GROMACS - Feature #1666

new approach for Verlet-scheme kernel generation

12/21/2014 02:00 PM - Mark Abraham

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

Description

This problem has been on the table for a while. We want to get rid of the use of the preprocessor for kernel meta-programming, because it's ugly to read, bug-prone to change, and there are no components amenable to unit tests. (That the kernels are too big to unit test is its own problem.)

Current plan is to evaluate and test the ability of compilers to do all the good constant propagation and inlining that would do the right thing with something like

```
// file_that_calls_kernels.cpp
tableOfFunctionPointers = { ... kernelWithFullName, ... };

// actual_kernel_file.cpp
#include kernel_template.h
void kernelWithFullName(args)
{
   kernel<theElecType, theVdwType, bNeedEnergy, etc.>(args);
}

// kernel_template.h
template <const bool bNeedEnergy>
void calcEwald(real qq, real rsq, real rinv, real gmx_restrict *V, rvec gmx_restrict *f)
{
    // compute forces
    ...
    if (bNeedEnergy)
    {
        // compute energy
    }
    ...
}

template <const int elecType, const int vdwType, const bool bNeedEnergy, etc.>
void kernel(args)
{
    ...
    switch(elecType)
    {
        case elecEwald : calcEwald<bNeedEnergy>(qq, rsq, rinv, V, f); break;
        ...
    }
    ...
}

// tests/kernel_template.cpp
#include "../kernel_template.h"
TEST_F(calcEwaldWorksWithEnergy)
{
    ...
}
```

Erik plans to have a go at that over Christmas, I understand.

Various types above would be the SIMD forms in those kernels; I suggest we do not attempt templating over the normal-vs-SIMD axis for now (see #1612).

If this doesn't work out on decently modern compilers, then we might fall back on some python scripts somewhat like we do for the group scheme. If this is not great on some niche compiler, then since it'll still be correct, I suggest we use that feedback as leverage on the compiler team rather than work around it. The required functionality seems like simple dead-code elimination after constant propagation, after all...

For the Verlet SIMD kernels, if we can put SIMD variables into an array, so that

```
for(int i = 0; i < unroll; ++i)
{
    calcEwald<bNeedEnergy>(qq[i], rsq[i], rinv[i], V, f);
}
```

generates the same instructions as such regions do now, then that is great. If so, we might be able to template over the unroll dimensions (in subsequent work?). I hope we're not going to need a global constant for declaring the maximum unroll size(s) for those arrays, though.

Related issues:

Related to GROMACS - Task #1211: improve use of preprocessor macros in CUDA k...	**New**	**03/27/2013**
Related to GROMACS - Task #1758: Verlet scheme reorganization / modularization	**New**	**06/28/2015**
Related to GROMACS - Task #1852: Remove group scheme	**New**	
Related to GROMACS - Feature #1665: improve free energy non-bonded kernel per...	**New**	
Related to GROMACS - Feature #1347: future of tables	**New**	

Associated revisions

Revision 65e33d97 - 02/21/2016 10:30 PM - Mark Abraham

Separate table construction

Construction of tables for the group scheme, pair interactions and
dispersion correction are now separated. The resulting tables are
never re-used for something else. This uses slightly more memory, but
makes the logic rather more simple. Some of the tables are now held by
reference by their owners, rather than by value, which might improve
cache locality a little.

With this change, we can implement the table support for the Verlet
scheme without getting involved with the group-scheme code, and will
have an easier time removing the group scheme.

Refs #1666, #1852

Change-Id: I8ca608f0e41b02723e6080b80b04d9e7ff048900

History

#1 - 01/08/2015 06:04 PM - Mark Abraham

- Related to Task #1211: improve use of preprocessor macros in CUDA kernels added

#2 - 04/09/2015 11:26 AM - Mark Abraham

- Target version changed from 5.1 to 5.x

#3 - 07/07/2015 11:17 PM - Roland Schulz

- Related to Task #1758: Verlet scheme reorganization / modularization added

#4 - 02/18/2016 11:16 PM - Gerrit Code Review Bot

Gerrit received a related patchset '8' for Issue #1666.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: I8ca608f0e41b02723e6080b80b04d9e7ff048900
Gerrit URL: https://gerrit.gromacs.org/5132

#5 - 02/26/2016 11:56 AM - Mark Abraham

- Related to Task #1852: Remove group scheme added

#6 - 07/11/2016 08:04 PM - Mark Abraham

- Target version changed from 5.x to 2018

#7 - 04/23/2017 12:07 PM - Mark Abraham

- Related to Feature #1665: improve free energy non-bonded kernel performance added

#8 - 04/23/2017 08:14 PM - Mark Abraham

- Related to Feature #1347: future of tables added

#9 - 06/27/2017 11:12 PM - Mark Abraham

- Target version changed from 2018 to 2019

Traditional annual bump to next year...

#10 - 10/08/2018 07:04 PM - Mark Abraham

- Target version changed from 2019 to future