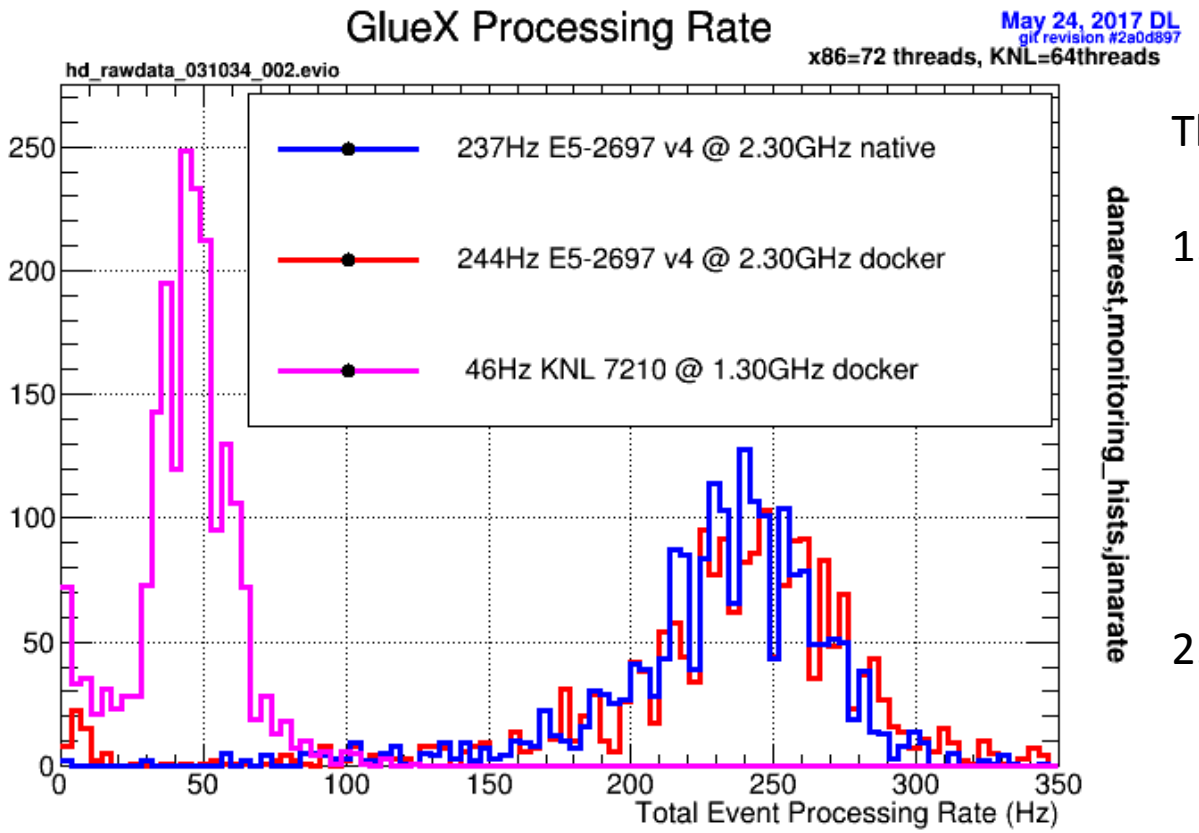
The distributions below were obtained by doing REST production with the standard *monitoring_hists* plugin. This is what is done during a GlueX Reconstruction launch on the farm. The data is the "high luminosity" (=150nA = 2.5x107 γ/s) data run 31034 taken in Spring 2017.



This shows two things:

1. Running reconstruction with 64 threads on a KNL machine is about 5 times slower than running 72 threads on a 2016 Broadwell

