

artdaq Utilities - Bug #8716

Ganglia metric plugin handling of EventStore run_id.subrun_id

05/12/2015 11:52 AM - Kurt Biery

Status:	Closed	Start date:	05/12/2015
Priority:	Normal	Due date:	
Assignee:	Eric Flumerfelt	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	artdaq_utilities v1_00_01	Spent time:	0.00 hour
Experiment:	-	Co-Assignees:	
Description			
<p>The 15-second interval for publishing metrics inside the <code>ganglia_metric::send_metric</code> code seems to be destructively interfering with the sending of the <code>run_id.subrun_id</code> value from the <code>EventStore</code> class when we automatically switch from one disk file to another by automatically pausing and resuming the run.</p> <p>This behavior is based on the latest code changes that I have made in <code>EventBuilderCore.cc</code> and <code>AggregatorCore.cc</code>. (currently on the <code>feature/inRunExit</code> branch)</p> <p>The observed behavior is that the <code>run_id.subrun_id</code> is correctly shown in the Ganglia plots for the EBs and the AGs only for the first subrun in a run. For subsequent subruns, the <code>run_id.subrun_id</code> is shown as zero.</p> <p>Here is what I think is happening:</p> <ul style="list-style-type: none">• the Aggregator sends Pause commands to all of the artdaq processes, in the appropriate order• the processFragments threads in the AGs and EBs exit and call <code>metricMan.do_stop()</code>• the <code>ganglia_metric::stopMetrics()</code> method sends zeroes to all metrics accumulators, including the ones for <code>run_id.subrun_id</code>• more than 15 seconds has passed since the latest updates to the <code>run_id.subrun_id</code> metrics, the zeroes get processed• the Aggregator sends Resume commands to all of the artdaq processes, in the appropriate order• the <code>start()</code> methods in <code>AggregatorCore</code> and <code>EventBuilderCore</code> call <code>metricMan.do_start()</code><ul style="list-style-type: none">◦ they then call <code>EventStore.startSubRun()</code>◦ <code>EventStore.startSubRun()</code> calls <code>metricMan.sendMetric()</code> for the new <code>run_id.subrun_id</code> combination• inside <code>ganglia_metric::sendMetric()</code>, less than 15 seconds has passed since the zero was passed in, so nothing is sent to Ganglia <p>One option would be to add an optional parameter to the <code>ganglia_metric::sendMetric</code> methods that would say "the zero that I'm sending you corresponds to a <code>stopMetrics</code> call". The <code>sendMetric</code> methods could use this flag to</p> <ul style="list-style-type: none">• circumvent the 15 second test (since we want the setting of parameters to zero at <code>stopMetrics</code> time to be reliable)• set the <code>lastSendTime</code> to zero (rather than now) so that the value that is sent next will always reliably be accepted/published <p>Of course, other options are possible.</p>			

History

#1 - 05/12/2015 12:04 PM - Kurt Biery

I should have said that I tested my "15 second" theory by bugging the `AggregatorCore` code so that it slept 16 seconds between the automatic pause and the automatic resume. In that situation, the `run_id.subrun_id` was successfully reported in the Ganglia plots for all subruns.

#2 - 06/05/2015 10:11 AM - Eric Flumerfelt

- Status changed from New to Assigned

This may necessitate changes to the `MetricPlugin` interface. It might be a good idea to move the "average over n seconds" code from the Ganglia metric to the `MetricPlugin` interface and then specify in the `artdaq` code whether the metric should be averaged over time or not (the time to average quantities over would become a configuration parameter for metrics, with 0 meaning "don't average"). This could be useful for high-rate metrics in Graphite, which uses fixed-size storage, and will only store a certain number of data points.

#3 - 06/08/2015 10:39 AM - Eric Flumerfelt

- % Done changed from 0 to 100

Code is in the repository, under the "TimeAveragingMetrics" feature branch, and has been tested to work. Specific metrics can be flagged in code to

be "non-accumulating" and are reported to the metric plugins immediately. Other metrics will accumulate for `metrics:plugin:reporting_interval` seconds, then the average value will be reported to the metric plugins.

#4 - 06/08/2015 10:39 AM - Eric Flumerfelt

- *Status changed from Assigned to Resolved*

#5 - 09/14/2015 08:27 PM - Kurt Biery

- *Status changed from Resolved to Closed*

The fix for this issue was included in v1_12_11 of artdaq.

#6 - 05/23/2016 10:06 AM - Eric Flumerfelt

- *Target version set to artdaq_utilities v1_00_01*

I think this was done before the project split, so I'm assigning it to v1_00_01.