Hi,

I sent this to the Gromacs mailing list and Szilárd Páll invited me to post this here:

I am having issues with checking the correctness of my mpi version of Gromacs 5.1. I have build and tested earlier versions without issues (e.g. 5.0.4 is the latest I did). I am also building float and double versions of the serial tools and they test without issue using: make check.

I tried two things to test my MPI build (I'll give a detailed account on how I build later):

```
make -j 8 check
```
on a back-end node (slurm sbatch job asking for 8 cores) of our cluster. Tests 19 to 25 pass, while the first 18 fail in MPI_Init. Is that how it is supposed to be?

The other test I tried was as follows:

- change into the regression test directory  
- source the GMXRC file of an install containing a float and a float mpi version  
- executing: ./gmxtest.pl all -np 4 (in a batch script asking for 4 cores)

I get the following:

```
gmx-completion.bash: line 258: `._gmx_convert-tpr_compl': not a valid identifier
```

So here comes the detail on the build:

```
gcc/5.2.0  
openmpi 1.10.0 build for the above gcc 5.2.0  
fftw 3.3.4 build for the above gcc with sse and avx  
boost/1.59.0
```

```
CMAKEFLAGS="-DCMAKE_C_COMPILER=gcc -DCMAKE_CXX_COMPILER=g++ -DGMX_FFT_LIBRARY=fftw3 -DBUILD_SHARED_LIBS=ON -DCMAKE_INSTALL_PREFIX=$PREFIX -DGMX_DEFAULT_SUFFIX=ON -DREGRESSION_TEST_PATH=$RTESTPATH -DGMX_SIMD=AVX_128_FMA -DCMAKE_PREFIX_PATH=$FFTW3_HOME"
```

I build the serial and parallel version:

1. Building single precision without MPI  
   BUILDDIR="bdir_float_boost"  
   mkdir -p $BUILDDIR  
   cd $BUILDDIR
cmake ../ $CMAKEFLAGS
make -j 16
make -j 4 check
make install
cd ..

1. Building single precision with MPI
   BUILDDIR="bdir_float_mpi_boost"
   mkdir -p $BUILDDIR
   cd $BUILDDIR

cmake ../ $CMAKEFLAGS -DGMX_MPI=ON -DBUILD_SHARED_LIBS=off
make -j 16
make install

Both installs go to the same directory. This procedure is essentially the same as I used for Gromacs 5.0.4.

I tested with gcc 4.9.0 and openmpi 1.8.3 (which I used for gromacs 5.0.4) and the results are similar, though the error message from
the MPI_Init is slightly different.

Szilárd also asked me to provide details on the bash we are using:

[gromacs]$ bash --version
GNU bash, version 4.1.2(1)-release (x86_64-redhat-linux-gnu)
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

Please let me know if you need anything further.

Thanks and best wishes
Joachim

Associated revisions

Revision cb4cc774 - 10/14/2015 05:58 AM - Teemu Murtola
Make bash completions work with bash --posix
Replace dashes in generated function names with underscores.
Related to #1840.
Change-Id: I0c74ba16f552f55900f99f11aa01ae830350f93c

History

#1 - 10/08/2015 04:45 PM - Teemu Murtola
The completion error likely comes from your bash being in posix mode (started with --posix, or configured to start like that by default). And it should
not prevent you from running the tests, except if you cannot make your shell ignore the error... We can possibly put some additional effort into making
the completion script work also in that case, but so far, there has not been any other complaints about this, so it's unlikely that many people would be
running bash configured like this.

But otherwise, this probably has very little to do with your problems. It certainly does not cause MPI_Init() failures, but there is very little we can do to
diagnose those unless you actually provide the error message that you get... I assume that in 5.1, the first 18 tests are unit tests using Google Test,
and tests from 19 onwards are the regression tests. All tests work fine on our CI build system, including with MPI, so likely the issue is somehow
related to configuration of the environment where you are running the tests.
#2 - 10/14/2015 05:59 AM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1840.
Uploader: Teemu Murtola (teemu.murtola@gmail.com)
Change-Id: I0c74ba16f552f55900f99f11aa01ae830350f93c
Gerrit URL: https://gerrit.gromacs.org/5205

#3 - 10/14/2015 08:42 PM - Teemu Murtola
- Assignee set to Joachim Hein

The completion issue should be fixed by the linked change.

