

## GROMACS - Bug #981

### CPU acceleration auto-detection fails with (some?) gcc <=v4.4

07/27/2012 05:14 AM - Szilárd Páll

<b>Status:</b>	Closed	
<b>Priority:</b>	High	
<b>Assignee:</b>	Rossen Apostolov	
<b>Category:</b>	build system	
<b>Target version:</b>	4.6	
<b>Affected version - extra info:</b>	4.6-dev	<b>Difficulty:</b> uncategorized
<b>Affected version:</b>		

#### Description

CPU acceleration auto-detection doesn't work with (some?) gcc v4.4 and earlier, the GMX\_ACCELERATION value defaults to "None".

Reproduced both on Linux and Mac OS Lion (with gcc obtained through MacPorts).

See the CMake outputs of configs on the build server (SSE2 support) failing:

- [gcc 4.1 on Ubuntu 10.04](#)
- [gcc 4.4 on Ubuntu 10.04](#),

while this one worked:

- [gcc 4.3 on Ubuntu 10.04](#)

Note that gcc 4.3 on another machine also fails, so the above pattern is just a coincidence.

#### History

##### #1 - 07/27/2012 04:10 PM - Erik Lindahl

Hi,

I think I already might have solved this; the previous inline ASM construct I had was VERY sensitive to compilers ignoring the volatile keyword when I was moving around the EBX register, and it might even be due to link-time optimization (which could explain why it isn't explained by the compiler alone).

However, apart from having a slightly smarter construct I also realized the EBX magic is only necessary when compiling PIC code in 32-bit mode. By adding `ifdefs` checking for this I no longer need to save/restore EBX on 99.9% of the platforms in practical use - there will be a patch later today.

##### #2 - 07/27/2012 08:20 PM - Szilárd Páll

Interestingly, while the `cmake-time` detection doesn't work in some cases the resulting binary still detects the correct acceleration at runtime and warns about the hardware-binary mismatch.

Have you tried your fixes for the above mentioned cases?

##### #3 - 07/27/2012 11:15 PM - Erik Lindahl

The bug occurs when we don't get the correct output of the EBX register from `cpuid`, which is because of the move-around magic not working. Since the contents of the register will be somewhat random, this is likely why you see it working in some cases. I'll try to test it, but since I completely avoid this on 64-bit (and non-PIC 32-bit) now I think the problem should be *much* less of an issue (if any).

##### #4 - 08/01/2012 03:45 PM - Mark Abraham

I found this independently. I tried to compile patch 16 the `nbnxn` branch in <https://gerrit.gromacs.org/#/c/1046/> with gcc 4.4.4 on Nehalem hardware (i.e. supporting SSE4.2). GROMACS auto-detection didn't work - CMake decides to use no acceleration, but will acquiesce if I choose `sse4.1` by hand. Then it dies during

```
$ make VERBOSE=1
```

```
...
```

```
cd /home/224/mxa224/git/nbnxn/build_cmake/src/gmxlib && /apps/gcc/wrapper/gcc -Dgmx_EXPORTS -DGMX_OPENMP -DTMPI_SET_AFFINITY -DHAVE_CONFIG_H -fopenmp -Wall -Wno-unused -Wunused-value -fomit-frame-pointer -funroll-all-loops -O3 -DNDEBUG -fPIC
```

```
-I/home/224/mxa224/progs/include/libxml2 -I/home/224/mxa224/git/nbnxn/build_cmake/src -I/home/224/mxa224/git/nbnxn/build_cmake/include
-I/home/224/mxa224/git/nbnxn/include -I/apps/fftw3/3.2.2/include -I/home/224/mxa224/git/nbnxn/src/gmxlib -o CMakeFiles/gmx.dir/bondfree.c.o -c
/home/224/mxa224/git/nbnxn/src/gmxlib/bondfree.c
In file included from /home/224/mxa224/git/nbnxn/include/gmx_x86_sse4_1.h:24,
from /home/224/mxa224/git/nbnxn/include/gmx_x86_simd_single.h:35,
from /home/224/mxa224/git/nbnxn/src/gmxlib/bondfree.c:61:
/apps/gcc/4.4.4/lib/gcc/x86_64-unknown-linux-gnu/4.4.4/include/smmintrin.h:32:3: error: #error "SSE4.1 instruction set not enabled"
In file included from /home/224/mxa224/git/nbnxn/include/gmx_x86_simd_single.h:35,
from /home/224/mxa224/git/nbnxn/src/gmxlib/bondfree.c:61:
/home/224/mxa224/git/nbnxn/include/gmx_x86_sse4_1.h:82: error: expected declaration specifiers or '...' before '___m128'
```

... and a host of SSE errors.

