Charm++ - Bug #1640

Segfault during migration for AMPI in SMP mode with "-tracemode projections"

07/22/2017 05:44 PM - Matthias Diener

Status: In Progress
Priority: Normal
Assignee: Sam White
Category: AMPI
Start date: 07/22/2017
Due date: 
% Done: 0%
Estimated time: 0.00 hour
Target version: 6.9.0
Spent time: 0.00 hour

Description

When running make OPTS="-tracemode projections" test in examples/ampi/Cjacobi3d, the application crashes after the first migration:

```
[0] RotateLB created
iter 1 time: 0.189357 maxerr: 2020.200000
iter 2 time: 0.180202 maxerr: 1696.968000
iter 3 time: 0.179076 maxerr: 1477.170240
iter 4 time: 0.179912 maxerr: 1319.433024
iter 5 time: 0.191689 maxerr: 1200.918072

CharmLB> RotateLB: PE [0] step 0 starting at 1.255664 Memory: 157.007812 MB
CharmLB> RotateLB: PE [0] strategy starting at 1.256069
CharmLB> RotateLB: PE [0] Memory: LBManager: 920 KB CentralLB: 2 KB
CharmLB> RotateLB: PE [0] #Objects migrating: 8, LBMigrateMsg size: 0.00 MB
CharmLB> RotateLB: PE [0] strategy finished at 1.256085 duration 0.000016 s
```

------------- Processor 2 Exiting: Caught Signal -------------
Reason: segmentation violation
Suggestion: Try running with '++debug', or linking with '-memory paranoid' (memory paranoid requires '+netpoll' at runtime).

```
[2] Stack Traceback:
[2:0] [0x6b526b]
[2:1] +0xf370 [0x7ffadf516370]
[2:2] CmiIsomallocBlockListPup+0x270 [0x6cc580]
[2:3] _ZN6TCharm9pupThreadERN3PUP2erE+0x69 [0x4ec539]
[2:4] _ZN6TCharm3pupERN3PUP2erE+0x334 [0x4eca34]
[2:5] _ZN8CkLocMgr14pupElementsForERN3PUP2erEP8CkLocRec19CkElementCreation_tb+0x1d5 [0x5d6605]
[2:6] _ZN8CkLocMgr8emigrateEP8CkLocReci+0xc9 [0x5d4bb9]
[2:7] _ZN8CkLocRec13staticMigrateE11LDObjHandlei+0x2d [0x5d507d]
[2:8] _ZN4LBDB7MigrateE11LDObjHandlei+0xe7 [0x657e57]
[2:9] LDMigrate+0x3c [0x637e4c]
[2:10] _ZN9CentralLB23ProcessReceiveMigrationEv+0x167 [0x669477]
[2:11] __ZN17CKIndex_CentralLB47_call_redn_wrapper_ProcessReceiveMigration_voidEPvS0_+0xc [0x66962c]
[2:12] CkDeliverMessageFree+0x39 [0x5abae9]
[2:13] [0x5abe35]
[2:14] __Z15_processHandlerPvP11CKCoreState+0x49c [0x5b10ec]
[2:15] CsdScheduleForever+0x70 [0x6bb760]
[2:16] CsdScheduler+0x2d [0x6bb29d]
[2:17] [0x6b88aa]
[2:18] ConverseInit+0x1b8 [0x6b9868]
[2:19] main+0x21 [0x4e83a1]
[2:20] __libc_start_main+0xf5 [0x7fade742b35]
[2:21] [0x4e83f0]
```

Fatal error on PE 2> segmentation violation

Happened in netlrts-{darwin,linux} and seems to be limited to AMPI. Maybe caused by some of the recent tracing changes.

History
Going back to commit 9df608634 (which is the one before AMPI tracing changes were merged) shows the same crash, so the error might be somewhere else. Happens only in SMP mode.

Subject changed from Segfault in pup routines when running with "-tracemode projections" and migrations to Segfault during migration for AMPI in SMP mode with "-tracemode projections"

I can reproduce this, and verify that the failure only happens in SMP mode with tracing on. Additionally, it only happens with the version of jacobi3D that uses PUP; if you use Isomalloc it passes, even though the failure for non-Isomalloc is happening inside the Isomalloc pup routines for migrating thread stacks.

