1 Introduction

We propose to hire a new postdoc at the University of Notre Dame, who will derive half of his/her support via the U.S. CMS operations program, to focus on R&D targeted at ensuring CMS can leverage a broad array of heterogeneous resources—in particular GPUs, FPGAs, and other accelerators—at large scale. To serve this goal, the postdoc would focus on two tasks, **Machine Learning Optimization** and **Data-Flow Processing with Accelerators** described in greater detail below.

As described in [1], the growing need for processing capacity during the HL-LHC era will drive CMS towards the use of accelerators, such as GPUs or FPGAs that can provide substantial speedups for computationally expensive processing steps. Already, CMS has demonstrated success using GPUs to accelerate tracking [2] and deep learning inference [3]. Nevertheless, it is not foreseen that all CMS algorithms will benefit from being ported to GPUs or other accelerators. Some fraction of the processing will remain best suited for CPU resources. Therefore, CMS (and other LHC experiments) will be driven towards a *heterogenous* computing model in which processing is distributed across a mixture of CPU and GPU or other accelerators. This shift from a homogenous, x86-based computing model to a heterogenous one will require evolution of the CMS software infrastructure from the reconstruction algorithms to the framework to workflow management tools.

The simplest model for a heterogenous computing system would be to have individual nodes consisting of an appropriate ratio of CPU to GPU resources At a fundamental level, however, this ideal ratio is not fixed. It depends on the type of processing—MC generation, detector simulation, reconstruction, miniAOD or nanoAOD production, or machine learning (ML) training—and will also evolve over time with changes of algorithms and computational capabilities. Therefore, it is impossible to design individual nodes whose resources will be used efficiently under all scenarios for the lifetime of the hardware. Furthermore, it is expected that a significant fraction of HL-LHC computing resources will be provided in the form of high performance computing (HPC) centers, where the resource mixture is not tuned specifically to meet CMS needs. In fact, already some current and many planned HPC facilities will provide the bulk of processing resources in GPU form. To face these needs, CMS will need to develop more flexible approaches to making use of accelerator technologies.

Two projects are planned for the proposed postdoc to address the challenges described above:

Machine Learning Optimization Frameworks: The "No Free Lunch" theorem [4] implies that the development of any powerful ML technique requires problem-specific optimization. While training for current ML models can be accomplished with the resources of a single GPU, because of the high dimensionality of hyperparameter space for most modern ML models, optimization requires a distributed GPU infrastructure to produce results in a timely fashion. There is no need for CMS or even the HEP community to write their own hyperparameter optimization framework; a number of these tools exist both for specific popular ML frameworks, as well as for more generic tools. However, these frameworks are often interfaced to computing resources like commercial cloud interfaces or HPC batch systems rather than the grid resources more readily available to high energy physicists. Therefore, the proposed postdoc would work to interface one or more existing optimization frameworks to the HTCondor batch system such that it is suitable for use with the CMS Connect service.

Data-Flow Processing with Accelerators: Since it is infeasible to rely on a tight coupling of CPU and GPU or other accelerators within a single physical node, it is essential that CMS explore models where these resources are less tightly coupled. The postdoc will contribute to this

R&D by measuring the performance of various approaches for linking distributed CPU and GPU resources in a single, multithreaded `cmsRun` processing instance. The technical viability of this approach has already been demonstrated in [3] and was further explored as an option for the HLT in [5]. This project would build on that work by measuring the processing throughput under various latency conditions arising from network distance between the CPU and GPU resources, ranging from having both devices in the same physical servers to have the resources distributed in different data centers. The project will also include examining the impact of different approaches to linking the distributed CPU and GPU resources, include various MPI, RPC, and other similar frameworks.

This research plan addresses several elements of the HEP software roadmap [1]. The projects will catalyze both the **Machine Learning** and the **Event Reconstruction** topics in the proposal by enabling CMS to leverage existing and future efforts in developing new ML and event reconstruction algorithms on heterogeneous architectures at scale. Furthermore, the **Data-Flow Processing with Accelerators** project directly addresses the **Data-Flow Processing Framework** topic in providing the necessary R&D to help determine the computing model and workflow management approaches that will allow the incorporation of heterogeneous resources within the practical constraints of budget and HPC requirements. Combining these two projects in the research plan of a single postdoc has some additional benefits. While it is generally not hard to find physicists interested in contributing to machine learning projects, it is less straightforward to attract young researches to projects involving the processing framework and workflow management topics. By combining machine learning, GPU and FPGA acceleration, framework, workflow management, and resource provisioning into the responsibilities for a single position, we maximize our chances of attracting a top candidate while also providing the postdoc with training in a broad skill set that will be useful for career paths both within and outside the field of HEP.

In what follows, we describe in greater detail the proposed research projects, including milestones for a one-year appointment and future directions for a longer term plan, should the postdoc's contract extend beyond one year. We then discuss the opportunities for **Mentorship** and **Community Interaction** that would be available for a postdoc placed within the Notre Dame team.

