Writing TNG files fails on Xeon Phi

07/10/2014 08:57 AM - Roland Schulz

Status: Closed
Priority: Normal
Assignee: Magnus Lundborg
Category: core library
Target version: 5.1
Affected version - extra info:
Affected version: 5.0

difficulty: uncategorized

Description
The unit tests:

- mdrun-test CanWrite/Trajectories.ThatDifferInNstxout/*
- legacy-tools-test NoFatalErrorWhenWritingFrom/TrjconvWithIndexGroupSubset.WithDifferentInputFormats/*

fail. And "gmx trjconv" fails with segfault if the output is TNG. This is without zlib. But running the tests with the same compiler (ICC14.0.2) on the CPU or without zlib seems fine.

Related issues:
- Related to GROMACS - Bug #1542: two unit tests fail on 32-bit ARM Closed 06/30/2014
- Related to GROMACS - Bug #1546: Legacy tools tests failed on armv7a with SIGBUS Closed 07/02/2014
- Related to GROMACS - Bug #1547: MDRUN tools tests failed on armv7a with SIGBUS Closed 07/02/2014

Associated revisions
Revision 898166c3 - 08/21/2014 05:33 PM - Magnus Lundborg
Fixed TNG memory alignment problems and reset pointers.

This is a temporary fix to the memory alignment problems on some platforms. In the main TNG repository the whole I/O system is rewritten to address this problem. This fix is to avoid the problems without making too large changes.

There are also some pointers that were not reset after memory was freed. This is also fixed here.

This commit does not correspond to any commit in the TNG repository.

Fixes #1542, #1546, #1547 and #1559.

Change-Id: i90a6406ccc8c343fd57d4423c2b661019c77763e8

History
#1 - 07/10/2014 10:14 AM - Mark Abraham
Roland Schulz wrote:

The unit tests:

- mdrun-test CanWrite/Trajectories.ThatDifferInNstxout/*
- legacy-tools-test NoFatalErrorWhenWritingFrom/TrjconvWithIndexGroupSubset.WithDifferentInputFormats/*

fail. And "gmx trjconv" fails with segfault if the output is TNG.

I don't think it makes sense to have I/O code run anywhere near a Phi. Presumably to reproduce one needs to do cmake ...
-DCMAKE_TOOLCHAIN_FILE=Platform/XeonPhi, but what was the run-time setup?

This is without zlib. But running the tests with the same compiler (ICC14.0.2) on the CPU or without zlib seems fine.
Mark Abraham wrote:

I don't think it makes sense to have I/O code run anywhere near a Phi. Presumably to reproduce one needs to do cmake ...
-DCMAKE_TOOLCHAIN_FILE=Platform/XeonPhi, but what was the run-time setup?

In native mode the I/O has to happen on the Phi. And native mode is currently the only supported option (we are still working on offload). Also the next generation will be self hosted. And even if we think that we don't care about TNG on Phi, the fact that this isn't working is hinting to some bug which probably will be present on some other architectures too. The Phi card has a standard Linux environment and I'm actually surprised that it isn't working. All other Gromacs I/O, unit- and regression tests work without any problems (with no changes). No special run-time setup is needed. To make it easy to run the unit-tests, the source and build folders should be located in a directory which is mounted to the same location on the Phi (e.g. /data/gromacs or /home/.../gromacs on both host and MIC - the reason is that the refdata classes use the CMAKE_SOURCE_DIR to find its xml files).

This is without zlib. But running the tests with the same compiler (ICC14.0.2) on the CPU or without zlib seems fine.

no. I tested on a standard CPU that the unit tests pass when one disables zlib. I also tested that the same compiler version is fine on a standard CPU.

Hopefully this is related to the other TNG issues. I hope to have a fix ready quite soon.

Yes this patch fixes it. It segfaulted in quantize_float because the compiler assumed that the float* passed to it is 4-byte aligned.

Thanks for the feedback.

Was the previous code valid C code or is OK for the C compiler to assume 4-byte alignment? If it was valid it might be good to report a compiler bug.

Was the previous code valid C code or is OK for the C compiler to assume 4-byte alignment? If it was valid it might be good to report a compiler bug.