

GROMACS - Bug #1988

Double-precision SIMD test failure on powerpc

06/09/2016 06:24 PM - Nicholas Breen

Status:	Closed	
Priority:	Low	
Assignee:	Mark Abraham	
Category:		
Target version:	2016	
Affected version - extra info:	2016-beta2	Difficulty: uncategorized
Affected version:	2016	

Description

Last beta 2 build issue for now!

This is a power8 PPC machine. These failures are only on the double-precision build. Single-precision passes all tests.

```
-- Detecting best SIMD instructions for this CPU
-- Detected best SIMD instructions for this CPU - IBM_VSX

...

-- Try C compiler IBM VSX SIMD flag = [-mvsx]
-- Performing Test C_FLAG_mvsx
-- Performing Test C_FLAG_mvsx - Success
-- Performing Test C_SIMD_COMPILE_FLAG_mvsx
-- Performing Test C_SIMD_COMPILE_FLAG_mvsx - Success
-- Try C++ compiler IBM VSX SIMD flag = [-mvsx]
-- Performing Test CXX_FLAG_mvsx
-- Performing Test CXX_FLAG_mvsx - Success
-- Performing Test CXX_SIMD_COMPILE_FLAG_mvsx
-- Performing Test CXX_SIMD_COMPILE_FLAG_mvsx - Success
-- Enabling IBM VSX SIMD instructions

...

[ RUN      ] SimdMathTest.log
/«PKGBUILDDIR»/src/gromacs/simd/tests/simd_math.cpp:289: Failure
Failing SIMD math function comparison due to sign differences.
Reference function: std::log
Simd function:      log
Test range is ( 1.00000000000000001e-30 , 1e+30 )
First sign difference around x={ 1e-30, 1e+26 }
Ref values:  { -69.0776, 59.8672 }
SIMD values: { -708.852, -708.833 }

[ FAILED  ] SimdMathTest.log (0 ms)

...

[ RUN      ] SimdMathTest.logSingleAccuracy
/«PKGBUILDDIR»/src/gromacs/simd/tests/simd_math.cpp:564: Failure
Failing SIMD math function comparison due to sign differences.
Reference function: std::log
Simd function:      logSingleAccuracy
Test range is ( 1.00000000000000001e-30 , 1e+30 )
First sign difference around x={ 1e-30, 1e+26 }
Ref values:  { -69.0776, 59.8672 }
SIMD values: { -708.852, -708.833 }

[ FAILED  ] SimdMathTest.logSingleAccuracy (0 ms)
```

Related issues:

Related to GROMACS - Bug #1808: error: 'asm' operand has impossible constrain...	Closed
Related to GROMACS - Bug #1997: big-endian power7 testbits is broken	Closed

Associated revisions

Revision 49b323e3 - 07/04/2016 04:45 PM - Mark Abraham

Fixes for Power7 big-endian

Now compiles and passes all tests in both double and single precision with gcc 4.9.3, 5.4.0 and 6.1.0 for big-endian VSX.

The change for the code in `incrStoreU` and `decrStoreU` addresses an apparent regression in 6.1.0, where the compiler thinks the type returned by `vec_extract` is a pointer-to-float, but my attempts a reduced test case haven't reproduced the issue.

Added some test cases that might hit more endianness cases in future.

We have not been able to test this on little-endian Power8; there is a risk the gcc-specific permutations could be endian-sensitive. We'll test this when we have hardware access, or if somebody runs the tests for us.

Fixes #1997.
Refs #1988.

Change-Id: [Iede0eac22504b22973f1a40d2b0180f10a34b7ed](#)

History

#1 - 06/14/2016 11:30 PM - Szilárd Páll

Did this pass with 5.1.3, I can't find the results? Could this be an endianness issue?

