Geosci. Model Dev. Discuss., 7, C3218–C3221, 2015
www.geosci-model-dev-discuss.net/7/C3218/2015/

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on “An approach to enhance pnetCDF performance in environmenta modeling applications” *by* D. C. Wong et al.

D. C. Wong et al.

wong.david-c@epa.gov

Received and published: 30 January 2015

We would like to thank the referee for a very thoughtful and detailed review of our manuscript. Incorporation of the reviewer’s suggestions has led to a much improved manuscript. Below we provide a point-by-point response to the reviewer’s comments and how we have addressed them in the revised manuscript.

[Comment]: First, altering IOAPI and PARIO to do true parallel I/O is a necessary engineering effort, but it not novel in 2014. (Authors do not spend a lot of time on this point, so I think they understand and would agree with me). [Response]: We agree with the reviewer that we had a bad choice of word. We have replaced novel with “an application level data aggregation approach” in the manuscript. As a matter of fact, re-engineering PARIO to make it to perform true parallel I/O operation is the only way

to overcome I/O bottleneck in the air quality model, CMAQ.

[Comment]: second, application level aggregation is not novel: in climate/weather it has been done/published in GCRM (a cloud resolving model) and PIO (for climate simulations). The approach described here, where the aggregation is done according to MPI processor topology, sounds a tiny bit novel, but does not get a lot of text.

[Response]: We agree with the reviewer that application level aggregation is not novel. Indeed, we were taught in MPI classes to aggregate data for message passing when it is possible. Performing true parallel I/O through pnetCDF and making use of data aggregation in the application level to increase pnetCDF performance in the air quality model, CMAQ is our unique contribution. Thanks to the reviewer bringing our attention to two additional recent papers: * Bruce Palmera, Annette Koontza, Karen Schuchardta, Ross Heikesb, David Randallb, "Efficient data IO for a Parallel Global Cloud Resolving Model", Environmental Modelling & Software, Volume 26, Issue 12, December 2011, Pages 1725–1735 * X. M. Huang, W. C. Wang, H. H. Fu, G. W. Yang, B. Wang, and C. Zhang, "A fast input/output library for high-resolution climate models", Geosci. Model Dev., 7, 93–103, 2014 The former paper stresses the importance of utilizing data aggregation to achieve high bandwidth and our work is also based on this fact. The latter paper focuses on having an extra set of processors dedicated for I/O operation to achieve overlapping of computational work and I/O task. We have compared the overall model run time when we considered those extra processors as part of the computational resource versus only dedicated for I/O process. Treating those extra processor for I/O only did poorly since I/O bound (I/O time with respect to the overall model run time is about 15 - 25%).

[Comment]: I am not sure how much tuning the authors did after adopting parallel-netCDF. Evalutaions suggest stripe size and stripe count were the two knobs chosen. As was demonstrated in Behzad and Lu’s 2013 SC paper (http://dl.acm.org/citation.cfm?id=2503210.2503278), tuning the I/O stack on machines like Edison and Kraken can have a 7-fold impact on performance. Now it must be said that a further point of the 2013 paper was that it’s a burden to expose these detailed tuning approaches to application scientists, so it’s ok if the authors only explored those two settings. I just want it explicilty mentioned.

[Response]: We know there are two parameters, “stripe count’ and “stripe size”, that users can adjust for performance purposes. Our intention was to try to obtain an "optimal" setting with respect to PE configuration and model domain size. In general, a user (scientist) does not know much about tuning I/O stack to obtain better performance. Here we try to provide some easy understandable way to improve I/O performance in scientific applications. Again thank you so much for bringing our attention to Lu’s paper which provides lots of useful information. It will be great if Lu’s work could be turned into some simple tools so scientists can use it to achieve optimal I/O performance for their model on different platforms.

[Comment]: Is the simplified CMAQ model used in these experiments available for others to use, or will it be made avaliable? The I/O community is a voracious consumer of such I/O kernels: if you publish the one you have created for CMAQ, then a small battalion of grad students and I/O researchers will add it to their list of kernels they consider when evaluating new i/o strategies and designing new i/o subsystems. [Response]: We are more than happy to share the simplified CMAQ model, we called it “pseudo” code, with you. Basically this striped down version of CMAQ looks like this (this pseudo code has been added to the manuscript):

```
DO I = 1, 3
Read in data
Perform numerical calculation (artificial work)
Output result
END DO
```

[Comment]: What aspects of the I/O stack made pnetcdf under-perform? Are there

lessons to be learned from CMAQ that could be applied to the I/O stack (pnetcdf, MPI-IO, and Lustre layers) that would benefit all applications on Edison and Kraken? [Response]: In this paper we did not attempt (in fact we don’t have such knowledge) to identify which aspect of the I/O stack made pnetCDF under-perform. It is known that pnetCDF and MPI-IO have aggregation capability with respect to I/O requests or messages. Our approach is to apply data aggregation on the application level. It will be difficult to adopt this approach to the I/O stack since this approach based on the knowledge of spatial domain decomposition which the I/O stack does not have. In this paper, we have demonstrated that scientists can adopt this application level data aggregation technique in an effective and straightforward manner.

Interactive comment on Geosci. Model Dev. Discuss., 7, 7427, 2014.