

artdaq - Bug #22451

DataSenderManager eliminates entry in routing_table_ after sending just one fragment

04/25/2019 05:12 PM - John Freeman

Status:	Resolved	Start date:	04/25/2019
Priority:	Normal	Due date:	
Assignee:	Eric Flumerfelt	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Co-Assignees:	
Experiment:	-		
Description			
<p>As part of my work on subsystems, I've modified Kurt's standard dune_sample_system subsystem arrangement, which in its original form is essentially a single parent/only child chain of subsystems:</p> <p>1 -> 2 -> 3 -> 4</p> <p>where subsystem 1 has a single boardreader in push mode and an eventbuilder, subsystems 2 and 3 are single eventbuilders each, and subsystem 4 has six boardreaders in pull (window) mode, as well as a routingmaster along with eventbuilders and a datalogger. This configuration runs just fine if you use commit 4ea76fe2f90b32e419adb2997aa12e670da34769 from branch feature/issue22388_multiple_parent_subsystems (the head of this branch contain some modifications to dune_sample_system which are irrelevant here).</p> <p>Subsystem 1 consists of a fragment generator in push mode (felixHF01) and an eventbuilder (TrigCand). If I copy them - modifying the file names and fragment_id - and use the copies to make a new subsystem which, like subsystem 1, is also a parent of subsystem 2, then subsystem 3 (a single eventbuilder, DFO) immediately complains when I try running. I can illustrate this with the file /tmp/trace_artdaq-demo_v3_04_01_jcfree.run2620 on mu2edeq01. In it, you'll see the following sequence of statements coming out of the DFO's DataSenderManager:</p> <pre>3566 04-25 16:39:49.796640 8 30595 31618 0 DFO_art1_DataSenderManager dbg . receiveTableUpdatesLoop_: (my_rank=10) received update: SeqID 1 -> Rank 13 3527 04-25 16:39:49.803514 444 30595 30595 5 DFO_art1_DataSenderManager d05 . DataSenderManager::sendFragment: Sending fragment with seqId 1 to destination 13 3526 04-25 16:39:49.803958 6 30595 30595 5 DFO_art1_DataSenderManager d05 . sendFragment: Done sending fragment 1 to dest=13 3523 04-25 16:39:49.803992 2 30595 30595 5 DFO_art1_DataSenderManager d13 . sendFragment start frag.fragmentHeader()=0x5bd1a00, szB=1040, seqID=1, type=2 3522 04-25 16:39:49.803994 515 30595 30595 5 DFO_art1_DataSenderManager d15 . calcDest_ use_routing_master check for routing info for seqID=1 routing_timeout_ms=2000 should_sto p_=0 3343 04-25 16:39:51.809527 196 30595 30595 5 DFO_art1_DataSenderManager wrn . Bad Omen: I don't have routing information for seqID 1 and the Routing Master did not send a table update in routing_timeout_ms window (2000 ms)!</pre> <p>Note that here, I've modified the ToySimulators in the boardreaders of both of subsystem 2's parents so they have 500 ADC counts rather than the traditional 40, i.e., fragments of about 1 kilobytes each. What seems to be happening is, the routing_table_ dictionary has an entry for seqID 1, and when the first of the two seqID fragments which make it to the DFO gets sent out, the entry in the dictionary is deleted, so when it's the second fragments turn, there's a failure. It looks like changes will need to be made to DataSenderManager to accommodate the setup I've described.</p>			
Related issues:			
Blocks artdaq Utilities - Feature #22388: DAQInterface should allow for a chi...		Resolved	04/16/2019

History

#1 - 04/26/2019 08:00 AM - Eric Flumerfelt

Most of the changes needed for this Issue are already on artdaq:feature/CFG_MultipleFragmentsPerRead. I'm not sure about the best strategy for moving forward, possibly create a new branch, do a no-commit merge, then discard the CFG changes...

#2 - 04/26/2019 09:10 AM - Eric Flumerfelt

Bugfix on artdaq:bugfix/22451_DSM_MultipleFragmentPerEventRouting

#3 - 04/26/2019 09:10 AM - Eric Flumerfelt

- Assignee set to Eric Flumerfelt

- Status changed from New to Resolved

#4 - 04/26/2019 10:37 AM - John Freeman

After merging bugfix/22451_DSM_MultipleFragmentPerEventRouting into v3_04_01 (commit b81e90096c8eae5f58601f4756193fd61ceeda2e), running with the subsystem layout described above and the feature/issue22388_multiple_parent_subsystems branch of DAQInterface works just fine. Details are in mu2eddaq01:/home/jcfree/run_records/2624. Not sure whether or not this qualifies as a "Review", but from the perspective of what I wanted, I'm satisfied.

#5 - 05/14/2019 08:38 AM - Eric Flumerfelt

- Blocks Feature #22388: DAQInterface should allow for a child subsystem to have more than one parent subsystem added