GROMACS - Task #2412

attempt to do better FFTW planning

02/14/2018 07:33 PM - Szilárd Páll

Status:	New	
Priority:	Normal	
Assignee:		
Category:	mdrun	
Target version:		
Difficulty:	uncategorized	

Description

The current FFTW performance is often highly unstable -- in particular on Skylake as recent benchmarks have shown -- likely due to:
- mdrun doing single-threaded planning and leaving the rest of the hwthreads/cores of the rank empty
- always using quick planning

What we could do to improve things:

- use exhaustive planning and recommend it in cases where it gives more reliable results (I've verified with fftw-wisomd that on Skylake it gives consistent and significantly different set of kernels;
- use multi-threaded planning, e.g. we might be able to plan concurrently on all threads of a rank; we could also try to repeat the planning to see if there are inconsistencies

History

#1 - 02/26/2018 03:14 PM - Erik Lindahl

My concern with different plan options is that it would lead to even more complications in the code and normal users now also having to care about the type of planning in the FFTs (in addition to MPI ranks, in addition to OpenMP, in addition to GPU setups, in addition to #ranks per GPU, in addition to possibly tuning PME, etc.). We urgently need to reduce the number of user-exposed such options rather than adding more of them.

We should also work hard keep external FFT libraries external modules, IMHO. If the FFTW planning can be improved with multiple threads, that seems like something we should try to contribute to FFTW instead of implementing in GROMACS. What is the state of things for MKL?

If we are going to spend significant effort, it might be wiser to invest those in our own SIMD-based FFT module.

#2 - 03/15/2018 05:01 PM - Szilárd Páll

Erik Lindahl wrote:

> My concern with different plan options is that it would lead to even more complications in the code and normal users now also having to care about the type of planning in the FFTs (in addition to MPI ranks, in addition to OpenMP, in addition to GPU setups, in addition to #ranks per GPU, in addition to possibly tuning PME, etc.). We urgently need to reduce the number of user-exposed such options rather than adding more of them.

We've just added a massive and complex option for manual task mapping, so I don't think we should worry about something that's for now posed as simply a question about outdated assumptions of the code, no option has been proposed.

> We should also work hard keep external FFT libraries external modules, IMHO. If the FFTW planning can be improved with multiple threads, that seems like something we should try to contribute to FFTW instead of implementing in GROMACS.

We use the serial FFTW (so again: the bug/misuse is the GROMACS) so there is nothing to contribute to FFTW -- unless we are willing take the hit of switching to using the multi-threaded FFTW (which is slower AFAIK). Additionally, somebody would likely have to put effort into reviving/reimplementing it.

> What is the state of things for MKL?

I don't know, but we use serial MKL, so I suspect similar issues arise with MKL too.

> If we are going to spend significant effort, it might be wiser to invest those in our own SIMD-based FFT module.

I agree, but I don't current resources do not permit such development.

However, neither of the proposed tests (especially testing exhaustive planning) should take significant effort.