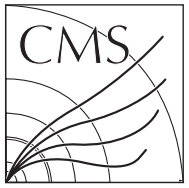
Available on CMS information server

CMS CR -2023/144



The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

31 August 2023 (v2, 21 September 2023)

Running GPU-enabled CMSSW workflows through the production system

Charis Kleio Koraka for the CMS Collaboration

Abstract

The CMS experiment at CERN accelerates several stages of its online reconstruction by making use of GPU resources at its High Level Trigger (HLT) farm for LHC Run 3. Additionally, during the past years, computing resources available to the experiment for performing offline reconstruction, such as Tier-1 and Tier-2 sites, have also started to integrate accelerators into their systems. In order to make efficient use of these heterogeneous platforms, it is essential to adapt both the CMS production system and the CMSSW reconstruction code to make use of GPUs. Ths CMSSW offline reconstruction can now partially run on GPUs, inheriting from the work done at the HLT. Parts of the production systems infrastructure have also been adapted to successfully map, schedule and run the available GPU-enabled workflows on different sites across the computing grid. This talk will describe the process of commissioning GPU-enabled CMSSW workflows through the production system and will present first results from the deployment of GPU-enabled offline reconstruction workflows.

Presented at *CHEP2023 26th International Conference on Computing in High Energy Physics and Nuclear Physics*

Running GPU-enabled CMSSW workflows through the production system

Charis Kleio Koraka[1,*] on behalf of the CMS collaboration.

[1]University of Wisconsin Madison

Abstract. The CMS experiment at CERN accelerates several stages of its online reconstruction by making use of GPU resources at its High Level Trigger (HLT) farm for LHC Run 3. Additionally, during the past years, computing resources available to the experiment for performing offline reconstruction, such as Tier-1 and Tier-2 sites, have also started to integrate accelerators into their systems. In order to make efficient use of these heterogeneous platforms, it is essential to adapt both the CMS production system and the CMSSW reconstruction code to make use of GPUs. The CMSSW offline reconstruction can now partially run on GPUs, inheriting from the work done at the HLT. Parts of the production systems infrastructure have also been adapted to successfully map, schedule and run the available GPU-enabled workflows on different sites across the computing grid. This talk will describe the process of commissioning GPU-enabled CMSSW workflows through the production system and will present first results from the deployment of GPU-enabled offline reconstruction workflows.

1 Introduction

The Compact Muon Solenoid (CMS) detector [1] at CERN's Large Hadron Collider (LHC) generates an immense data volume, with collisions happening at a rate of 40 MHz. In the High Luminosity phase of LHC (HL-LHC), the peak luminosity is projected to be 2.5 times larger than what it currently is, as shown in Figure 1. These challenging conditions, along with the increased granularity of the CMS detector and the higher complexity of the collision events generated by the accelerator will pose major challenges in the data acquisition, offline data processing, simulation, and analysis. Figure 2 shows estimates of the annual projected needs of CPU resources during the HL-LHC for two scenarios, one where no R&D is considered and one that incorporates the most probable outcome of ongoing R&D activities [2]. To address this challenges, the adoption of GPUs has emerged as a crucial strategy. The successful use of GPUs at the High Level Trigger (HLT) farm for LHC Run 3 along with the visible shift of the hardware market towards GPUs and heterogeneous platforms have showcased the need for exploring GPU-enabled offline workflows and enabling support for running GPU workflows on the computing grid.

*e-mail: charis.kleio.koraka@cern.ch

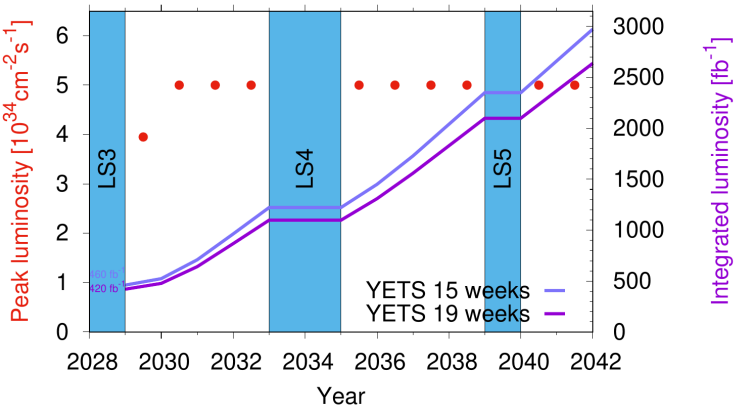


Figure 1. Evolution of the projected peak luminosity (red dots) and the total integrated luminosity (solid lines) during the High Luminosity phase of the LHC. Taken from [3].