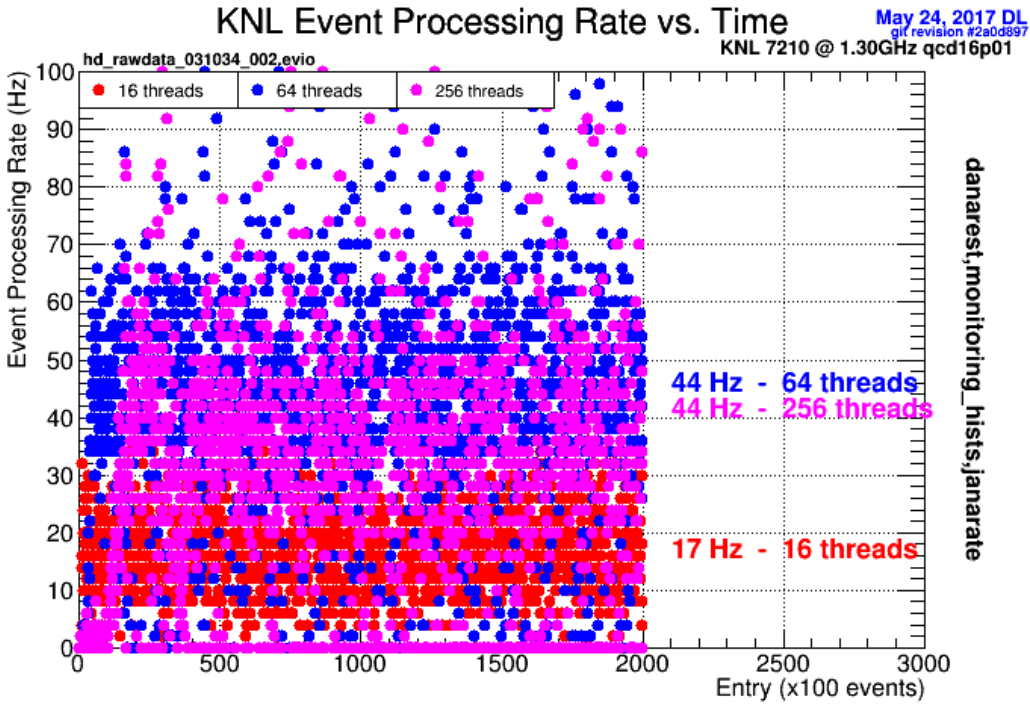
2. Running the reconstruction in a Docker container on the Broadwell gives the same performance as running natively

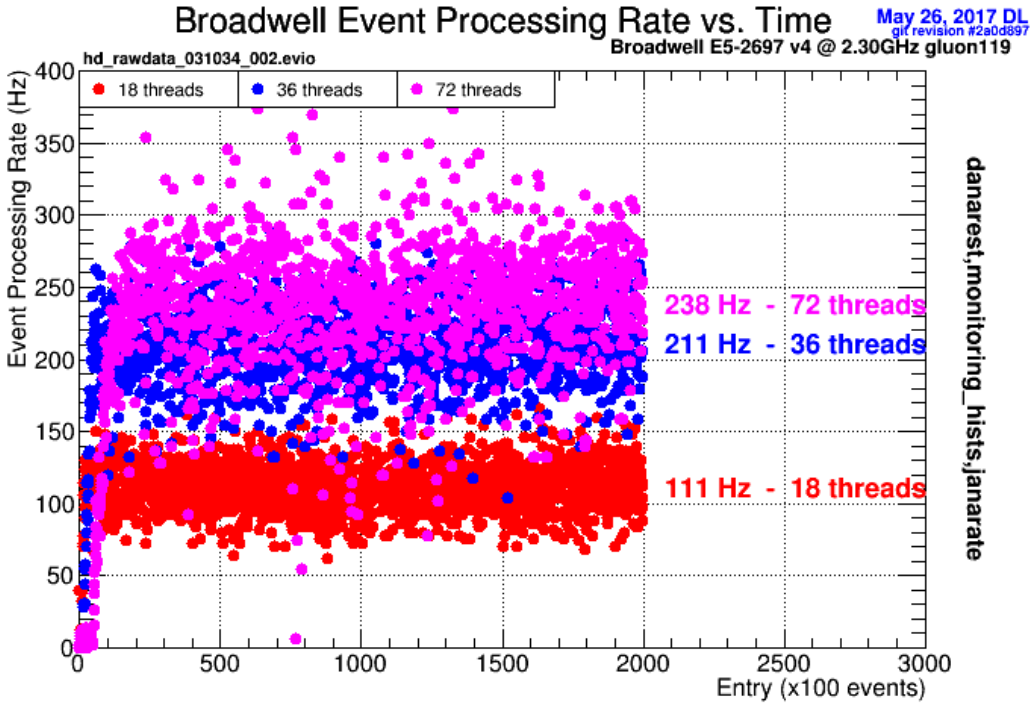
These plots show the event processing rate for every 100 events. The x-axis of each plot is the number of events divided by 100. (Each plot represents 200k events)

The top plot shows that the steady state of the KNL machine has no performance improvement when going from 64 to 256 threads*. It also shows that the scaling from 16 to 64 threads is not very good (44/17=2.6 compared to 4)

The bottom plot shows good scaling for the Broadwell machine (211/111=1.9 compared to 2). It also shows a benefit from the hyperthreading region equivalent to about 13% of a core per hyperthread.

