

artdaq - Bug #23045

Confusing error from event builder when too many fragments are returned

08/02/2019 08:58 AM - Philip Rodrigues

Status:	Closed	Start date:	08/02/2019
Priority:	Low	Due date:	
Assignee:	Eric Flumerfelt	% Done:	100%
Category:	Known Issues	Estimated time:	0.00 hour
Target version:	artdaq v3_07_00	Co-Assignees:	
Experiment:	DUNE		

Description

Running on ProtoDUNE SP with artdaq version v2019_07_17_protoDFO_pduneHacks, I inadvertently set `generated_fragments_per_event` in one of the fragment generators to 1 when it should have been 2. There are 10 instances of this board reader in my configuration.

The event builders timed out waiting for the correct number of fragments for each event, and printed the following slightly confusing message:

```
Active event 87 is stale. Scheduling release of incomplete event (missing 18446744073709551606 Fragments) to art
```

After a while, I realised that 18446744073709551606 is -10 interpreted as a `uint64_t`, but it would be nice to have the error message indicate a bit more clearly what has gone wrong.

History

#1 - 10/18/2019 02:28 PM - Eric Flumerfelt

- Assignee set to Eric Flumerfelt
- Status changed from New to Resolved
- Category set to Known Issues

I have committed a fix to `artdaq:bugfix/23045_SMEM_HandleTooManyFragments`. The key here was that SMEM did not consider the possibility that more fragments could be received than declared in the configuration. I have added test cases to `SharedMemoryFragmentManager_t` which illustrate the problem.

I also have included checks in `SharedMemoryEventManager` which check for already-released events when allocating a buffer, and automatically discards data for events which have already been released (this was already in place for incomplete events, but not complete ones).

#2 - 10/22/2019 10:44 PM - John Freeman

I've confirmed that if I cherry-pick commit `541d65dee6b40c5ee7e6baf51c598091ade92732` from `bugfix/23045_SMEM_HandleTooManyFragments` onto develop, and then run "mrb t", that the `SharedMemoryFragmentManager_t` test fails. I've also confirmed that if I rebuild from the head of `bugfix/23045_SMEM_HandleTooManyFragments (0a1b812afa1f4d2562648b64a53c89d3783572ee)` the test passes. I've also been performing runs (`cluck:/home/jcfree/run_records/11`) where I've played around with having `expected_fragments_per_event` in the eventbuilder not fully account for the # of fragments per event coming from the ToySimulator, but haven't (yet) been able to recreate what Phil was seeing back on August 2.

#3 - 10/24/2019 03:27 PM - John Freeman

- % Done changed from 0 to 100
- Status changed from Resolved to Reviewed

I performed a run on `mu2edaq01 (/home/jcfree/run_records/3013)` which used the head of artdaq's develop branch, and where I used two ToySimulators, one in push mode producing one fragment per event, one in window mode producing two fragments per event...but with `generated_fragments_per_event` set only to "1" so that after DAQInterface's bookkeeping the eventbuilder expected two total fragments per event rather than three. Sure enough, the eventbuilder log file was chock full of:

```
Active event 3 is stale. Scheduling release of incomplete event (missing 1 Fragments) to art.
```

Now, in run 3014 (`/home/jcfree/run_records/3014`), I switched over to the head of `bugfix/23045_SMEM_HandleTooManyFragments`, and with everything else the same I wound up with messages like:

```
Event 1 has already been completed and released to art! Check configuration for inconsistent Fragment count per event!
```

instead of the "Active event <N> is stale" type messages.

Between this and the SharedMemoryFragmentManager_t test working, I consider this issue reviewed.

#4 - 11/21/2019 02:47 PM - Eric Flumerfelt

- *Target version set to artdaq v3_07_00*

- *Status changed from Reviewed to Closed*