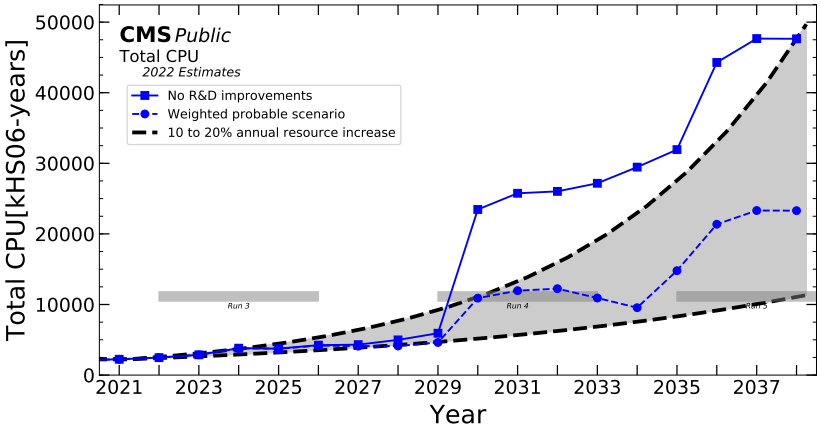


Figure 2. Estimates of the annual projected needs of CPU resources during the HL-LHC. Two scenarios are considered, the first one being a baseline scenario with no improvement due to ongoing R&D factored in (squares), and the second one which incorporates the most probable outcome of the ongoing R&D activities (circles). The gray band shows the projected resource availability for a scenario that extrapolates the 2021 CMS pledged resources using an annual increase in available resources of between 10% and 20%.

2 GPUs at the WLCG

The Worldwide LHC Computing Grid (WLCG) [4] is a global network of computer centers aimed at managing and analyzing data collected from the different LHC experiments. The WLCG employs a tiered structure to efficiently handle data processing and distribution, composed of several levels, each with a specific set of services. The Tier-0, which is situated at

CERN, receives, process and stores raw data directly from the different LHC experiments. Tier-1 and Tier-2 centers, located in various countries, store significant portions of data and provide substantial computing resources for data processing and analysis. A schematic representation of the tiered structure of the WLCG is shown in Figure 3. Traditionally, these centers primarily utilized CPU-based systems for their computational needs but over the past years, there has been a notable shift in the computing landscape. Many of these centers have already integrated GPUs into their computing infrastructure as can be seen in Figure 4.

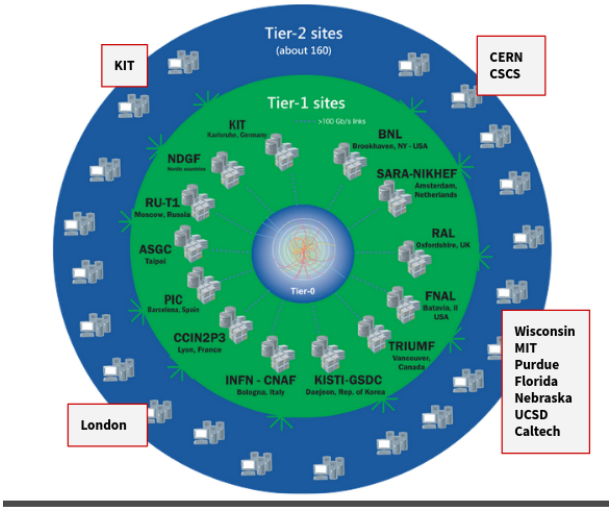


Figure 3. Schematic representation of the tiered structure of the Worldwide LHC Computing Grid (WLCG). Lists of Tier-1 and Tier-2 sites that are equipped with GPUs are shown inside the gray boxes.

