Hi, I am trying to compile GROMACS with xlc/xlC in the simplest way I can. From the build directory:

cmake3 -DCMAKE_C_COMPILER=xlc -DCMAKE_CXX_COMPILER=xlC ..

Output:

[u0017592@sys-85310 build]$ cmake3 -DCMAKE_C_COMPILER=xlc -DCMAKE_CXX_COMPILER=xlC ..

-- No compatible CUDA toolkit found (v5.0+), disabling native GPU acceleration
CMake Error at cmake/gmxManageSimd.cmake:90 (message):
Cannot find IBM VSX compiler flag. Use a newer compiler, or disable SIMD support (slower).
Call Stack (most recent call first):
cmake/gmxManageSimd.cmake:414 (gmx_give_fatal_error_when_simd_support_not_found)
CMakeLists.txt:665 (gmx_manage_simd)

CMakeError.log & CMakeOutput.log attached.

My xlc version is 13.1.5. Many of the flags checked for in the CMakeError.log are no longer supported.


- -mabi=altivec: Not in any xlc doc. Exists in gcc. Maybe in previous versions this didn't throw errors. "-qaltivec" Enables the compiler support for vector data types and operators. Effects VSX/VMX load & store functions. This might be functionally equivalent to -maltivec, which is supported for gcc compatibility
- -mvsx: not in any doc. See above - can this be replaced by -qaltivec (or -maltivec)?
- -qsupport: Exists in 13.1 but not 13.1.5
- -qhalt=e: The flag exists, but "w" is the only valid option in 13.1.5. In 13.1 more values were allowed, e.g. e, s

The invalid options throw warnings but not errors, so maybe it's not necessary to fail the build if they are raised when compiling with xlc/xlC.

Related issues:
Related to GROMACS - Bug #2103: xlc/xlC: shellfc.cpp: empty array initializer...
Related to GROMACS - Task #2102: decide future of xlc+power support

Associated revisions
Revision deb27c17 - 02/09/2017 04:45 PM - Erik Lindahl
Work around false xlc-13.1.5 bug in SIMD test
atan2(0,0) should return 0.0, which the Gromacs simd implementation does. However, since at least one compiler produces -nan for the standard library version it's better to compare with the known correct value rather than calling std:atan2(0,0).

Refs #2102.
Change-Id: ibo449e9f16fb1ab79486927a3e9993da0c3e937f

Revision f05743b0 - 03/04/2017 12:25 PM - Mark Abraham
Fixes for xlc compilers 13.1.5 on Power8

Applied Erik's recent fix also for the atan2SinglePrecisionAccuracy test.

Refs #2102

Change-Id: i230dac8084be2d0693cb616b5a5951b0ae4b71a6

Revision a590e91 - 12/08/2017 11:58 AM - Paul Bauer
Add information regarding xlc compiler
Added information concerning that the xlc compiler is neither supported nor tested.

Refs #2102
Change-Id: i1963a2fdaae627f4d9521c2808f1c17eabe97

History

#1 - 01/27/2017 02:57 PM - Szilárd Páll
- Status changed from New to Accepted

Our build system seems to
i) lack xlc robust (due to the lack of hw/sw access and lack of maintainers)
ii) have no chance in keeping up with changes in compiler flags given the changes you claim between patch versions.

Given that we have no access and little incentive (until xlc proves to be superior than gcc), I'm not sure we can commit much resources to this issue. However, contributions would be welcome and we can certainly provide guidance!

If you'd test/contribute the xlc flags, consider adding the general flags not directly related to SIMD support in prepare_power_vsx_toolchain(). SIMD flags required/useful for VMX are tested next here.

#2 - 01/27/2017 05:20 PM - Mark Abraham
In the last fortnight, I finally got access to such a compiler (on JURON at Juelich). I can reproduce something like James report. (Previously the only interesting platform for xlc was BlueGene, and that was locked on the xlc 12 major version.)

I assume that xlc is likely slower than gcc on any platform where both it and gcc are supported, and I'm sure gcc was the only compiler used when Erik implemented the SIMD layer for VMX and VSX. But I can look into making it work with xlc, at least.

#3 - 01/27/2017 05:56 PM - Mark Abraham
GROMACS deliberately checks for a range of flags that might activate SIMD, so that multiple compilers can be supported with minimal fuss on our end. So we expect a bunch of the tests to fail in most cases.

