Geosci. Model Dev. Discuss., 7, C2869–C2870, 2015
www.geosci-model-dev-discuss.net/7/C2869/2015/

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.

R. Latham (Referee)

robl@mcs.anl.gov

Received and published: 5 January 2015

I would say the authors have done a better-than-average job of addressing the I/O overhead in CMAQ. Often, applications will encounter I/O problems and throw up their hands. These authors do a bit more, but I would like to see even more detail.

I live in the parallel-netcdf and MPI-IO layer, so my tendency is to encourage a more technical evalutaion than the authors perhaps intended.

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).

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.

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.

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.

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?

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