Deploying GPU algorithms through SONIC

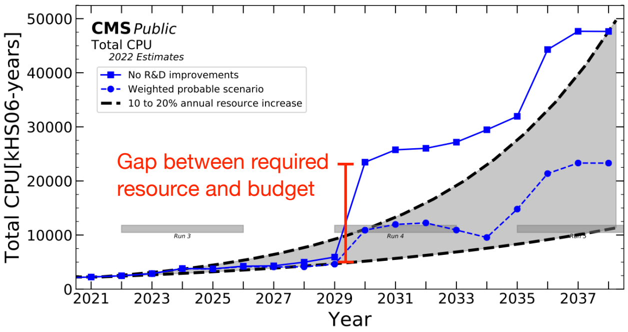
Project goals

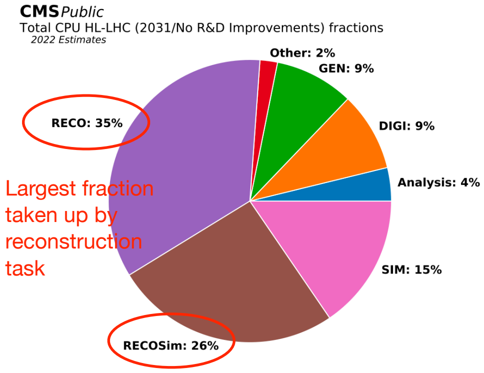
The ultimate goal of the project is to demonstrate at a sufficiently large scale the reconstruction algorithm workflow within CMSSW to be processed, where the client jobs are running on one site, while the Line Segment Tracking (LST) algorithm will be executed on GPUs on computing nodes at another site connected through SONIC (Services for Optimized Network Inference on Co-processors) framework.

Dr. Kelci Mohrman, the postdoc candidate for the R&D initiative project, will first work on integrating the LST workflow with SONIC framework within CMSSW. Dr. Mohrman will carry out deployment of the algorithm at small scale at one single site (i.e., UF T2) as a first milestone, then carry over to two sites (e.g., UF T2 and Purdue T2), and then scale up the test to a larger number of CPU and GPU nodes in various combinations and measure the performance of the throughput and CPU and GPU resource utilization. In particular, careful timing and resource utilization measurements will be performed in order to also provide a step towards measuring the GPU resource needs for the HL-LHC.

GPU algorithms and SONIC

The CMS computing model study shows that the projected required computing resources in Run 4 will exceed the computing resources of the budget. This is shown in the figure below [1].





The largest component of the computing resources is taken up by the reconstruction task of the CMS data processing [1].

There is various algorithm innovation work, including LST [2, 3], that aims to offload reconstruction tasks to GPU co-processors. However, even if the algorithm innovation is successful, how the reconstruction tasks that now demand GPU co-processors in each workflow will be deployed and processed is an equally important problem to be addressed.

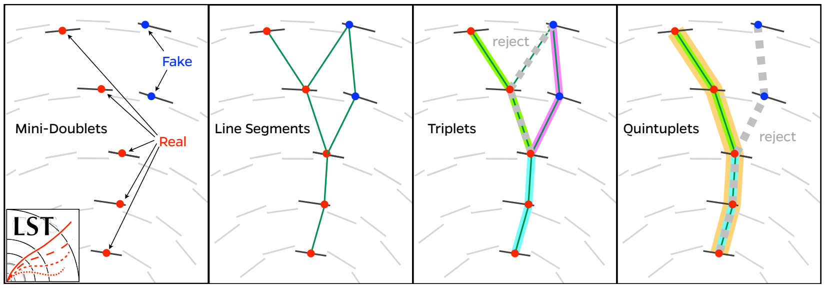
SONIC is a project focused on running machine learning inference on co-processors situated in a different site but provided to the client as a service. In a recent study [4], it was shown that the network protocols set up through SONIC framework can successfully aid in processing machine learning inference for the MiniAOD production task. This demonstration shows that SONIC is primed to try out reconstruction tasks, the largest portion of the computing tasks.

In light of this, **I propose to demonstrate that the LST algorithm can be run as-a-service through SONIC and ultimately lay the groundwork of porting other reconstruction algorithms that use GPUs to a service through SONIC.** If successful, the project will demonstrate that innovative algorithms developed to run on GPUs to address CPU timing resources can be deployed through SONIC and demonstrate that the HL-LHC operations will gain in efficiency of the GPUs that CMS collaboration purchases via through flexibility.

Line Segment Tracking

In HL-LHC, the CMS outer tracker will feature "p_T modules" that are double-layered silicon detectors. The double-layer design allows for pair of hits in each layer to be correlated, providing a rough p_T estimate on the particle that presumably created the hit signals. This opens up a good opportunity to utilize parallel processing architecture to correlate hits between the silicon doublet layers in parallel thus quickly filtering interesting hits first.