I thought adding `-msse4.1` might help things out, but it segfaults gcc!

```
$ (cd /home/224/mxa224/git/nbnxn/build_cmake/src/gmxlib && /apps/gcc/wrapper/gcc -Dgmx_EXPORTS -DGMX_OPENMP -DTMPI_SET_AFFINITY
-DHAVE_CONFIG_H -fopenmp -Wall -Wno-unused -Wunused-value -fomit-frame-pointer -funroll-all-loops -O3 -DNDEBUG -fPIC
-I/home/224/mxa224/progs/include/libxml2 -I/home/224/mxa224/git/nbnxn/build_cmake/src -I/home/224/mxa224/git/nbnxn/build_cmake/include
-I/home/224/mxa224/git/nbnxn/include -I/apps/fftw3/3.2.2/include -I/home/224/mxa224/git/nbnxn/src/gmxlib -o CMakeFiles/gmx.dir/bondfree.c.o -c
/home/224/mxa224/git/nbnxn/src/gmxlib/bondfree.c -msse4.1)
/home/224/mxa224/git/nbnxn/src/gmxlib/bondfree.c: In function 'calc_bonds_omp_fn.0':
/home/224/mxa224/git/nbnxn/src/gmxlib/bondfree.c:3749: internal compiler error: Segmentation fault
Please submit a full bug report,
with preprocessed source if appropriate.
See <http://gcc.gnu.org/bugs.html> for instructions.
```

```
$ gcc -v
Using built-in specs.
Target: x86_64-unknown-linux-gnu
Configured with: ../gcc-4.4.4/configure --prefix=/apps/gcc/4.4.4 --with-local-prefix=/apps/gcc/4.4.4 --disable-multilib --enable-threads=posix
--with-gmp=/apps/gmp/4.3.2 --with-gmp-include=/apps/gmp/4.3.2/include --with-gmp-include=/apps/gmp/4.3.2/lib --with-mpfr=/apps/mpfr/2.4.2
--with-mpfr-include=/apps/mpfr/2.4.2/include --with-mpfr-lib=/apps/mpfr/2.4.2/lib --enable-languages=c,c++,fortran,java,objc,obj-c++
Thread model: posix
gcc version 4.4.4 (GCC)
```

I worked backwards, and commit  
7ba20b409d6e3cfd6839c4cee29cae47314351b Fixed gcc inline assembly issue with PIC and older gcc compilers  
seems to introduce the above regression, as commit  
5ba7125c5972f2aafde2310eaa4a345cbac55da5 New CPU detection & AVX/SSE code, removed raw assembly files.  
(which comes two before 7ba20b409) detects and compiles fine.

icc 11.1 on the same machine detects, compiles and runs fine.

So it seems to me that we either

- have (the potential for) known issues with older versions of gcc, which we need to prohibit with CMake, or
- have a bug in 7ba20b409

#### #5 - 08/01/2012 08:31 PM - Teemu Murtola

Mark Abraham wrote:

So it seems to me that we either

- have (the potential for) known issues with older versions of gcc, which we need to prohibit with CMake, or
- have a bug in 7ba20b409

I think that Christoph reported problems also on gcc 4.6 and 4.7 that were fixed by the patch, so it's not limited to older gcc. But I think it's pointless to speculate before we actually have Erik's latest fix (which is not included in the mentioned patch set 16, nor in any other change in gerrit AFAIK).

#### #6 - 12/20/2012 07:34 PM - Erik Lindahl

- Status changed from New to Closed

This should all be fixed in the latest git version.