

## artdaq - Support #22159

### Determine why artdaqDriver performance suffers with many shared memory buffers

03/19/2019 09:22 AM - Eric Flumerfelt

<b>Status:</b>	Closed	<b>Start date:</b>	03/19/2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Eric Flumerfelt	<b>% Done:</b>	100%
<b>Category:</b>	Additional Functionality	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	artdaq_core v3_05_09	<b>Co-Assignees:</b>	
<b>Experiment:</b>	-		
<b>Description</b>			
Gennadiy and Ron reported that when using many buffers (O(1000)) in the SBND system, the performance was actually worse than when using O(100) buffers. This is mostly likely due to inefficiencies in SharedMemoryManager.			

#### History

##### #1 - 03/20/2019 12:37 PM - Eric Flumerfelt

- Assignee set to Eric Flumerfelt
- Status changed from New to Resolved
- Category set to Additional Functionality

I have made a small set of improvements in artdaq-core:feature/22159\_SMM\_ManyBufferImprovements and artdaq:feature/22159\_SMEM\_PerformanceImprovements using profile-guided optimization.

I was specifically targeting the SBND case where O(1 kHz) of O(100 KB) Fragments was desired. The changes on the artdaq-core branch, in particular, drastically improve performance when using many ( $\geq 1000$ ) shared memory buffers.

##### #2 - 11/01/2019 01:34 PM - John Freeman

- File driver\_test20000.fcl added
- File driver\_test2000.fcl added
- File driver\_test200.fcl added

I've performed comparisons of artdaq-core and artdaq at the head of their develop branches (2c73f1ce0d9e66ea6e7c302b1fd563ec73beb61a and 8d9c7a305666b65d22dfecaf557f1e2e3c8009d4, respectively) versus at the head of artdaq-core's feature/22159\_SMM\_ManyBufferImprovements branch (16589d86080ba8769613d324a51a81dd74372523) and the head of artdaq's feature/22159\_SMEM\_PerformanceImprovements branch (210ac95358a329c2939ab760a4f3204ec76757c2). Some results.

First, for artdaqDriver: taking Eric's original test FHiCL documents, but reducing the events processed from 1M to 100k (and attached as files to this Issue), I found the following for the develop branches:

```
driver_test200.fcl (buffer_count 200):
real0m19.130s
user0m24.206s
sys0m2.222s
```

```
driver_test2000.fcl (buffer_count 2000):
real0m52.138s
user1m22.932s
sys0m3.085s
```

```
driver_test20000.fcl (buffer_count 20000):
real7m9.167s
user12m10.183s
sys0m10.494s
```

...in other words, things slowed down drastically with a higher buffer count. And then, if I go to the feature branches:

```
driver_test200.fcl (buffer_count 200):
real0m16.127s
user0m17.897s
sys0m1.638s
```

driver\_test2000.fcl (buffer\_count 2000):  
real1m4.124s  
user1m20.900s  
sys0m3.345s

driver\_test20000.fcl (buffer\_count 20000):  
real0m49.137s  
user1m11.282s  
sys0m4.669s

...what I see is that while performance is pretty similar for a buffer count of 200 and 2000, it's drastically better for a buffer count of 20000.

To generate a slightly messier, more real-world scenario, I also performed runs of 60 seconds each using different buffer counts in the eventbuilder (but not the datalogger), with two toy simulators running in push mode with no built-in pauses (throttle\_usecs and usecs\_between\_sends both set to 0). Here's what I found:

Develop branches:

Run	buffer_count	events
3078	200	129996
3079	600	117075
3080	1000	88646
3081	2000	57865
3082	10000	16452

Feature branches:

Run	buffer_count	events
3073	200	136342
3074	600	129826
3075	1000	122304
3076	2000	74467
3077	10000	29153

Where I should point out that (A) as always for my tests, the run records can be found in /home/jcfree/run\_records, and (B) I used only 10 ADC counts per event, rather than the hundreds-of-thousands in the driver\*.fcl scripts. As you can see, while in both cases performance degraded considerably with higher buffer counts, the degradation was significantly less severe using the feature branches than using the develop branches.

### #3 - 11/04/2019 09:26 AM - John Freeman

- % Done changed from 0 to 100

- Status changed from Resolved to Reviewed

Eric's happy, I'm happy, and this issue is reviewed.

### #4 - 11/21/2019 02:35 PM - Eric Flumerfelt

- Status changed from Reviewed to Closed

### #5 - 11/21/2019 02:35 PM - Eric Flumerfelt

- Target version set to artdaq\_core v3\_05\_09

## Files

driver_test2000.fcl	1.93 KB	11/01/2019	John Freeman
driver_test200.fcl	1.93 KB	11/01/2019	John Freeman
driver_test20000.fcl	1.93 KB	11/01/2019	John Freeman