The unrecognized suppression flags don't seem to be a serious problem. They were working around bugs in previous versions of the compiler...

The `-qhalt=e` compiler tolerates `-qhalt=e` with only a warning, so the GROMACS check for `-qarch=auto -qaltivec` succeeds there. The matching C compiler does not tolerate `-qhalt=e`, which seems like a behaviour worth reporting, if someone knows where to file that. We no longer require the C toolchain to support SIMD on Power machines, so the GROMACS build system can be hacked to get around that (remove the check related to SIMD$[GMX_SIMD]_C_flags on line 421 of cmake/gmxManageSimd.cmake).

The `-qhalt=e` is generated by CMake (even for its master branch), so until someone proposes a fix for that, xlc 13.1.5 will not be usable for any C application that uses CMake. However, I agree with Szilard that such changes in point release versions make attempting to support this compiler seem more trouble that it is worth.

There are other bugs in our implementation of our SIMD layer for xlc, so I will attempt to fix those and see what may be a reasonable way forward.

#4 - 01/27/2017 07:08 PM - Mark Abraham
A large bunch of our unit tests fail, even after the above mentioned fixes, so xlc 13 isn't supported on POWER* until we have time to get to the bottom of the issues (and I'm making no promises, this compiler is not a high priority for us).

I suggest trying gcc on these platforms - we'll likely fix any issue with that quickly.

#5 - 01/27/2017 07:08 PM - Mark Abraham
- Related to Bug #2103: xlc/xlc: shellfc.cpp: empty array initializer "expression not supported" added

#6 - 01/31/2017 04:57 PM - James Ostrander
Would you accept pull requests for fixes for this group of compilers? You mentioned that there was further work to do even after making the above changes. Are there discrete chunks that work could be broken into so that I could assign someone else to work on those tasks?

#7 - 01/31/2017 05:03 PM - James Ostrander
BTW: https://gitlab.kitware.com/cmake/cmake/issues/16605
Thanks for the tip!

#8 - 01/31/2017 09:42 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #2102.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~release-2016~I230dac8084be2d0693cb616b5a5951b0ae4b71a6
Gerrit URL: https://gerrit.gromacs.org/6444

#9 - 01/31/2017 09:44 PM - Mark Abraham
James Ostrander wrote:

Would you accept pull requests for fixes for this group of compilers?

Very likely, yes.

You mentioned that there was further work to do even after making the above changes. Are there discrete chunks that work could be broken into so that I could assign someone else to work on those tasks?

I've got most things in working shape. There are some TODOs in the patch for our 2016 branch that I've just uploaded.

The test binaries tend to segfault from a double free after completing the tests, which is irritating but not actually a problem.

#10 - 02/07/2017 03:14 PM - Erik Lindahl
For the record, we actually did test xlc extensively when developing SIMD code - both on big and little-endian machines.

However, the was >1 year ago, and compared to gcc the xIC compiler is not stellar. There were several SIMD-related bugs, and performance was clearly lower than for gcc. Even the IBM people I interacted with seemed to prioritize gcc at least as much (if not more) when it comes to VSX and general optimization for Power8.

Previous versions of xlc also had an issue where our unit tests failed because there was a sign error in the reference value for tan(0) provided by the compiler.

#11 - 02/07/2017 03:26 PM - Erik Lindahl
Update, mostly for reference. When compiling a trivial program (which properly includes cmath) that prints tan(0) we get the following with gcc-4.8.5 and xlc-13.1.5:

```
gcc -O2 0
gcc -O3 0
xlc -O2 0
xlc -O3 -0
```

According to the tan() man page, the correct value is 0.

#12 - 02/07/2017 03:42 PM - Mark Abraham
Output for the atan2 tests:

```
[02/20/2020 3/9]
```
The reference value comes from std::atan2(+0, +0), and should be +0, but perhaps is getting turned into -nan from a buggy implementation of std::tan. Regardless, our SIMD result is correct, so we should probably disable this test for this compiler (family).

