LArSoft - Bug #16778

Memory deallocation error running MicroBooNE reconstruction

06/06/2017 01:46 PM - Gianluca Petrillo

Status:	Resolved	**Start date:**	06/06/2017
Priority:	Normal	**Due date:**	
Assignee:	Tracy Usher	**% Done:**	100%
Category:	Reconstruction	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	8.00 hours
Occurs In:	v06_26_01_02	**Co-Assignees:**	
Experiment:	MicroBooNE		

Description

Reported via e-mail by Herbert Greenlee:

> In testing mcc8, I am observing a crash in about 1% of events in the destructor of the sparse_vector owned by recob::Wire. The crash is due to a bad deallocate.
>
> Steps to reproduce.
>
> I am running off of base release larsoft v06_26_01_02. I have checked out uboonecode branch v06_26_01_br and larreco branch v06_26_01_01_branch, each with some local updates, which may or may not be affecting the crash.
> I do not have lardataobj checked out.
>
> My source area is here:
>
> /uboone/app/users/greenlee/mcc8rel/srcs
>
> A tarball containing compiles binaries is here:
>
> /uboone/app/users/greenlee/mcc8rel/local3.tar

and following:

> Here are two crashing events:
>
> /pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17183444_1004/PhysicsRun-2016_4_8_4_5_37-0005824-00307_20160408T150623_bnb_20160408T175930_merged_20160409T090100_reco1_20160409T145657_reco2_20160530T201704_filter_20170520T004002_wflash.root
> /pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17185615_1795/PhysicsRun-2016_3_13_9_33_25-0005412-00070_20160322T071700_bnb_20160323T214727_merged_20160409T134756_reco1_20160412T053429_reco2_20160531T020148_filter_20170520T010806_wflash.root

Associated revisions

Revision 8e8c7b51 - 06/09/2017 03:38 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision ead990c1 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

Revision f9d16c55 - 06/09/2017 04:14 PM - Gianluca Petrillo

Baseline algorithm now accommodates the case of integral dynamic range.

This should solve issue #16778.

History

#1 - 06/06/2017 01:51 PM - Gianluca Petrillo

- Occurs In v06_26_01_02 added

#2 - 06/06/2017 08:59 PM - Herbert Greenlee

Latest fcl file is reco_uboone_data_mcc8_driver_stage1and2_test4.fcl

in directory /uboone/app/users/greenlee/mcc8rel/srcs/uboonecode/fcl/reco

Crashing files from with this fcl file are as follows.

/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17187014_450/PhysicsRun-2016_4_6_20_41_21-0005804-00167_20160407T064154_bnb_20160407T080221_merged_20160409T083941_reco1_20160410T050412_reco2_20160530T204501_filter_20170520T011906_wflash.root
/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17186833_412/PhysicsRun-2016_2_23_23_0_18-0005127-00114_20160225T021639_bnb_20160226T175217_merged_20160409T150345_reco1_20160412T235028_reco2_20160531T074224_filter_20170520T011701_wflash.root
/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17188543_917/PhysicsRun-2016_4_10_15_14_18-0005850-00100_20160412T095516_bnb_20160413T182800_merged_20160413T223336_reco1_20160414T105500_reco2_20160530T184455_filter_20170520T012743_wflash.root
/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17185387_1717/PhysicsRun-2016_3_27_5_29_27-0005607-00064_20160327T140931_bnb_20160328T231737_merged_20160409T091655_reco1_20160411T110144_reco2_20160530T223237_filter_20170520T010540_wflash.root

#3 - 06/07/2017 02:04 PM - Gianluca Petrillo

- Category set to Reconstruction

- Status changed from New to Assigned

- Assignee set to Gianluca Petrillo

So far, I failed to reproduce the crash with a debug version using the public v06_26_01_br branch and the FHiCL file specified, as copied from its location, running on the first input file in the first report and on the first file of note 2.
I still have to try with a profiling version, and then with the full working area specified in the report, before claiming irreproducibility.

