Geosci. Model Dev. Discuss., 7, C3214–C3217, 2015
www.geosci-model-dev-discuss.net/7/C3214/2015/

Geoscientific Model Development Discussions

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]: The authors do not provide sufficient information to reproduce the results of their work. What version of pnetcdf was used and and what versions of supporting software such as MPI and Lustre were used? What version of the CMAQ models was used and is the pseudo-code used to perform the experiments publicly available? No reference was provided for either netcdf or pnetcdf libraries. [Response]: We agree with the reviewer's point of view that there is insufficient software information

describing our experiments in the paper. Here is additional information. These are the software packages we used on Edison and Kraken, respectively and will be added to the manuscript in section 4.1: Edison: cray-mpich/7.0.4, cray-netcdf/4.3.0, parallel-netcdf/1.3.1, lustre: 2.5.0 Kraken: Cray MPT 5.3.5, netcdf 3.6.3, pnetcdf 1.2.0, lustre 2.5.0 We used CMAQ version 5.0.2 for the one day simulation and we have constructed a small scale test code based on the data flow of regular CMAQ: a cycle of input data, data calculation, and output data. In the manuscript, section 4.2, we will provide the pseudo code of this small scale test code. This small scale test code can be obtained upon request.

[Comment]: The data aggregation technique that the authors refer to as a novel new approach is in essence the same technique that pnetcdf, MPIIO and Lustre apply at lower levels in the software stack so the question becomes - why does doing this aggregation at a higher level in the software stack work better than it does at a lower level? [Response]: Based on the pnetcdf documentation, data aggregation can be done on sequences of small requests with non-blocking I/O. Our data layout is column, row, layer, variable while the spatial domain is decomposed. The amount of data in each processor is not "small" even with smallest domain described in this article and the number of processors is over 128. Hence aggregate data in the software level with respect to the spatial domain, in particular along the column dimension, makes more sense: not only the output data chunk is larger but also the contiguousness of the data.

[Comment]: Total time to perform parallel IO includes both communication time and IO time but the authors make no attempt to separate these factors. In section 5.3 the authors claim that increasing the data size on the processor which is responsible for I/O will translate into higher I/O rates. If this is the case why not just use the original serial approach in which one I/O processor is responsible for all of the data? [Response]: In our approach, the process consists of two parts: data aggregation (communication time) and data write to disk (I/O time) by a subset of processors. So the sum of the communication time and the I/O time is the true representation of time to move data

in each processor to the disk. This timing can be used to compare directly with the parallel I/O approach implemented with pnetCDF which sends data to the disk collectively without any communication. Figure 11 illustrates the notion of larger chunk has a better I/O rate, we argue that the benefit of data aggregation as a basis for the success of our approach. We won't go to the extreme to have one processor to do the I/O (this is the technique being employed in the current CMAQ model) but rather try to strike for a balance by utilizing parallel I/O technology. Indeed, in this article, we have demonstrated our approach out performs the serial approach which is currently used in the CMAQ model, substantially.

[Comment]: The pnetcdf library contains a number of IO interfaces, collective vs independent and synchronous vs asynchronous, and supports several techniques for data aggregation. The authors do not indicate which pnetcdf interfaces they are using, or whether they experimented with others. [Response]: In this work, we are using only collective parallel netcdf API. We can take a look at other types of interfaces that pnetcdf provides as future work.

[Comment]: Figures 3-8 are too small and too busy to convey any meaningful information, perhaps tables would be better? [Response]: We agree that Figures 3 - 8 are too small and too busy however, we believe a 3D plot is the best way to convey performance information with respect to two different variables: stripe count and stripe size, at the same time. In the original plots, positive (solid color) and negative (checkered pattern) bars which denotes the performance comparison of a specific technique, can be clearly distinguished in each scenario. After the publisher's typesetting process of the entire paper, such clear distinction is gone and that is unfortunate. We have re-plotted these graphs with only two colours: red which denotes positive value and blue which denotes negative value. We have replaced Figures 3 – 8 with these new ones in the manuscript. Here we have attached Figure 3 as example what the new figure looks like.

Please also note the supplement to this comment:
http://www.geosci-model-dev-discuss.net/7/C3214/2015/gmdd-7-C3214-2015-supplement.pdf

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