Charm++ - Bug #2030

tests/ampi/megampi sometimes fails on mpi-win-x86_64-smp

11/19/2018 04:28 PM - Evan Ramos

<table>
<thead>
<tr>
<th>Status:</th>
<th>New</th>
<th>Start date:</th>
<th>11/19/2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Evan Ramos</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>6.10.0</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

Description
This failure shows up in autobuild every few days.

```
../../bin/testrun ./pgm +p2 +vp4
Running on 2 processors: ./pgm +vp4
charmrun> /cygdrive/c/Program Files/Microsoft MPI/Bin/mpiexec -n 2 ./pgm +vp4
Charm++> Running on MPI version: 2.0
Charm++> level of thread support used: MPI_THREAD_FUNNELED (desired: MPI_THREAD_FUNNELED)
Charm++> Running in SMP mode: 2 processes, 1 worker threads (PEs) + 1 comm threads per process, 2 PEs total
Charm++> The comm. thread both sends and receives messages
Charm++ warning> fences and atomic operations not available in native assembly
Converse/Charm++ Commit ID: v6.9.0-0-gc3d50ef
Charm++> Disabling isomalloc because mmap() does not work.
CharmLB> Load balancer assumes all CPUs are same.
Charm++> Running on 1 hosts (1 sockets x 4 cores x 2 PUs = 8-way SMP)
Charm++> cpu topology info is gathered in 0.016 seconds.
CharmLB> RandCentLB created.

job aborted:
[ranks] message
[0] terminated
[1] process exited without calling finalize

---- error analysis ----

[1] on CS-DEXTERITY
./pgm ended prematurely and may have crashed. exit code 0xc0000005

---- error analysis ----

make[1]: *** [Makefile:37: test-ampi] Error 2
make[3]: Leaving directory '/home/nikhil/autobuild/mpi-win-x86_64-smp/charm/mpi-win-x86_64-smp/tests/ampi/megampi'
```

http://charm.cs.illinois.edu/autobuild/old.2018_11_06__01_01/mpi-win-x86_64-smp.txt
http://charm.cs.illinois.edu/autobuild/old.2018_11_10__01_01/mpi-win-x86_64-smp.txt
http://charm.cs.illinois.edu/autobuild/old.2018_11_14__01_01/mpi-win-x86_64-smp.txt

History
#1 - 11/29/2018 12:57 PM - Eric Bohm
- Assignee set to Evan Ramos

03/14/2019
- Target version deleted (6.9.1)

I think this showed up in a multicore-win-x86_64 build today, in addition to mpi-win-x86_64-smp:

http://charm.cs.illinois.edu/autobuild/old.2019_01_11__01_07/multicore-win-x86_64.txt
http://charm.cs.illinois.edu/autobuild/old.2019_01_11__01_07/mpi-win-x86_64-smp.txt

It happened on mpi-win-smp today, and generally seems to happen somewhat frequently though not everytime.

Target version set to 6.10.0

I managed to catch this crash in Visual Studio’s debugger. ampi::getRank() is called with this pointing to garbage.

```
pgm.exe!ampi::getRank() Line 2588
  at tmp\libs\ck-libs\ampi\ampiimpl.h(2588)
pgm.exe!MPI_Comm_free(int * comm) Line 8994
  at tmp\libs\ck-libs\ampi\ampi.c(8994)
pgm.exe!AMPI_Main_cpp(int argc, char ** argv) Line 494
  at tests\ampi\megampi\test.c(494)
pgm.exe!AMPI_Fallback_Main(int argc, char * argv) Line 830
  at tmp\libs\ck-libs\ampi\ampi.c(830)
pgm.exe!MPI_threadstart_t::start() Line 1059
  at tmp\libs\ck-libs\ampi\ampi.c(1059)
pgm.exe!MPI_threadstart(void * data) Line 1076
  at tmp\libs\ck-libs\ampi\ampi.c(1076)
pgm.exe!startTCharmThread(TCharmInitMsg * msg) Line 164
  at tmp\libs\ck-libs\tcharm\tcharm.c(164)
pgm.exe!FiberSetUp(void * fiberData) Line 1371
  at tmp\threads.c(1371)
[External Code]
```

I tried the following change to help diagnose the problem:

```
index dad98cf50..8f32e8e77 100644
--- a/src/libs/ck-libs/ampi/ampi.C
+++ b/src/libs/ck-libs/ampi/ampi.C
@@ -8990,7 +8990,9 @@ AMPI_API_IMPL(int, MPI_Comm_free, MPI_Comm *comm)
 // get = parent->freeUserKeyvals(*comm, parent->getKeyvals(*comm));
 if (*comm != MPI_COMM_WORLD && *comm != MPI_COMM_SELF) {
     ampi* ptr = getAmpiInstance(*comm);
+    CmiEnforce(*comm == ptr->getCommStruct().getComm()); // assertion 1
    ptr->barrier();
+    CmiEnforce(*comm == ptr->getCommStruct().getComm()); // assertion 2
    if (ptr->getRank() == 0) {
        CProxy_CkArray(ptr->ckGetArrayID()).ckDestroy();
    }
```

The odd thing is that assertion 1 succeeds but assertion 2 fails.

ptr->barrier() calls thread->suspend(), which calls CthSuspend(). I suspect the problem is there.

Alternatively, there is the following comment in tcharm_impl.h:
/* SUBTLE: We have to do the get() because "this" may have changed  
 * during a migration-suspend. If you access *any* members  
 * from this point onward, you'll cause heap corruption if  
 * we're resuming from migration!  (OSL 2003/9/23) */

I tried changing assertion 2 to CmiEnforce(*comm == getAmpiInstance(*comm)->getCommStruct().getComm()); but it still failed, just with a null pointer deference here:

```
pgm.exe!CkArray::lookup(const CkArrayIndex & idx) Line 595
   at tmp\ckarray.h(595)
pgm.exe!CProxyElement_ArrayBase::ckLocal() Line 743
   at tmp\ckarray.c(743)
pgm.exe!CProxyElement_ArrayElement::ckLocal() Line 1031
   at include\ckarray.decl.h(1031)
pgm.exe!ampiParent::comm2ampi(int comm) Line 2170
   at tmp\ck-lib\ampi\ampiimpl.h(2170)
pgm.exe!getAmpiInstance(int comm) Line 3799
   at tmp\lib\ck-libs\ampi\ampi.c(3799)
pgm.exe!MPI_Comm_free(int * comm) Line 8996
   at tmp\lib\ck-libs\ampi\ampi.c(8996)
pgm.exe!AMPI_Fallback_Main(int argc, char * * argv) Line 494
   at tests\ampi\megampi\test.c(494)
pgm.exe!AMPIClient:: AMPIClient(int argc, char * * argv) Line 830
   at lib\ck-libs\ampi\ampi.c(830)
pgm.exe!MPI_threadstart_t::start() Line 1059
   at lib\ck-libs\ampi\ampi.c(1059)
pgm.exe!AMPIService::AMPIService(void * data) Line 1076
   at lib\ck-libs\ampi\ampi.c(1076)
pgm.exe!startTCharmThread(TCharmInitMsg * msg) Line 164
   at lib\ck-libs\tcharm\tcharm.c(164)
pgm.exe!FiberSetUp(void * fiberData) Line 1371
   at tmp\threads.c(1371)
   [External Code]
```

I am more inclined to believe the problem is in CthSuspend(); because this failure does not always occur, and it only occurs on Windows.