#4 - 06/07/2017 02:47 PM - Gianluca Petrillo

- Status changed from Assigned to Feedback

I could not reproduce the problem using either:

```
PhysicsRun-2016_4_6_20_41_21-0005804-00167_20160407T064154_bnb_20160407T080221_merged_20160409T083941_reco1_20160410T050412_reco2_20160530T204501_filter_20170520T011906_wflash.root
PhysicsRun-2016_4_8_4_5_37-0005824-00307_20160408T150623_bnb_20160408T175930_merged_20160409T090100_reco1_20160409T145657_reco2_20160530T201704_filter_20170520T004002_wflash.root
```

as input files and a working area with only larreco (branch v06_26_01_01_branch) and uboonecode (branch v06_26_01_br), running the FHiCL file copied from /uboone/app/users/greenlee/mcc8rel/srcs/uboonecode/fcl/reco/reco_uboone_data_mcc8_driver_stage1and2_test4.fcl on a SCD machine (woof.fnal.gov).

Please publish feature branches with the complete code reproducing the issue. That will also give me a place where to push fixes.

#5 - 06/08/2017 09:42 AM - Herbert Greenlee

I will also post this message to issue 16778.

I think the reason you could not reproduce this crash is because you
didn't have all of the local patches and upates from my test release. As
you requested, I have now pushed these, along with all job fcl files, to
the following branches in the following git repositories.

uboonecode greenlee_deallocate
larreco v06_26_01_01_branch

If you check out these two branches and build against base release larsoft
v06_26_01_02, you should have the identical code as is crashing for me.

I have a set of crashing input files from my latest set of tests using job
fcl file reco_uboone_data_mcc8_driver_stage1and2_test6.fcl

/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17186319_221/PhysicsRun-2016_4_12_7_8_49-0005877-00025_20160412T161908_bnb_20160413T201332_merged_20160414T023354_reco1_20160414T074940_reco2_20160531T071821_filter_20170520T011238_wflash.root
/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17185034_1603/PhysicsRun-2016_4_10_15_14_18-0005850-00006_20160412T131747_bnb_20160414T173755_merged_20160415T055329_reco1_20160415T100428_reco2_20160530T185628_filter_20170520T010244_wflash.root
/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17188196_837/PhysicsRun-2016_4_7_20_48_20-0005820-00011_20160408T040322_bnb_20160408T062331_merged_20160409T091541_reco1_20160409T150857_reco2_20160530T202718_filter_20170520T012658_wflash.root
/pnfs/uboone/persistent/users/mdeltutt/v06_26_01_01/prod_bnb_reco_neutrino2016_beamfilter_goodruns_v5_tpcneutrinoidselectionii/17187417_589/PhysicsRun-2016_4_7_16_9_35-0005819-00042_20160407T2

35446_bnb_20160408T022431_merged_20160409T082635_reco1_20160409T180308_reco2_20160530T202550_fil
ter_20170520T012134_wflash.root

I verified that the first file crashes for me reliably interactively for
both prof and debug build. Here is the error message.

```
================================================================================================
====

MemoryTracker General SUMMARY (all numbers in units of Mbytes)

  Peak virtual memory usage (VmPeak)  : 2258.38 Mbytes
  Peak resident set size usage (VmHWM): 1476.57 Mbytes

ProcessStep          Module ID                                                    Vsize
RSS
================================================================================================
======
Module Construction  out1:RootOutput                                               0
0.121
Module Construction  rns:RandomNumberSaver                                         0
0.090
   .
   .
   .
Module Construction  pmtrackT0RecoLoose:T0RecoAnodeCathodePiercing                 0
0
Aborted
```

#6 - 06/08/2017 11:30 AM - Gianluca Petrillo

- % Done changed from 0 to 10

I can obtain an equivalent message now.

