

art - Bug #5664

Support # 7559 (Closed): Memory check issues for Sirius A

The summary of @SimpleMemoryCheck@ looks garbaged if there are few items

03/15/2014 03:53 PM - Gianluca Petrillo

Status:	Closed	Start date:	03/15/2014
Priority:	Low	Due date:	
Assignee:	Kyle Knoepfel	% Done:	100%
Category:	Infrastructure	Estimated time:	4.00 hours
Target version:	1.13.00	Spent time:	24.00 hours
Occurs In:		Experiment:	LArSoft
Scope:	Internal	SSI Package:	art

Description

The summary of SimpleMemoryCheck looks garbaged if there are less than 8 valid items:

```
MemoryReport> Peak virtual size 2937.95 Mbytes
Key events increasing vsize:
[0] run: INVALID subRun: INVALID event: INVALID vsize = 0 deltaVsize = 0 rss = 0 delta 0
[1] run: 1 subRun: 5 event: 41 vsize = 2816.05 deltaVsize = 0 rss = 2140.74 delta 0
[2] run: 1 subRun: 5 event: 42 vsize = 2937.95 deltaVsize = 121.895 rss = 2263.19 delta 122.453
[0] run: INVALID subRun: INVALID event: INVALID vsize = 0 deltaVsize = 0 rss = 0 delta 0
[0] run: INVALID subRun: INVALID event: INVALID vsize = 0 deltaVsize = 0 rss = 0 delta 0
[0] run: INVALID subRun: INVALID event: INVALID vsize = 0 deltaVsize = 0 rss = 0 delta 0
[1] run: 1 subRun: 5 event: 41 vsize = 2816.05 deltaVsize = 0 rss = 2140.74 delta 0
[2] run: 1 subRun: 5 event: 42 vsize = 2937.95 deltaVsize = 121.895 rss = 2263.19 delta 122.453
```

History

#1 - 03/17/2014 12:02 PM - Christopher Green

- Category set to Infrastructure
- Status changed from New to Feedback
- Experiment LArSoft added
- Experiment deleted (-)
- SSI Package art added
- SSI Package deleted ()

In order to get some context for this, could we ask you to attach the full log, and an explanation or illustration of what you expected to see?

#2 - 04/09/2014 11:21 AM - Christopher Green

- Target version set to 1.10.00

#3 - 05/05/2014 11:53 AM - Christopher Green

- Target version changed from 1.10.00 to 1.14.00

#4 - 05/05/2014 11:59 AM - Christopher Green

- Target version changed from 1.14.00 to 1.13.00

#5 - 12/22/2014 09:09 AM - Kyle Knoepfel

- Assignee set to Kyle Knoepfel
- Estimated time set to 4.00 h
- Parent task set to #7559

#6 - 12/22/2014 09:10 AM - Kyle Knoepfel

- Status changed from Feedback to Assigned

#7 - 01/09/2015 10:56 AM - Kyle Knoepfel

- Status changed from Assigned to Resolved

- % Done changed from 0 to 100

The reason for the spurious output was due to a hard-coded way of keeping track of a limited number of events that were causing the most significant virtual memory problems. Based on looking at some vestigial parts of the service, it looks like the original code was meant to produce something akin to this:

```
MemoryReport> Peak virtual size 476.781 Mbytes
Key events increasing vsize:
[1] run: 1 subRun: 0 event: 10 vsize = 476.781 deltaVsize = 1.14844
[2] run: 1 subRun: 0 event: 9 vsize = 475.633 deltaVsize = 1.14844
[3] run: 1 subRun: 0 event: 8 vsize = 474.484 deltaVsize = 1.14844
[4] run: 1 subRun: 0 event: 7 vsize = 473.336 deltaVsize = 1.14844
[5] run: 1 subRun: 0 event: 6 vsize = 472.188 deltaVsize = 1.14844

Key events with largest vsize:
[1] run: 1 subRun: 0 event: 10 vsize = 476.781
[2] run: 1 subRun: 0 event: 9 vsize = 475.633
[3] run: 1 subRun: 0 event: 8 vsize = 474.484
```

This is the new summary format users will see for SimpleMemoryCheck.

For the first category, there were always 5 printed lines, and for the latter category ("Key events with largest vsize", which was not printed out), 3 lines were always printed. This was true regardless of the actual number of events that were logged. For cases with fewer than 5/3 events in their respective categories, invalid data were printed.

This problem has been resolved by introducing containers (of type `constrained_multimap`) that cannot exceed a certain size. For the first category, up to 5 **unique** keys (value of `deltaVsize`) are stored. For events that have a `deltaVsize` that is the same as one of the keys already stored in the container, an extra entry is inserted. Formally speaking the size of the container is unconstrained, but the number of unique keys cannot exceed 5. For the second category, up to 3 unique keys are stored.

The default summary is to show only 5/3 entries, respectively. However, if the full printout corresponding to the entire `constrained_multimap` object is desired, one can specify the following in his/her FHiCL file:

```
services:
{
  SimpleMemoryCheck: {
    truncateSummary: false
  }
}
```

Implemented with [fbc282dd](#).

#8 - 02/16/2015 10:32 AM - Christopher Green

- Status changed from Resolved to Closed