Add a warning for or document which architectures / compilers not tested

I has been suggested as part of https://gerrit.gromacs.org/#/c/3740 to add a warning in cmake if the user compiles with a compiler or on a architecture which hasn't been tested. The warning could suggest to prefer a tested compiler (on x86), to run all unit/regression tests, and compare results of one's simulation to a known compiler/architecture.

- Should we add such a warning?
- For which compilers, architectures?
- For everything not tested by Jenkins? E.g. also 32bit x86, Xeon Phi, and Cuda 6? Or only those we know are not tested by Jenkins or our core-developers?
- Should we also warn for all compiler versions not tested?
- Should we also warn for cmake options not tested? E.g. some of the QM options are probably not tested by any configuration.
- Anything else?

At least on architectures we don't test, it is necessary that the user runs the tests. Because we don't have a way to query whether it is a tested architecture every user should run the tests.

Also add a warning if unit tests cannot be build because libmxl2 wasn't found.

Partial fix for #1548

Change-Id: I8b87cc77f909d5cf2a4ee70151096312e3191af2

I think this is doomed on multiple fronts.

1) We can't specify what our test configurations are with detail that is both useful and maintainable with no ongoing effort. Even writing CMakery to manage and compare with the database seems like overkill to me.
2) Specifying that (say) icc 12.0.1 is tested helps only the rare individual that has a problem, and ready access to alternative versions of icc, and the problem is not caused by the interaction of icc with some other component that still differs
3) We all know that "Yeah I tested md-vv / BlueGene / GPU" means "I tested it on the few things that were meaningful to me; I can't possibly test it in concert with everything, not least because I don't know how everything else works"
07/02/2014 11:18 PM - Roland Schulz
Subject changed from Add a warning for architectures / compilers not tested to Add a warning for or document which architectures / compilers not tested

07/02/2014 11:33 PM - Roland Schulz
I think, your point 1 is the most important. Do we want to put effort into maintaining a list of architectures we feel confident about. I think this would have value to both user and ourselves (e.g. it might have helped us realize we don't know what going on with 32bit earlier). If we have such a list we can just put it into the documentation or use it for a cmake warning.

Alternative if we think users should always run the tests on any architecture (your point about that even with known compiler/arch/OS it could still be something else different), then we might want to advertise that more. I have the feeling that most people don't run the tests. But I might be wrong.

Things we could do:

- Add running tests to the "Quick and dirty installation" section of the install-guide.
- print a warning if unit-tests are not build (because libxml wasn't found) and if the regression tests cannot be run (neither REGRESSIONTEST_PATH nor REGRESSIONTEST_DOWNLOAD is set).
- print a message that one should now run the tests at the end of "make" or even run them automatically at the end of "make" (create a new target (e.g. named all-no-tests) which doesn't execute the tests)

07/03/2014 09:56 AM - Erik Lindahl
The solution we have for this is the regression test suite, and increasingly the unit tests.

In general, we are very good at coming up with new handy ideas that in principle would be nice to have, but that would take huge amounts of time for somebody. That only works provided somebody has that time to spend. Real in-depth testing would easily be 1-2 hours per platform, and then we would rapidly be talking weeks of time before each release.

07/03/2014 05:45 PM - Szilárd Páll
I suggested adding a warning regarding random123 because the developers explicitly warn against using the code without thoroughly testing it.

I do think there is value in documenting:

- what do we test regularly (in jenkins);
- what do we test occasionally (HPC platforms, exotic stuff, etc.);
- known issues with compilers, libraries, platforms, etc.

IIRC we discussed this in detail on gmx-dev and perhaps even during developer meetings in the past. I thought we actually agreed on putting together the above information, but I think not much has happened since. There is very little public information of this kind on the wiki except the jenkins machine/software config information which has been kept more or less up-to-date, but this does not explicitly state the exact configurations used.

Not documenting anything because it's too complex seems to only serve the purpose of saving some effort and clouding things - which I don't think is beneficial.

11/25/2015
Do we want to put effort into maintaining a list of architectures we feel confident about.

I think we do and as I suggested above we have two clear "levels of confidence". Alternative if we think users should always run the tests on any architecture (your point about that even with known compiler/arch/OS it could still be something else different), then we might want to advertise that more. I have the feeling that most people don't run the tests. But I might be wrong.

I think you're right. Given that nobody complained about the tests failing even though we have two known and not uncommon cases that cause a regression tests error each seems to indicate that.

Things we could do:

- Add running tests to the "Quick and dirty installation" section of the install-guide.

+1

We should somehow advocate the use of the regression tests much more. I'd definitely add this step to the install guide and I think it would not hurt issuing a CMake message at some point which instructs the user what/how to test an installation.

#7 - 07/03/2014 06:22 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1548.
Uploader: Roland Schulz (roland@rschulz.eu)
Change-Id: I8b87cc77f909d5cf2a4ee70151096312e3191af2
Gerrit URL: https://gerrit.gromacs.org/3748

#8 - 07/04/2014 05:38 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1548.
Uploader: Roland Schulz (roland@rschulz.eu)
Change-id: I8b87cc77f909d5cf2a4ee70151096312e3191af2
Gerrit URL: https://gerrit.gromacs.org/3757

#9 - 07/12/2014 07:48 AM - Roland Schulz
- Status changed from New to Closed

The patch which has been merged does as much as we can without adding maintenance work.

#10 - 07/15/2014 12:11 PM - Teemu Murtola
- Category set to documentation
- Target version set to 5.0.1