GROMACS - Bug #1811

Extrae build issues

08/20/2015 05:31 PM - Szilárd Páll

Status:	New		
Priority:	Normal		
Assignee:			
Category:	build system		
Target version:			
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:	5.1		

Description

I have tried to build with the Extrae support in 5.1 (wanting to get simple MPI/OpenMP event traces and ran into the following issues:

- The mapping of the features to library names is not entirely correct, AFAIK the ompitrace library does MPI+OpenMP and ompicudatrace does MPI+OpenMP+CUDA.
- The build scipts make no attempt to generate linker flags to at least try to link against the Extrae dependecies (nor does the script make any attempt to automatically detect these dependencies); hence, I get tons of link-stage warnings about libxml, BFD (which are AFAIK mandatory dependencies) as well as PAPI.

Given these issues the current GMX_EXTRAE CMake feature is hardly of any use, one may often have to override the library selection and assemble the link arguments manually.

The former should be fixed by:

- fixing the features to tracing library mapping (suggest in 5.1)
- allowing manual override (see attached patch) (maybe post 5.1 if the above is fixed?)

The latter needs more attention:

- for 5.1: could a helper message should be issued with some suggested linker arguments (e.g. "Bla-bla-bla: -lpapi -lxml -lbfd);
- post-5.1 a proper FindExtrae could actually look at the library dependecies and issue a more educated guess/suggestion for the link options the user has to use (looking for them manually may be an overkill).

History

#1 - 08/21/2015 02:27 PM - Szilárd Páll

Szilárd Páll wrote:

> - The build scipts make no attempt to generate linker flags to at least try to link against the Extrae dependecies (nor does the script make any attempt to automatically detect these dependencies); hence, I get tons of link-stage warnings about libxml, BFD (which are AFAIK mandatory dependencies) as well as PAPI.

Looks like I was wrong about this because when I use some Extrae libraries other than the ones I just built (and manually pick the tracing library) I did get the linking to succeed. Do the extrae libs always contain full RPATH?

#2 - 08/21/2015 02:42 PM - Szilárd Páll

Szilárd Páll wrote:

> - The mapping of the features to library names is not entirely correct, AFAIK the ompitrace library does MPI+OpenMP and ompicudatrace does MPI+OpenMP+CUDA.

One more correction:

- ompitrace is correctly mapped and ompicudatrace is in fact cudaompitrace.
- standard single-node tMPI+OpenMP+CUDA build however looks for cudaomptrace but that does not exists (and technically even this should be cudaptomptrace, right?), so omptrace would be more appropriate.

Additionally, I noticed that OpenCL is supported by extrae, so that may work too?

#3 - 08/21/2015 02:50 PM - Szilárd Páll

One more remark:

```
$ cmake . -DGMX_MPI=ON -DGMX_GPU=OFF
[...]
$ grep "EXTRAE_LIBRARY:" CMakeCache.txt 
EXTRAE_LIBRARY:FILEPATH=/opt/tcbsys/extrae/3.0.1/lib/libompitrace.so

$ cmake . -DGMX_MPI=ON -DGMX_GPU=ON
EXTRAE_LIBRARY:FILEPATH=/opt/tcbsys/extrae/3.0.1/lib/libompitrace.so
```

So if the ${extraelib} variable changes, the detection is not rerun with the new value.

#4 - 08/24/2015 11:50 AM - Rossen Apostolov

Szilárd Páll wrote:

> - standard single-node tMPI+OpenMP+CUDA build however looks for cudaomptrace but that does not exists (and technically even this should be cudaptomptrace, right?), so omptrace would be more appropriate.

The only PT libs are lib*pt*trace and lib*ptmpi*trace. And there is no lib*cudaomp*trace. Thus we're left with just lib*omp*trace for the standard single-node setup. would it suffice though?

> Additionally, I noticed that OpenCL is supported by extrae, so that may work too?

good point, I'll check that

#5 - 08/24/2015 11:57 AM - Rossen Apostolov

There is a special flag for building PT support in all libs:

--enable-pthread-support-in-all-libs
Allows all the instrumentation libraries to work
with pthreads. Caution! May add dependencies with
pthread library (disabled by default)

That might solve the pt support problem

#6 - 09/01/2015 09:41 PM - Gerrit Code Review Bot

Gerrit received a related patchset '3' for Issue #1811.
Uploader: Rossen Apostolov (rossen@kth.se)
Change-Id: Ie88c1d1a9e3d38a221c7d5a423ed29e512b1e72b
Gerrit URL: https://gerrit.gromacs.org/5043

#7 - 11/23/2015 05:48 PM - Mark Abraham

- Target version deleted (5.1.1)

#8 - 12/11/2017 05:22 PM - Erik Lindahl

Has anybody actually used Extrae for the last two years? If not (i.e., if there's no feedback here :-), it sounds like a feature begging to be killed.

#9 - 12/11/2017 09:13 PM - Szilárd Páll

Erik Lindahl wrote:

> Has anybody actually used Extrae for the last two years? If not (i.e., if there's no feedback here :-), it sounds like a feature begging to be killed.

Yes, I do use it regularly (what are others using?), but I often had to default to setting up the build semi-manually due to some of the issue I documented in detail. If I'm really the only one using it, I can set up linking myself and we can just remove this report.

#10 - 12/13/2017 12:12 AM - Erik Lindahl

How much work is it to fix this bug report instead?

There's not much point in keeping the option around if it doesn't work, so we should decide to either fix it or remove the support.

#11 - 12/13/2017 03:43 AM - Mark Abraham

I could fix it, but I doubt we'll get much new actionable insight. And if we want it, then someone else should probably work on the build system, because our bus factor is too low there.

Files

FindExtrae.cmake.patch	692 Bytes	08/20/2015	Szilárd Páll