BAYLOR
UNIVERSITY

Research Proposal for the 2021 U.S. CMS HL-LHC S&C Program

PI: **Kenichi Hatakeyama**, Associate Professor
for **Mark Saunders**, Postdoctoral Research Associate
Department of Physics, Baylor University

1 Introduction

This is my proposal for the 2021 U.S. CMS HL-LHC S&C program. The goal of the proposal is to **develop heterogeneous particle flow reconstruction for the CMS Phase 2 detector**. This goal would be well aligned with one of the priority areas of this program, *i.e.* adapting reconstruction algorithms to heterogeneous computing architectures.

The particle flow (PF) algorithm [1] plays a central role in CMS event reconstruction. The present PF reconstruction uses information from the tracker, ECAL, HCAL, and muon spectrometer to build a comprehensive list of final-state particles, called PF candidates. This list of PF candidates provides a global event description, which yields excellent performance for measuring jets, missing E_{T}, hadronic τ decays, electrons, and muons.

Although PF reconstruction in CMS was established during Run 1, it needs significant updates as we upgrade our detectors. PF reconstruction for the Phase 2 detector needs to integrate HGCAL reconstruction in the endcap region, incorporate forward tracks up to $|\eta| \sim 4$, and perform 4-dimensional reconstruction by incorporating the precision timing provided by e.g. the MIP timing detector (MTD). It also needs to be able to cope with the increased data rates without loss of physics performance by efficiently exploiting the expected heterogeneous computing resources offered by GPUs and/or FPGAs.

The PI has spent the past couple of years setting the foundation for the evolution of the PF reconstruction by extensively refactoring the PF code and improving the PF validation package. The first PF implementation in the HCAL forward calorimeter (HF) region ($3 \lesssim |\eta| \lesssim 4$), matching forward tracks and calorimeter clusters, was incorporated in CMSSW by the PI with collaborators, and the HGCAL and MTD reconstruction is incorporated into the CMS PF reconstruction as a part of the effort for the Phase 2 high-level trigger (HLT) technical design report (TDR) [2], though in a somewhat ad-hoc manner. Outside the acceptance of the HGCAL detector, PF candidates are reconstructed using the standard PF algorithm; for the HGCAL detector, a simplified PF-based event interpretation is applied to "The Iterative CLustering" (TICL) candidates as described in the HLT TDR [2]. We need a harmonized and uniform PF-based global event description by the time of HL-LHC, and **establishing a coherent PF reconstruction across all calorimeters** is one of the goals of this proposal.

With expected significantly increased pileup interaction rates, we need a computationally efficient PF reconstruction algorithm to process events with a sufficient rate. Therefore, another main goal of this proposal is **to update the time consuming parts of PF reconstruction to parallel-processing friendly algorithms that run on heterogeneous computing resources.**

I am grateful for the support from the 2020 U.S. CMS HL-LHC S&C program. Dr. Mark Saunders, who graduated from the Florida Institute of Technology (FIT), joined Baylor to work on this project beginning of February 2021 based on support from this program, and he began contributing to this effort immediately. I would like to request support for him from the 2021 U.S. CMS HL-LHC S&C program as well to enable him to continue on this project.

2 Heterogeneous particle flow reconstruction

A simplified workflow diagram of the PF reconstruction chain is shown in Fig. 1, and the more complete workflow can be found in [3]. First, a list of PF-specific rechits is created from rechits of each calorimeter subsystem, and it is used to produce calorimeter clusters. Then, a list of PF clusters as well as PF tracks and several other sets of information, called PF elements, are sent to the PFBlockProducer to put all linked elements in a PFBlock. Then each block is processed by PFAlgo to produce a list of PF candidates.

