SelectionUnitTests fail on i686 with SIMD disabled

09/03/2015 01:32 AM - Dominik Mierzejewski

Status: Rejected  Priority: Normal  Assignee:  Category:  Target version:  Affected version -  extra info:  Affected version: 5.1  Difficulty: uncategorized

Description

When built with -DGMX_SIMD=None, gromacs-5.1 internal testsuite fails Test #16: SelectionUnitTests.

I think the failure is in HandlesUnsortedIndexGroupsInSelectionsDelayed as that's the last test run displayed by ctest.

Full build log attached.

History

#1 - 09/03/2015 05:44 AM - Teemu Murtola
It is unlikely that GMX_SIMD=None has any effect on this issue. If you can reproduce the issue (by just running the bin/selection-test binary from the build), it would be great if you can provide a stack trace of the crash.

#2 - 09/03/2015 06:51 PM - Dominik Mierzejewski
Indeed, I'm unable to reproduce it again. Sorry for the noise. Please close. I'll reopen if it's reproducible.

#3 - 09/03/2015 08:49 PM - Teemu Murtola
- Status changed from New to Rejected

If I read you build log right, the build that fails is actually configured for a release build, but does not define -DNDEBUG. This makes it enable a few types of floating point exceptions, which may not be robust with certain optimizations. There are two possibilities:

- There is some uninitialized memory access in the test, and that causes it to randomly (and very rarely) generate fp exceptions. But I do not remember seeing such issues in our CI system, which runs the test quite extensively all the time, but it is still possible that these have gone unnoticed. If this is the case, the uninitialized memory is probably never used for anything except for some arithmetic operations (whose result is not used further), so this should not be any major issue. But in order to fix such an issue, a stacktrace of the crash is necessary to even try to localize the issue.
- Your combination of compiler flags might cause the code to generate an fp exception from some code that with strict fp arithmetic would not do it.

I moved the issue to rejected, but please reopen if you actually can reproduce the issue and can provide additional information. If you ensure that -DNDEBUG is actually passed to non-debug builds, you should not see these fp exceptions (which are there mainly as a debugging aid).

#4 - 09/22/2015 03:16 PM - Dominik Mierzejewski
I can reproduce it almost 100% of the time, but only by running ctest -I 16,16 in the build directory:

```
$ pwd
/builddir/build/BUILD/gromacs-5.1/serial
$ LD_LIBRARY_PATH=$PWD/lib ctest -I 16,16
Test project /builddir/build/BUILD/gromacs-5.1/serial
  Start 16: SelectionUnitTests
1/1 Test #16: SelectionUnitTests ............... ***Exception: Numerical 0.09 sec
0% tests passed, 1 tests failed out of 1
```

Label Time Summary:
GTest = 0.09 sec
UnitTest = 0.09 sec
Total Test time (real) = 0.10 sec

The following tests FAILED:
16 - SelectionUnitTests (NUMERICAL)

Errors while running CTest

However, running the bin/selection-test binary in a loop does not fail (I waited while it kept running for about 5 minutes). Could this be a bug in ctest?

Also, I'm running cmake using the %cmake rpm macro, which does define -DNDEBUG, so I'm not sure your initial analysis is correct.

```
$ rpm --eval '%cmake'
[...]
/usr/bin/cmake \
-DCMAKE_C_FLAGS_RELEASE:STRING="-DNDEBUG" \
-DCMAKE_CXX_FLAGS_RELEASE:STRING="-DNDEBUG" \
-DCMAKE_Fortran_FLAGS_RELEASE:STRING="-DNDEBUG"
[...]
```

Do you have any suggestions how to debug this?

#5 - 09/23/2015 05:54 AM - Teemu Murtola

Dominik Mierzejewski wrote:

I can reproduce it almost 100% of the time, but only by running ctest -l 16,16 in the build directory:

[...]
However, running the bin/selection-test binary in a loop does not fail (I waited while it kept running for about 5 minutes). Could this be a bug in ctest?

Do you have any suggestions how to debug this?

I would suggest using ulimit to allow creation of core files, and then look at the stack trace from a crash. That would give at least some hint of what is happening.

Also, I'm running cmake using the %cmake rpm macro, which does define -DNDEBUG, so I'm not sure your initial analysis is correct.

[...]

Looking at your log, the cmake command line specifies the flags twice, and the second set of flags does not include -DNDEBUG, so I think this overrides your first ones.

#6 - 09/23/2015 02:21 PM - Dominik Mierzejewski

Teemu Murtola wrote:

Dominik Mierzejewski wrote:

I can reproduce it almost 100% of the time, but only by running ctest -l 16,16 in the build directory:

[...]
However, running the bin/selection-test binary in a loop does not fail (I waited while it kept running for about 5 minutes). Could this be a bug in ctest?

Do you have any suggestions how to debug this?

I would suggest using ulimit to allow creation of core files, and then look at the stack trace from a crash. That would give at least some hint of what is happening.

No core files are produced, even with ulimit -c unlimited.

Also, I'm running cmake using the %cmake rpm macro, which does define -DNDEBUG, so I'm not sure your initial analysis is correct.

[...]

Looking at your log, the cmake command line specifies the flags twice, and the second set of flags does not include -DNDEBUG, so I think this overrides your first ones.

You're right, I've just fixed it in the build script and that seems to have fixed the issue. Thanks!

Files