Dear all,
I am running Mac OS Catalina 10.15.1 (Xcode 11.2 11B52). I have successfully compiled several versions of Gromacs (5.1.5, 2018.8, or 2019.4) patched with plumed (v.2.4.6 for 5.1.5, or v.2.5.3 for 2018/2019 versions). The compilations are completed without any errors, employing the configuration:

1. `cmake .. -DGMX_BUILD_OWN_FFTW=ON -DGMX_MPI=on
   -DCMAKE_INSTALL_PREFIX:PATH=/usr/local/gromacs -DCXX="$MPICXX"

For example Gromacs v. 5.1.5 (but the same is happening on the other versions), I get segmentation faults, when I just run some tools, like:

```
Process received signal

Signal: Segmentation fault: 11 (11)
Signal code:  (0)
Failing at address: 0x0
 0   libsystem_platform.dylib
 0x00007fff69e47b1d _sigtramp + 29
 1   libdyld.dylib
 0x00007fff69c3730e dyld_stub_binder + 282
 2   libgromacs_mpi.1.5.0.dylib
 0x000000010dd60f2e _ZN3gmx24CommandLineModuleManager3runEiPPc + 798
 3   gmx_mpi
 0x000000010dcc39d1 main + 129
 4   libdyld.dylib
 0x00007fff69c462e5 start + 1
 5   ???
 0x0000000000000002 0x0 + 2
```

End of error message

```
zsh: segmentation fault  gmx_mpi trjconv
```

Just running the command e.g. `gmx(_mpi) grompp`, or `gmx(_mpi) trjconv` without any input files, a segmentation fault is produced, as it is in the case also with input files. Some tools, like `editconf` (v. 2019.4), just complain about missing input files, if none is given, but without a segmentation fault, as it should be. Switching MPI support on/off on the configuration/compilation stage does not change the situation. The vanilla versions are also behaving the same way (without the plumed patch). Some tools are running smoothly, as `editconf`. However others, have segmentation faults. Grompp is running on v.5.1.5, but gives a segmentation fault on v. 2019.4.

I would appreciate some help, thank you,
Vangelis.
Work around broken Apple compilers in Mac OS 10.15

The Xcode compiler in Mac OS Catalina checks and enforces stack alignment by default, but embarrassingly enough the C library provided by Apple fails to adhere to the aligned stack for AVX despite bug reports being filed during their beta cycle. We work around this sloppiness by removing the check, since AVX does not require the alignment.

Fixes #3199.

Revision b68d25f6 - 11/06/2019 02:22 PM - Erik Lindahl

Work around broken Apple compilers in Mac OS 10.15

The Xcode compiler in Mac OS Catalina checks and enforces stack alignment by default, but embarrassingly enough the C library provided by Apple fails to adhere to the aligned stack for AVX despite bug reports being filed during their beta cycle. We work around this sloppiness by removing the check, since AVX does not require the alignment.

Fixes #3199.

History

#1 - 11/06/2019 12:24 PM - Erik Lindahl
- Status changed from New to Resolved

This is not a GROMACS bug, but due to Apple first deciding to enable strict stack checking for 16-byte-alignment by default in Xcode 11, and then failing to have their own C library align the stack properly.

For more details, see https://forums.developer.apple.com/thread/121887

It is possible to work around by setting the environment variable "MACOSX_DEPLOYMENT_TARGET=10.14" before compiling. In theory one could also work around it by adding -fno-stack-check to the compiler flags, which we might do for the latest 2019/2020 releases, but it's low priority and we definitely won't do any extra work to fix older versions because a sloppy vendor ships a completely broken development environment despite getting multiple bug reports about it during their beta.

Apart from having decided to move away from open standards such as OpenCL and OpenGL, Apple also explicitly removes OpenMP support from their tree of the LLVM compilers just to push their own proprietary technologies.

It was a long time ago that Apple could be recommended as a nice open source player. Today we STRONGLY recommend against using their platform for anything.

#2 - 11/06/2019 02:02 PM - Erik Lindahl
- Status changed from Resolved to Fix uploaded

Workaround added for release-2019 in https://gerrit.gromacs.org/c/gromacs/+/14145, which will also be pulled into master/2020.

However, until those versions are out and apply it by default, the easiest option is likely to just set MACOSX_DEPLOYMENT_TARGET=10.14 before compiling.

#3 - 11/06/2019 03:25 PM - Eric Irrgang

Erik Lindahl wrote:

However, until those versions are out and apply it by default, the easiest option is likely to just set MACOSX_DEPLOYMENT_TARGET=10.14 before compiling.

Note, also, that GROMACS 2020 provides a default value for CMAKE_OSX_DEPLOYMENT_TARGET of 10.9, so some mitigation should be present in the 2020 beta 2 release.

#4 - 11/06/2019 05:45 PM - Erik Lindahl
- Status changed from Fix uploaded to Resolved

Applied in changeset 2d92a6f84330b033fa2508f355f1138e379a2b42.

#5 - 11/06/2019 05:45 PM - Erik Lindahl
Applied in changeset b68d2566a84a200b75a895af9c5887ce4a149bc.

#6 - 11/13/2019 02:44 PM - Paul Bauer
- Status changed from Resolved to Closed

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMakeOutput.log</td>
<td>100 KB</td>
<td>11/06/2019</td>
<td>Vangelis Daskalakis</td>
</tr>
<tr>
<td>Makefile.cmake</td>
<td>28.5 KB</td>
<td>11/06/2019</td>
<td>Vangelis Daskalakis</td>
</tr>
<tr>
<td>Makefile</td>
<td>45.8 KB</td>
<td>11/06/2019</td>
<td>Vangelis Daskalakis</td>
</tr>
</tbody>
</table>