DEPARTMENT OF PHYSICS
PO BOX 97316 · WACO, TEXAS 76798-7316 · (254) 710-2511 · FAX (254) 710-3878

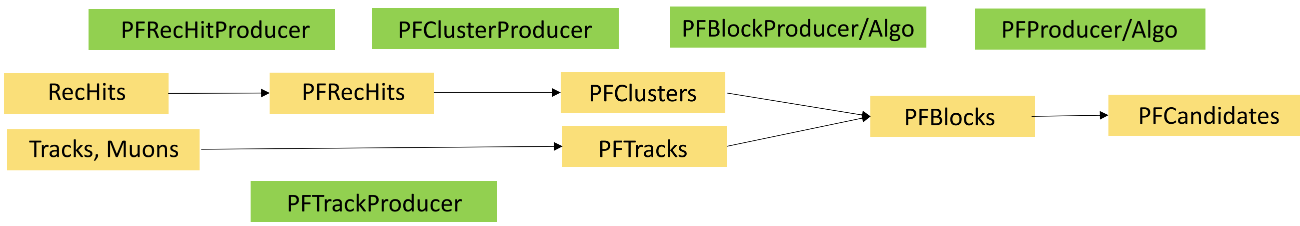

Figure 1: Simplified diagram of the particle flow reconstruction workflow.

Fig. 2 shows the breakdown of processing time estimates for HLT in Run 3 with the so-called "scouting" (high-rate data-taking with reduced event content) path and for the Phase 2 offline processing of $t\bar{t}$ simulated events and 200 pileup interactions. Fig. 2(a) is not an estimate for HL-LHC operation, which is the main theme of this program; however, it illustrates current progress on the use of heterogeneous computing resources for HLT. CMS will introduce the heterogeneous HLT farm for Run 3 by equipping all HLT nodes with one GPU each. In preparation for Run 3, the pixel tracking is already ported to run on GPU [4], and already good progress has been made on porting the ECAL and HCAL local reconstruction to GPU. The natural next target to be ported to GPU is PF reconstruction, as many inputs to PF will be reconstructed on GPU. For the Phase 2 offline reconstruction, the PF reconstruction takes about 7% of the total processing time as shown in Fig. 2(b), which is not a major portion; however, the coherent PF reconstruction over the whole detector could increase this number, and we need a conscious effort to keep this percentage well under control.

Currently there are two general classes of approach being considered for acceleration of particle flow: **(1) improve parallelization of PF modules**, and **(2) develop ML-based fast approximation**. I believe both should be pursued, and there are good R&D activities for both approaches. The ML-based approach, in particular the end-to-end ML approach from a list of PF elements to a list of PF candidates is interesting, and initial results are encouraging [5]. It would also be important to have well-understood algorithmic approaches that run fast enough by exploiting parallelization and heterogeneous computing. In this proposal, I elaborate more on the first approach.

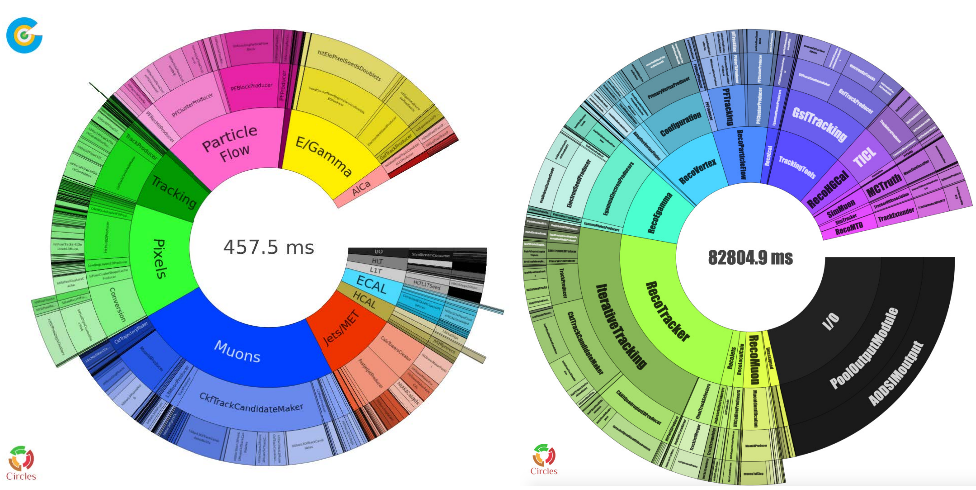

Figure 2: (a) Processing time profile of 30kHz PF scouting with pixel-only tracks for a Run 3 scenario. (b) Processing time profile of the Phase 2 offline reconstruction scenario with $t\bar{t}$ events with 200 pileup interactions.