**The 256 thread case (magenta) appears visually to have a lower average, but there are points cut off on the top of the screen that pull up the average. (n.b. the zeros are also included in the average).*



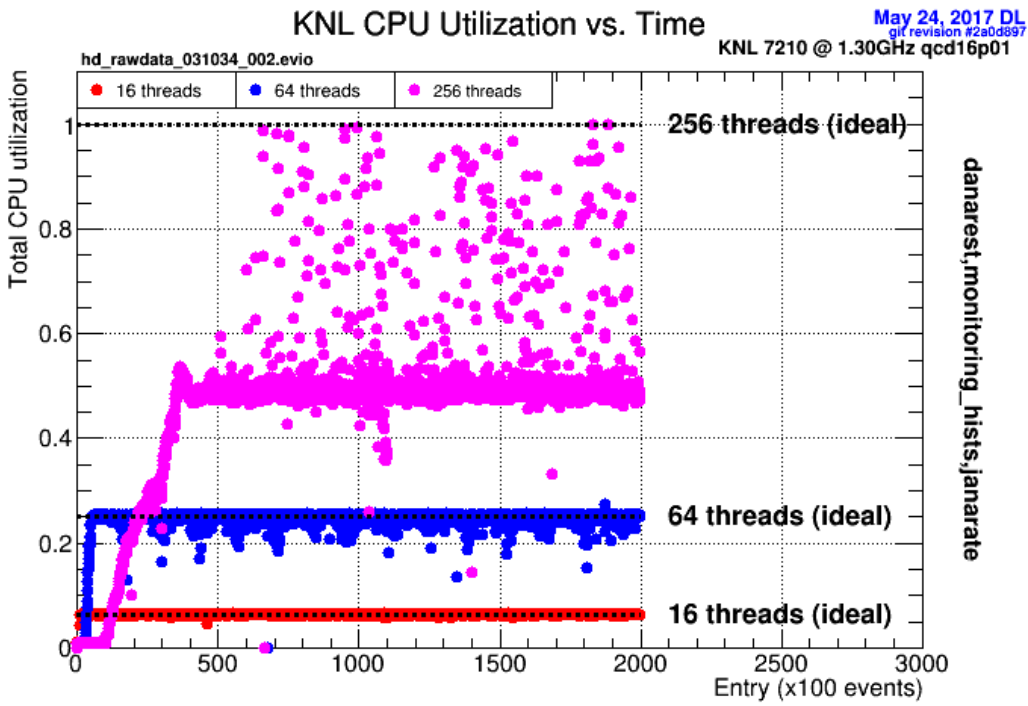


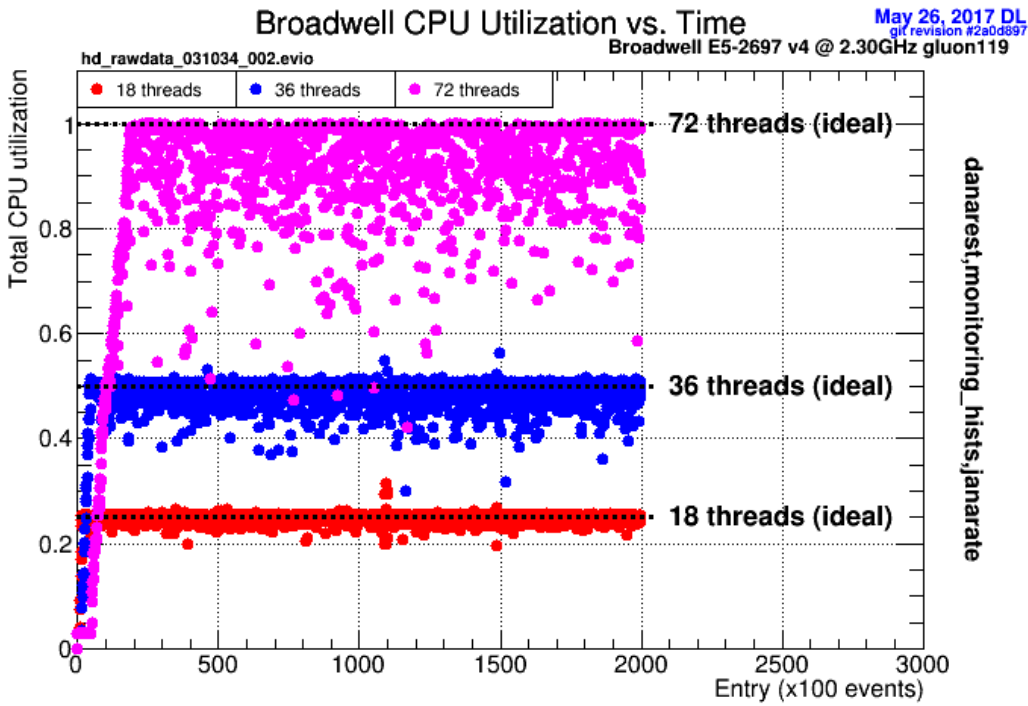
These show the CPU utilization vs number of events. The utilization is reported as a fraction of the number of cores + hyperthreads. The dotted black lines indicate the “ideal” utilization if all threads are continuously busy.