For the MPI_Init() problem, I suspect that your MPI environment/library does not allow you to run single-rank jobs without mpirun or similar, which make check tries to do. For the record, passing -j 8 to make check has no effect on parallelization of the tests. Nearly all the tests will always run in serial, with the exception of the regression tests. But there is very little we can do without additional information on the problem, or confirmation that this indeed is the reason.

#4 - 11/18/2015 05:24 PM - Mark Abraham
- Status changed from New to Feedback wanted

This fix is now in 5.1.1

#5 - 11/18/2015 10:37 PM - Teemu Murtola

Yes, the completion issue is fixed in 5.1.1. But there is no clarity on the actual MPI_Init() issue.

#6 - 11/19/2015 04:20 PM - Joachim Hein

Hi,

First of all sorry for not being more responsive to this. I got swamped in a number of even higher priority tasks than testing the gromacs. I am really sorry for this unfortunate situation.

When I got the notice that gromacs 5.1.1 should fix some (all?) of the issues, I downloaded and build. The situation around MPI_Init seems unchanged.

Using OpenMPI 1.10.0 I get for the first failed test:

@############# start quote #1
Start 1: TestUtilsUnitTests
1/25 Test #1: TestUtilsUnitTests ...............***Failed 0.55 sec
/common/sw/alarik/src/gromacs/gromacs-5.1.1/bdir_float_mpi_boost/bin/testutils-test: Symbol `fftwf_version' has different size in shared object, consider re-linking
[an046][63975,1][0][.......///omp/mca/btl/openib/btl_openib_proc.c:157] ompi_modex_recv failed for peer [63975,1].
At least one pair of MPI processes are unable to reach each other for MPI communications. This means that no Open MPI device has indicated that it can be used to communicate between these processes. This is an error; Open MPI requires that all MPI processes be able to reach each other. This error can sometimes be the result of forgetting to specify the "self" BTL.

Process 1 ([63975,1],[0]) is on host: an046
Process 2 ([63975,1],[1]) is on host: unknown!
BTLs attempted: self openib

11/25/2015
Your MPI job is now going to abort; sorry.

With OpenMPI 1.8.3 (from the 5.1 attempts) I got slightly different output. It could figure the hostname on Process 2:

Test project /common/sw/alarik/src/gromacs/gromacs-5.1/bdir_float_mpi_boost_g490
Start 1: TestUtilsUnitTests
1/25 Test #1: TestUtilsUnitTests ...............***Failed 0.55 sec
/common/sw/alarik/src/gromacs/gromacs-5.1/bdir_float_mpi_boost_g490/bin/testutils-test: Symbol 'fftwf_version' has different size in shared object, consider re-linking
[an075:02349] [db_pmi.c:454:commit] PMI_KVS_Commit: Operation failed
[an075][35672,1][btl_openib_proc.c:157:ompi_modex_recv failed for peer ][35672,1]
At least one pair of MPI processes are unable to reach each other for MPI communications. This means that no Open MPI device has indicated that it can be used to communicate between these processes. This is an error; Open MPI requires that all MPI processes be able to reach each other. This error can sometimes be the result of forgetting to specify the "self" BTL.

Process 1 ([35672,1],0) is on host: an075
Process 2 ([35672,1],1) is on host: an075
BTLs attempted: openib self

Your MPI job is now going to abort; sorry.

Considering testing using the ./gmxtest.pl file, the situation changed, but still no joy. I now get:

Will test on 4 MPI ranks (if possible)
ERROR: Can not find executable gmx_mpi pdb2gmx in your path.
Please source GMXRC and try again.

I have sourced the GMXRC prior to executing the test and checked with "which" that it does something. Prior to sourcing gmx_mpi is not on the path, after sourcing it is. There is no pdb2gmx in the binary directory of the install created (the serial gromacs still tests ok using make check). FYI: I install the builds from 4 gromacs compiles (single/double, with and without MPI) into a single directory.

