

artdaq - Idea #15791

To prevent 100% CPU usage by fragment generators, users should have to specify a sleep time in getNext_()

03/08/2017 05:13 PM - John Freeman

Status:	New	Start date:	03/08/2017
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Needed Enhancements	Estimated time:	1.00 hour
Target version:			
Experiment:	-		
Description			
<p>This issue grew out of some troubleshooting Kurt and I performed for protoDUNE yesterday. In a nutshell, the problem is as follows: according to the "top" and "top -H" commands, BoardReaders were using 100% of the CPU. The reason is that in the relevant fragment generators, chunks of data were being collected from hardware and put into a queue on a thread separate from the thread running getNext(); all getNext_() did was check to see if any new data was on the queue, and if so, to package them in fragments and send them downstream. However, getNext_() was being called a couple thousand times per second, and the vast majority of the time, it didn't find any new data on the queue and therefore returned...only to be called again, immediately. By sticking a sleep of just 1 millisecond into the getNext_() call, we found BoardReader CPU usage to drop to less than 10%, with no actual data loss.</p> <p>This sort of scenario can and will come up again whenever the fragment generator collects data from hardware on a thread separate from the thread which sends fragments downstream. We're thinking a good way to handle this would be to REQUIRE users to explicitly set a sleep time (which could be 0, if they so choose) to execute in each call to getNext(), where the sleep time typically would be long enough that we wouldn't expect the majority of calls to getNext() to not return any new data.</p>			