

Geosci. Model Dev. Discuss., referee comment RC4
<https://doi.org/10.5194/gmd-2021-183-RC4>, 2021
© Author(s) 2021. This work is distributed under
the Creative Commons Attribution 4.0 License.

Reply on AC2

Anonymous Referee #3

Referee 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-RC4>, 2021

Since you have experimented with RAM-disks, perhaps you can mention that as another speed comparison.

Concerning the "R" comparison, what I meant by a "real" programming language is that I wrote a big java program to walk through the giant input dataset one pixel/location at a time. That giant program builds the climate or weather files in the appropriate format, the fileX, runs DSSAT, and scoops up the output files. Your system pulls some of those pieces inside DSSAT which is great. So, similarly, the data handling part will be sped up by using FORTRAN (as you have done) or java or another "real" language rather than, for example, my pathetic shell scripts.

Still, I think the point that we are coming to a consensus on is that the computation, as such, is often NOT the issue. The real issues are ease of use by the researcher (e.g., being able to use netCDF files directly instead of requiring conversion to some other format and yet another conversion immediately before running DSSAT) and the ability to game the management system of whatever computer you have access to.

Again, my personal experience (the only experience I have, naturally) is that total wall time to finish is faster when using more "limited" computing resources that you have complete control of than on the incredibly large shared resources. Success on the big machines requires gaming the system somehow (flooding with lots of small jobs to claim any and all backfill opportunities, using do-nothing placeholder jobs so you can recycle your access to the resources, having massive funding to buy dedicated time, complaining to the administrators that if they want to include your "helping real poor people" project in their glossy brochure to obtain their own funding that maybe they should give you a dedicated queue so you can actually do the work, etc.). Thus, we circle back to structuring the computer program in a way that makes it useful on the computing resources available, that is easy for the researcher to use, and allows the researcher to manage the human/social barriers to accessing the appropriate computing resources.

My comment to the "unwary readers" was poorly introduced. I did not mean that there was a shortcoming in the manuscript; I wished to preemptively defend you against anyone who might try to criticize your implementation as being too narrow. That said, if there is a way to slide those ideas into the manuscript, it might help reassure those who come along later who will wrestle with the same issues.