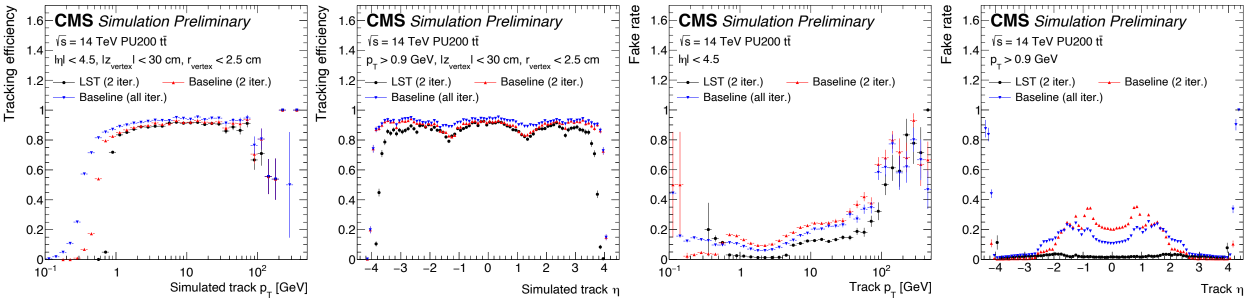
The LST algorithm performs such hit correlations in parallel on GPUs and creates *Mini-Doublets*, that consist of two hits. Subsequently, the pair of mini-doublets are linked to create line segments. Then pairs of line segments are considered to create good candidates of triplets and further quintuplets. An illustration of the steps are shown in the figure below.

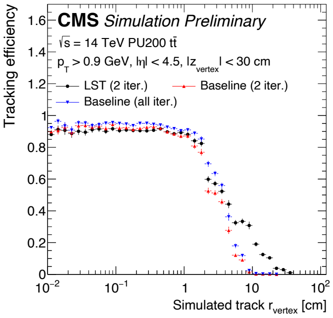


LST in CMSSW

The algorithm has recently been integrated into CMSSW [3]. In CMS iterative tracking, there are more than ten seeds types. Two important seeds are the quadruplet and triplet seeds[1]. These two iteration seeds have been consumed by the LST and track candidates are produced.

When the LST algorithm produced track candidates performance was compared to that of the baseline default tracking algorithm with the two aforementioned seeds, the results show that efficiency is on par while the fake rate is much lower. The efficiency result is shown in the figure below, which can also be found in Ref. [3]. (cf. comparison is to be made between black and red results, to be close to apples-to-apples comparison)





Also, purely outer tracker-based tracks that are pertinent to displaced tracks show efficiency far better than the full iterative tracking results. This can be seen in the figure on the left as well as in Ref. [3]. (cf. blue and black results in the figure on the left.) For tracks with high displaced vertex, the default tracking efficiency drops while the LST performs several factors higher in efficiency. The rest of the integration work is ongoing, and tuning and optimization will take place in the near future.

Expected impact on the timing

As of yet, no comprehensive timing measurement has been carried out, and therefore, it is hard to estimate the final impact to the full reconstruction timing utilizing the LST. Despite being integrated into the CMSSW, the timing measurement cannot account for GPU usage properly.

However, there are some numbers that gives an indication that with LST algorithm there are gains to be expected. Currently, in CMSSW 13.0.0 RelVal profile shows that the reconstruction step for Phase 2 takes 54 seconds on average per event. Among them 21 seconds are spent on tracking. If Phase 1 is taken to be as a guide, the outer-tracker only tracking is expected to take about the same as all pixel-based tracking (i.e. roughly half of all tracking). However, with the LST algorithm a relatively pure outer-tracker tracks have been created on GPU with an average of ~10 ms per event.

Also, the displaced tracking efficiency has greatly been improved over baseline tracking, as mentioned in the previous section, indicating that the LST algorithm is performing unique tracking tasks that the baseline tracking algorithm has not been performing. In other words, if the baseline tracking were to be reconfigured to also target the displaced tracks, the tracking timing would have been larger than the current average of 21 seconds. Therefore, LST is expected to at least do more with the same amount of resources.

Therefore, it will be important in this project to focus on proper timing results. Dr. Mohrman will pay close attention to measuring fair timing measurement and take a first step towards a benchmark standard for future GPU reconstruction algorithm development that will be integrated for HL-LHC.

Project Milestones

Milestone #1: (Month 1, 2)

The first milestone is to run the LST code on SONIC on local CPU nodes with another CPU node with a GPU and successfully obtain input and output of track candidates through SONIC. Towards the first milestone, the first step is to adopt the LST code for SONIC. There are minor code changes to handle sending of the inputs and receiving of the outputs through the Triton server that will be set up on the GPU nodes. The postdoc will develop the code in the LST package to handle inputs and outputs with SONIC. The UF T2 Operations support computing professionals who have root access who can help us with the process if any root privilege is required to set up Triton servers. However, it is currently foreseen to not be necessary or be minimal.

