GROMACS - Task #2216

GROMACS SIMD acceleration: generation 3

07/17/2017 06:57 PM - Erik Lindahl

Status:	New
Priority:	Normal
Assignee:	Erik Lindahl
Category:	core library
Target version:	future
Difficulty:	hard

Description

The architecture-agnostic SIMD interface works quite well, but there are a number of challenges we should start discussing for the future:

1. GROMACS has a nice reputation for squeezing out any performance available on all hardware, which we should try to maintain. However, as all SIMD instructions sets grow (and diverge) in size, features, or width, there is an inherent conflict between wanting to exploit each architecture fully and maintaining a small SIMD interface that doesn't lead to separate code paths.

2. For the foreseeable future, I expect x86 to remain by far the most important CPU architecture, and CUDA to similarly be the most important GPU architecture. Just as we will likely move to a CUDA-specific code path so we can execute entirely on GPUs, should we create an architecture where we can allow more things for specific CPU architectures - for instance for the Nonbonded & PME kernels - while we restrict things to a **much** smaller SIMD interface (smaller than our current one) in the rest of the code?

3. The new ARM Scalable Vector Extensions will make things even more complicated, since you won't know the SIMD width at compile time.... I'm not even sure we can handle that at all in our current SIMD layer.

4. For Skylake things will be really complex since only the highest-end CPUs can do two AVX512 FMAs per cycle, while the cheaper CPUs can do one AVX512 or two AVX256 FMAs. As Szilard has mentioned, we will likely prefer to stick to AVX256 for GPU runs. I also suspect many login nodes (where we compile...) will only have a lower-end CPU, which will screw up our auto-detection. I think this is just another indication that it's time to plan for a future where we have multiple SIMD instruction sets compiled and choose either a full library or modules (say, nonbonded, PME, and integration) based on benchmarks at the start of a run...

5. There are also good things with all modern x86 hardware now supporting FMA and AVX2 (and no other architecture is that diverse). Maybe we could restrict the separate SIMD paths to 3-4 key modules so we can have it all in a single binary?

Related issues:

Related to GROMACS - Task #2221: Avoid preprocessor for SIMD functions	**New**

History

#1 - 08/02/2017 01:41 AM - Roland Schulz

- Related to Task #2221: Avoid preprocessor for SIMD functions added