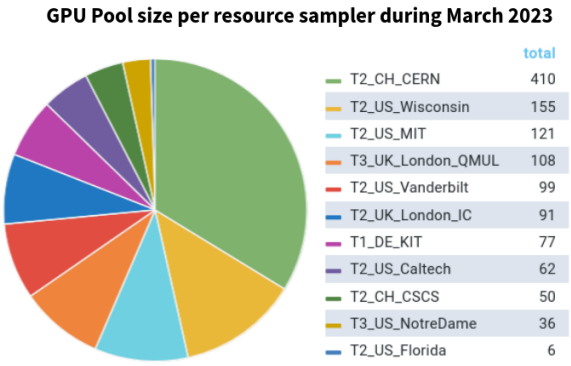


Figure 4. Number of GPU resources available to CMS during March 2023. The information in the pie chart is acquired via the periodic submission of probe jobs (pilots) that request and run on grid sites that have GPU resources available.

3 A GPU-enabled High Level Trigger

Since the beginning of Run 3, the CMS experiment has leveraged GPUs for the online reconstruction at HLT. Currently, several HLT algorithms can be executed on GPU, including the tasks of the ECAL local reconstruction [5], the HCAL local reconstruction [6] and the pixel tracking and vertex reconstruction [7]. The offloading of these algorithms on the GPU has resulted in a significant improvement in the average time spent to reconstruct each event [8]. This is shown in Figure 5, where the CPU only reconstruction timing is shown against the timing of the CPU and GPU combined reconstruction. The reduced average timing also results to an increased throughput of the order of 80% as shown in Figure 6.

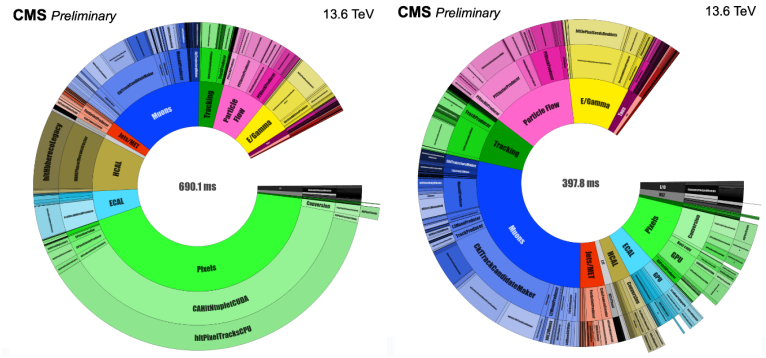


Figure 5. Measurement of average reconstruction time spent per event for two different configurations. The pie chart on the left shows the average time and the breakdown of the time spent in each CMSSW module when a CPU-only configuration is utilized. The pie chart on the right shows the average time and the breakdown of the time spent in each CMS software (CMSSW) module in a CPU and GPU combined configuration. The HLT timing improves by 40% when part of the CMSSW is offloaded to GPU.

4 Overview of CMS offline workflows

CMS executes a variety of tasks on its distributed computing infrastructure in order to reconstruct and simulate collision data and analyze both. These tasks can be categorized into three distinct workflow types, each serving a specific purpose. Figure 7 illustrates the various workflow types and the corresponding steps encompassed by each. The McM workflows yield simulated collision data, while the ReReco workflow's objective is the re-reconstruction of collision data through refined calibrations and corrections. The purpose of RelVal workflows is to validate changes in new releases of the CMS software and can either run on data or produce simulated collision data. The initial step in simulating collision data involves the generation task, which employs Monte Carlo event generators to generate events based on theoretical principles. Following this, the simulation task takes the outcomes of the generation task and emulates the energy deposition resulting from particle interactions with the material within the CMS detector while the digitization task follows and is used to model the electronic processing of the signal produced by the energy depositions in the detector. A common step for all workflows is the offline reconstruction task which includes all algorithms necessary to decipher signals as a result of interactions involving identifiable particles within the detector. For this step, the CMS software (CMSSW) is utilized. Part of the offline reconstruction can be offloaded to GPUs inheriting from what is done online at the HLTs