DEPARTMENT OF PHYSICS
PO BOX 97316 · WACO, TEXAS 76798-7316 · (254) 710-2511 · FAX (254) 710-3878

2.1 Porting the early steps of PF to GPU

Among several steps of PF reconstruction shown in Fig. 1, one of the most time consuming steps is **PFClusterProducer**, which groups calorimeter hits to form clusters corresponding to showers created by incident particles. Clustering is a common exercise in HEP, but it is often hard to parallelize. The HGCAL DPG has introduced 2-dimensional layer-wise rechit clustering, CLUE (CLUstering of Energy), in a GPU-ready framework [6], which is based on an imaging algorithm. This algorithm could be an good option for very highly segmented calorimeters such as HGCAL; however, it is unlikely to work well for more coarsely segmented ECAL and HCAL barrel and HF where particle showers naturally overlap more significantly. In the current PFClusterProducer, an **expectation-maximization (EM) algorithm** based on a **Gaussian-mixture model (GMM)** is used to reconstruct the layer-wise clusters, which accounts for energy sharing from different showers in individual calorimeter cells [1].

We built a small team consisting of collaborators from Baylor, CERN, and Kuwait to implement this algorithm on GPUs, and we recently created a first prototype PF layer-wise clustering algorithm based on the **Cellular Automaton (CA) algorithm** with the von Neumann and Moore neighborhoods [7]. The alpha version of the GPU accelerated algorithm yields identical PF clusters as the current CPU version with processing time comparable to the current CPU time. At present, copying PFRecHit data from CPU to GPU memory is the largest bottleneck for this approach, but once **PFRecHitProducer** is also ported to GPU this intermediate copy step will no longer be necessary and will lead to a significant acceleration. We are also continuing to optimize the current implementation including data format definitions and algorithm configurations and exploring ideas for further improvements based on e.g. pixel hit GPU clustering [4].

After the layer-wise clusters are formed, 3-dimensional multi-depth clusters should be formed for HCAL barrel and endcap sections [3]. We have not started on this step, but I think we will be able to use the **CA based hit chain-maker approach** used for GPU pixel tracking [4] and also the **CA based pattern recognition method** of forming “tracksters“ [2], reconstructed particle showers, from layer-wise clusters as prototypes.

As already discussed above, we also need to port PFRecHitProducer to GPU, which is critical to minimize necessary data transfer between CPU and GPU and to make the PF GPU workflow efficient. The main roles of this step are to apply quality cuts, to associate navigator information to individual hits to aid the following clustering steps, and for HF, to compute the EM and hadronic energies based on long and short readout fibers. The hit quality evaluation that relies on neighbors will need some careful thought, but this step should be relatively straightforward for parallelization and porting to GPU.

The software discussed above for GPU is being written in CUDA (Compute Unified Device Architecture, a parallel computing platform) and utilizes the CUDA service integrated in CMSSW. It means that CMS will have two versions of the code for the same (or similar) function, one written in C++ for CPU and another written in CUDA for GPU, which is not an ideal situation for code maintenance. And, in the future, there will be a wider range of different computing architectures and different vendors that CMS reconstruction code needs to run across in order to profit from better performance and energy efficiency at a lower cost. CMS is investigating ways for achieving **performance portability** by exploring libraries and frameworks (e.g. Alpaka or Kokkos) for getting comparable levels of efficiency with respect to native performance over different target architectures with a single code basis. The key for performance portability is that we already have data-parallel code, and the patatrack pixel track reconstruction code for GPU is being tested on some of these portability libraries. The GPU PF software should be relatively straightforward to be converted to work with portability libraries. When the above GPU code is ready for commissioning for Run 3 HLT, if CMS’s strategy for the portability library is clear, I plan to add adopting the CUDA-based code to the portability library as one of the future goals of this project.

