Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., author comment AC2
https://doi.org/10.5194/gmd-2021-183-AC2, 2021

Reply on RC3

Phillip Alderman

Author comment on "Parallel gridded simulation framework for DSSAT-CSM (version 4.7.5.21) using MPI and NetCDF" by Phillip D. Alderman, Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-183-AC2, 2021

Thank you for your comments. I found them to be fair and constructive. I offer here a few thoughts in response and some questions.

I have tried experimenting with RAM-disk with some limited success, but have not found the gains there to be very large or consistent.

I do agree with you that most of the speedup is probably related to the generation of the text input files rather than the execution time of DSSAT-CSM itself. However, it seems to me that the speed of the system should be evaluated on the process from gridded inputs to gridded outputs rather than only on the DSSAT-CSM simulation in isolation. I thought you did make a valid point with your comment "If R is slow to prepare the inputs, the comparison may be detecting R's attributes rather than DSSAT's." The implementation of the text input in the present case is somewhat naive based on an existing R interface for DSSAT-CSM. No doubt one could develop a faster implementation. What do you mean by a "real" programming language?

That said, the main objective was documenting the framework itself rather than providing a comprehensive comparison of text vs. NetCDF per se. The comparison of text vs NetCDF was meant to provide a rough contrast of execution time. I do plan to perform a more comprehensive and systematic analysis of execution time and scaling of the framework in a subsequent study. Even so, I certainly do not want to be disingenuous in my treatment of the topic here. Do you think that the current discussion on this point is misleading? If so, what points would you suggest that should be added in the discussion to clarify the situation?

I may need to think some more about your comments vis-a-vis MPI. However, my initial reaction is that I think MPI's utility may depend on whether the simulations are being run on a dedicated cluster rather than on a shared resource. Although it would be possible to submit a large number of small jobs to a job scheduler, users of shared resources (e.g. university clusters) often have a limit on the number of jobs they can submit at a given time. The advantage of using MPI is that you can have parallelization within jobs that allows you to run tens of thousands of grid points without needing to submit that many jobs. I welcome your response if you have a different opinion or think I am missing something.

Your comments on parallelization vs. portability are helpful. I will consider where/how to

add clarification there for "unwary readers." Perhaps my perspective is a bit different since I'm looking at the issue from within the DSSAT-CSM development community. When I see legacy artifacts like "clunky text files," I ask myself whether there is a better, more efficient way to achieve the same objective while at the same maximizing portability. I realize that portability and optimization for a specific problem/system architecture will always be in tension. Nevertheless, the work I have presented in this manuscript is meant to move things forward within the DSSAT-CSM community.