#2 - 06/20/2016 01:54 AM - Nicholas Breen

It did not pass with 5.1.3 either. Build logs: <https://buildd.debian.org/status/fetch.php?pkg=gromacs&arch=powerpc&ver=5.1.2-3&stamp=1461287488>

The most relevant part in 5.1.3 appears to be

```
/«PKGBUILDDIR»/src/gromacs/simd/impl_ibm_vsx/impl_ibm_vsx.h: In member function 'virtual void gmx::test::{anonymous}::SimdFloatingpointTest_gmxSimdGetExponentR_Test::TestBody()':  
/«PKGBUILDDIR»/src/gromacs/simd/impl_ibm_vsx/impl_ibm_vsx.h:452:80: error: 'asm' operand has impossible constraints  
    __asm__ ("xvcvsxwdp %0,%1 : "=ww" (x) : "ww" ((__vector signed int) (ix)));  
                                                                    ^
```

#3 - 06/28/2016 02:04 AM - Mark Abraham

- Status changed from New to In Progress

- Assignee set to Mark Abraham

Thanks for the report. I don't have access to a power8 machine, but I observe these to fail on big-endian power7 in the same way. I was able to track down the fix to an conversion of a 32-bit integer to a double that was not appropriately versioned with endianness. Apparently we didn't test this case when we implemented this code. I imagine the incoming fix may work for big-endian power8 also.

5.1 has a different implementation, and has previously had a similar issue at [#1808](#), but we haven't tried to backport it. Perhaps the most practical approach is to de-support VSX in 5.1, now that 2016 is nearly ready.

#4 - 06/28/2016 02:04 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1988](#).

Uploader: Mark Abraham (mark.j.abraham@gmail.com)

Change-Id: [Iede0eac22504b22973f1a40d2b0180f10a34b7ed](#)

Gerrit URL: <https://gerrit.gromacs.org/5993>

#5 - 06/28/2016 02:04 AM - Mark Abraham

- Related to Bug [#1808](#): error: 'asm' operand has impossible constraints when compiling gromacs 5.1 on PPC64 and PPC64LE with VSX SIMD added

#6 - 06/28/2016 02:07 AM - Mark Abraham

- Related to Bug #1997: big-endian power7 testbits is broken added

#7 - 06/30/2016 08:41 PM - Szilárd Páll

Mark Abraham wrote:

5.1 has a different implementation, and has previously had a similar issue at [#1808](#), but we haven't tried to backport it. Perhaps the most practical approach is to de-support VSX in 5.1, now that 2016 is nearly ready.

I suggest to take a refined a slightly more refined approach to de-supporting, e.g. we could refuse to build with VSX on big endian. Otherwise comparing our progress on Power8 will be hard to assess.

#8 - 06/30/2016 10:54 PM - Mark Abraham

Yes, we don't have to de-support the whole thing. But if we have no access to test some architecture/Endian/OS combination, and there's a problem reported on the bugfix branch, and months go by without someone putting up their hand... that's starting to sound like unsupported. Right now we only have access to Power7 as a side effect of the BG/Q support.

#9 - 07/04/2016 07:02 PM - Mark Abraham

- Status changed from In Progress to Feedback wanted

This may have been fixed by the commit now submitted to release-2016 branch, but we can't test it on Power8 at this time.

#10 - 07/05/2016 12:50 AM - Erik Lindahl

Nicholas: I **might** be able to debug this by proxy if you can test the latest version in git.

#11 - 07/06/2016 03:44 AM - Nicholas Breen

All "make check" tests pass successfully with a release-2016 checkout as of Mark's commit 49b323e on a ppc64 Power8 host, using `-mcpu=power8 -mpower8-vector -mpower8-fusion -mdirect-move -mvsx`. I don't have ready shell access to a 32-bit Power8 system, the earlier log was from a restricted build host, but tests on 32-bit Power7 also pass. (Actually, I should probably force the Debian build to **avoid** `-mcpu=power8`, in favor of a more common baseline.)

#12 - 07/06/2016 02:15 PM - Mark Abraham

- Status changed from Feedback wanted to Resolved

Excellent, sounds like we've got things working for now.

#13 - 07/07/2016 08:38 PM - Erik Lindahl

- Status changed from Resolved to Closed

Thanks Nicholas, much appreciated!