2.2 Coherent PF for the entire Phase 2 detector

As discussed earlier, the current Phase 2 PF reconstruction consists of two almost separate paths for inside and outside the HGCAL detector acceptance. The plan has been to establish a coherent PF reconstruction across all calorimeters when the HGCAL local reconstruction TICL becomes mature, and we aim to tackle this challenging topic starting from year 2 of this project. Two foreseen approaches are: (1) feed TICL outputs into the PFBlockPro-

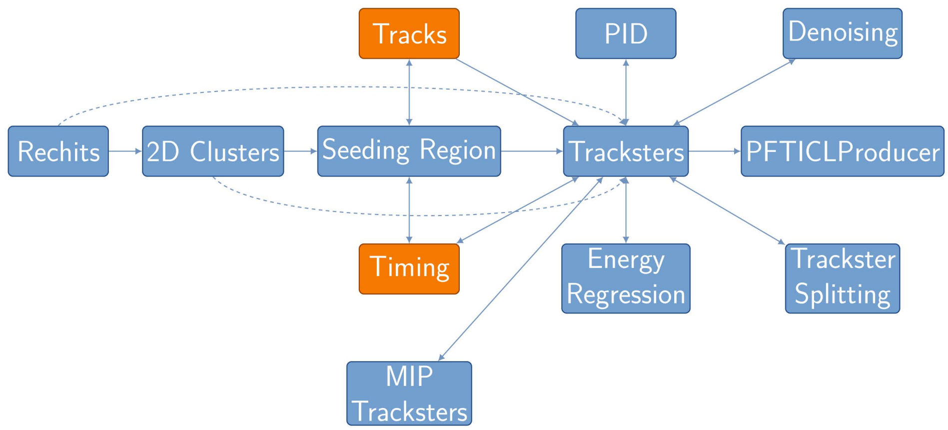

Figure 3: Illustration of the building blocks of the HGCAL TICL reconstruction workflow.

ducer and PFProducer, (2) migrate other calorimeters into the TICL-like workflow (see Fig. 3), which is developed primarily for the HGCAL region reconstruction, but could be adopted to take input from other calorimeters. Regardless of which of these two general approaches CMS will take, much of the work above of porting PF reconstruction modules to GPU will stay relevant. Even for the TICL-like approach, I expect that the PF clusters reconstructed on GPU PFClusterProducer will be used as inputs to **TICL Trackster producer** and **PFTICLProducer**.

At the very basic level, after input tracks and calorimeter clusters are formed, the two approaches perform similar tasks in similar ways. The TICL software is more modular in general and written from the beginning with parallel-processing in mind, and for some modules, both CPU and GPU versions are already available, so from the code structure and organization point of view, it offers advantages and it could help to establish heterogeneous particle flow reconstruction. Also, in the TICL-like approach, we can more naturally use tracks to guide 3-D multi-depth clusters than the current PF workflow, which could help for reconstruction in the Phase 1 upgraded HCAL region. However, from the physics performance point of view, there is large room for improvements. For example, so-called "MIP" iteration seeded by muon detector hits is ready for general use for PF muon reconstruction, and much more importantly track-HGCAL cluster energy reweighting is not worked out yet, which leads to sub-optimal energy assignment for low-energy TICL candidates, and this is where PF reconstruction should bring the largest benefit. The PI is closely communicating with the HGCAL reconstruction group as a PF group convener and as one of the HGCAL collaboration members. In case some of these TICL development items have not yet been worked out yet by the end of this year to a satisfactory level, we would like to contribute to these critical TICL development items based on our in-depth PF reconstruction experience.

