artdaq Utilities - Feature #21235

DAQInterface should be able to launch processes by a method other than pmt.rb

10/25/2018 05:36 PM - John Freeman

Status:	Resolved	**Start date:**	10/25/2018
Priority:	Normal	**Due date:**	
Assignee:	John Freeman	**% Done:**	100%
Category:	artdaq-daqinterface	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Experiment:	-	**Co-Assignees:**	

Description

One of the disadvantages of the traditional, MPI-based method of launching artdaq processes via pmt.rb is that if one artdaq process goes down, they all go down. Particularly with a view toward DUNE's future, with its model of "the DAQ is always running", this is undesirable. DAQInterface should provide an option in which it launches processes without using pmt.rb.

History

#1 - 12/12/2018 06:25 PM - John Freeman

- Status changed from New to Resolved

- % Done changed from 0 to 100

With commit 969df6c705b2e8ae29f6805e6a4477856a680c02 on the develop branch, DAQInterface has the following capabilities:

If the environment variable "DAQINTERFACE_PROCESS_MANAGEMENT_METHOD" is set to "direct" - it defaults to "pmt" in the generic DAQInterface source_me file - then rather than using artdaq_mpich_plugin's pmt.rb script to launch processes, DAQInterface will simply launch them directly. Note that just as the DAQ setup script defined in the boot file traditionally needs to make pmt.rb available when sourced, if we're using direct process management, DAQInterface won't expect pmt.rb to be available but it **will** require artdaq's boardreader, etc. executables to be available, otherwise it throws an exception. In general, this is already available using the standard setupARTDAQDEMO script created on artdaq-demo's installation.

A feature available when we're using direct - rather than pmt.rb-based - management of artdaq processes is that when a process dies when DAQInterface is in the running state, rather than all the other processes dying as well and getting returned to DAQInterface's "stopped" state, instead DAQInterface will just print a message and continue running. The message looks, e.g., like the following:

```
Wed Dec 12 17:42:44 CST 2018: Appear to have lost process with label component01 on host mu2edaq05

Processes remaining:
component02
EventBuilder1
EventBuilder2
DataLogger1
Dispatcher1

Exception caught in DAQInterface attempt to query status of artdaq process
BoardReader at localhost:10100; most likely reason is process no longer
exists
```

Note that this is **not** a guarantee that everything will turn out fine - e.g., if all your eventbuilders die, artdaq process errors will result in your boardreaders! Empirically, I ran some tests using the latest artdaq/artdaq-demo (bb4b22e35a6bfa97499cc6179b15f07a2797a98a/b717ae4a5b945f4ca7d25fb22cb76386dbb8d35d). These correspond to runs 1748 through 1753 on the mu2edaq cluster as described in mu2edaq01:/home/jcfree/run_records . To get specific, all runs were performed with the demo configuration, using a ToySimulator in push mode on mu2edaq01, a ToySimulator in push mode on mu2edaq05, one eventbuilder on mu2edaq01 and one on mu2edaq05, and a datalogger and dispatcher on mu2edaq01.

First things first: for every run (i.e., every time we made it into the running state), a *.root file was saved regardless of what happened. Also, at no point during the runs did I kill and restart DAQInterface; the DAQInterface logfile is mu2edaq01:/tmp/daqinterface_jcfree/DAQInterface_partition0.log.

Run 1748: Just a regular run: boot-config-start-stop-terminate. Everything went fine.

Run 1749: When running, I killed the boardreader on mu2edaq01, and then ended the run shortly afterwards. A rawEventDump of /home/jcfree/daqdata//artdaqdemo_r001749_sr01_20181212T233623_1_dl1.root indicates 616 events with both fragments, 54 with just one fragment, as would be expected.

Run 1750: During running, I first killed the boardreader on mu2edaq05, and a little while later, the boardreader on mu2edaq01. After that, when I issued a stop, an exception was thrown. A rawEventDump of /home/jcfree/daqdata/artdaqdemo_r001750_sr01_20181212T234159_1_dl1.root indicated 452 events with both fragments, 329 with just one.

Run 1751: During running, I killed the eventbuilder on mu2edaq05. Later, when I issued a stop, a timeout (30 seconds) occurred when the stop was sent to the mu2edaq05 boardreader.

On the next attempted boot, an error occurred in which it appeared there were two eventbuilders on mu2edaq05 trying to use the same port. Will keep an eye out for this in the future.

Run 1752: Same test as run 1751, with the same timeout sending a stop to the mu2edaq05 boardreader.

Run 1753: first, didn't see the problem with the boot that I saw in my first attempt at run 1752. Killed mu2edaq01's eventbuilder during running. Timeout occurred sending stop to mu2edaq01's eventbuilder.

Bottom line: at this point, with the support now available from DAQInterface, it will be possible to develop artdaq code to handle scenarios in which individual processes die (e.g., ignoring dead eventbuilders, etc.). Also, later, we can investigate getting DAQInterface to relaunch dead artdaq processes and take them through individual transitions to bring them in line with other artdaq processes.