GROMACS - Bug #1471

cmake 2.8.12.2 breaks linking on Mac OS X.

03/27/2014 08:34 PM - Anders Gabrielsson

Status: Closed
Priority: Normal
Assignee: Mark Abraham
Category: build system
Target version: 5.0
Affected version - extra info:
Affected version: 4.6.6
Difficulty: uncategorized

Description
On Mac OS X, binaries are broken during "make install" using cmake version 2.8.12.2.
Downgrading to cmake 2.8.11.2 or GMX_GPU=OFF (then tested compiling with clang) are working walk-arounds.

Possible fix (but not tested): http://cmake.org/gitweb?p=cmake.git;a=commit;h=08141a5

Associated revisions

Revision d9d2b0aa - 06/18/2014 02:17 PM - Erik Lindahl
Fixed CMake 2.8.12+ CUDA dylib bugs and warnings on OS X
This patch fixes up a number of minor compilation issues with CUDA on OS X. The STRICT_ANSI define has been added to make CUDA work with gcc version 4.8 and 4.9, a dummy variable has been added to the dummy c++ gpu utils file to avoid warnings about empty objects, and we now properly handle both RPATH (newer CMake) and INSTALL_NAME_DIR (older CMake) without warnings. CMake-2.8.12 introduced a bug that caused Gromacs executables to be malformed since the rpath to the CUDA libraries was added twice. This is caused by fairly deep internal changes in CMake, but it has been worked around by removing the rpath (which FindCUDA.cmake only adds on OS X) manually for CMake>2.8.11.
Fixes #1471.
Change-Id: I8b01204dc07678bc305821162f20db65e0e7b88d

History

#1 - 06/12/2014 02:05 AM - Erik Lindahl
- Priority changed from Normal to High
- Target version set to 5.0

#2 - 06/13/2014 04:56 PM - Erik Lindahl
- Status changed from New to Feedback wanted
- Priority changed from High to Normal

Anders - please provide more details.
The description about testing with clang instead hints that you are using gcc or something else in the first place. Exactly what configuration settings were you using? What computer are you compiling on? OS version? CUDA version? Compiler versions?
It is not clear to me that this is actually a CMake, rather than Cuda, bug; builds both with clang and gcc work fine for me without Cuda.
The Mac CUDA situation is unfortunately a mess since NVIDIA only supports old versions of GCC while the system default clang does not support OpenMP.

#3 - 06/13/2014 05:22 PM - Anders Gabrielsson
As far as I can remember, I tried building (mdrun) using then up-to-date versions of both apple-clang and macports-gcc along with CUDA 6.

05/04/2020
The thing is that the binaries in the build tree works fine, but the ones put in your path after 'make install' have broken dylib headers. (otool cant read them anyway.)
This makes me think it is related to the new rpath features of cmake 2.8.12 for macos. http://www.kitware.com/blog/home/post/510

I can take a few minutes to try and reproduce this with my current setup.

#4 - 06/13/2014 06:00 PM - Erik Lindahl
- Status changed from Feedback wanted to In Progress

OK, then I can reproduce it - thanks!

#5 - 06/13/2014 06:08 PM - Anders Gabrielsson
Yes, I still get the same thing both for recent branches of both 4.6 and 5.0.

This is with following setup:
MacOS X 10.9.3
Clang/Xcode 5.1.1
CUDA (driver) 6.0.46
Cmake 2.8.12.2 (from macports)

#6 - 06/17/2014 05:31 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1471.
Uploader: Erik Lindahl (erik@kth.se)
Change-id: I8b01204dc07678bc305821162f20db65e0e7b88d
Gerrit URL: https://gerrit.gromacs.org/3610

#7 - 06/17/2014 05:32 PM - Erik Lindahl
- Status changed from In Progress to Resolved

#8 - 06/17/2014 05:32 PM - Erik Lindahl
- Status changed from Resolved to Fix uploaded

#9 - 06/18/2014 02:30 PM - Erik Lindahl
- Status changed from Fix uploaded to Resolved
- % Done changed from 0 to 100

Applied in changeset d9d2b0aa743becd6f40864c478e83007689f8cd5.

#10 - 06/19/2014 03:17 PM - Rossen Apostolov
- Status changed from Resolved to Closed