#13 - 02/08/2017 03:10 AM - Mark Abraham

There are further issues trying to use pthreads mutexes for thread MPI, e.g. the following (with or without CUDA in the build, and also in release mode)

$ (cd build-cmake-xlc-gpu-debug; ninja && gdb bin/mdrun-test)
[1/30] Generating git version information
GNU gdb (GDB) 7.11.1.20160801-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "powerpc64le-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
Find the GDB manual and other documentation resources online at:
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from bin/mdrun-test...done.
(gdb) run
[==========] Running 25 tests from 9 test cases.
[----------] Global test environment set-up.
[----------] 6 tests from BondedInteractionsTest
[ RUN     ] BondedInteractionsTest.NormalBondWorks
NOTE 1 [file /gpfs/homeb/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/src/programs/mdrun/tests/Testing/Temporary/BondedInteractionsTest_NormalBondWorks_input.mdp, line 1]:
/gpfs/homeb/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/src/programs/mdrun/tests/Testing/Temporary/BondedInteractionsTest_NormalBondWorks_input.mdp did not specify a value for .mdp option "cutoff-scheme". Prob
alty it was first intended for use with GROMACS before 4.6. In 4.6, the Verlet scheme was introduced, but the group scheme was still the default. The default is now the Verlet scheme, so you will observe different behaviour.

Program received signal SIGSEGV, Segmentation fault.
__GI___pthread_mutex_lock (mutex=0x80) at ../nptl/pthread_mutex_lock.c:68
68 ../nptl/pthread_mutex_lock.c: No such file or directory.
(gdb) bt
#0 __GI___pthread_mutex_lock (mutex=0x80) at ../nptl/pthread_mutex_lock.c:68
#1 0x000003fffb62d1fd0 in tMPI_Thread_mutex_lock ()
    from /gpfs/home/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/lib/libgromacs.so.2
#2 0x000003fffb58f702c in gmx_fio_open ()
    from /gpfs/home/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/lib/libgromacs.so.2
#3 0x000003fffb58f75c0 in gmx_fio_fopen ()
    from /gpfs/home/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/lib/libgromacs.so.2
#4 0x000003fffb591709c in write_inpfile(char const*, int, t_inpfile*, int, warninp*) ()
    from /gpfs/home/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/lib/libgromacs.so.2
#5 0x000003fffb5c25c24 in get_ir(char const*, char const*, t_inputrec*, t_gromppopts*, warninp*) ()
    from /gpfs/home/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/lib/libgromacs.so.2
#6 0x000003fffb5be73c0 in gmx_grompp(int, char**) ()
    from /gpfs/home/padc/padc010/git/r2016/build-cmake-xlc-gpu-debug/lib/libgromacs.so.2
#7 0x00000000010052ae4 in gmx::test::SimulationRunner::callGromppOnThisRank(gmx::test::CommandLine const&) ()
#8 0x00000000010052c58 in gmx::test::SimulationRunner::callGrompp() ()
#9 0x00000000010075c0 in gmx::test::BondedInteractionsTest_NormalBondWorks_Test::TestBody ()
#10 0x000000000100708c in void testing::internal::HandleExceptionsInMethodIfSupported<testing::Test, void>(testing::Test*, void (testing::Test::*)(), char const*) ()
#11 0x000000000100708c in testing::Test::Run() ()
#12 0x000000000100708c in testing::TestInfo::Run() ()
#13 0x000000000100708c in testing::TestCase::Run() ()
#14 0x000000000100708c in testing::internal::UnitTestImpl::RunAllTests ()
#15 0x000000000100708c in bool testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::UnitTestImpl, bool>(testing::internal::UnitTestImpl*, bool (testing::internal::UnitTestImpl::*)(), char const*) ()
#16 0x000000000100708c in testing::UnitTest::Run() ()
#17 0x000000000100708c in main ()

Given that the code works on POWER with gcc, and on every other platform we know of that supports pthreads, it seems likely there is an issue related to xlc. Any ideas, James? Can you reproduce this?

#14 - 02/08/2017 09:43 PM - James Ostrander

Thanks for the updates, Mark. I was able to reproduce that issue. Here is my output without gdb (I tried release mode)

[u0017592@sys-85390 build-mpi]$ ./bin/mdrun-test

[==========] Running 24 tests from 9 test cases.
[----------] Global test environment set-up.
[----------] 6 tests from BondedInteractionsTest
[ RUN ] BondedInteractionsTest.NormalBondWorks

NOTE I [file /home/u0017592/projects/gromacs/build-mpi/src/programs/mdrun/tests/Testing/Temporary/BondedInteractionsTest_NormalBondWorks_input.mdp, line 1]:
/home/u0017592/projects/gromacs/build-mpi/src/programs/mdrun/tests/Testing/Temporary/BondedInteractionsTest_NormalBondWorks_input.mdp did not specify a value for the .mdp option "cutoff-scheme". Probably it was first intended for use with GROMACS before 4.6. In 4.6, the Verlet scheme was introduced, but the group scheme was still the default. The default is now the Verlet scheme, so you will observe different behaviour.