2 Machine Learning Optimization

Modern approaches to ML, especially deep learning involve a significant number of hyperparameters, which are model parameters that cannot be learned, but instead must be determined through some optimization procedure. The process of optimizing model hyperparameters is sometimes referred to as metalearning. Simple deep neural networks already have a significant number of hyperparameters, including the number of hidden layers in the model, the number of hidden nodes in each layer, activation and loss functions, and regularization techniques. Certain fields, such as computer vision and natural language processing, have demonstrated that specialized deep learning models can dramatically improve performance, but these network types introduce additional hyperparameters. As HEP researchers explore deep learning approaches specifically adapted to physics problems, including graph-based neural networks (GNNs) and networks incorporating directly physics domain knowledge, the space of network hyperparameters only grows further. To succeed at developing ML approaches powerful enough to address the needs of HL-LHC, CMS physicists must be prepared to explore this high dimensional hyperparameter space, requiring the deployment of hyperparameter optimization frameworks at scale.

Not all existing hyperparameter optimization frameworks are currently interfaced to grid resources. More commonly, these frameworks are interfaced to cloud resources and batch systems common at HPC sites. Although these resources can be used in CMS, they require either funds (for commercial cloud resources) or access to allocations (in the case of HPC sites). Meanwhile, there are a growing number of GPU resources being made available via traditional sources of HEP computing such as through the OSG and at WLCG sites (e.g. Tier-2 or Tier-3 sites). As these resources are freely available to CMS collaborators, it is important to make sure hyperparameter optimization frameworks easily interface with these resources. The proposed postdoc would address this issue through two main activities.

By consulting with the CMS ML Forum and surveying the broader CMS community, the postdoc will determine which ML frameworks are most prevalent. The postdoc will then survey available hyperparameter optimization packages, including those targets specific frameworks like Tensorflow, Keras, or Pytorch as well as general purpose tools, for example SHADHO [6]. The postdoc will summarize which methods of distributed processing are supported by each framework. **Deliverable:** A summary of the ML and hyperparameter optimization framework options, including links to the pages for the various tools, in a web or Twiki page format to allow for regular updating as the landscape of tools evolves with time.

Based on the most results of the information gathering described above, the postdoc will select at least one hyperparameter optimization framework and implement the necessary interface so that it can be deployed on CMS and OSG grid resources through the CMS Connect platform. This will involve making sure the tool is interfaced to HTCondor batch system and that the interface does not assume a "local cluster" model (such as making use of a shared file system among nodes). **Deliverable:** (1) The interface software and relevant documentation delivered either via a pull request against the repository of the relevant hyperparameter optimization GitHub repository or as a separate software package hosted in GitHub, as appropriate.

If this project continues beyond the posdoc's first year, the natural progress would be to continue to expand the set of tools interfaced with grid resources via CMS Connect and to maintain the documentation summarizing the most prevalent ML frameworks and which hyperparameter optimization tools have been interfaced with CMS resources. Longer term, the postdoc's effort should move beyond hyperparameter optimization to distributed learning. As models grow more powerful, they also require more resources, dramatically increasing the number of trainable parameters and the size of the datasets needed for training. At some scale, these models grow beyond the resources available to a single GPU and must be distributed across multiple GPUs. Existing ML frameworks incorporate support for distributing training across multiple GPU resources, but like the hyperparameter optimization packages, will need to be interfaced to CMS and OSG grid resources to provide the maximum benefit for CMS researchers. The proposed postdoc would work on provide the necessary interfaces.

3 Data-Flow Processing with Accelerators

For the foreseeable future, only a fraction CMS processing will be able to be accelerated by a GPU. One way to address this challenge is through multithreaded parallel processing of events. If enough events are processed in parallel streams, then the the aggregate work from all the streams may keep the GPU fully occupied. Maximizing throughput, therefore, is a matter of finding the right ratio of CPU threads per GPU. Of course, this ratio will depend on the specific capabilities of the CPU and GPU, as well as the fraction of CMS processing capable of using the GPU. If CPUs and GPUs from different physical machines can be harnessed for a single processing run, it would be much easier to achieve optimal resource loading. Of course, with this model, there is latency in transmitting data from the CPU to the GPU on another machine and back again. It is unknown how that latency would impact the overall throughput; therefore, we propose that the postdoc funded by this proposal would measure the throughput and resource utilization under different scenarios, from CPU and GPU on the same physical node, to CPU and GPU located on different nodes within the same data center, to CPU and GPU nodes located in different data centers.

Notre Dame's Center for Research Computing (CRC) provides an ideal laboratory for performing these throughput measurements. The CRC hosts the Notre Dame Tier-3 (NDT3) cluster consisting of roughly 1,000 CPU cores. It also hosts the NSF funded Cyberinfrastructure to Accelerate Machine Learning (CAML) cluster. CAML nodes provide 24 CPU cores and four NVidia RTX6000 GPU cards. There are two such nodes dedicated to CMS processing, and CMS can access up to an additional 17 nodes in the shared part of the cluster. ND is centrally located within the US, close to FNAL and the Purdue Tier-2 cluster, and has a 100 Gb/s WAN connection, allowing tests of scenarios where the CPU and GPU resources are nearby data centers, as well as data centers that are separated by up to half the width of the US.

