artdaq - Feature #10146

Provide moderate synchronization when sending fragments from multiple BoardReaders

09/14/2015 08:23 PM - Kurt Biery

Status:	Closed	**Start date:**	09/01/2015
Priority:	Normal	**Due date:**	
Assignee:	Kurt Biery	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	v1_12_13		
Experiment:	-	**Co-Assignees:**	

Description

In systems that don't have hardware triggers, it's possible for the sending of fragments from some of the BoardReaders to get rather far ahead of other BoardReaders.

The place where we first noticed this (long ago) was the artdaq-demo before we implemented the 0.1 second sleep in the BRs. This can easily be reproduced in the demo with version 1.12.12 of artdaq by setting the ToyGenerator sleep times to zero. To see the problem, look in the EventBuilder DAQ metrics that are sent to the FileMetric reporting destination (e.g. /tmp/eventbuilder/evb_*_metrics.log and search for the reporting of "Incomplete Event Count" and see that it can become quite large. Also, when you try to stop the run, there is a good chance that it will take a really long time to complete because the lagging BoardReader has to finish sending all of its fragments, complete events need to be made in the EBs, etc.

We have also seen the problem in real experiment environments. For example, in the DUNE 35-ton DAQ, if the readout of the data from one of the RCEs or SSPs stops for some reason, the readout of the data from the remaining components can continue to run, and the EventBuilder EventStores can consume lots of memory for incomplete events. As in the demo, stopping the run can take a long time, and often fails, when this happens.

For quite a while, we were thinking that the way to address this problem was to time out incomplete events (Issue #3187) or exert backpressure when a configurable number of incomplete events was reached in the EventStore (Issue #6727). Given our experience with the 35t DAQ, the first option seemed like yet another silent failure that shifters could fail to notice. The second option might be better, because it could provide warning messages when the EventStore is full, but selectively throttling fragments from certain BoardReaders would probably be tricky to implement.

An easier solution is to use MPI_Barrier(s) to periodically synchronize the sending of fragments from the BoardReaders. Historically, this option was not used because of the blocking nature of MPI_Barrier and the possibility that an EndRun message might not get processed because a BoardReader's process_fragments thread was blocked on the barrier. However, the latest versions of MPICH and mvapich2 have a non-blocking barrier call available: MPI_Ibarrier.

This issue requests that we investigate whether MPI_Ibarrier can be used to implment a non-blocking synchronization of the sending of fragments from the BoardReaders.

Related issues:

Related to artdaq - Feature #3187: Add a timeout for incomplete events in Eve...	**Closed**	**12/18/2012**
Related to artdaq - Feature #6727: Add a bound on the number of incomplete ev...	**Closed**	**08/01/2014**
Related to artdaq - Idea #7245: Investigate whether we can use an MPI_Barrier...	**Closed**	**10/30/2014**

History

#1 - 09/14/2015 08:23 PM - Kurt Biery

- Related to Feature #3187: Add a timeout for incomplete events in EventStore. added

#2 - 09/14/2015 08:24 PM - Kurt Biery

- Related to Feature #6727: Add a bound on the number of incomplete events that can be kept in the EventStore added

#3 - 09/14/2015 08:25 PM - Kurt Biery

- Status changed from New to Assigned

- Assignee set to Kurt Biery

- Target version set to v1_12_13

- Start date changed from 09/14/2015 to 09/01/2015

#4 - 09/14/2015 08:30 PM - Kurt Biery

In the Description, I should have said that we may want to implement an incomplete event timeout in the EventStore as an alternative to BoardReader synchronization and let individual experiments choose how they want to handle these types of problems.

#5 - 09/15/2015 09:28 AM - Kurt Biery

- Status changed from Assigned to Resolved

Changes have been made to BoardReaderCore.h/cc to implement this feature.

The scheme that was implemented calls MPI_Ibarrier every N fragments but allows more fragments to flow until a threshold is reached (before the next N fragments are processed). At that point, the code waits for the other BoardReaders to catch up (all BoardReaders call MPI_Ibarrier), and then it continues. During the time that it is waiting, the code prints out configurable messages to say that it is waiting.

There are a number of configuration parameters that control how this feature behaves:

- mpi_sync_interval - this is the number of fragments that will be used by the BoardReaders to decide when to call MPI_Ibarrier. That is, MPI_Ibarrier is called every N fragments, where N==mpi_sync_interval. A value of zero for this parameter is the default, and this value turns off the Ibarrier functionality.
- mpi_sync_wait_threshold_fraction - this is the fraction of the N fragments per Ibarrier call that are allowed to be processed before the BoardReader decides to wait for its siblings to catch up. This value should be GE 0.0 and LT 1.0. The default value is 0.5.
- mpi_sync_wait_interval_usec_ - when a BoardReader is waiting for its siblings to catch up, this is the number of microseconds that it will wait between calls to MPI_Test to see if all BoardReaders have called MPI_Ibarrier. The default value is 100 usec. Users should be careful not to make this value too large, lest it affect overall performance.
- mpi_sync_wait_log_level - this parameter specifies whether the BoardReader should log the "I'm waiting" messages as Warnings (value=2) or Errors (value=3). The default is 2.
- mpi_sync_wait_log_interval_sec - the time interval between the "I'm waiting" messages. The default value is 10 seconds. Actually, each time the code prints out a message, it increases the interval by this amount. So, a value of 10 will cause messages to be printed out after 10 seconds, then after 20 seconds, then after 30 seconds, etc.

#6 - 10/30/2015 08:51 AM - Kurt Biery

- Related to Idea #7245: Investigate whether we can use an MPI_Barrier to synchronize BoardReaders with a newer version of MPI added

#7 - 05/23/2016 10:31 AM - Eric Flumerfelt

- Status changed from Resolved to Closed