GROMACS - Bug #1110

GMX_FFT_LIBRARY=mkl does not work

01/09/2013 03:22 AM - Szilárd Páll

Status:	Closed		
Priority:	Normal		
Assignee:	Mark Abraham		
Category:	build system		
Target version:	4.6.4		
Affected version - extra info:	4.6-beta3	**Difficulty:**	uncategorized
Affected version:	4.6.1		

Description

Invoking the build generator with cmake -DGMX_FFT_LIBRARY=mkl will result in the FindMKL throwing an error about not being able to find MKL, even though the Intel Compiler toolchain (including MKL) is correctly set up.

Related issues:

Related to GROMACS - Bug #1186: install guide should mention how to configure...	**Closed**	**03/11/2013**

Associated revisions

Revision 28d2a6d1 - 04/26/2013 08:37 AM - Mark Abraham

Update linking to MKL and document same

Works using nifty feature from icc 11 and up, or any other compiler if
the user does the legwork (which is all we ever used to offer).

Code that used HAVE_LIBMKL had a bug, which we never saw because
the top-level CMakeLists.txt set HAVE_MKL. Fixed that.

Removed unused TextMKL.c code - check_function_exists() is sufficient.

Refs #1110,#1186

Change-Id: I39a66673e5fe571a5f8b0691bbe2ec619cd60778

Revision 236275d9 - 07/25/2013 01:51 PM - Mark Abraham

Find mkl.h on more icc versions

Refs #1110

Change-Id: I0b6bc2497fc2a504b6c29b0697ee9e354fe6cffd

History

#1 - 01/09/2013 03:22 AM - Szilárd Páll

- Affected version - extra info set to 4.6-beta3

#2 - 01/09/2013 03:34 AM - Erik Lindahl

- Priority changed from Normal to Low

Related to #1067, but since MKL is anyway slower than FFTW3 I would be willing to bump this to 4.6.1 if we have to...

#3 - 01/09/2013 03:56 AM - Szilárd Páll

- Priority changed from Low to Normal

Erik Lindahl wrote:

> Related to #1067, but since MKL is anyway slower than FFTW3 I would be willing to bump this to 4.6.1 if we have to...

Agreed about bumping. Can't confirm whether MKL is slower, I've only tried on Westmere with 1-2 systems and based on that I would not jump to conclusions.

Here's the workaround:

```
cmake -DGMX_FFT_LIBRARY=mkl
-DCMAKE_PREFIX_PATH=/opt/intel/composer_xe_2011_sp1/mkl:/opt/intel/composer_xe_2011_sp1/mkl/lib/intel64
```

This is not enough, though, because the linking will still fail. To fix that one has to set CMAKE_EXE_LINKER_FLAGS=-mkl. Actually, this flag might be the only thing needed when using MKL with icc (v12+).

The users should not have the do any of the above with the toolchain correctly set up (everything is in the path). This is a FindMKL issue and we might be able to find a version that could replace the current buggy one.

#4 - 01/14/2013 09:30 PM - Christoph Junghans

I think we should drop **direct** mkl support.

1.) Newer mkl wrap fftw's fft function, so one uses effectivly fftw
1b.) >mkl=10.2 comes with fft interface, so that one can use mkl as fftw
2.) Find{BLAS,LAPACK} can use mkl as Blas/Lapack implementation

I guess there could be cases, where pure mkl is faster than fftw, but this should be rare, I will look into that a bit more.

#5 - 01/14/2013 10:22 PM - Szilárd Páll

How much is the overhead of the fftw wrapper, if any? If it is close to none even at high parallelization, I am all for it. However, if that's not the case, using mkl in a native manner, at least with the current code, requires only minor fixes (that my above workarounds hint).

#6 - 01/14/2013 10:26 PM - Szilárd Páll

As it's not a crucial issue, I'm lowering the priority of this.

