GROMACS - Task #3189

Task # 2792 (New): Improvement of PME gather and spread CUDA kernels

implement heuristics for switching between different spread/gather kernel layouts

10/30/2019 01:29 PM - Szilárd Páll

Status:	In Progress	
Priority:	High	
Assignee:	Jonathan Vincent	
Category:	mdrun	
Target version:	2021-infrastructure-stable	
Difficulty:	uncategorized	

Description

Based on benchmarking data, we need to implement some heuristics that allow switching to the best performing kernel setup.
Cases to consider:
TODO

The task depends on #3188 which will shift the crossover between 4 / 16 threads/atom.

Associated revisions

Revision d5072ffa - 12/02/2019 09:01 PM - Jonathan Vincent

Heuristics for switching between CUDA spread/gather kernels

The various CUDA spread/gather kernels perform better in different circumstances,
so heuristics are used to control which one is selected.

Refs #3189

Change-Id: I32c0726021a48dc8721e337f8f41e9c9d334e05c

Revision e08e6a6f - 02/28/2020 08:19 AM - Jonathan Vincent

Moves call to choose best performaing kernels to pme_gpu_reinit_atoms

atoms.nAtoms is zero when pme_gpu_reinit is called, resulting in
the useOrderThreadsPerAtom and recalculateSplines always being set to false.

Refs #3189

Change-Id: I36d4be71565cfe8cd8e50fbe6cfe1035f3e15c8e

History

#1 - 11/06/2019 04:27 PM - Jonathan Vincent

Ok did an initial version of this, where we are just keying off the number of atoms https://gerrit.gromacs.org/#/c/gromacs/+/14147/

My feeling is that potentially we could change optimization if we have a decomposition and the number of local atoms changed, which is why I put it in reint_atoms. Maybe there is a better place.

If we want to use GPU information then we need to get it to there which I am looking at now. We should have the GPU information available at least even if we do not use it in the first implementation.

Not sure how complicated we want to go right now with the heuristics. My preference would be to start with something simple and try to refine it later given the time constraints. For sizes above about 20,000 atoms on the water boxes the general best case is 4 threads per atom with recalculate splines. The smaller sizes is more complicated for sure.

Just did this as a stand alone patch to start with, so we can discuss by itself. Expanding it to cover the PME_PARALLEL_SPLINE stuff is relatively simple once we agree on the way to control everything and where the routines should be placed in the call tree.

#2 - 11/15/2019 06:51 PM - Szilárd Páll

Jonathan Vincent wrote:

> Ok did an initial version of this, where we are just keying off the number of atoms https://gerrit.gromacs.org/#/c/gromacs/+/14147/

> My feeling is that potentially we could change optimization if we have a decomposition and the number of local atoms changed, which is why I put it in reint_atoms. Maybe there is a better place.

Perhaps this should be based on grid size -- hence the heuristic may change with PME tuning (as it changes the cutoff and grid spacing)?

> If we want to use GPU information then we need to get it to there which I am looking at now. We should have the GPU information available at least even if we do not use it in the first implementation.

Sure, that can be done in the follow-up, just need to pass the GPU detection info which has the compute capability.

> Not sure how complicated we want to go right now with the heuristics. My preference would be to start with something simple and try to refine it later given the time constraints. For sizes above about 20,000 atoms on the water boxes the general best case is 4 threads per atom with recalculate splines. The smaller sizes is more complicated for sure.

I suggest to add first a "dummy" heuristic with the current default.

We need the parallel spline kernels benchmarked before we can tweak the crossover points.

Also, we may want use some additional systems with more sizes between 5-50k (if nothing to avoid the current grid shape bias).

#3 - 11/16/2019 10:03 AM - Szilárd Páll

Szilárd Páll wrote:

> Jonathan Vincent wrote:
>
> > Ok did an initial version of this, where we are just keying off the number of atoms https://gerrit.gromacs.org/#/c/gromacs/+/14147/
> >
> > My feeling is that potentially we could change optimization if we have a decomposition and the number of local atoms changed, which is why I put it in reint_atoms. Maybe there is a better place.
>
> Perhaps this should be based on grid size -- hence the heuristic may change with PME tuning (as it changes the cutoff and grid spacing)?

Correction: actually, the number of atoms is more relevant, though heuristics might still be slightly affected by significant cutoff/grid scaling.

#4 - 11/16/2019 08:50 PM - Jonathan Vincent

Szilárd Páll wrote:

> Correction: actually, the number of atoms is more relevant, though heuristics might still be slightly affected by significant cutoff/grid scaling.

Ok, that is what I thought as well. But clearly you know the code better than me.

That implies to me that reinit_atoms is the correct place to do the update?

The other question is how we make sure everything is turned off for OpenCL. The only way I could think of doing it was using the pre-processor. You did highlight this as something that is best avoided, but then we need a better way. I guess something could be done with function pointers in a similar way to the different kernels are handled, but I am not sure it helps. Anyway if you have an idea of a better way let me know.

#5 - 12/02/2019 01:15 PM - Paul Bauer

- Target version changed from 2020-beta3 to 2021-infrastructure-stable

not going to be in 2020

#6 - 12/04/2019 12:20 AM - Szilárd Páll

- Status changed from New to In Progress

- Target version changed from 2021-infrastructure-stable to 2020-beta3

Paul Bauer wrote:

> not going to be in 2020

already merged, though uncertain about whether the current version is final.

#7 - 12/04/2019 10:47 AM - Szilárd Páll

- Target version changed from 2020-beta3 to 2020-rc1

bumped because it requires more work

#8 - 12/20/2019 08:32 AM - Paul Bauer

- Target version changed from 2020-rc1 to 2021-infrastructure-stable

@Szilard I bumped the remaining work to 2021

#9 - 01/29/2020 04:18 PM - Szilárd Páll

What are the plans for this? Having better tuned spread/gather will become even more important with PME decomp, so I think we should plan to get these heuristics reassessed and improved.