Geosci. Model Dev. Discuss., 6, C1740–C1742, 2013
www.geosci-model-dev-discuss.net/6/C1740/2013/

Geoscientific Model Development Discussions

Interactive comment on "A fast input/output library for high resolution climate models" *by* X. Huang et al.

R. Latham (Referee)

robl@mcs.anl.gov

Received and published: 9 December 2013

Overall:

Observing that climate simulations have regular phases of computation and I/O, the authors implemented a framework whereby an application can dispatch I/O to "I/O processes".

Specific comments:

The Interactive Discussion has already touched on this a bit, but I think the paper needs to be more clear about the role of CFIO in the climate software stack. Phrases like "CFIO decreases the I/O overhead by a factor of 5.1 compared to pnetcdf" give the impression that this project replaces pnetcdf, when really it augments. Perhaps

phrases like that could be re-worded to say "decreases the I/O overhead compared to pnetcdf alone"?

How are you launching extra i/o processes? The paper makes no mention of the mechanism, but if you are, or were, using MPI dynamic process management routines, that would be remarkable (they do not see much if any use, and I'm sure the MPI forum would welcome any papers discussing experiences of that facility). On the other hand, dynamic processes don't work on blue gene, so an alternate approach would be more portable. In the github code, it looks like you assign some processes in COMM_WORLD to be i/o processes. More portable, but now you've got servers and clients contending? I think your paper and your code have diverged a bit. Have you learned something worth adding to the paper?

(Ah the perils of publishing your source code. I was happy to find the code on github, and only raise these questions because I like your ideas and want them to work.)

Some codes, like GCRM, have pretty specific compute node requirements (gcrm: must be "10 times a power of two" processes), so spawning extra procs makes sense. on the other hand, you simply cannot spawn extra procs on Blue Gene, so stealing from COMM_WORLD might make more sense.

In section 3.1 you mention "For data writing operations, data aggregation is performed to gather subarray data". Only writes? Would reads not benefit from this approach? Or, as I think is the case, do you imagine this framework rarely if ever being used for reads?

I had to laugh at your understatement that determining the ideal balance of compute and i/o resources was "ongoing work". to say the least!

I'm worried about your test environment. Intel MPI requires extra steps to turn on MPI-IO parallel I/O for lustre, I think: - http://software.intel.com/en-us/forums/topic/286114 - lustre default striping is something tiny like 4?

but i think from the other parts of your paper you have already dealt with those details? Lustre complicates the I/O story enough that they are worth mentioning.

related work: I found your related work section to be a simple laundry list of technologies. For each paper you mention, it would be good to explicitly discuss how your work relates.

I think it's worth noting that CFIO implements a form of the two-phase optimization implemented in ROMIO. ROMIO two phase designates some MPI ranks to be the I/O aggregators, though ROMIO's aggregators are assigned to file regions, not to clients. ROMIO's approach also has no information about what's being done (data model is "linear stream of bytes"), whereas your approach has an array-oriented model.

Jing Fu's 2010 paper is noteworthy because it quantifies the overhead of overlaping i/o and computation.

I'd like more clarification of how you are overlapping I/O and computation – you discuss how an asynchronous approach introduces too much overhead, and that a synchronous approach scales better. If so, where is the overlap/benefit happening now?

Your baseline applications collect data on rank 0. this is of course a stupid way to do i/o in 2013, so it's not surprising you get remarkable improvements. What if those baseline applications used parallel-netcdf? Again, it's not clear to me where the overlap is happening if i/o is happening synchronously

Technical Corrections:

I did not notice any typos or grammar mistakes. Well done.

Interactive comment on Geosci. Model Dev. Discuss., 6, 4775, 2013.