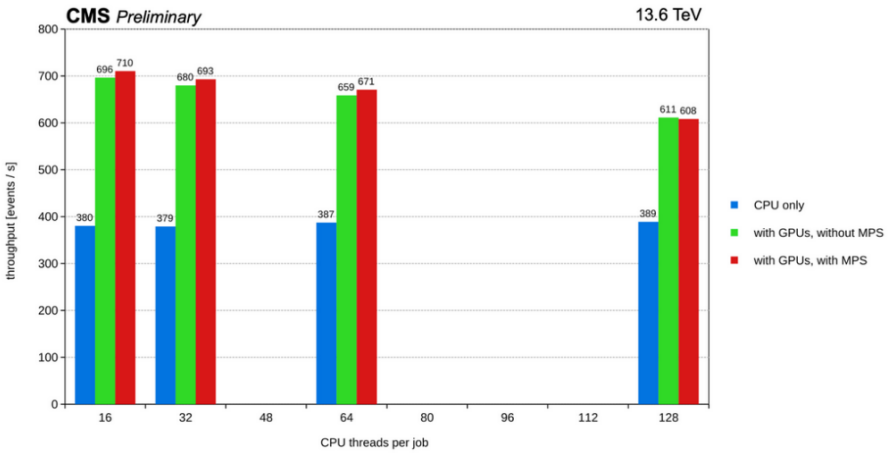


Figure 6. HLT throughput measured under different conditions. The blue bars correspond to jobs running only on CPU, the green bars correspond to jobs offloading part of the computations to GPU while the red bars correspond to jobs offloading part of the computation to the GPU using the NVIDIA Multi-Process Server. The different groups of bars correspond to measurements with different number of jobs (16, 8, 4, 2) and threads per job (16, 32, 64, 128).

streamlined version of the reconstruction, as described in Section 3. Finally the reduced data format step is responsible for producing smaller data formats optimized for space saving.

5 Adding GPU support to the Workload Management System

In order to submit jobs to all the grid sites available, CMS has created a Global Pool of resources, based on the HTCondor and GlideinWMS software technologies [2]. Each job is described in terms of the required computing resources, such as the number of CPU cores, memory or hardware accelerator. Through a process of matchmaking, suitable resources are assigned to the corresponding jobs, according to the needs of each workflow. Initially, the workflow is created, submitted and directed to the request manager. The request manager is then responsible for creating the request, validating the request parameters and finally placing the request into the global work-queue. Several adaptations have taken place in the workload management framework in order to support GPUs in central production workflows. Two new request parameter options were created enabling the discovery of GPUs during the resource provisioning and matchmaking process. These include the RequiresGPU parameter which defines whether the workflow requires GPU resources or not and the GPUParams parameter, which is a set of key/value pairs defining what are the GPU hardware requirements. After the request is placed in the global work-queue, the work-queue elements are split into jobs and are sent to the condor pool. The workflow level GPU configuration, defined within the GPUParams parameter, gets reflected in the job classad description. The final step of the workflow management is the HTCondor matchmaking, where resources are finally attached to jobs. A simplified schematic representation of the GPU workload management is pictured in Figure 8.

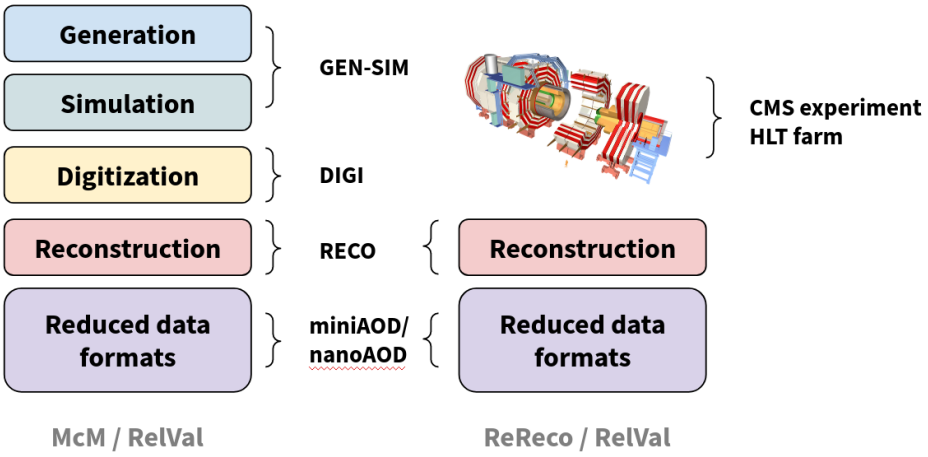


Figure 7. Illustration of the various CMS workflow types and the corresponding steps involved in each workflow. The McM workflows yield simulated collision data while the ReReco workflows utilize the data collected at the HLT and re-reconstruct them making use of refined corrections and calibrations. RelVal workflows can either run on data or produce simulated collision data and are used to validate changes in new releases of the CMS software.