The proposed work would proceed as follows: (1) the postdoc would take baseline measurements of throughput and utilization in the scenario where the CPU and GPU reside on the same physical node. The ratio of CPU threads to GPU will be varied to attempt to determine the ratio that provides the best throughput. (2) Next the postdoc will using the existing, MPI-based implementation to measure throughput performance for situations in which the CPU and GPU are located on different machines but within the same datacenter (making use of NDT3 and CAML resources at the CRC), in nearby datacenters (making use of CAML and CPU resources at either FNAL or Purdue), and in well separated datacenters (making use of CAML and CPU resources at the MIT, Florida, or UCSD Tier-2 sites). With each stage of tests, the postdoc would iterate with CMS framework experts (Chris Jones and Matti Kortelainen at FNAL) to insure proper optimization and interpretation of the test results. The **deliverables** from this research would be a document describing the testing procedure and methods, summarizing and detailing all test results, and providing recommendations regarding whether further exploration of this data-flow processing model is warranted. If further exploration is warranted and if the project is extended past its first year, the following directions would be pursued: (1) Compare the MPI based approach to an alternative communication protocol, such as RPC. (2) Modify the transmitter-receiver framework to handle multiple transmitters communicating to a single receiver to enable explorations involving larger ratios of CPU cores per GPU. (3) Perform scale testing with many independent instances of `cmsRun` on different CPU nodes communicating with multiple GPU nodes. (4) Investigate, with Jones and Kortelainen, whether performance could be improved by tuning the characteristics of communicating with or scheduling work on the GPUs or arranging processing in the multithreaded framework to hide latency. The ultimate **deliverable** from this longer term body of work would be a strategy document outlining whether this model of data-flow processing with remote GPUs is a fruitful direction for CMS to pursue.

4 Milestones

The table below details milestones for the first year of the postdoc's contract. All dates are given in terms of months after the postdocs start date. If the postdoc's contract is extended past one year, additional milestones would be defined for the subsequent years.

Date	Description
2 months	Survey of ML frameworks and hyperparameter optimization tools provided.
4 months	Baseline throughput measurements with CPU and GPU in same server completed.
7 months	Implementation of HTCondor backend for one hyperparameter optimization tool.
9 months	Throughput tests with CPU and GPU in different nodes within CRC completed.
10 months	Hyperparameter optimization at scale demonstrated with CMS Connect.
11 months	Throughput tests with CPU and GPU in different datacenters completed.
12 months	Report detailing all throughput tests completed.

5 Mentorship, Supervision, and Community Interaction

The proposed postdoc would be jointly supervised by PIs Mike Hildreth and Kevin Lannon. Following the model successfully employed with previous ND postdoc Kenyi Hurtado, the postdoc would be given office space both in the physics department and at the CRC and would spend at least one day per week at the CRC interacting with other CRC staff members. The postdoc would also join in weekly meetings including ND CMS faculty, ND CS faculty (Doug Thain, Walter Scheirer, David Chiang), and CRC staff discussing a range of CMS-related computing R&D projects, including the NDT3 evolution (Kubernetes), CMS workflow management (WMAgent and Lobster), and future analysis facilities (Coffea). CRC staff member Hurtado is the primary developer of CMS Connect and can directly support the ML optimization work. ND's proximity to FNAL will make it easy for the postdoc to collaborate with framework developers Jones and Kortelainen. It will also facilitate engagement by the postdoc in the LPC community, especially for the purpose of educating LPC users about the opportunities the postdoc will be enabling through work on ML optimization. ND's proximity to University of Chicago will also facilitate collaboration with the CI Connect team from OSG lead by Rob Gardner. Collaboration with the HTCondor and CMS workflow management experts residing at the University of Wisconsin (UW) will also be possible given the four-hour drive time between ND and UW. The PIs have a history of successfully developing computing skills in young researchers, including former graduate students Anna Woodard (now a computational postdoc at the University of Chicago) and Matthias Wolf (now a computatinal scientist at ETH/Lausanne) and former postdoc Kenyi Hurtado, now on the staff of ND's CRC. We plan to continue this successful approach with this postdoc if funded.

References

[1] Johannes Albrecht et al. A Roadmap for HEP Software and Computing R&D for the 2020s. *Comput. Softw. Big Sci.*, 3(1):7, 2019.

[2] Patatrack Website. `https://patatrack.web.cern.ch/patatrack/index.html`.

[3] Javier Duarte et al. FPGA-accelerated machine learning inference as a service for particle physics computing. *Comput. Softw. Big Sci.*, 3(1):13, 2019.

[4] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transaction on Evolutionary Computation*, 1(1):67–82, 1997.

[5] Mario Gonzalez Carpintero. Remote GPU Offloading in CMSSW. `https://cds.cern.ch/record/2690695`, Sep 2019.

[6] Jeff Kinnison, Nathaniel Kremer-Herman, Douglas Thain, and Walter J. Scheirer. SHADHO: massively scalable hardware-aware distributed hyperparameter optimization. *CoRR*, abs/1707.01428, 2017.