GROMACS - Task #2135

check non-Jenkins compilers work

03/08/2017 05:32 PM - Mark Abraham

Status:	Closed	
Priority:	Normal	
Assignee:	Mark Abraham	
Category:	testing	
Target version:	2018	
Difficulty:	uncategorized	

Description

Szilard has observed Apple Clang from XCode fail one of the C++11 tests (the library one I think). We ought to expect e.g. distro compiler maintainers to provide some way for a standard toolchain to get hooked up to an installed standard library. We should test it works out of the box, and/or fix and/or document.

Further thought: it's easy to accidentally name clang as the compiler, and even with clang -std=c++11 you get mysterious looking errors that go away when you do clang++ -std=c++11, so perhaps we should add a hint when that test fails that naming the compiler clang++ is necessary.

Related issues:

Related to GROMACS - Task #2161: update the way testing matrices are specified	**Closed**

History

#1 - 03/14/2017 02:38 PM - Mark Abraham

Another thought - clang now supports OpenMP, but e.g. the package from the LLVM project running on bs_mac does not detect OpenMP support. Find out what is the issue here.

#2 - 05/12/2017 07:34 PM - Szilárd Páll

Mark Abraham wrote:

> Another thought - clang now supports OpenMP, but e.g. the package from the LLVM project running on bs_mac does not detect OpenMP support. Find out what is the issue here.

Perhaps it was build without OpenMP?

In my experience, I always had to set up clang with some external OpenMP lib (libomp or libgomp) and did not find any way to set a default such that I would not need to pass extra -fopenmp=libomp to CMake nor set the LD_LIBRARY_PATH every time when using a separate libomp.

#3 - 05/15/2017 01:41 AM - Mark Abraham

- Related to Task #2161: update the way testing matrices are specified added

#4 - 01/04/2018 04:20 PM - Mark Abraham

- Status changed from New to Closed

- Assignee set to Mark Abraham

Post submit now tests linux clang with openmp. Since Apple XCode clang still doesn't support OpenMP, it's not yet an interesting performance target.

We have done some testing of cray wrapper compilers, and gcc for power 8 compilers, so I regard this as complete enough.