

artdaq - Idea #21923

With a few code changes, we could support multiple logger processes sending events to the Dispatcher

02/19/2019 08:28 AM - Kurt Biery

Status:	Closed	Start date:	03/05/2019
Priority:	Normal	Due date:	
Assignee:	Kurt Biery	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Before DAQInterface v3_07_00		
Experiment:	-		
Description			
<p>After talking with protoDUNE folks about their needs for distributing events to Online Monitoring processes, I realized that the limitation that only one logger process can send events to the Dispatcher is solely due to the fact that different logger processes often have different patterns for the filenames that they write.</p> <p>This got me thinking about the possibility of providing a single configuration document for all logger processes, that has substitution strings to take care of the different filename patterns that we want for the different logger processes.</p> <p>I have developed some prototype code for this that I will describe here.</p>			
Subtasks:			
Idea # 22059: Implement filename wildcards in RootDAQOut			Closed
Feature # 22129: Please add artdaq_daqinterface support for book-keeping firstLoggerRank			Closed

History

#1 - 02/19/2019 08:44 AM - Kurt Biery

The sample modifications to RootDAQOut_module.cc are on the feature/21923_MultipleLoggersToDispatcher branch in the artdaq repo.

The scheme that I used for substitutions is to use \${<keyword>} in the filename pattern. Other ideas for the substitution pattern may be better.

The currently supported substitutions are the following:

- \${Rank} is replaced with the process rank
- \${ZeroBasedRelativeRank} is replaced with the zero-based relative rank of the current logger process. This substitution needs the "firstLoggerRank" parameter to be set in the config document.
- \${OneBasedRelativeRank} is replaced with the one-based relative rank of the current logger process. This substitution needs the "firstLoggerRank" parameter to be set in the config document.

I will commit sample mediumsysteM_with_routing_master config files that use some of these substitutions, and I also need to talk with John about the possibility of book-keeping the firstLoggerRank config parameter.

In addition to the 'fixed' substitutions mentioned above, user-defined substitutions are also supported. Here is a sample:

```
normalOutput: {
  module_type: RootDAQOut
  fileName: "/tmp/artdaqdemo_r%06r_sr%02s_%#_dl${OneBasedRelativeRank}_${DataDisk}_${ExtraSuffix}.root"
  fileProperties: { maxSubRuns: 1 maxRuns: 1 maxSize: 8192000 maxEvents: 1000 }
  fastCloning: false
  compressionLevel: 0
  firstLoggerRank: 6
  fileNameSubstitutions:
  [
    {
      targetString: "DataDisk"
      replacementList:
      [
        { appName: "DataLogger1_art1" newString: "data0" },
        { appName: "DataLogger2_art1" newString: "data1" }
      ]
    },
    {
      targetString: "${ExtraSuffix}"
      replacementList:
```

```
[
  { appName: "DataLogger1_art1" newString: "hello" },
  { appName: "DataLogger2_art1" newString: "world" }
]
}
]
```

#2 - 02/19/2019 09:22 AM - Kurt Biery

The sample changes to the sample mediumsyste_m_with_routing_master configuration and daqinterface.py are on the feature/21923_MultipleLoggersToDispatcher branch in the artdaq-utilities-daqinterface repo.

The command that I used to test this 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 component03 --runduration 40 --partition=4`

And, in the tests, I modified the mediumsyste_m_with_routing_master/boot.txt file to include the second DataLogger.

#3 - 02/20/2019 10:36 AM - Eric Flumerfelt

I have reviewed the code on the feature/21923_MultipleLoggersToDispatcher, and performed differential testing to demonstrate the new functionality.

I have also verified that the code in artdaq works the same as before if not accompanied by the DAQInterface changes or configuration updates.

#4 - 02/20/2019 10:58 AM - Kurt Biery

Thanks, Eric.

Something to keep in mind is that, while the code changes are backward-compatible with existing configurations, the old code is **not** forward-compatible with new configurations.

That is, if we try to run the old code with a newer-style configuration (with firstLoggerRank or fileNameSubstitutions parameters), the existing code will complain because it doesn't know about those parameters.

#5 - 03/05/2019 08:32 AM - Eric Flumerfelt

- Start date changed from 02/19/2019 to 03/05/2019

- Due date set to 03/05/2019

due to changes in a related task: [#22059](#)

#6 - 03/13/2019 01:23 PM - Kurt Biery

- Due date set to 03/13/2019

due to changes in a related task: [#22129](#)

#7 - 05/14/2019 08:43 AM - Eric Flumerfelt

I'm noting that when I merged develop into this branch, I got a merge error because code around line 1790 had been commented out. From the context, however, I think that instead of commenting out the code, the "disable_unique_rootfile_labels" DAQInterface setting should be set.

This actually doesn't matter for v3_05_00, which separately resolves this issue through changes to the artdaq->art interface.

#8 - 05/14/2019 08:43 AM - Eric Flumerfelt

- Status changed from Assigned to Resolved

#9 - 05/14/2019 08:44 AM - Eric Flumerfelt

- Status changed from Resolved to Reviewed

#10 - 12/10/2019 11:49 AM - Eric Flumerfelt

- Target version set to Before DAQInterface v3_07_00

- Status changed from Reviewed to Closed