[sys-85390:01731] *** Process received signal ***
[sys-85390:01731] Signal: Segmentation fault (11)
[sys-85390:01731] Signal code: Address not mapped (1)
[sys-85390:01731] Failing at address: 0x90
[sys-85390:01731] [ 0] [0x3fffbea180478]
[sys-85390:01731] [ 1] [0x3fffbea180480]
[sys-85390:01731] [ 2] /home/u0017592/projects/gromacs/build-mpi/lib/libgromacs.so.2(tMPI_Thread_mutex_lock+0x40c)
[sys-85390:01731] [ 3] /home/u0017592/projects/gromacs/build-mpi/lib/libgromacs.so.2(gmx_fio_open+0x1cc)
[sys-85390:01731] [ 4] /home/u0017592/projects/gromacs/build-mpi/lib/libgromacs.so.2(gmx_fio_fopen+0x20)
[sys-85390:01731] [ 5] /home/u0017592/projects/gromacs/build-mpi/lib/libgromacs.so.2(_Z13write_inpfilePKcIpirftmpfile_iEiPiPKt18gromppoptionsPKt19warninpobj+0x3dc)
[sys-85390:01731] [ 6] /home/u0017592/projects/gromacs/build-mpi/lib/libgromacs.so.2(_Z6get_irPKcS0_P10t_inputrecP12t_gromppoptsP7warninp+0x358c)
[sys-85390:01731] [ 7] /home/u0017592/projects/gromacs/build-mpi/lib/libgromacs.so.2(_Z10gmx_gromppiPPc+0x40c)

02/20/2020
05/09
Segmentation fault

I configured with:

```
[u0017592@sys-85390 build-mpi]$ cmake3 -j 2 -DREGRESSIONTEST_DOWNLOAD=ON -DCMAKE_BUILD_TYPE=Release -DCMAKE_COMPILER=mpicc -DCMAKE_CXX_COMPILER=mpicxx -DGMX_MPI=ON ..
```

#15 - 02/08/2017 11:53 PM - Mark Abraham
OK thanks, James. If you know where to lodge an effective bug report, then it should be easy for us to isolate repro cases for these pthread_mutex_lock and tan (and maybe atan2) bugs. Do you?

#16 - 02/09/2017 02:44 PM - James Ostrander
If you can add your repro cases here, I'll gladly see to it that they're forwarded along to the XL team.

#17 - 02/09/2017 03:46 PM - Gerrit Code Review Bot
Gerrit received a related patchset `1` for Issue #2102.
Uploader: Erik Lindahl (erik.lindahl@gmail.com)
Change-Id: gromacs~master~I60449e9f16fb1ab79486927a3e9993da0ce937f
Gerrit URL: https://gerrit.gromacs.org/6465

#18 - 02/09/2017 03:48 PM - Gerrit Code Review Bot
Gerrit received a related patchset `1` for Issue #2102.
Uploader: Erik Lindahl (erik.lindahl@gmail.com)
Change-Id: gromacs~release-2016~I60449e9f16fb1ab79486927a3e9993da0ce937f
Gerrit URL: https://gerrit.gromacs.org/6466

#19 - 02/10/2017 04:24 PM - Mark Abraham
I'm in the process of testing those changes now. I get the following when compiling with GCC+MPI (same result Release vs. Debug build type):

```
[u0017592@sys-85390 build-gcc]$ make
Scanning dependencies of target git-version-info
[ 0%] Generating git version information
[ 0%] Built target git-version-info
[ 0%] Generating baseversion-gen.c
Scanning dependencies of target libgromacs
[ 0%] Building CXX object src/gromacs/CMakeFiles/Libgromacs.dir/listed-forces/bonded.cpp.o
In file included from /home/u0017592/projects/gromacs/src/gromacs/simd/impl_ibm_vsx/impl_ibm_vsx.h:46:0,
 from /home/u0017592/projects/gromacs/src/gromacs/simd/simd.h:123,
 from /home/u0017592/projects/gromacs/src/gromacs/simd/impl_ibm_vsx/impl_ibm_vsx_simd_double.h:157,
```

