

artdaq - Support #23918

Investigate CPU usage of DataLogger process at SBN

01/17/2020 07:44 AM - Eric Flumerfelt

Status:	Closed	Start date:	01/17/2020
Priority:	Normal	Due date:	
Assignee:	Eric Flumerfelt	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	artdaq-core v3_05_10	Co-Assignees:	
Experiment:	-		
Description			
<p>Wes has mentioned a few times that the ICARUS DAQ appears to be limited by the DataLogger, to the point where they have added additional DataLoggers to handle the load. I have done some perf and callgrind-based profiling of an artdaqDriver job using similar Fragment sizes and the RootDAQOut module to try and identify places where there is too much CPU usage.</p> <p>I did find a few minor performance improvements, but the bulk of the art process time (>90%) was spent in ROOT libraries involved in writing data (i.e. WriteFastArray).</p> <p>I still intend to collect perf recordings of processes <i>in situ</i> at ICARUS, just to make sure there isn't anything specific to their environment.</p>			
Related issues:			
Related to artdaq - Support #23839: Remove metric aggregation from SharedMemo...		Closed	01/07/2020

History

#1 - 01/17/2020 07:48 AM - Eric Flumerfelt

artdaq-core:feature/23918_SMM_ImproveBufferReportPerformance improves the performance of retrieving the buffer status report, which is called in SMEM::check_pending_buffers.

#2 - 01/17/2020 08:33 AM - Eric Flumerfelt

artdaq-utilities:feature/23918_MetricPerformanceImprovements further reduces the burden of the metric subsystem on the thread generating the metrics, at a minor expense to the metric-sending thread within MetricManager. Total work done remains constant.

#3 - 01/17/2020 09:16 AM - Eric Flumerfelt

For comparison, here is the Metric Sending rate test from MetricManager_t

With change in this issue:

01-17 09:00:29.001114	MetricManager_t nfo Time for One Metric: 6.925e-06 s.
01-17 09:00:29.001157	MetricManager_t nfo Time for Ten Metrics: 3.0576e-05 s.
01-17 09:00:29.001162	MetricManager_t nfo Time for One Hundred Metrics: 9.7372e-05 s.
01-17 09:00:29.001168	MetricManager_t nfo Time for One Thousand Metrics: 0.000704977 s.
01-17 09:00:29.001173	MetricManager_t nfo Time for Ten Thousand Metrics: 0.00658637 s.

Without change in this issue (i.e. artdaq_utilities v1_05_03):

01-17 09:10:03.204633	MetricManager_t nfo Time for One Metric: 9.597e-06 s.
01-17 09:10:03.204690	MetricManager_t nfo Time for Ten Metrics: 1.5383e-05 s.
01-17 09:10:03.204695	MetricManager_t nfo Time for One Hundred Metrics: 0.000130032 s.
01-17 09:10:03.204701	MetricManager_t nfo Time for One Thousand Metrics: 0.00104086 s.
01-17 09:10:03.204706	MetricManager_t nfo Time for Ten Thousand Metrics: 0.00900333 s.

#4 - 01/27/2020 10:03 AM - Eric Flumerfelt

- Related to Support #23839: Remove metric aggregation from SharedMemoryEventManager added

#5 - 01/27/2020 10:05 AM - Eric Flumerfelt

- Status changed from New to Resolved

#6 - 01/30/2020 01:51 PM - Ron Rechenmacher

- Status changed from Resolved to Reviewed

ran demo in several configs, noting throughput/CPU.

For example, with 2 BR's generating approx 1 MB fragments at 50 Hz gives 100 MB/s at data logger using approx. 73% CPU (data logger art writing to /dev/null) on mu2eddaq13.

```
...
  max_fragment_size_bytes: 1001024
    generator: ToySimulator
    fragment_type: TOY2
    fragment_id: 1
    board_id: 1
    starting_fragment_id: 1
    random_seed: 2899
    sleep_on_stop_us: 500000
    nADCcounts: 500000 # approx. 1 MB
    throttle_usecs:0 usecs_between_sends:20000
    distribution_type: 4
...
```

No Datalogger nor EventBuilder CheckIntegrity. Also, no EventBuilder prescaling.

#7 - 02/20/2020 11:45 AM - Eric Flumerfelt

- *Target version set to artdaq-core v3_05_10*

- *Status changed from Reviewed to Closed*