LArSoft - Bug #21041

Segfault in BlurredClusteringAlg (larreco)

10/05/2018 09:43 AM - Dominic Brailsford

Status:	Resolved	**Start date:**	10/05/2018
Priority:	Normal	**Due date:**	
Assignee:	Kyle Knoepfel	**% Done:**	100%
Category:	Reconstruction	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	32.00 hours
Occurs In:		**Co-Assignees:**	
Experiment:	SBND		

Description

sbndcode/larsoft version: v07_06_01

A small subset of SBND's test reconstruction jobs (3/100 jobs) fail with a segfault.

Here is the full gdb bt:

```
#0  0x00007ffff75da839 in std::_Bit_reference::operator bool (this=0x7fffffe53f0)
    at /scratch/workspace/canvas-products/vcheckpoint/e17/SLF6/debug/build/gcc/v7_3_0/Linux64bit+2
.6-2.12/include/c++/7.3.0/bits/stl_bvector.h:81
#1  0x00007fffe07c4d7a in cluster::BlurredClusteringAlg::GaussianBlur (this=0xd284b20, image=...)
    at /scratch/workspace/build-larsoft/v07_06_01/SLF6/debug/build/larreco/v07_04_01/src/larreco/R
ecoAlg/BlurredClusteringAlg.cxx:576
#2  0x00007fffe0ef0016 in cluster::BlurredClustering::produce (this=0xd284790, evt=...)
    at /scratch/workspace/build-larsoft/v07_06_01/SLF6/debug/build/larreco/v07_04_01/src/larreco/C
lusterFinder/BlurredClustering_module.cc:215
#3  0x00007ffff646f8d7 in art::EDProducer::doEvent (this=0xd284790, ep=..., cpc=..., counts=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Core/E
DProducer.cc:25
#4  0x00007ffff6505261 in art::WorkerT<art::EDProducer>::implDoProcess (this=0xd289ef0, ep=..., cp
c=0x7fffffe6840, stats=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Core/W
orkerT.h:88
#5  0x00007ffff72bb575 in art::Worker::ImplDoWork<(art::BranchActionType)2>::invoke<art::EventPrin
cipal> (w=0xd289ef0, p=..., cpc=0x7fffffe6840)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Princi
pal/Worker.h:201
#6  0x00007ffff72b0d63 in art::Worker::doWork<art::ProcessPackage<(art::Level)4> > (this=0xd289ef0
, p=..., cpc=0x7fffffe6840)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Princi
pal/Worker.h:259
#7  0x00007ffff72bb728 in art::WorkerInPath::runWorker<art::ProcessPackage<(art::Level)4> > (this=
0xd28c3d0, ep=..., cpc=0x7fffffe6840)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Core/W
orkerInPath.h:107
#8  0x00007ffff72b1c41 in art::Path::process<art::ProcessPackage<(art::Level)4> > (this=0xd28b9d0,
 ep=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Core/P
ath.h:148
#9  0x00007ffff72bb3fe in bool art::Schedule::runTriggerPaths_<art::ProcessPackage<(art::Level)4>
 >(art::ProcessPackage<(art::Level)4>::MyPrincipal&)::{lambda(auto:1)#1}::operator()
<art::Path*> (__closure=0x7fffffe68c0, p=0xd28b9d0) at /scratch/workspace/art-release-build/SLF6/
debug/build/art/v2_11_03/src/art/Framework/Core/Schedule.h:156
#10 0x00007ffff72bb4c4 in art::Schedule::doForAllEnabledPaths_<bool art::Schedule::runTriggerPaths
_<art::ProcessPackage<(art::Level)4> >(art::ProcessPackage<(art::Level)4>::MyPrinci
pal&)::{lambda(auto:1)#1}>(bool art::Schedule::runTriggerPaths_<art::ProcessPackage<(art::Level)4>
 >(art::ProcessPackage<(art::Level)4>::MyPrincipal&)::{lambda(auto:1)#1}) (
    this=0x7789080, functor=...) at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_1
1_03/src/art/Framework/Core/Schedule.h:180
#11 0x00007ffff72b0a65 in art::Schedule::runTriggerPaths_<art::ProcessPackage<(art::Level)4> > (th
```

```
is=0x7789080, ep=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Core/S
chedule.h:156
#12 0x00007ffff729e1db in art::Schedule::process<art::ProcessPackage<(art::Level)4> > (this=0x7789
080, principal=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Core/S
chedule.h:127
#13 0x00007ffff72908a2 in art::EventProcessor::process_<art::ProcessPackage<(art::Level)4> > (this
=0x7ffffffe6f80, p=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.h:205
#14 0x00007ffff727fa80 in art::EventProcessor::processEvent (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:892
#15 0x00007ffff727ccf9 in art::EventProcessor::process<(art::Level)4> (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:426