#20 - 02/10/2017 08:04 PM - James Ostrander
I'm the process of testing those changes now. I get the following when compiling with GCC+MPI (same result Release vs. Debug build type):
m[0] = vec_extract(x.simdInternal_, 0);

/home/u0017592/projects/gromacs/src/gromacs/simd/impl_ibm_vsx/impl_ibm_vsx_simd_double.h:157: confused by earlier errors, bailing out

Preprocessed source stored into /tmp/ccXvhzo0.out file, please attach this to your bug report.

make[2]: *** [src/gromacs/MakeFiles/libgromacs.dir/listed-forces/bonded.cpp.o] Error 1
make[1]: *** [src/gromacs/MakeFiles/libgromacs.dir/all] Error 2
make: *** [all] Error 2

I won't be able to test GPU yet, since the VMs I'm currently working from don't have hardware acceleration.

#21 - 02/10/2017 08:52 PM - James Ostrander

With XLC+MPI in release mode, it failed to generate shell completions:

```plaintext
[100%] Building CXX object src/programs/CMakeFiles/gmx.dir/legacymodules.cpp.o
warning: 1540-5203 Unrecognized value "e" specified with option "halt".
1 warning generated.

[100%] Linking CXX executable ../../bin/gmx_mpi

[100%] Built target gmx

Scanning dependencies of target gmx

[100%] Generating command-line completions for programs

[[sys-85390:02528] ] *** Process received signal ***

[[sys-85390:02528] ] [Signal: Segmentation fault (11)]

[[sys-85390:02528] ] [Paging at address: 0x500000014]

[[sys-85390:02528] ] [0x3fff8617fa10]

[[sys-85390:02528] ] [1] /home/u0017592/projects/gromacs/build-release/lib/libgromacs_mpi.so.2(_ZN3gmx7CpuInfo16s_vendorStrings_E+0x0)

[[sys-85390:02528] ] [2] /home/u0017592/projects/gromacs/build-release/lib/libgromacs_mpi.so.2(tMPI_Thread_mutex_destroy+0x28)

[[sys-85390:02528] ] [3] /home/u0017592/projects/gromacs/build-release/lib/libgromacs_mpi.so.2(_ZN4tMPI5mutexD2E+0x4c)


[[sys-85390:02528] ] [5] /lib64/libc.so.6(__cxa_finalize+0x10c)

[[sys-85390:02528] ] [6] /lib64/libc.so.6(+0x44000)

Passed - End of error message ***

Failed to generate shell completions, will build GROMACS without. Set GMX_BUILD_HELP=OFF if you want to skip this notification and warnings during installation.

[100%] Built target completion

Scanning dependencies of target mdrun_test_objlib

[100%] Building CXX object src/programs/mdrun/tests/CMakeFiles/mdrun_test_objlib.dir/mdruncomparisonfixture.cpp.o
warning: 1540-5203 Unrecognized value "e" specified with option "halt".
1 warning generated.

[100%] Building CXX object src/programs/mdrun/tests/CMakeFiles/mdrun_test_objlib.dir/moduletest.cpp.o
warning: 1540-5203 Unrecognized value "e" specified with option "halt".
1 warning generated.

[100%] Building CXX object src/programs/mdrun/tests/CMakeFiles/mdrun_test_objlib.dir/terminationhelper.cpp.o
warning: 1540-5203 Unrecognized value "e" specified with option "halt".
1 warning generated.

[100%] Built target mdrun_test_objlib

#22 - 02/10/2017 08:54 PM - James Ostrander

The GCC error I mentioned in #20 occurs regardless of whether MPI is used.

#23 - 02/16/2017 02:42 PM - Mark Abraham

James Ostrander wrote:

I'm in the process of testing those changes now. I get the following when compiling with GCC+MPI (same result Release vs. Debug build type):

