artdaq - Support #21621

Notes from testing the killing of an EB process on a teststand

01/03/2019 10:54 AM - Kurt Biery

Status:	Assigned	**Start date:**	01/11/2019
Priority:	Normal	**Due date:**	
Assignee:	Kurt Biery	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
Experiment:	-	**Co-Assignees:**	

Description

One of Giovanna's suggestions in the list of lessons learned from protoDUNE was to check into whether a protoDUNE-like system can keep taking data after an EB crashes in the middle of a run. (Of course, we are interested in this, too.)

I've run some tests on mu2edaq01, and I wanted to capture some information from those tests in this Issue.

At a high level, I was hoping to see the system continue running and the remaining EventBuilders take over the extra rate that had been handled by the failed EB.

In general, this is exactly what I observed. More details below.

Subtasks:

Feature # 21666: Add periodic reporting to SharedMemoryEventManager	**Closed**
Bug # 21671: FileMetric does not write timestamps to metric file	**Closed**

Related issues:

Related to artdaq - Support #23111: Restarting a crashed EventBuilder doesn't...	**Closed**	**08/13/2019**

History

#1 - 01/03/2019 11:03 AM - Kurt Biery

I ran the tests on mu2edaq01 using an artdaq-demo system with the following software versions:

- artdaq-utilities-daqinterface, branch develop
- artdaq, branch for_dune-artdaq
- artdaq_core, HEAD detached at 582b1ec (v3_04_04 with merge from bugfix/SMER_CheckCurrentDataSourceBeforeMarkBufferEmpty)
- artdaq_utilities, branch develop
- artdaq_core_demo, v1_06_12a with a modification to product deps to use artdaq_core v3_04_04
- artdaq_demo, v3_03_01

#2 - 01/03/2019 11:05 AM - Kurt Biery

The command that I used to run the tests was the following:

- sh ./run_demo.sh --config mediumsystem_with_routing_master --bootfile `pwd`/artdaq-utilities-daqinterface/simple_test_config/mediumsystem_with_routing_master/boot.txt --comps component01 component02 component03 component04 component05 component06 component07 component08 component09 component10 --runduration 300 --no_om --partition=3

To change the number of EventBuilders from 3 to 4, I modified the boot.txt file in artdaq-utilities-daqinterface/simple_test_config/mediumsystem_with_routing_master.

#3 - 01/03/2019 11:10 AM - Kurt Biery

To watch the rate of events through the system, I looked at the File Metric files in /tmp/eventbuilder and /tmp/aggregator. (I needed to create the aggregator subdirectory, since that hadn't been done before.)

The command that I used to watch the rates was:

- while _1_; do date; find . -amin -1 -type f -print | xargs -I % sh -c ' tail --lines=200 % | grep "Event Rate"| tail -1' ; sleep 15 ; done

I was a little surprised to realize that the metrics in the File Metrics files don't seem to have a timestamp. That it something that I'd like to look into a little bit to understand why not and if those can be added.

#4 - 01/03/2019 11:11 AM - Kurt Biery

In order to prevent DAQInterface from automatically recovering from the EB failure, I hacked a couple of DAQInterface source files:

```
diff --git a/rc/control/daqinterface.py b/rc/control/daqinterface.py
index f53c2e2..c16e5e2 100755
--- a/rc/control/daqinterface.py
+++ b/rc/control/daqinterface.py
@@ -1962,7 +1962,7 @@ class DAQInterface(Component):
                 self.do_disable()

             elif self.manage_processes and self.state(self.name) != "stopped" and self.state(self.name) != "b
ooting" and self.state(self.name) != "terminating":
-                self.check_proc_heartbeats()
+                self.check_proc_heartbeats(False)
                 self.check_proc_exceptions()

         except Exception:
diff --git a/rc/control/manage_processes_pmt.py b/rc/control/manage_processes_pmt.py
index 0ced68b..677f54c 100644
--- a/rc/control/manage_processes_pmt.py
+++ b/rc/control/manage_processes_pmt.py
@@ -80,7 +80,7 @@ def launch_procs_base(self):
     # 30-Jan-2017, KAB: increased the amount of time that pmt.rb provides daqinterface
     # to react to errors.  This should be longer than the sum of the individual
     # process timeouts.
-    self.launch_cmds.append("export ARTDAQ_PROCESS_FAILURE_EXIT_DELAY=120")
+    self.launch_cmds.append("export ARTDAQ_PROCESS_FAILURE_EXIT_DELAY=3600")

     if self.have_artdaq_mfextensions():
```

#5 - 01/03/2019 11:21 AM - Kurt Biery

As a baseline, I ran the mediumsystem_with_routing_master with 3 and 4 EventBuilders (without killing any processes), and I observed the expected rates:

- approximately 2.5 Hz event rate per EB when using 4 EBs
- approximately 3.3 Hz event rate per EB when using 3 EBs
- approximately 10 Hz event rate at the DataLogger

There were occasions, though, when the event rate would drop noticeably for a few 10s of seconds. This happened both in the DL and the EBs. My sense was that it would show up first in the DL, so I wonder if it is simply occasional slowness in disk writing? I want to look more into this later.

#6 - 01/03/2019 11:30 AM - Kurt Biery

In a system with 4 EventBuilders, I killed one of them a little while after the run began ('kill <pid>').

