**Gromacs - Bug #1127**

**MPI_Init_thread**

01/22/2013 10:13 PM - Roland Schulz

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Low</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Berk Hess</td>
</tr>
<tr>
<td>Category:</td>
<td>mdrun</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Affected version - extra info:</td>
<td>Affected version: 4.6</td>
</tr>
</tbody>
</table>

**Description**

With OpenMP we are suppose to call MPI_Init_thread not MPI_Init. At least OpenMPI can be compiled without thread support and then it doesn't even support MPI_THREAD_FUNNELED (MPI only called from one thread - our approach). I'm not aware of any problem, but it seems rather dangerous. There could be some MPI implementation with e.g. a non-thread safe malloc (which is often overwritten by the MPI lib) or some other global function, which makes it non-thread safe even if no MPI function is called within OpenMP. So we probably should call MPI_Init_thread. It returns the provided thread level and if that is not at least MPI_THREAD_FUNNELED we should disable OpenMP.

I just asked on the OpenMPI user list ([http://www.open-mpi.org/community/lists/users/2013/01/21196.php](http://www.open-mpi.org/community/lists/users/2013/01/21196.php)) why MPI_Init_thread return MPI_THREAD_SINGLE when requesting MPI_THREAD_FUNNELED if compiled without enable-mpi-thread-multiple.

**Associated revisions**

Revision 12768d14 - 06/22/2015 02:20 PM - Erik Lindahl

Use MPI_THREAD_FUNNELED when available

We have never observed any problems with MPI and OpenMP, but for compliance we should call MPI_Init_thread() and try to get MPI_THREAD_FUNNELED support level. However, if that level is not supported we simply call the old MPI_Init() instead - at least for Gromacs that seems fine.

If we get an error return code we warn the user, but if MPI_Init_thread() still worked we hope for the best and don't bother the user.

Fixes #1127.

Change-Id: I11b81a65125e32b95255dbb769cf86b835bd62ab

**History**

#1 - 02/05/2013 05:59 AM - Roland Schulz

- Priority changed from Normal to High

Increasing to high priority because it seems that this is a real issue. It is unsupported to use MPI_THREAD_FUNNELED with OpenMPI when compiled without enable-mpi-thread-multiple. See: [http://www.open-mpi.org/community/lists/users/2012/10/20423.php](http://www.open-mpi.org/community/lists/users/2012/10/20423.php) and [http://www.open-mpi.org/community/lists/users/2013/01/21256.php](http://www.open-mpi.org/community/lists/users/2013/01/21256.php). I still haven't noticed this to produce a problem, but of course that doesn't mean it is OK, given that threading bugs are often non-trivial to notice.
This whole issue is made worse by the fact that compiling with enable-mpi-thread-multiple has a performance impact and silently disables OpenFabrics support: [http://www.open-mpi.org/community/lists/users/2012/10/20587.php](http://www.open-mpi.org/community/lists/users/2012/10/20587.php). enable-opal-multi-threads is sufficient, but I don't know whether it has any performance impact. Also it is made worse by OpenMPI v1.2 returned MPI_THREAD_SINGLE even when FUNNELED was supported: [http://www.open-mpi.org/community/lists/users/2007/05/3330.php](http://www.open-mpi.org/community/lists/users/2007/05/3330.php).

#2 - 02/05/2013 01:16 PM - Berk Hess

This looks like a very problematic issue.

My Open MPI 1.5.4 installation on OpenSUSE reports MPI_THREAD_SINGLE, but I never noticed any issues running OpenMP parallel code. So adding this check might reduce performance on many installations which actually do (seem to?) work with only one thread doing MPI calls.

#3 - 02/05/2013 04:32 PM - Roland Schulz

I think we should definitely change our code, because independent of the OpenMPI issue, our code isn't MPI compliant and there could be some MPI implementation which does things different with MPI_THREAD_SINGLE/MPI_THREAD_FUNNELED (e.g. (non)-thread safe malloc).

My question is, whether for some OpenMPI versions, we should ignore that the provided thread level and use OpenMP even if MPI_THREAD_SINGLE is returned. Given that enable-opal-multi-threads documentation isn't really helpful and questions about this ([http://www.open-mpi.org/community/lists/devel/2010/01/7275.php](http://www.open-mpi.org/community/lists/devel/2010/01/7275.php), [http://www.open-mpi.org/community/lists/users/2011/05/16451.php](http://www.open-mpi.org/community/lists/users/2011/05/16451.php)) haven't been answered, I really don't know. Looking at the code enable-opal-multi-threads OPAL_HAVE_THREAD_SUPPORT activates volatiles and mutexes in the communication layer. I'm see how this might be required for MPI_THREAD_SERIALIZED but I can't see how this is required for MPI_THREAD_FUNNELED. So I suspect this to be a bug in OpenMPI and that MPI_THREAD_FUNNELED is usually OK even without enable-opal-multi-threads. But I'm not sure it is always OK, e.g. when OpenMPI is compiled without thread support (doesn't find neither posix threads but with ptmalloc2 then it seems like the malloc isn't thread safe. Unless we get a better answer from the OpenMPI developers, maybe the best approach would be:
- ask for MPI_THREAD_FUNNELED
- disable OpenMP if provided level is only MPI_THREAD_SINGLE
- unless MPI is OpenMPI. Then only print a warning.

#4 - 04/29/2013 07:53 PM - Mark Abraham

- Target version deleted (4.6.1)
- Affected version set to 4.6

#5 - 06/14/2014 09:12 PM - Roland Schulz

- Priority changed from High to Low

Since no one has observed a problem - it seems only a compliance issue -> priority low.

#6 - 06/18/2015 10:03 PM - Gerrit Code Review Bot

Gerrit received a related patchset ‘1’ for Issue #1127.
Uploader: Erik Lindahl (erik.lindahl@gmail.com)
Change-id: 11b61a65125e32b95255dbb769c66b835bd62ab
Gerrit URL: [https://gerrit.promacs.org/4742](https://gerrit.promacs.org/4742)

#7 - 06/18/2015 10:04 PM - Erik Lindahl
- Status changed from New to Fix uploaded

#8 - 06/29/2015 03:25 PM - Erik Lindahl
- Status changed from Fix uploaded to Resolved

#9 - 06/29/2015 03:25 PM - Erik Lindahl
- Status changed from Resolved to Closed