```
Yes, I get that also (in a non-MPI build, but that fact doesn't matter for this) for gcc 4.8.5, but gcc 5.3.1 in at9.0 is fine. From our git history, it looks like the testing on Power8 was done with gcc 4.9, so if you're also using 4.8.5 then we can just require 4.9 and move on.
```

#24 - 02/16/2017 02:48 PM - Mark Abraham
James Ostrander wrote:

With XLC+MPI in release mode, it failed to generate shell completions:

[...]

Yes, this is running one of the GROMACS binaries to generate user-facing conveniences that are guaranteed to match what the code can do. If you had a deeper backtrace, I expect it would look similar to those of comments 13 and 14. GROMACS requires that standard pthread mutexes work. Clearly they can be made to work, else you couldn't have a conformant C++11 xlC compiler with std::mutex, but something about the toolchain setup isn't working correctly yet.

#25 - 02/16/2017 03:31 PM - James Ostrander

You're right, I'm running gcc 4.8.5, which is the latest rpm for RHEL 7.3. I'll try on Ubuntu ppc64le, where I have a more up to date gcc available.

RE: The pthread issue, is there a simple and isolated way to reproduce the problem, or should I send that along to the XL team with instructions to reproduce it with a GROMACS build?

#26 - 02/16/2017 06:41 PM - Mark Abraham

James Ostrander wrote:

You're right, I'm running gcc 4.8.5, which is the latest rpm for RHEL 7.3. I'll try on Ubuntu ppc64le, where I have a more up to date gcc available.

OK, I'll patch release 2016 to require gcc 4.9.

RE: The pthread issue, is there a simple and isolated way to reproduce the problem, or should I send that along to the XL team with instructions to reproduce it with a GROMACS build?

I still suspect it is a toolchain configuration issue, e.g. the xlC binary is getting the pthread implementation from at9 with obvious potential for disaster. I have a request open at JURON to explore this, to which other IBM people have contributed (e.g. [https://trac.version.fz-juelich.de/hbp-ppc/ticket/30](https://trac.version.fz-juelich.de/hbp-ppc/ticket/30)) if that's publicly visible. We have a cmake-time compile tests for pthreads functionality which must have passed for both of us. But it does not check that linking works, which suggests that the problem does not arise until linking a multi-source binary. I'll see what they say and consider making a repro case. But really this should be IBM's job to make work, even in pre-production access :-)

#27 - 02/17/2017 02:57 PM - James Ostrander

On the subject of XLC causing printf(tan(0)) to return -0 at -O3, -O3 in XLC enables -qnostrict, which apparently relaxes the restrictions enough to change the sign of the zero.

#28 - 03/29/2017 10:57 AM - Mark Abraham

Nobody knows how to get a reliable xlC toolchain working at the only site where I have access to xlc+power, so for now that combination is not supported.

#29 - 12/03/2017 11:55 AM - Mark Abraham

- Tracker changed from Bug to Task
- Subject changed from xlc/xlC: cmake fails due to invalid flags to decide future of xlc+power support
- Target version set to 2018
- Affected version deleted (git master)

Decide how to document that xlC is not tested / not supported on POWER architectures in GROMACS 2018.

My suggestion is to say it is unsupported, recommend gcc, and leave it open to anybody who wants to make it work to show that the performance of xlC over e.g. gcc is worth the work.

#30 - 12/08/2017 11:59 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2102.
Uploader: Paul Bauer (paul.bauer.q@gmail.com)
Change-Id: gromacs~release-2018~i1f963a2fd6e274df9521c28088f1c1f1e7eae97
Gerrit URL: [https://gerrit.gromacs.org/7310](https://gerrit.gromacs.org/7310)

#31 - 12/11/2017 04:45 PM - Paul Bauer

As this one has been added now, should the issue be closed or left open if someone wants to step up the support in the future?

#32 - 12/11/2017 04:46 PM - Paul Bauer
#33 - 12/12/2017 08:55 AM - Mark Abraham
- Status changed from Accepted to Closed

Closed for now - if there's reason for someone to do the work, then we're still open to supporting xlc in future.

#34 - 03/21/2018 01:02 PM - Mark Abraham
Note that various gcc+POWER fixes have happened in release-2018 branch, but we don't have any access to a working xlc compiler at the moment.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMakeError.txt</td>
<td>18.9 KB</td>
<td>01/26/2017</td>
<td>James Ostrander</td>
</tr>
<tr>
<td>CMakeOutput.txt</td>
<td>76.2 KB</td>
<td>01/26/2017</td>
<td>James Ostrander</td>
</tr>
</tbody>
</table>