For the longer term goal of establishing a coherent PF across all calorimeters, first we need to carefully evaluate the pros and cons of two general approaches discussed above, but as of now I expect that the eventual form of the coherent PF across all calorimeters is closer to the TICL trackster producer/PFTICLProducer than the current PFBlockProducer/PFProducer. In case we decide to adopt the TICL-like approach for other calorimeters, first I will try to extend it to the barrel region by feeding in the PF clusters from the PFClusterProducer, which we are currently working on. The current PF works reasonably well in the barrel region even in the anticipated case of 200 pileup interactions, so this will provide a stringent testbed for the TICL-like approach.

3 About the postdoctoral candidate

Dr. Mark Saunders joined Baylor at the beginning of February 2021 after graduating from FIT with his thesis on the top quark mass measurement using leptonic observables. He has had a strong interest in computing from an early age, and he has a wide range of software experience including C, C++, Python, Bash, CUDA, ML applications using

Keras and Pytorch, and development in a Linux environment. Before starting his doctoral program, he worked as a software engineer developing interfaces between C++ and Python for in-flight networking in commercial aviation at LiveTV, Melbourne, FL. He has experience with programmable GPU interfaces to visualize complex datasets such as fractals in real time. He has limited prior experience with PF reconstruction, but the PI will be able to help him to gain necessary knowledge. During only one month and a half on the PF project, he already improved the processing time of one of the hot spots of PFBlockProducer by a factor of 6, addressed the reproducibility issue of the parallel GPU PFClusterProducer code outputs, and helped to address output differences between the CPU and GPU versions of the PFClusterProducer. He currently serves as a CMS PF group RECO contact.

4 Timeline and milestones

The timeline and the milestones of the proposed project are presented below. Please note that the timeline and milestones, in particular those beyond March 2022, will need to be adjusted based on outcomes of the earlier steps and various activities in the CMS Collaboration as a whole.

June 2021: optimize the GPU version of the layer-wise PFClusterProducer and PFCluster CUDA data format.

September 2021: set up the GPU version of the PFRecHitProducer and the multi-depth PFClusterProducer.

December 2021: integrate the GPU PF reconstruction chain up to producing PF clusters into CMSSW, review the TICL status at that time and start contributing to critical areas e.g. charged candidate energy assignment.

March 2022: commission the PF on GPU for Run 3 HLT, continue on the TICL development in the HGCAL region.

June 2022: evaluate pros and cons of different approaches for a coherent PF reconstruction for the entire detector.

September 2022: establish the first prototype of coherent PF reconstruction across the barrel and HGCAL regions.

January 2023: continue on coherent PF reconstruction, perform physics-level validation of this approach against the current two-path PF approaches. When the core software group settles on the CMS approach for performance portability, start adopting the GPU version of PF softwares to a portability library.

5 Summary

In summary, the particle flow algorithm plays a central role in CMS event reconstruction, and its upgrade for Phase 2 is of paramount importance for the CMS experiment's success during HL-LHC. I think the proposed work on the development of the heterogeneous particle flow reconstruction for Phase 2 has significant impacts on the U.S. CMS HL-LHC S&C program and the CMS Collaboration as a whole. Thank you for your consideration.

References

[1] A. M. Sirunyan *et al.* (CMS Collaboration), JINST **12**, no. 10, P10003 (2017), arXiv:1706.04965.

[2] CMS Collaboration, "The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger TDR," draft.

[3] K. Hatakeyama, "Introduction of Particle Flow," CMS HCAL Days at Fermilab (Sep. 19, 2019).

[4] F. Pantaleo, "The challenge of Heterogeneous Computing," INFN Seminar (Nov. 22, 2017).

[5] J. Pata (for the CMS PF Group), "Machine-Learning based Particle Flow," CMS week (Dec. 2, 2020).

[6] M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo and C. Seez, "CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics," arXiv:2001.09761.

[7] J. Salfeld-Nebgen, "Particle Flow on GPUs," HLT upgrade meeting (Jul. 28, 2020); also PFClusteringOnGPU twiki.

DEPARTMENT OF PHYSICS
PO BOX 97316 · WACO, TEXAS 76798-7316 · (254) 710-2511 · FAX (254) 710-3878