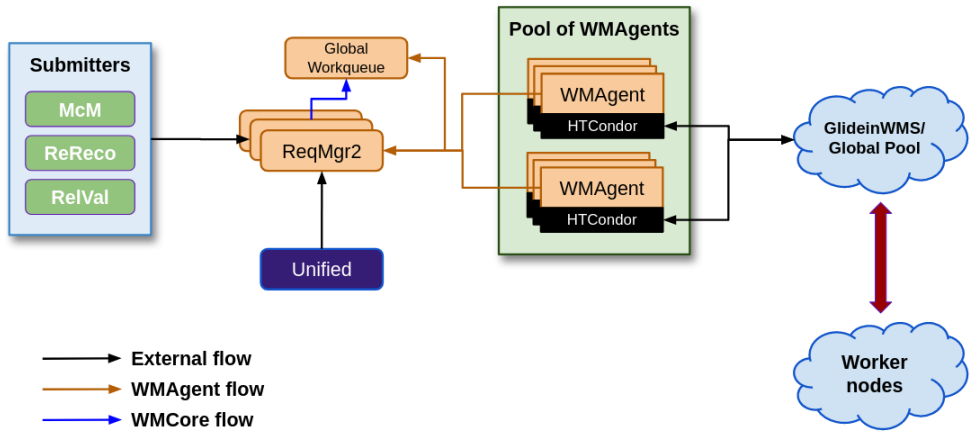


Figure 8. Simplified diagram that shows the workload management of central production workflows that require GPU resources.

6 Usage of GPUs in production

The addition of GPU support to the Workload Management System allowed for the submission of offline GPU workflows through central production. One of the first offline GPU workflows submitted to the grid was a ReReco workflow, goal of which was to perform a large

scale validation of the CMMSW code before it was deployed online and used for data-taking in mid-2022. The resource utilization of this workflow can be seen in Figure 9, which shows the number of GPU hours spent by the corresponding CMS jobs running on the Wisconsin Tier-2 site as a function of time. In addition to the large scale ReReco workflows, several release validation workflows (RelVals) were submitted through central production to validate updates in GPU reconstruction code and make sure that the CPU-only and the combined CPU and GPU configuration was giving equivalent results. Examples of the distributions comparing the CPU-only and combined CPU and GPU results are shown in Figure 10.

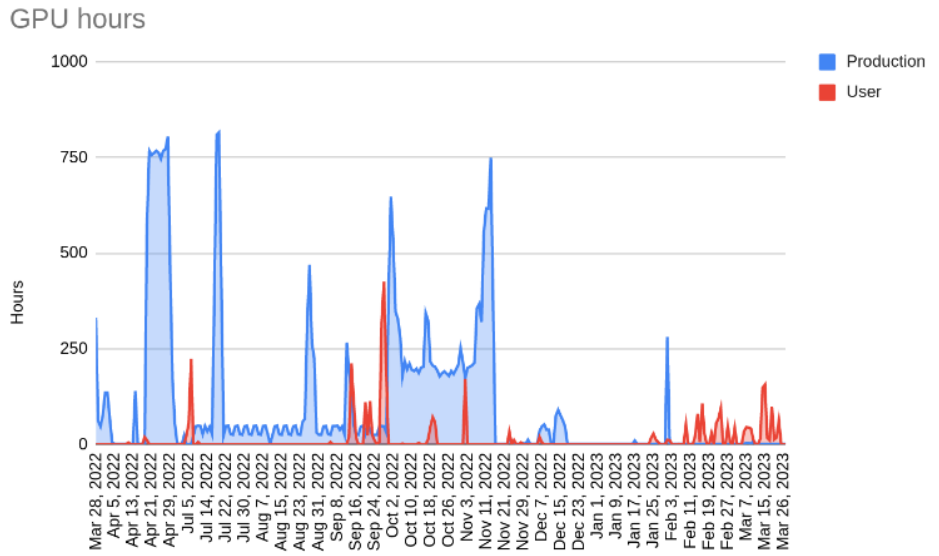


Figure 9. Number of GPU hours spent by CMS jobs running on the Wisconsin Tier-2 shown as a function of time. The blue curve shows the number of GPU hours corresponding to jobs submitted to the grid through the central CMS production system while the red curve shows the number of GPU hours corresponding to jobs submitted to the grid by users.