Megampi exhibits the same problem.

It looks like this only happens when a message that is for a recipient VP on the same PE as the sender is sent inline via direct invocation of ampi::genericRdma or ampi::generic. I'm not sure what exactly is going wrong here yet, but for the release we could at least add a build option like AMPI_LOCAL_IMPL. It's a narrow enough case (AMI + SMP mode + tracemode projections + not Isomalloc) that we shouldn't hold up the release for a full fix.

Without digging into the code, I'm guessing the issue is that the tracing code allocates stack objects to track function entry/exit, and those objects end up retaining pointers to PE-local state that won't migrate along with the object. Possibly some interaction with the nesting potentially created by [inline] calls.

We're not actually using [inline] entry methods, we're just calling the C++ methods directly on the object returned by ckLocal().

I tested this with 6.7.1, which crashes as well. So it is definitely not a regression. Looking at Phil's suggestion, when defining AMPIAPI as empty, the crash does not occur, so the bug seems indeed related to the TCharmAPIRoutine object created on the stack.

EDIT: This comment is probably wrong.

Hmm when I had tested it with 6.7.1 on netlrts-darwin-x86_64-smp, I think it passed, but maybe I didn't run it enough times to trigger. If it does crash, then my diagnosis is off, there was no PE-local optimization in AMPI in 6.7.1.

Hmm, testing it again, the crash does seem somewhat different (no mention of Isomalloc), and happens only rarely, so it might be a different issue.

6.7.1 shipped with a bug in MPI_Info's handling of strings, which would show up in AMPI_Migrate(MPI_Info) calls. That might be what you are seeing,
but it's not dependent on SMP mode or tracing or Isomalloc...

#11 - 07/31/2017 11:55 AM - Matthias Diener
I bisected this bug to the following commit:
22ac66875b1b90c52c54b1327efddd5f5816abfcd
AMPI: execute local sends of contiguous data inline using direct memcpy

https://charm.cs.illinois.edu/gerrit/#/c/2450/

#12 - 07/31/2017 02:43 PM - Sam White
This gives users a way to build with inline messaging disabled as a workaround: https://charm.cs.illinois.edu/gerrit/#/c/2849/

#13 - 07/31/2017 04:04 PM - Matthias Diener
Does disabling inline messaging fix this bug? If yes, should we disable inline messaging also when CMK_TRACE_ENABLED is true?
Edit: Building with -DAMPI_LOCAL_IMPL=0 did not seem to fix this bug.

#14 - 07/31/2017 06:16 PM - Sam White
Hmm, it fixed the issue for me on netlirts-darwin-x86_64-smp. What build are you running?

#15 - 07/31/2017 06:20 PM - Matthias Diener
I also tried with netlirts-darwin-x86_64-smp. Could you post your full build line? Maybe there is an issue with argument ordering.

#16 - 07/31/2017 06:28 PM - Sam White

./build AMPI netlirts-darwin-x86_64 smp -j16 -g -DAMPI_LOCAL_IMPL=0

#17 - 07/31/2017 10:14 PM - Matthias Diener
Edit:
Applying https://charm.cs.illinois.edu/gerrit/#/c/2849/ and compiling with -DAMPI_LOCAL_IMPL=0 indeed seems to fix this crash. We should consider making AMPI_LOCAL_IMPL=0 when running with CMKTRACE_ENABLED for 6.8.0.

#18 - 08/01/2017 12:05 PM - Eric Bohm
- Assignee set to Matthias Diener

#19 - 08/01/2017 12:06 PM - Eric Bohm
- Status changed from New to In Progress

#20 - 08/01/2017 02:58 PM - Matthias Diener
- Status changed from In Progress to Implemented

#21 - 08/01/2017 02:58 PM - Matthias Diener
https://charm.cs.illinois.edu/gerrit/#/c/2849/ works around this issue.
#22 - 08/01/2017 07:32 PM - Sam White

- Priority changed from High to Normal
- Target version changed from 6.8.0 to 6.8.1
- Status changed from Implemented to In Progress
- Assignee changed from Matthias Diener to Sam White

The above workaround was merged for 6.8.0, but we still need to fix the underlying issue.

#23 - 08/22/2017 07:59 PM - Sam White

- Target version changed from 6.8.1 to 6.9.0