#16 0x00007ffff72b9ebd in void art::EventProcessor::process<(art::Level)3>()::{lambda()#2}::operat
or()() const (__closure=0x7ffffffe6cc0)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:450
#17 0x00007ffff72c127f in art::detail::ExceptionCollector::call<void art::EventProcessor::process<
(art::Level)3>()::{lambda()#2}>(void art::EventProcessor::process<(art::Level)3>()::{lambda()#2})
(this=0x7ffffffe6fa0, f=...) at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03
/src/art/Framework/EventProcessor/detail/ExceptionCollector.h:38
#18 0x00007ffff72b9fbd in art::EventProcessor::process<(art::Level)3> (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:449
---Type <return> to continue, or q <return> to quit---
#19 0x00007ffff72ab1db in void art::EventProcessor::process<(art::Level)2>()::{lambda()#2}::operat
or()() const (__closure=0x7ffffffe6d40)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:450
#20 0x00007ffff72b9ff9 in art::detail::ExceptionCollector::call<void art::EventProcessor::process<
(art::Level)2>()::{lambda()#2}>(void art::EventProcessor::process<(art::Level)2>()::{lambda()#2})
(this=0x7ffffffe6fa0, f=...) at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03
/src/art/Framework/EventProcessor/detail/ExceptionCollector.h:38
#21 0x00007ffff72ab2db in art::EventProcessor::process<(art::Level)2> (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:449
#22 0x00007ffff729bfb7 in void art::EventProcessor::process<(art::Level)1>()::{lambda()#2}::operat
or()() const (__closure=0x7ffffffe6dc0)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:450
#23 0x00007ffff72ab317 in art::detail::ExceptionCollector::call<void art::EventProcessor::process<
(art::Level)1>()::{lambda()#2}>(void art::EventProcessor::process<(art::Level)1>()::{lambda()#2})
(this=0x7ffffffe6fa0, f=...) at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03
/src/art/Framework/EventProcessor/detail/ExceptionCollector.h:38
#24 0x00007ffff729c0b7 in art::EventProcessor::process<(art::Level)1> (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:449
#25 0x00007ffff728efb3 in void art::EventProcessor::process<(art::Level)0>()::{lambda()#2}::operat
or()() const (__closure=0x7ffffffe6e40)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:450
#26 0x00007ffff729c0f3 in art::detail::ExceptionCollector::call<void art::EventProcessor::process<
(art::Level)0>()::{lambda()#2}>(void art::EventProcessor::process<(art::Level)0>()::{lambda()#2})
(this=0x7ffffffe6fa0, f=...) at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03
/src/art/Framework/EventProcessor/detail/ExceptionCollector.h:38
#27 0x00007ffff728f0b3 in art::EventProcessor::process<(art::Level)0> (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:449
#28 0x00007ffff727cd51 in art::EventProcessor::<lambda()>::operator()(void) const (__closure=0x7ff
ffffe6ec0)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:468
#29 0x00007ffff727fe94 in art::detail::ExceptionCollector::call<art::EventProcessor::runToCompleti
```

```
on()::<lambda()> >(art::EventProcessor::<lambda()>) (this=0x7ffffffe6fa0, f=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/detail/ExceptionCollector.h:38
#30 0x00007ffff727cdd9 in art::EventProcessor::runToCompletion (this=0x7ffffffe6f80)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/EventP
rocessor/EventProcessor.cc:467
#31 0x00007ffff7d4ff1c in art::run_art_common_ (main_pset=...) at /scratch/workspace/art-release-b
uild/SLF6/debug/build/art/v2_11_03/src/art/Framework/Art/run_art.cc:307
#32 0x00007ffff7d4f0bf in art::run_art (argc=7, argv=0x7ffffffe9aa8, in_desc=..., lookupPolicy=...
, handlers=...)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/src/art/Framework/Art/ru
n_art.cc:206
#33 0x00007ffff7d4abf3 in artapp (argc=7, argv=0x7ffffffe9aa8)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/build-Linux64bit+2.6-2.1
2-e17-debug/art/Framework/Art/artapp.cc:51
#34 0x00000000004015dc in main (argc=7, argv=0x7ffffffe9aa8)
    at /scratch/workspace/art-release-build/SLF6/debug/build/art/v2_11_03/build-Linux64bit+2.6-2.1
2-e17-debug/art/Framework/Art/lar.cc:9
```

