IRIS-HEP Fellowship Proposal

Sean Condon

Duration: June 1st to September 1st, 2020

Developing selection algorithms to reduce output data rate from the Large Hadron Collider

As an IRIS-HEP fellow, I will work on the problem of selecting the best data to store from the Large Hadron Collider (LHC) sensors to assist with the third run of the LHCb experiment. While fully operational, this detector can output as much as 5 terabytes of data per second; because it is infeasible and unnecessary to store all of this information, this data needs to be examined in real-time by a data reduction pipeline to reduce to about 10 gigabytes of output per second. The *Allen* project, which I will be assisting, is a proposal to carry out this extraordinary computational task on only a few hundred GPUs with the help of inventive algorithm design and new machine learning techniques.

I will be working with Professor Mike Williams, Dr. Dan Craik, Tom Boettcher, and all members of the Allen project based at MIT and in Europe to develop and refine the selection algorithms that make this real-time data reduction challenge possible with a relatively small amount of GPUs. The bulk of this work is developing machine learning classifiers that can distinguish useful data in large amounts of detector noise, and then flag this data to be saved during the data reduction process, for which I will use machine learning libraries in Python like TensorFlow and Pytorch. We specifically want classifiers that are able to pick out useful data without large levels of correlation between event variables like invariant mass and transverse momentum. We also want to program these algorithms such that they can be run easily and efficiently on the GPU clusters they will eventually be installed on.

The end goal of this fellowship will be to commit these refined selection algorithms to the official *Allen* repository. Here, they will become the baseline for trigger studies in run three of the LHCb

experiment at CERN. The entirety of this work will be done remotely due to the ongoing COVID-19 pandemic, and I will be working on it full-time for the months of June, July, and August. Below is an expected timeline for this project.

End of June

* In Run 2, in HLT1 LHCb used both one-track and two-track inclusive heavy-flavor selections, and both relied on ML classifiers. Our primary goal by the end of June is to train new classifiers for these algorithms that are tuned for the Run 3 data-taking conditions, which will be 5x higher pile up and all new tracking systems.

* Prepare data samples for both signal and background that can be used for training classifiers used in Run 3 selections. The Monte Carlo itself has been produced. This step requires us to build the objects that will be fed to the classifiers in the trigger, and write them out into a data format we can use in the training / testing.

* Train the classifiers and study their performance. Choose classifiers for the one-track and two-track selections to be used as the baseline in LHCb HLT studies, in coordination with the responsible LHCb working group.

End of July

* Develop software required to run the June algorithms within the LHCb fully GPU-based HLT1 application.

* Deploy this software in the official GPU-based HLT1 software stack. (This is our primary goal for achieving by the end of July.)

* In Run 2, LHCb used two-track, three-track, and four-track inclusive b-physics selections in HLT2 (the second HLT stage). The GPU-based HLT1 is so fast that it can find tracks as soft as were used in HLT2 in Run 2 (even though the input data rate is much higher). We want to study running these b-physics selections in HLT1 for Run 3, which would permit reducing the output bandwidth. (The inputs to these algorithms will be the outputs from the June selections with additional tracks added.)

* Prepare the MC samples for training and testing the classifiers for these b-physics selections.

End of August

* Train the classifiers for the b-physics selections and study their performance. Choose classifiers to be used as the baseline in LHCb HLT studies, in coordination with the responsible LHCb working group.

* Develop and deploy software to run these algorithms within the LHCb fully GPU-based HLT1 application.

* If time permits, we want to study using bespoke loss functions in the training that will allow us to control the performance of these algorithms to deal with online running issues. In Runs 1 and 2, this was done following https://arxiv.org/abs/1210.6861 (developed in 2010). However, we now want to do something better, and the MIT-LHCb group has an ongoing AI project that would be perfect for this.