able to compile both the c++ and fortran sections, the link step failed and so I was not able to check the running code. However the main time stepping loop appears to run on a single processor and it is only the difference and averaging operators, in the c++ code, which make use of the multi-processor architecture. This is surprising given the authors emphasis on the importance of parallel computing.

I am also a bit wary about all the details being lost in the c++ compiler/interpreter code. The authors emphasis the possibility of portability but this implies a large organisation continually keeping such a code up to date and adapted to the latest hardware. If not, the climate modelling groups have to do it themselves in which case the effort required will be much the same as now except for the addition of the compiler/interpreter and associated packages.

Both of the architectures discussed appear to be cache based but as I understand it the next major advance will come from better use of gpus. In these systems I expect variables will stay in gpu memory or be swapped between gpus and rarely return to the main cpu memory. For such systems the structure proposed here appears unsuitable.

There is also the question of how researchers might try out new code with the prosed library and debug the resulting runs. No user manual is provided and it is difficult to see how bugs can be traced, especially if they involve the c++ section of the code.

I also do not understand why a just-in-time compiler is used, given that the model grids do not change in time so that both human and computer effort would be better spent optimising the fixed grid code. And on this theme I am also concerned, although I would

like to be proved wrong, that what has been achieved here is little different from what might be achieved with a good fortran coarray code, together with statement functions or cpp define statements to take the place of the operators.

One reason that these are not used in ocean models comes from the fact that in a typical ocean model only about half the theoretical 3-D grid is involved in the calculation, the rest representing land or ocean topography. When computer memory and power is readily available a factor of two does not really matter but given the computational cost of many ocean models, spending time on such cells, as usually happens when coarays are used, can be critical.

Anyway - you can see that I am unhappy. However I must emphasise that I can also see that the paper represents a lot of hard work and I accept that as an example of an attempt to tackle good computational problem it is worthwhile. For this reason I think that publication in a journal more closely linked to the development of artificial intelligence would be more suitable.

If on the basis of the other referees reports, the authors are asked to provide a revised version then there are two extra documents I would like to see in the additional documentation section. The first is a user manual. The second is a full list of the include files and libraries needed (i.e. all those which are not the fortran or c++ compiler files) - to help with the problems I had.

In such a case I would also like to see some good answers to the concerns detailed above and, although I have not mentioned it so far, a few fewer buzz-phrases.

David Webb.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-28, 2019.