GROMACS - Feature #1187
Optimized Verlet SIMD kernels for native MIC (Xeon Phi co-processor)
03/12/2013 10:08 AM - Mikhail Plotnikov

| Status:     | Closed                      |
| Priority:   | Normal                      |
| Assignee:   | Roland Schulz              |
| Category:   | mdrun                       |
| Target version: | uncategorized          |

**Description**

As MIC supports native (MPI ranks only on MIC) and symmetric (MPI ranks distributed between HOST and MIC) execution there is a need to implement optimized Verlet SIMD kernels for MIC. This should be done on intrinsics. I have already single precision version of it, but it is based on the beta1 code. Currently, Berk is doing clean up of SIMD kernels to remove all x86 specifics. After this work is properly done there shouldn’t be much trouble to enable any new arch by defining HW-specific intrinsics in gmx_simd_macros.h. Building occurs on Host, so it is reasonable to configure and build 2 binaries – for Host and for MIC, if there is a related Cmake option defined, e.g. --DGXM_MIC_NATIVE=ON. For building native MIC binary compiler flag "-mmic" is required instead of "-mavx".

**Related issues:**
Related to GROMACS - Feature #1181: Implementing asymmetric offload to MIC (X... Closed
Related to GROMACS - Feature #1394: 16-wide MIC SIMD bonded interaction kernels Closed
Related to GROMACS - Feature #1420: Fast force reduction for large number of ... Closed

**Associated revisions**
Revision d28336a2 - 01/09/2014 10:12 AM - Roland Schulz
Add SIMD support for Intel MIC
Only single precision is supported so far.

To compile in native mode:
CFLAGS="-mmic" CXXFLAGS="-mmic" cmake

Regression tests pass with ICC 14.

Instructions are not updated because only offload mode (#1181) is useful to the user.

Part of #1187
Change-Id: I81a2022cfcecf634fdfaff5ce63ad82f0a5d4dee

**History**

#1 - 03/22/2013 03:58 PM - Mark Abraham
Sounds good. We're still adding various acceleration paths for the 4.6.x series, so we'll be happy to consider this code also. The impetus for separating the code into generalized and specialized layers is very valuable.

#2 - 03/24/2013 10:53 AM - Mikhail Plotnikov
Could you please give an approximation for timelines when this feature will be implemented?

#3 - 03/24/2013 06:04 PM - Mark Abraham
I'm not quite sure what you mean, Mikhail. Berk has done https://gerrit.gromacs.org/#/c/2180/ trying to set the stage for this and any similar development. There's no plans from the core GROMACS team to implement the intrinsics for MIC into gmx_simd_macros.h, or write MIC kernels - our priorities are elsewhere (and AFAIK we don't have the hardware). But, if there is enthusiasm from you (or others) for writing these kernels, then we are prepared to facilitate.

Would you like me to prepare a stub commit based on the release-4-6 branch that would set up the build and execution context? I know almost nothing about MIC, but from the above description it seems like we would treat MIC somewhat like we do now for GPUs. The Verlet kernels would run on MIC and if there was FFT work for PME, that would be done on the CPU. IIRC we can shift some Verlet kernel load off the CPU to the GPU if the CPU is under-utilized (but I'd have to check that one). If so, we might want the same for MIC?

If so, I can do that in the next few days. I'd plan to upload a draft to gerrit and anybody interested can use that as a base for kernel development.
Mikhail, the change which did reorganization of the NxN kernels was merged and the bonded force computation is also fully SIMD-ized now. While there are plans (as well as a WIP change in gerrit) to split up the NxN kernels and auto generate the source, I think the code is stable enough for the MIC porting (IMO setting up the auto-generation can be done later).

Is there anything we can do to facilitate the MIC kernel porting? What do you think about Mark's suggestions?

Szilárd, as the code has been stabilized I'm going to implement MIC kernels based on the current release-4-6 branch and a fix
https://gerrit.gromacs.org/#/c/2326/26
It would be great to have a stub commit to a separate branch with this code for kernel development as Mark suggested earlier.

First part has been submitted: https://gerrit.gromacs.org/#/c/2822

Remaining issues:
- add prefetch instructions
- double precision support

One could also explorer whether a 4x4 (with swizzle) or 4x4x4 is faster, then the 2x4x8 currently implemented.

I think whatever we might do has been done.

The prefetch it is still open in gerrit. Also we are still working on the 4x4x4. But I guess neither has to be tracked as this issue.

- Target version deleted (5.x)