7 Conclusions

The HL-LHC will pose a significant challenge for the offline computing world of CMS. The successful use of GPUs at the HLT as well as the shift of the hardware market towards GPUs and heterogeneous platforms have showcased the need for exploring GPU-enabled offline workflows. In this contribution, the process of enabling GPU support for the CMS workload management system was described. The efficient distribution of GPU enabled jobs across the computing grid has been facilitated by the addition two new request parameter options allowing the discovery and matchmaking of jobs to available GPU resources. These developments have allowed the submission and execution of large-scale GPU-enabled workflows on the computing grid, allowing for the validation of the CMS reconstruction code before being deployed and used for data-taking.

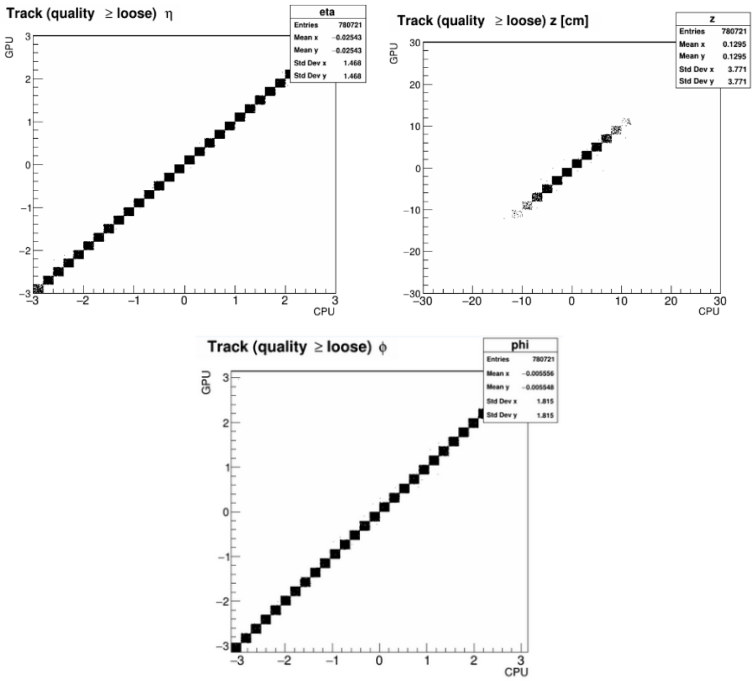


Figure 10. Example of plots produced by the Release Validation monitoring system of CMS. The plots are used to check whether the event reconstruction on CPU-only and on CPU and GPU gives similar results. From top left to bottom, the figures show a comparison of the reconstructed tracks pseudo-rapidity η, z position and *phi* angle when the event reconstruction is performed on CPU-only (x-axis) vs CPU and GPU (y-axis).

References

[1] S. Chatrchyan et al. (CMS), JINST **3**, S08004 (2008)

[2] C.O. Software, Computing, Tech. rep., CERN, Geneva (2022), `https://cds.cern.ch/record/2815292`

[3] *HL-LHC Project*, `https://espace.cern.ch/HiLumi/WP2/Wiki/HL-LHC%20Parameters.aspx`

[4] *LHC Computing Grid (LCG) project*, `http://www.cern.ch/lcg/`

[5] T. Reis, for the CMS Collaboration, Journal of Physics: Conference Series **2438**, 012027 (2023)

[6] A. Massironi, V. Khristenko, M. DAlfonso, Journal of Physics: Conference Series **1525**, 012040 (2020)

[7] A. Bocci, V. Innocente, M. Kortelainen, F. Pantaleo, M. Rovere, Frontiers in Big Data **3**, 601728 (2020)

[8] *Commissioning CMS online reconstruction with GPUs* (2022), `https://cds.cern.ch/record/2851656`