#7 - 06/08/2017 04:12 PM - Gianluca Petrillo

So far, I have collected the following evidence:

- the crash is caused within CalWireROI module
- the crash that happens when using as input the raw digits from noise removal does *not* happen when using the original raw digits from SimWireMicroBooNE
- no crash happens when using uboonecode v06_26_01_04
- valgrind/memcheck does not expose any foul play

Now bisecting.

#8 - 06/08/2017 04:52 PM - Gianluca Petrillo

Bisection revealed that the crash does not happen in uboonecode:39607de940f656226c8dad07fbc90f78ab5c0794, does happen in uboonecode:3cc0b0a511e697a8a5878883c49302553f54c381 .
Unfortunately the three commits in between are not compilable.
Working on making them compilable.

#9 - 06/08/2017 07:23 PM - Gianluca Petrillo

- Experiment MicroBooNE added

- Experiment deleted (-)

The crash happens only when using BaselineMostProbAve baseline algorithm.
An unrelated small bug prevented BaselineStandard algorithm from being selected; that has been fixed in commit:ce9854a9e65d4045f65dce9adc0b3f8abc16641d .
I don't know yet if BaselineMostProbAve is bad by itself or if it just triggers something else.

#10 - 06/08/2017 07:23 PM - Gianluca Petrillo

- Status changed from Feedback to Work in progress

#11 - 06/09/2017 12:38 PM - Gianluca Petrillo

- Assignee changed from Gianluca Petrillo to Tracy Usher

I have pushed an updated version of greenlee_deallocate branch with some recommended changes (the bug fix mentioned above, and added constantness to input arguments of the tools).
This will **not** solve the issue. See the next note...

#12 - 06/09/2017 12:45 PM - Gianluca Petrillo

- % Done changed from 10 to 50

The issue is triggered by uboonecode/uboone/CalData/DeconTools/BaselineMostProbAve_tool.cc line 61 and following.
This can be observed replacing roiHistVec[std::floor(2. * (holder[binIdx] - min))]++; with roiHistVec.at(std::floor(2. * (holder[binIdx] - min)))++; (change that is **not** recommended unless debugging:
The vector holder, of size nbin, can be not large enough to host all the entries. This turns into the code trying to write beyond the limits of that array ("overrun").
The mode of failure for a buffer overrun can be anything. In the reported case, the overrun area hosted heap pointers that are used when deallocating memory, and that memory was assigned to a region of a sparse_vector object, therefore destructing that object generated an invalid access. It is possible that the overrun has no consequence, if instead the overrun memory happens to be not allocated; the vector will still be shorter than due, so the physics results will be incorrect.
It is also not obvious to me that this overrun happens every time, nevertheless the most conservative assumption is that even if a job has succeeded, its results might be wrong.

Reassigning to the author.

#13 - 06/09/2017 03:54 PM - Gianluca Petrillo

- Status changed from Work in progress to Resolved

Picking it back after Tracy request for convenience.
I just extended the too-short vector by one unit.
It turns out the failure comes when the range of the ADC counts in the region of interest is exactly an integer (probably less than just an accident). In that case, the histogram, implemented as a vector, is not large enough to accommodate the maximum value, and overrun occurs.

Pushed on the branch greenlee_deallocate as uboonecode:8e8c7b51c3ed406f906fb47f844e7e8720b8be92 (together with a slight modernisation of the code).
I have tested it with a single failing file among the ones specified by Herbert, also with a modified code which would throw an exception instead of silently overrunning the array boundary.
Ready for extensive test.
The code in develop also needs to be fixed, which I did not.

#14 - 06/09/2017 03:55 PM - Gianluca Petrillo

- % Done changed from 50 to 100

#15 - 06/09/2017 04:00 PM - Tracy Usher

Thank you Gianluca for finding and recommending fix!
I have patched the version on the feature branch off develop (using art tools). I didn't include exception checking but hopefully this is no longer necessary.
Ah, I'm reminded to make the input vector const... will do that straight away.