artdaq Utilities - Bug #22869

MetricManager_t fails occasionally

07/05/2019 12:35 PM - Eric Flumerfelt

Status:	Closed	**Start date:**	07/05/2019
Priority:	Normal	**Due date:**	
Assignee:	Eric Flumerfelt	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	artdaq_utilities v1_05_01	**Spent time:**	0.00 hour
Experiment:	-	**Co-Assignees:**	

Description

I have been trying to run test builds of artdaq to make sure that all of the features recently merged into develop work together and that the basic validation procedures still work. I have been encountering failures of MetricManager_t because of an apparent timing issue between MetricManager, the TestMetric plugin, and MetricManager_t itself. This has led to some minor alterations to MetricManager_t and a new option in MetricPlugin which allows MetricPlugins to specify whether they should receive zeros when registered metrics are missing and at the end of the run.

History

#1 - 07/08/2019 11:48 AM - Eric Flumerfelt

Resolution on artdaq-utilities:bugfix/22869_MetricManager_t_Reliability

#2 - 07/08/2019 11:48 AM - Eric Flumerfelt

- Assignee set to Eric Flumerfelt

- Status changed from New to Resolved

#3 - 09/10/2019 03:01 PM - John Freeman

- % Done changed from 0 to 100

- Status changed from Resolved to Reviewed

This issue has been reviewed, via code examination. More on that in a moment. I'd hoped to recreate the problem, but after using the develop branch (commits e969250edbf3bec2811fff1a6dbea681c9d9c460 or c7fad57247beb4f3282cec22ef7ccc094a5e09b3, dated Sep 5/ Sep 6) of artdaq-utilities and seeing "Passed" every time I ran MetricManager_t (both on mu2edaq nodes and on a single-thread virtual machine on my laptop), I switched to code review.

It appears that the major commits of interest are the following:

96bbd95684dc4f3a672edb404943d9972cd08f09: here, in various places, where a simple "usleep" or "sleep" had previously used between a call to sendMetric() in an instance of MetricManager and a call to the static function artdaq::TestMetric::received_metrics(), instead this was used:

```
while (mm.metricManagerBusy()) usleep(1000);
```

(where mm is the instance name).

metricManagerBusy() itself was implemented in commit 87c4cc9ec9d437b572f03a38b3118eb5bf62dc6b:

```
bool artdaq::MetricManager::metricManagerBusy()
{
        bool pluginsBusy = false;

        for (auto& p : metric_plugins_)
        {
                if (p->metricsPending())
                {
                        pluginsBusy = true;
                        break;
                }
        }

        TLOG(TLVL_TRACE) << "Metric queue empty: " << metricQueueEmpty() << ", busy_: " << busy_ << ", Plugins busy: " << pluginsBusy;
        return !metricQueueEmpty() || busy_ || pluginsBusy;
}
```

which appears to be the right approach. The "new option in MetricPlugin which allows MetricPlugins to specify whether they should receive zeros when registered metrics are missing and at the end of the run" appears to explicitly be disabled for those instances of MetricManager in MetricManager_t for which the timing was altered as described above, and hence is not reviewed here.

#4 - 09/24/2019 08:52 AM - Eric Flumerfelt

- Target version set to artdaq_utilities v1_05_01

- Status changed from Reviewed to Closed