GROMCS

Jiarong song

December 2023

Contents

0.1 Answer

0.1.1 Question 1

1. Include your plots of benchmarking the performance and efficiency of the simulations of the protein+ligand and ligand-only systems at varying thread counts.

a. What are the general trends in performance (ns/day)?

- Performance (nanoseconds/day of simulation) generally improves as the number of threads increases, meaning the simulation runs faster. However, this improvement may become less noticeable after a certain point, as the overhead of parallel processing may begin to outweigh its benefits.

b. What are the trends in efficiency ((ns/day)/(Core t))?

protein+ligand	performance	effencicy
1 thread	1.632	1.632/5293.022
2 threads	5.976	5.976/2891.526
4 threads	7.984	7.984/4328.492
8 threads	8.327	8.327/8300.783
12 threads	7.997	7.997/12964.362
16 threads	4.985	4.985/27729.526

Table 1: Protein+Ligand

lig-only	performance	efficiency
1 thread	39.161	39.161/220.630
2 threads	63.967	63.967/270.144
4 threads	102.475	102.475/337.260
8 threads	109.510	109.510/631.185
12 threads	94.439	94.439/1097.871
16 threads	24.716	24.716/5593.233

Table 2: Ligand-only

- As the number of threads increases, efficiency may decrease. This is because, although using more threads can improve performance, this usually results in a decrease in efficiency per thread. This is because the overhead of parallel processing (e.g., communication and synchronization between threads) increases with the number of threads.

Note: Because the version of the program loaded on the computer is too old, it cannot truly simulate the image. The above answer is consistent with the Internet search.
The failure content is as follow:

Info) VMD for LINUXAMD64, version 1.9.4a57 (April 27, 2022)
Info) http://www.ks.uiuc.edu/Research/vmd/
Info) Email questions and bug reports to vmd@ks.uiuc.edu
Info) Please include this reference in published work using VMD:
Info) Humphrey, W., Dalke, A. and Schulten, K., 'VMD - Visual
Info) Molecular Dynamics', J. Molec. Graphics 1996, 14.1, 33- 38.
Info) ————————————————————-
Info) Multithreading available, 12 CPUs.
Info) CPU features: SSE2 SSE4.1 AVX AVX2 FMA F16 HT
Info) Detected VM or hypervisor execution environment
Info) Free system memory: 146MB (2
Info) Creating CUDA device pool and initializing hardware...
Info) Unable to load NVML library, GPU-CPU affinity unavailable.
Info) Detected 1 available CUDA accelerator::
Info) [0] NVIDIA GeForce MX450 14 SM 7.5 1.6 GHz, 2.0GB RAM SP32 KT AE2 ZC
OptiXRenderer) ERROR: Failed to load OptiX shared library.
OptiXRenderer) NVIDIA driver may be too old.
OptiXRenderer) Check/update NVIDIA driver
Info) Dynamically loaded 3 plugins in directory:
Info) /usr/local/lib/vmd/plugins/LINUXAMD64/molfile

0.1.2 Question 2

2. Look at the atom counts of the final set up protein+ligand and ligand-only systems.

	protein+ligand	ligand-only
solvent molecules	15792	1606
chloride ions	15855	1612
sodium ions	15822	1609

Table 3: Caption

a. How many atoms does each system have?

$$lig-only: 4876 \qquad protein+other: 49552$$

b. How many solvent molecules, chloride ions, and sodium ions does each system have?

0.1.3 Question 3

3. In the benchmarking of the protein+ligand system and ligand-only system, why is there such a large disparity in ns/day between the two systems when the same number of threads are used?

The difference in performance between protein+ligand systems and ligand-alone systems is primarily due to differences in the complexity and communication overhead of the two systems. Protein + ligand systems contain more atoms and interactions and therefore require more computational resources and time to simulate. On the other hand, ligand-alone systems are relatively simple and therefore simulations can be completed faster. In addition, when using multi-threading for simulation, data exchange and synchronization are required between different systems, which will also increase communication overhead and delay, affecting performance.

0.1.4 Question 4

4. Take a look at the log for the 48 thread protein+ligand system (in $fak_example/mdrun_48_threads/$).

Note: Due to the influence of computer performance, 48 threads are not running, so this question is not done.

a. How many ranks did GROMACS assign to PP?

b. How many ranks did GROMACS assign to PME?

c. How many threads were assigned per rank?

d. If you wanted to get better performance, how would you change the ratio of ranks?

0.1.5 Quertion 5

5. Let's look at those failed ligand-only runs.

a. At higher thread counts, why did the ligand-only system become sensitive to the number of threads used? Specifically, why were there cases where a lower thread count failed, but a higher thread count would run normally?

Because its calculation or simulation may involve a large number of parallel operations. In some cases, a lower thread count may result in uneven task distribution, low resource utilization, or even cause the program to fail in some extreme cases. At a higher thread count, the program may run normally due to more even distribution of tasks and higher resource utilization.

b. Why didn't you see this behavior in the protein+ligand system?

c. Are there specific differences between the log files of the protein+ligand and ligand-only runs at the same thread count that you can point to that explain the change in behavior?

d. In the cases where the ligand-only system failed to run, what changes to your mdrun command could you make to fully utilize your requested thread-count?

I can add –oversubscribe to the directive.

0.1.6 Question 6

6. Include a screenshot (including the Display, graphical representations, and VMD main windows, similar to slide 49 of the powerpoint) of your VMD visualization of the protein+ligand system from the first frame and the last frame. Keep the axis in the lower left corner (this is done by default, just don't get rid of it).

a. How many frames does your trajectory have?