Christoph & others, please switch the status to "In progress" if you are working on it. Otherwise, we can bump this to 4.6.1 (AND note this in the "known issues").

#7 - 01/18/2013 03:34 PM - Erik Lindahl

- Target version changed from 4.6 to 4.6.1

#8 - 03/06/2013 09:12 PM - Szilárd Páll

- Target version changed from 4.6.1 to 4.6.2

#9 - 03/07/2013 12:47 PM - Mikhail Plotnikov

I was able to build MKL interface simply by adding –mkl=sequential to linker option:
-DGMX_FFT_LIBRARY=mkl -DMKL_LIBRARIES="$MKLROOT/lib/intel64" -DCMAKE_EXE_LINKER_FLAGS="-mkl=sequential"

This works for ICC >=v12.0.
Sequential MKL libraries are needed, because OpenMP parallelization is enabled on a level above MKL.
So, linking MKL should be as simple as defining:
-DGMX_FFT_LIBRARY=mkl -DMKL_LIBRARIES="$MKLROOT/lib/intel64"
and adding "-mkl=sequential" to linker options. This should be not much trouble to implement in 4.6.2.

#10 - 03/07/2013 05:04 PM - Mark Abraham

Thanks, Mikhail. That looks great.

Our install doocumentation will suggest either using the "source compilervars.sh intel64" that does the right environment magic, or the -DMKL_LIBRARIES approach you suggest, or the local equivalent solution. I expect your approach has the advantage of still working from a subsequent shell.

#11 - 03/08/2013 05:23 PM - Szilárd Páll

Mikhail Plotnikov wrote:

> So, linking MKL should be as simple as defining:
> -DGMX_FFT_LIBRARY=mkl -DMKL_LIBRARIES="$MKLROOT/lib/intel64"
> and adding "-mkl=sequential" to linker options. This should be not much trouble to implement in 4.6.2.

That can't be correct because MKL_LIBRARIES is supposed to contain the **libraries** needed for linking against MKL and not the path.

Note that our current FindMKL cmake module sucks badly. We should really replace it asap. In fact, in my testing I had to either comment out the find_package_handle_standard_args() in FindMKL or set set stuff like CMake prefix path or MKL_INCLUDE_DIR just to make FindMKL happy and not fail.

#12 - 03/11/2013 02:25 PM - Mark Abraham

Ja, working on it.

#13 - 03/23/2013 06:04 PM - Mark Abraham

Note to self - Windows probably needs /Qmkl, not -mkl.

#14 - 04/25/2013 09:12 PM - Mark Abraham

- Affected version set to 4.6.1

Assuming https://gerrit.gromacs.org/#/c/2229/9 gets merged, once https://gerrit.gromacs.org/2230 stabilizes, fix the management of the cached MKL_LIBRARIES variable so that TEST_MKL is the result of a test when MKL_LIBRARIES (etc.) changes

#15 - 04/30/2013 03:57 PM - Mark Abraham

- Status changed from New to Accepted

- Assignee set to Mark Abraham

- Target version changed from 4.6.2 to 4.6.3

#16 - 04/30/2013 03:57 PM - Mark Abraham

- Status changed from Accepted to In Progress

- % Done changed from 0 to 90

#17 - 07/05/2013 06:29 PM - Mark Abraham

- Target version changed from 4.6.3 to 4.6.4

Fixed, awaiting code review

#18 - 07/25/2013 12:41 PM - Mark Abraham

Clarifying: -DGMX_FFT_LIBRARY=mkl now Just Works in 4.6.3. Waiting on https://gerrit.gromacs.org/2230 for elegance in the implementation.

#19 - 07/26/2013 01:30 AM - Mark Abraham

- Status changed from In Progress to Resolved

#20 - 08/07/2013 08:19 PM - Szilárd Páll

- Status changed from Resolved to Closed

Change 2230 got merged, but unfortunately it did not ref this issue. Works well for me now, closing the issue.