GROMACS examiners notes

Michael Robo • Oct 30

Here's some feedback based on what I saw in the homework. I've updated the assignment with a rubric if you want the answers or granular details on grading.

General Notes
Not all of you posted your logs from mdrun, and many of those of you who did didn't do in the exact format that I asked. I was lenient about the organization of the mdrun logs for the homework, but your logs will need to be organized exactly as I ask in the final assignment.

Question 1
Because the benchmarking was done on a wide variety of systems, I got a big range of performance numbers in the responses. Obviously, this is something to be expected. Less obviously, some systems had at least 48 physical cores, and others had less than 48 cores but at least 48 threads (using multithreading). People who ran simulations on multithreaded systems saw their performance *decline* after 24 threads, but increase again to the 24 thread value at 48 threads. This is consistent with the notion of multithreading not providing value for HPC applications such as GROMACS. If you did the homework on a system with 48 or more physical cores, I would recommend trying some runs on a Chameleon system with 24 cores to see if you also experience this behavior.

Question 2
A lot of people were tripped up by the difference between atoms and molecules. For reference, a molecule is a group of atoms, and the number of molecules in the system is explicitly listed in the topol.top file. See slide 24 of the webinar slides for reference.

Question 3
Some people answered that the ligand-only runs needed ~10x less FLOPS than the protein+ligand runs. Other noted that the protein+ligand system has 10x more atoms than the ligand-only system. Both answers are correct, as the number of atoms in the system will directly influence the total FLOPS needed for the simulation.

Question 4
By default, GROMACS should have created 48 ranks, with 1 thread each, but this didn't happen for all students. You can use the -ntmpi and the -ntomp options to specifically control how many ranks, and how many OMP threads per rank you are using.

Question 5
This was the tricky question, and I hope you guys had as much fun grappling with it as I had coming up with it! Here's the answer:

The best way to look at a job that failed is to look at the log that succeeded and note the differences.
Let's use the -nt 24 and -nt 28 versions of the lig-only runs as an example.

If 28 threads are used, and GROMACS creates 28 ranks with 1 thread per rank, then 24 of them automatically get assigned to PP ranks.
In the lig-only system, the maximum decomposition per dimension is 4 X 4 Y 4 Z. Because domain decomposition involves literally splitting up space into chunks, we have to come up with 3 integers that we can multiply together to get to 24 in order to utilize 24 ranks. Due to the factorization of 24 (2*2*2*3), this is possible, you can use a 4 X 2 Y 3 Z setup, for instance.

However, when 24 threads are used, GROMACS creates only 20 PP ranks. Due to the factorization of 20 (2*2*5), we can't come up with 3 integers of value 4 or less that we can multiply together to get to 20.

Question 6
This was a relatively straightforwad question, but not everyone followed the directions and made a trajectory of the complete protein+ligand system as asked. The fact that the frames are 0-ordered in VMD tripped some people up, there should have been 11 frames in the complete trajectory total.