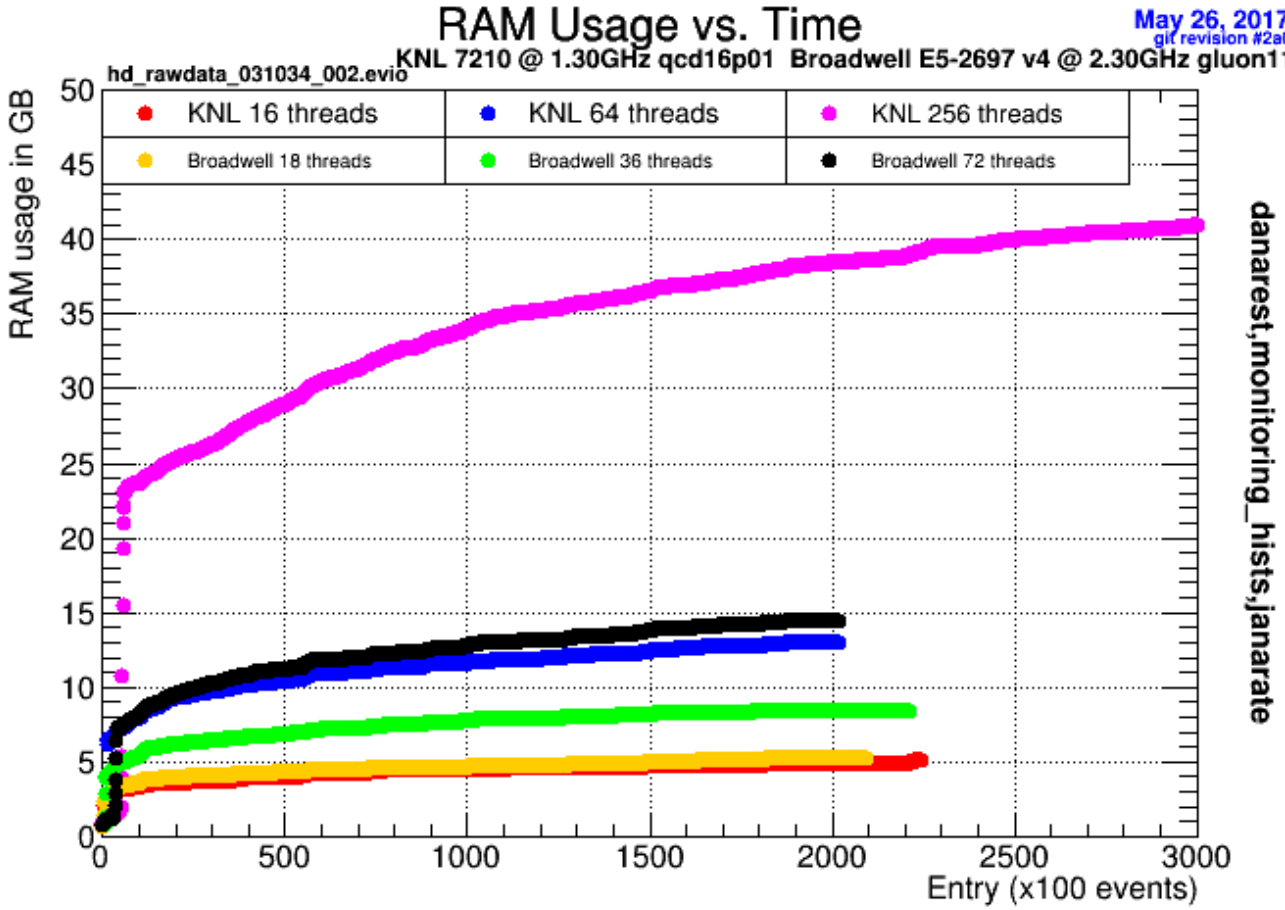
The most interesting feature is that when running 256 threads on the KNL machine, most of the time only one core and one hyperthread are being utilized. Note that on the previous page, this extra hyperthread utilization did not actually seem to translate to any increase in performance.

