Charm++ - Bug #1543

Memory leaks in asynchronous array creation

04/28/2017 05:06 PM - Sam White

<table>
<thead>
<tr>
<th>Status:</th>
<th>Merged</th>
<th>Start date:</th>
<th>04/28/2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</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></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
AMPI uses the asynchronous array creation API to create split comm's.
Running valgrind on megampi with 4 vp's gives the following:

```
/usr/bin/valgrind --tool=memcheck --leak-check=yes -v --log-file=myvalg_txt.%p --trace-children=yes ./pgm +vp4
```

`==27289== 16,416 bytes in 19 blocks are definitely lost in loss record 614 of 627
==27289== at 0x4C2AB80: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==27289== by 0x6D237D: malloc_nomigrate (libmemory-default.c:724)
==27289== by 0x6E1A23: CmiAlloc (convcore.c:2930)
==27289== by 0x5EC913: envelope::alloc(unsigned char, unsigned int, unsigned short) (envelope.h:312)
==27289== by 0x5ECC45: _allocEnv(int, int, int) (envelope.h:485)
==27289== by 0x6002FA: CkAllocMsg (msgalloc.C:21)
==27289== by 0x5659E8: CMessage_CkCreateArrayAsyncMsg::alloc(int, unsigned long, int*, int) (CkArray.def.h:1978)
==27289== by 0x565AA98: CMessage_CkCreateArrayAsyncMsg::operator new(unsigned long, int) (CkArray.def.h:1965)
==27289== by 0x561F02: CkSendAsyncCreateArray(int, CkCallback, CkArrayOptions, void*) (ckarray.C:944)
==27289== by 0x5AE527: CProxy_ami::ckNew(CkArrayID const&, amiCommStruct const&, CkArrayOptions const*, CkCallback, CEntryOptions const*) (ami.def.h:2916)
==27289== by 0x58AF6B: ami::createNewChildAmpiSync() (ami.C:2049)
==27289== by 0x58BF6F: ami::splitPhase1(CkReductionMsg*) (ami.C:2082)
```

History

#1 - 04/28/2017 05:17 PM - Phil Miller
Could you run that with the option --track-origins=yes (IIRC) to get fuller detail on the leak event?

#2 - 04/28/2017 05:25 PM - Sam White
- Target version set to 6.8.0
- Assignee set to Sam White
- Status changed from New to Implemented

https://charm.cs.illinois.edu/gerrit/#/c/2465/

#3 - 04/28/2017 07:02 PM - Sam White
- Subject changed from Memory leak in AMPI split communicator creation to Memory leaks in asynchronous array creation

#4 - 05/02/2017 03:35 PM - Phil Miller
- Status changed from Implemented to Merged

05/12/2017 1/1