Charm++ - Bug #1447

AMPI_Reduce is broken for derived datatypes in user-defined reductions

02/27/2017 11:21 AM - Sam White

<table>
<thead>
<tr>
<th>Status:</th>
<th>Implemented</th>
<th>Start date:</th>
<th>02/23/2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Sam White</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td>AMPI</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>6.8.0</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

Description

Users are allowed to use derived datatypes in custom reduction functions, but AMPI does not support that and might silently corrupt memory in such a case.

History

#1 - 03/02/2017 01:31 PM - Phil Miller
As noted in meeting, there are two potential quick fixes to at least get codes hitting this case running or failing clearly:

- Abort if this case arises
- Implement the reduction as a gather to the root, and reduce locally. In allreduce, a root rank can then broadcast the result.

#2 - 03/02/2017 01:40 PM - Sam White
Abort: https://charm.cs.illinois.edu/gerrit/#/c/2280/
I can probably get to the 'gather' implementation tomorrow or this weekend.

#3 - 03/04/2017 08:27 PM - Sam White
- Status changed from In Progress to Implemented

Implemented user-defined reductions on non-contiguous derived datatypes using the non-commutative reduction code path:
https://charm.cs.illinois.edu/gerrit/#/c/2289/

#4 - 03/05/2017 10:06 AM - Sam White
- Status changed from Implemented to Merged

#5 - 03/05/2017 10:07 AM - Sam White
- Status changed from Merged to Implemented

Oops, not yet