The disconnection was reported in the MsgFac messages:

```
%MSG-w component01_TCPSocketTransfer:  Early pre-events TCPSocket_transfer.cc:718
2019-01-03 11:25:07 -0600: transfer_between_6_and_10_SEND: sendFragment_: WRITE ERROR: Broken pipe
2019-01-03 11:25:07 -0600: %MSG
2019-01-03 11:25:07 -0600: %MSG-e component01_DataSenderManager:  Early pre-events DataSenderManager.cc:516
2019-01-03 11:25:07 -0600: sendFragment: Sending fragment 303 to destination 10 failed! Data has been lost!
```

And, there were follow-up messages for the events which get lost:

```
2019-01-03 11:25:12 -0600: %MSG-w component08_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component07_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component06_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component04_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component05_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
```

```
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component02_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component03_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component10_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:12 -0600: %MSG-w component09_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:12 -0600: Missed request for sequence ID 303! Will not send any data for this sequence ID!
2019-01-03 11:25:12 -0600: %MSG
2019-01-03 11:25:13 -0600: %MSG-w component09_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:13 -0600: Missed request for sequence ID 308! Will not send any data for this sequence ID!
2019-01-03 11:25:13 -0600: %MSG
2019-01-03 11:25:13 -0600: %MSG-w component06_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:13 -0600: Missed request for sequence ID 308! Will not send any data for this sequence ID!
2019-01-03 11:25:13 -0600: %MSG

...

2019-01-03 11:25:18 -0600: %MSG-w component04_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:18 -0600: Missed request for sequence ID 364! Will not send any data for this sequence ID!
2019-01-03 11:25:18 -0600: %MSG
2019-01-03 11:25:18 -0600: %MSG-w component04_CommandableFragmentGenerato:  Early pre-events CommandableFragme
ntGenerator.cc:1127
2019-01-03 11:25:18 -0600: Missed request for sequence ID 365! Will not send any data for this sequence ID!
2019-01-03 11:25:18 -0600: %MSG
```

However, after things settled down, the rate per EventBuilder became ~3.3 Hz, and the overall rate to the DL went back to ~10 Hz.

#7 - 01/03/2019 01:04 PM - Kurt Biery

03-Jan-2019, KAB
Some things that I'd like to follow up on later:

- look into why the EventBuilders and DataLogger don't output periodic stats to the MessageFacility like the BoardReaders and RoutingMaster do
- look into why File Metrics don't have timestamps in the files
- look into why the event rate sometimes goes to zero during a test run like what was used here [04-Jan-19, KAB: I realized today that this is caused by one raw data output file closing and another opening.]
- think about how to give feedback to users that indicates **why** the rate is dropping (if/when it drops and recovers)
- run similar tests (of killing an EB process) at protoDUNE when we get access to that cluster again

#8 - 01/11/2019 11:18 AM - Eric Flumerfelt

- Due date set to 01/11/2019

- Start date changed from 01/03/2019 to 01/11/2019

due to changes in a related task: #21666

#9 - 01/11/2019 04:07 PM - Eric Flumerfelt

- Due date set to 01/11/2019

due to changes in a related task: #21671

#10 - 08/01/2019 03:13 PM - Kurt Biery

I want to revisit these tests with the additional goal of restarting the crashed EventBuilder process and re-including it in the run.

Here are the steps that I'm following for this:

- installed the demo on mu2edaq12
  - wget https://cdcvs.fnal.gov/redmine/projects/artdaq-demo/repository/revisions/develop/raw/tools/quick-mrb-start.sh
  - chmod +x quick-mrb-start.sh

    - ./quick-mrb-start.sh --tag=v3_05_00
- changed the artdaq-utilities-daqinterface/source_me script to use "direct" process management as the default
- added a fourth and fifth EventBuilder to the mediumsystem_with_routing_master boot.txt file; created the FCL file for EventBuilder5 in that area
- reduced the event creation rate to 2 Hz
- started a half-hour run with the following command
    - sh ./run_demo.sh --config mediumsystem_with_routing_master --bootfile `pwd`/artdaq-utilities-daqinterface/simple_test_config/mediumsystem_with_routing_master/boot.txt --comps component01 component02 component03 component04 component05 component06 component07 component08 component09 component10 --runduration 1800 --partition 5 --no_om
- killed the EventBuilder3 process with "kill"
- killed the art process associated with EventBuilder3 with "kill -9"
- restarted EB3 with a command like the following:
    - eventbuilder -c "id: 5237 commanderPluginType: xmlrpc rank: 12 application_name: EventBuilder3 partition_number: 5"
- xmlrpc http://localhost:5237/RPC2 daq.status
- cd run_records/<run number>
- export EB3_CFG=`cat EventBuilder3.fcl | grep -v '^\#' | sed 's/\:/\\\\:/g' | sed 's/{/\\\\{/g' | sed 's/}/\\\\}/g' | sed 's/"/\\\\"/g' | sed 's/,/\\\\,/g'`
- xmlrpc http://localhost:5237/RPC2 daq.init "$EB3_CFG"
- xmlrpc http://localhost:5237/RPC2 daq.start <run number>

{more on this soon}

#11 - 08/13/2019 01:36 PM - Kurt Biery

- Related to Support #23111: Restarting a crashed EventBuilder doesn't automatically result in data flowing through that EB again added