Milestone #2: (Month 3 - 5)

The second milestone is to achieve running reconstruction workflow where the client CPU jobs are running the workflow on one site, and the GPU server that serves the LST tracks are running the LST algorithm on another site. The plan is to work with Purdue T2 as the client, and UF T2 GPUs as the service servers. This milestone will demonstrate a step that is close to a realistic scenario for HL-LHC, and will be a major milestone, albeit only achieved on a small scale at first. The postdoc will also focus on producing timing and throughput results during this phase of the work.

Milestone #3: (Month 6 - 8)

The next step will be focused on scaling it up to a larger number and studying the impact on the throughput as a function of the number of GPUs. The first goal would be to start applying to a few clients with one GPUs and slowly scaling up to O(10) clients to few GPUs. There may be various issues with having multiple client jobs sending LST algorithm tasks to the same GPU. Various debugging is likely necessary. The postdoc will again focus on the timing and throughput measurements.

Milestone #4: (Month 9 - 10)

The last step of the milestone is to scale up the result to a much larger size with O(100-1000) clients. The postdoc will study how things scale with CPU-to-GPU ratios and find an optimal number of GPU resources for maximum throughput.

Milestone #5: (Month 11 - 12)

In this phase, the postdoc will try to reach out to other GPU reconstruction algorithm developers to see if synergies can be made. For example, there is a vertexing algorithm that runs on GPU that could be integrated into the work. This could lay a groundwork for other algorithms to also utilize SONIC to deploy their GPU reconstruction algorithms.

Dr. Kelci Mohrman

Kelci Mohrman graduated from the University of Notre Dame in 2023 and has joined PI Chang's group starting April of 2023. Her thesis topic was "Search for new physics impacting associated top production in multilepton final states using the framework of effective field theory," for which she earned the Department of Physics and Astronomy Research & Dissertation Award from the University of Notre Dame.

Dr. Mohrman has extensive experience in dealing with large-scale computing and deploying software workflow over a large number of computing resources. She has collaborated with the University of Notre Dame's Cooperative Computing Lab (CCL) in the past, focusing on large-scale parallel computing for CMS analysis and workflows, and has made an IEEE publication in the past [5, 6]. Her skillsets and insights in large-scale computing will be useful for this project. For her thesis analysis, a large number of signal events had to be generated. Dr. Mohrman has run millions of CMSSW tasks extensively through workflow management tools from CCL utilizing O(10k) cores opportunistically. Her experience in handling a large amount of CMSSW tasks will be useful for this project.

Dr. Mohrman has also demonstrated analytical skills in investigating a complex piece of code. Dr. Mohrman studied how false minima in likelihood fits of the HiggsCombine tool could affect the fits and developed an approach to navigate around the false minima and produced a solution and contributed by adding a feature to the HiggsCombine tool, which in the near future will be merged to the tool [7].

Since the start of her postdoc appointment (mid-April 2023) Dr. Mohrman has already demonstrated versatility in applying her insights and skillset to the University of Florida T2 computing center and quickly made progress. She has already learned the LST CMSSW workflow that utilizes the GPU and has started to reproduce the results presented in [3]. PI Chang and collaborators can provide technical help on the LST algorithm details and bring her up to speed on the algorithm details. Dr. Mohrman has also ported over the scaling up the coffea analysis framework workflow setup in Notre Dame T3 center utilizing workqueue software from CCL to the University of Florida T2. In the process have already been engaged closely with the computing professionals at UF T2 and have demonstrated an excellent understanding of the computing infrastructure at UF T2 and the Research Computing of UF in general.

Dr Mohrman's past accomplishments prove the competency and expertise necessary for the project's success.

References

[1] CMS Offline Software and Computing, *CMS Phase-2 Computing Model: Update Document*, Tech. rep. CERN-CMS-NOTE-2022-008, 2022. https://cds.cern.ch/record/2815292

[2] Philip Chang et al. *Segment Linking: A Highly Parallelizable Track Reconstruction Algorithm for HL-LHC*. J. Phys. Conf. Ser., 2375(1):012005, 2022.

[3] CMS Collaboration, *Performance of Line Segment Tracking algorithm at HL-LHC*, CMS-DP-2023-019, https://cds.cern.ch/record/2857438?ln=en

[4] Patrick McCormack, *Portable Acceleration of CMS Mini-AOD Production with Coprocessors as a Service*, https://indico.jlab.org/event/459/contributions/11816/

[5] B. Tovar, B. Lyons, K. Mohrman, B. Sly-Delgado, K. Lannon and D. Thain, "Dynamic Task Shaping for High Throughput Data Analysis Applications in High Energy Physics," *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Lyon, France, 2022, pp. 346-356, doi: 10.1109/IPDPS53621.2022.00041.

[6] Kevin Lannon, Paul Brenner, Mike Hildreth, Kenyi Hurtado Anampa, Alan Malta Rodrigues, Kelci Mohrman, Doug Thain, Benjamin Tovar, Snowmass Whitepaper: Analysis Cyberinfrastructure: Challenges and Opportunities, https://arxiv.org/abs/2203.08811

[7] https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit/pull/713