Charm++ - Bug #1520

Bug # 259 (In Progress): Bugs exposed by use of randomized Q

**multicore-darwin-x86_64 megatest hangs when built with --enable-randomized-msgq --with-prio-type=int --enable-error-checking -debug**

04/20/2017 11:16 AM - Jim Phillips

<table>
<thead>
<tr>
<th>Status</th>
<th>New</th>
<th>Start date:</th>
<th>04/20/2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td></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>6.8.1</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

**Description**

`multicore-darwin-x86_64 --no-build-shared --enable-randomized-msgq --with-prio-type=int --enable-error-checking -debug`

jim@roswell% /Projects/namd2/charm-6.8.0-debug/multicore-darwin-x86_64/bin/megatest +p1
Charm++: standalone mode (not using charmrn)
Charm++ Running in Multicore mode: 1 threads
Converse/Charm++ Commit ID: v6.8.0-beta1-60-g4b5b444-namd-charm-6.8.0-debug-build-2017-Apr-20-133877
Warning> using Isomalloc in SMP mode, you may need to run with '+isomalloc_sync'.
Charm++ Using STL-based msgQ:
Charm++ Using randomized msgQ. Priorities will not be respected!
CharmLB> Load balancer assumes all CPUs are same.
Charm++ Running on 1 unique compute nodes (16-way SMP).
Charm++ cpu topology info is gathered in 0.000 seconds.
Megatest is running on 1 nodes 1 processors.
test 0: initiated [groupring (milind)]
test 0: completed (0.00 sec)
test 1: initiated [nodering (milind)]
^C

--no-build-shared --enable-tracing --enable-tracing-commthread -optimize works fine.

--no-build-shared --with-production works fine.

netlrts-darwin-x86_64 works fine.

**History**

#1 - 04/20/2017 11:21 AM - Jim Phillips
Hangs in test 0 on multicore-linux64-iccstatic --no-build-shared --enable-randomized-msgq --with-prio-type=int --enable-error-checking -debug

jim@sunnyvale%/Projects/namd2/charm-6.8.0-debug/multicore-linux64-iccstatic/bin/megatest +p1
Charm++: standalone mode (not using charmrn)
Charm++ Running in Multicore mode: 1 threads
Converse/Charm++ Commit ID: v6.8.0-beta1-60-g4b5b444-namd-charm-6.8.0-debug-build-2017-Apr-20-133877
Warning> Randomization of stack pointer is turned on in kernel, thread migration may not work! Run 'echo 0 > /proc/sys/kernel/randomize_va_space' as root to disable it, or try run with '+isomalloc_sync'.
Charm++ Using STL-based msgQ:
Charm++ Using randomized msgQ. Priorities will not be respected!
CharmLB> Load balancer assumes all CPUs are same.
Charm++ Running on 1 unique compute nodes (52-way SMP).
Charm++ cpu topology info is gathered in 0.000 seconds.
Megatest is running on 1 nodes 1 processors.
test 0: initiated [completion_test (phil)]
^C
Also, why are the tests being run in reversed order on darwin vs linux?

#2 - 04/20/2017 12:27 PM - Sam White
- Parent task set to #259

There are multiple other bugs in our test suite noted in issue #259.

#3 - 04/25/2017 04:12 PM - Sam White
- Target version changed from 6.8.0 to 6.8.1
- Category deleted (Machine Layers)

The other related test failures with randomized queues are targeted to 6.8.1.