and here is last larsoft/larreco line called before STL takes over and complains:

```
#1  0x00007fffe07c4d7a in cluster::BlurredClusteringAlg::GaussianBlur (this=0xd284b20, image=...)
    at /scratch/workspace/build-larsoft/v07_06_01/SLF6/debug/build/larreco/v07_04_01/src/larreco/R
ecoAlg/BlurredClusteringAlg.cxx:576
576               if (blurx < 0 and fDeadWires[x+blurx])
```

and here are the values of x and blurx

```
(gdb) p x
$1 = 20
(gdb) p blurx
$2 = -26
```

It looks like further defence is needed in the if logic (e.g. blurx < 0 && blurx < x ...) assuming that the values of blurx and x are even sensible in the file I'm looking at.

Here is the file: /sbnd/data/users/dbrailsf/bugs/tp0.4_reco_segfault/detsim-20d1b113-9db3-4f77-a03d-99c218ef3713.root
The segfault happens on event 15 so

```
lar -c standard_reco_sbnd_basic.fcl /sbnd/data/users/dbrailsf/bugs/tp0.4_reco_segfault/detsim-20d1
b113-9db3-4f77-a03d-99c218ef3713.root --nskip 14
```

Should get you right to the problematic event

History

#1 - 10/08/2018 10:36 AM - Kyle Knoepfel

- Subject changed from Segfault in BlurrerClusteringAlg (larreco) to Segfault in BlurredClusteringAlg (larreco)

- Status changed from New to Assigned

- Assignee set to Kyle Knoepfel

#2 - 10/10/2018 07:59 AM - Kyle Knoepfel

I have been able to confirm the segfault. Perhaps tellingly, I am unable to reproduce the error if I use the '--nskip 14' command-line option. Investigating further.

#3 - 10/10/2018 08:32 PM - Kyle Knoepfel

Triggering the segmentation violation is difficult--I am able to trigger it by taking the standard_reco_sbnd_basic.fcl configuration from v07_06_01, making sure it has the blurredcluster module on its path, and running that configuration against the develop branches of LArSoft.

valgrind points to various memory errors that need to be resolved within LArSoft. It is unclear at this point, how much more analysis will need to be done to understand what is going on. In an earlier test, I was unable to get a debug build of the code to trigger the segmentation violation. I will need to follow up on this.

In the end, the BlurredClusterAlg functionality **must be unit-tested** so this type of error can be avoided in the future. Stay tuned.

#4 - 10/12/2018 12:52 PM - Kyle Knoepfel

- Status changed from Assigned to Resolved

- % Done changed from 0 to 100

It was sufficient to skip any iteration steps where 'x+blurx' is less than 0. This may not be what was intended by the module author, but it seems to make sense based on the code. The code has been significantly adjusted to adopt better C++ practices, including changing signed integer types to unsigned types.

Implemented with commit larreco:e360ed2.

#5 - 10/16/2018 11:44 AM - Kyle Knoepfel

- Status changed from Resolved to Assigned

- % Done changed from 100 to 50

The commit I pushed in fact caused another problem, so this issue is not yet resolved, and I have reverted the commit.

#6 - 10/18/2018 11:00 AM - Kyle Knoepfel

- Status changed from Assigned to Resolved

- % Done changed from 50 to 100

After meeting with Mike Wallbank, the original author of the code in question, we decided that a simple check on the value of 'x+blurx' was sufficient and that migrating from signed integer to unsigned integer types creates problems that are not readily soluble.

Fixed with commit larreco:c370a6d.