Let me know (step by step, I am not a gromacs user myself) what further info you require.

Thanks for everything so far.

Best wishes
Joachim
This looks like an issue in your MPI environment/configuration. Something makes OpenMPI think that you are running with (at least) two ranks, even though you just start a single process without mpirun. As said earlier, most of the tests will just run in serial.

Hi,

Thanks for your comment.

I checked with a simple MPI hello world. On our frontend one can start mpi jobs without mpiexec or mpirun. They will run ok on single core. On the backend of our cluster this will not work and gives an error similar to what I have seen when testing gromacs. I assume it is some interaction between SLURM and Openmpi. On the frontend, the

make check

now passes all tests. Though frontend running is not really liked here (and any other HPC centre I know about). For a widely used package such as gromacs, it could be worthwhile to consider upgrading the test to use the job launcher (e.g. have an environment variable).

So this leaves the issue with the ./gmxtest.pl script. I don't regard this as critical for us, since I now have a pass on the make check.

Thanks again for your help.

Joachim Hein wrote:

Hi,

Thanks for your comment.

I checked with a simple MPI hello world. On our frontend one can start mpi jobs without mpiexec or mpirun. They will run ok on single core. On the backend of our cluster this will not work and gives an error similar to what I have seen when testing gromacs. I assume it is some interaction between SLURM and Openmpi. On the frontend, the

make check

now passes all tests. Though frontend running is not really liked here (and any other HPC centre I know about). For a widely used package such as gromacs, it could be worthwhile to consider upgrading the test to use the job launcher (e.g. have an environment variable).

So this leaves the issue with the ./gmxtest.pl script. I don't regard this as critical for us, since I now have a pass on the make check.

Thanks again for your help.

Getting make check to work on the back end requires giving CMake a bit of help to know how to get the implementation of make check to call MPI programs there, as you suggested. Some details at http://manual.gromacs.org/documentation/5.1.1/install-guide/index.html#testing-gromacs-for-correctness. I suspect you just need cmake

-DMPIEXEC=whatever. Whether you'd want to bother depends whether the toolchains differ on front and back end...

The dynamic linking failures suggest there are multiple FFTW installs available, and a different one is being found at configure, build and/or run time.
Joachim Hein wrote:

now passes all tests. Though frontend running is not really liked here (and any other HPC centre I know about). For a widely used package such as gromacs, it could be worthwhile to consider upgrading the test to use the job launcher (e.g. have an environment variable).

Possibly, but that would require that there is robust detection for the job launcher that works everywhere. If the build system guesses the name of the launcher or its parameters wrong, then that will break all the tests just as well. Given that gmxtest.pl, or the test executables, have never worked when used this way, and this is the first report that they do not work, it might break more than it fixes to try to make things work automatically...

So this leaves the issue with the ./gmxtest.pl script. I don't regard this as critical for us, since I now have a pass on the make check.

There is no separate issue; this is exactly the same: gmxtest.pl will need to execute other commands than just gmx mdrun, but it will only use mpirun for running mdrun. And since your environment does not support executing any of the other commands, it just fails. The error message is not really accurate, but it comes from the fact that the script just tries to execute the command and if it fails (like it does in your case), it assumes that it does not exist.

Mark Abraham wrote:

Getting make check to work on the back end requires giving CMake a bit of help to know how to get the implementation of make check to call MPI programs there, as you suggested. Some details at http://manual.gromacs.org/documentation/5.1.1/install-guide/index.html#testing-gromacs-for-correctness. I suspect you just need cmake -DMPIEXEC=whatever. Whether you'd want to bother depends whether the toolchains differ on front and back end...

Setting MPIEXEC has no effect on the serial tests. And as I said above, trying to use it automatically might break more than it fixes. We could possibly add a separate CMake variable that forces the use of MPIEXEC also for the serial tests, default that to off, and just leave it to the users to find this and set all the necessary variables.