Charm++ - Bug #1956

tests/charm++/sdag/migration and tests/charm++/sdag/anytimeMigration fail on mpi-win-x86_64-smp with debug options (-g -O0)

08/07/2018 03:24 PM - Nitin Bhat

Status: New  Start date: 08/07/2018
Priority: High  Due date: 
Assignee: Eric Mikida  % Done: 0%
Category:  Estimated time: 0.00 hour
Target version:  Spent time: 0.00 hour

Description
Charm build command: ./build LIBS mpi-win-x86_64 smp --enable-error-checking --without-romio --suffix=debug -j8 -g -O0 |& tee build_result_debug

tests/charm++/sdag/migration:

make -C migration test OPTS='' TESTOPTS=''
make[1]: Entering directory '/home/nikhil/nitin/charm/mpi-win-x86_64-smp-debug/tests/charm++/sdag/migration'
../.../..../bin/testrun +p2 ./test1

Running on 2 processors: ./test1
charmrun> /cygdrive/c/Program Files/Microsoft MPI/Bin/mpiexec -n 2 ./test1

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, 0 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: v6.8.2-853-g4146bf788
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 0.000 seconds.
Running Parallel on 2 processors for 25 elements

Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0
Migrating from PE 0

(0, 0): received finished message
(0, 1): received finished message
(0, 2): received finished message
(0, 3): received finished message
(0, 4): received finished message
(1, 0): received finished message
(1, 1): received finished message
(1, 2): received finished message
(1, 3): received finished message
(1, 4): received finished message
(2, 0): received finished message
(2, 1): received finished message
(2, 2): received finished message
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell unpacking or sizing
called PUP for cell packing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell unpacking or sizing
called PUP for cell packing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell unpacking or sizing
called PUP for cell packing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
Migrating from PE 1
called PUP for cell unpacking or sizing
called PUP for cell packing
called PUP for cell unpacking or sizing
job aborted:
[ranks] message
[0] terminated
[1] process exited without calling finalize

---- error analysis ----
[1] on CS-DEXTERTY
./test1 ended prematurely and may have crashed. exit code 0xc0000417

---- error analysis ----
m[1]: *** [Makefile:28: test] Error 127
make[1]: Leaving directory '/home/nikhil/nitin/charm/mpi-win-x86_64-smp-debug/tests/charm++/sdag/migration'
tests/charm++/sdag/anytimeMigration:

make[1]: Entering directory '/home/nikhil/nitin/charm/mpi-win-x86_64-smp-debug/tests/charm++/sdag/ anytimeMigration'
./..././bin/charm migration.ci
./..././bin/charm migration.C -o migration
moduleinit5796.C
Ignored Unrecognized argument -Wl,--export-dynamic
Running on 2 processors: ./migration
charmrun> /cygdrive/c/Program Files/Microsoft MPI/Bin/mpiexec -n 2 ./migration
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, 0
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: v6.8.2-853-g4146bf788
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 0.000 seconds.
running SDAG migration test

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

----- error analysis -----

[1] on CS-DEXTERITY
./migration ended prematurely and may have crashed. exit code 0xc0000417

----- error analysis -----
make[1]: Leaving directory '/home/nikhil/nitin/charm/mpi-win-x86_64-smp-debug/tests/charm++/sdag/anytimeMigration'

History
#1 - 08/08/2018 12:51 PM - Eric Bohm
On a related note, the format of testing for mpi-win-smp seems to be running all tests with PPN = 1. Which means its not really exercising the multiple worker thread aspect that motivates use of the smp target. So all of PPN>1 could be broken and nightly build would not tell us. This creates a strong expectation shortfall. When looking at autobuild for mpi-win-smp we are misled into believing we have a good test for it, when we are in fact barely hitting any of the interesting smp use cases.

#2 - 08/21/2018 03:10 PM - Evan Ramos
- Target version deleted (6.9.0)

#3 - 09/20/2018 01:32 PM - Eric Bohm
- Assignee set to Eric Mikida