Geosci. Model Dev. Discuss., 7, C3656–C3658, 2015
www.geosci-model-dev-discuss.net/7/C3656/2015/

Geoscientific Model Development Discussions

Interactive comment on "gpuPOM: a GPU-based Princeton Ocean Model" *by* X. Huang et al.

D.J. Webb (Referee)

djw@soton.ac.uk

Received and published: 20 April 2015

Overall the article is well written and the content is of potential interest to people considering further developing their ocean models of the ocean. This is because GPU hardware, similar to that discussed, may soon be used for many large scale ocean and atmospheric models.

As someone who, twenty years ago, developed a version of the MOM model for use on multi-processor machines I was interested to learn more about GPU systems and how they could be best used for ocean models. The paper shows that in the present case, each GPU unit can replace a large number of standard processors and also lead to almost an order of magnitude reduction in the power consumption when running the model.

Unfortunately I also have a large number of concerns. After a fairly standard introduc-

tion, the key description of the Nvidia K20X unit and the CUDA low level programming model, involving warps and streaming multiprocessors, is poorly written and confusing. There is also very little on the K20X memory and cache hardware although these will always have a major impact on the structure of the optimum code.

Another concern, not all the responsibility of the authors, arises from the fact that the POM model appears never to have been rewritten for cache based CPUs or multi-processor systems. As a result the sections of loop unrolling and overlapping I/O are not new and should have been included in the base code years ago.

I am also concerned that there is no proper discussion about how best to deal with the large ocean model arrays in a cache based system. The code continues to use the east-west index as the innermost array index, although with a cache it may be more efficient to use the vertical index. Although not mentioned in the paper, the code shows that many of the innermost loops have been changed to vectorise in the vertical However because the main subroutines updating the tracer and velocity fields have been split into a number of small GPL kernels, many opportunities to make better use of the cache and to reduce cache loads have been missed.

The amount of effort spent converting a Fortran code to C and CUDA-C is also odd, given that a CUDA-Fortan compiler has been available since 2009 and that in future POM is likely to stay a Fortran code - if for no other reason than the simplicity of loop optimisation with this compiler.

Because of these concerns I think that the paper cannot be published in its present form. The authors need to improve their description of the hardware and software models. They also need a proper quantitative discussion of how the ocean model is fitted into memory and cache, and where the bottlenecks are when running the model. Finally I think the code needs to be rewritten to drastically reduce the number of independent kernels. This is because once a cache contains the temperature, salinity, velocity and grid arrays for a small region of ocean, it appears senseless not to update

the values of all of these variables.

Interactive comment on Geosci. Model Dev. Discuss., 7, 7651, 2014.