The Broadwell performs as expected.

n.b. the slow rise to full rate on the left side of the plot is due to calibration constants being read in by all threads. They contend with one another for access to the SQLite file lock so it takes a bit to get going at full rate.



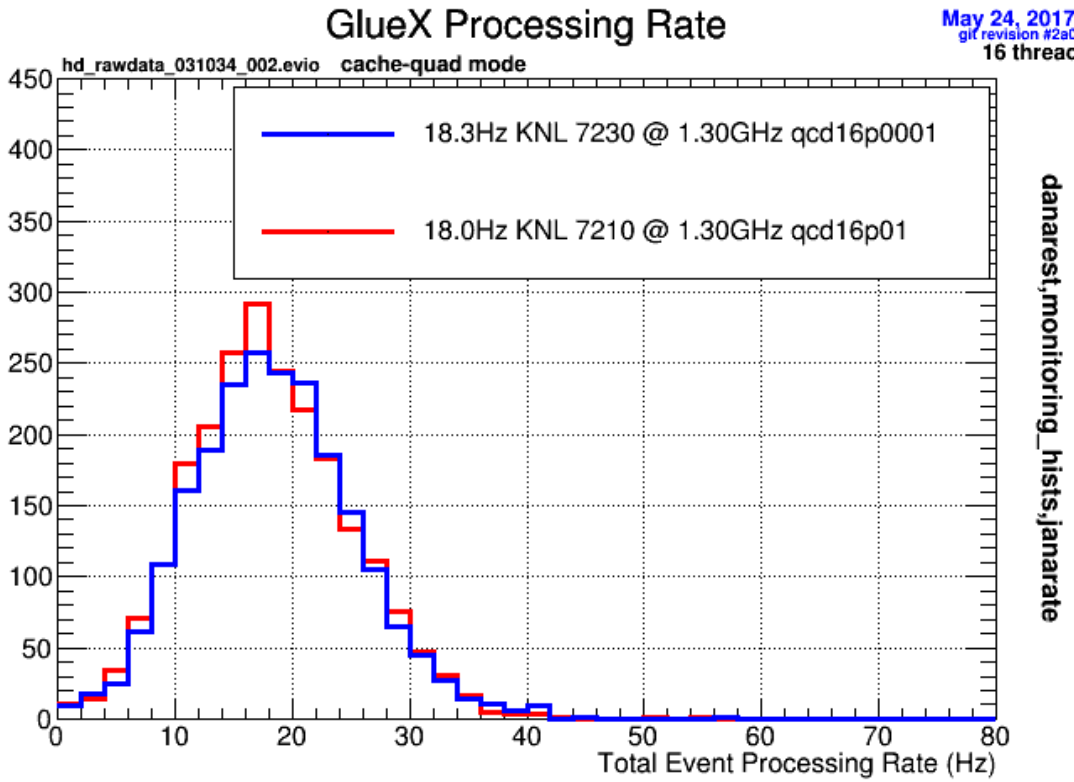




This shows the RAM usage vs. events processed for both KNL and Broadwell architectures. One would not expect the RAM usage to differ and this just confirms it.

The RAM usage rises as the program progresses due to the use of various object pools maintained by each thread. One can see from this that RAM usage for this type of reconstruction will be roughly:

2GB + 170MB/thread



This shows a comparison between on older KNL node (qcd16p01) with the default mode (all to all ?) and a new KNL node (qcd16p0001) running with the quad-cache mode.

16 threads were used in both cases.

This shows no significant difference in the performance between the two modes.