Exploiting GPU Resources at VEGA for CMS Software Validation

Daniele Spiga[1,*] *Adriano* Di Florio[2], *Andrea* Bocci[3], *Antonio* Pérez-Calero Yzquierdo[4,6], *Christoph* Wissing[5], and *José M.* Hernández[6]
for the CMS Collaboration

[1]INFN Sezione di Perugia, Via A. Pascoli 23c, 06123 Perugia, Italy
[2]CC-IN2P3
[3]CERN (Switzerland)
[4]Port d'Informació Cientifica (PIC), Barcelona, Spain
[5]Deutsches Elektronen-Synchrotron, Notkestr. 85, 22603 Hamburg, Germany
[6]Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain

Abstract. In recent years, the CMS experiment has expanded the usage of HPC systems for data processing and simulation activities. These resources significantly extend the conventional pledged Grid compute capacity. Within the EuroHPC program, CMS applied for a "Benchmark Access" grant at VEGA in Slovenia, an HPC centre that is being used very successfully by the ATLAS experiment. For CMS, VEGA was integrated transparently as a sub-site extension to the Italian Tier-1 site at CNAF. In that first approach, only CPU resources were used, while all storage access was handled via CNAF through the network. Extending Grid sites with HPC resources was an established concept for CMS, however, in this project, HPC resources located in a different country from the Grid site were first integrated. CMS used the allocation primarily to validate a recent CMSSW release regarding its readiness for GPU usage. Former developments in the CMS workload management system that allow the targeting of GPU resources in the distributed infrastructure turned out to be instrumental and jobs could be submitted like any other release validation workflow. The presentation will detail aspects of the actual integration, some required tuning to achieve reasonable GPU utilisation, and an assessment of operational parameters like error rates compared to traditional Grid sites.

1 Introduction

Over the last few years, the CMS [1] experiment has begun to evolve its computing system to integrate heterogeneous resource providers such as High Performance Computing (HPC) centers. This is one of the anticipated solutions to help the experiment manage the increased demands of the future HL-LHC. For now, this resource acquisition focuses solely on computing, excluding storage, and is primarily intended for opportunistic use. As such, it extends beyond the capacity provided by the Grid. HPC integration has increasingly become a key asset of CMS Computing. Since 2019, continuous efforts have led to the successful integration of multiple HPC machines, which are now consistently used in production [2].

*e-mail: daniele.spiga@pg.infn.it

Given the heterogeneous nature of the environment, no single solution can address all challenges. Instead, CMS Computing, in collaboration with local teams, has developed and deployed a diverse portfolio of technical strategies, enabling seamless integration between the experiment and various HPC systems. This strategically positions CMS towards the exploitation of additional centers.

CMS aims to further expand its use of HPC resources, particularly in the EU, where its current utilization remains lower compared to the US. Towards the end of 2023, CMS successfully applied for a EuroHPC benchmarking grant on VEGA, the first of eight peta and pre-exa-scale EuroHPC Joint Undertaking machines, hosted at IZUM in Maribor. The grant gave CMS access to the VEGA GPU Partition, whose nodes have two AMD Epyc 7H12 2.6 GHz CPUs with a total of 128 core, 512 GB RAM and 4 NVIDIA A100 GPUs. Although Slovenia is not a member of the CMS Collaboration, the grant was an excellent opportunity to prove the flexibility of the CMS computing system across several dimensions, in particular:

- To integrate a world-class HPC center as a logical partition of an existing WLCG site, demonstrating the feasibility of a transnational site extension.
- To exploit a large number of opportunistic accelerated nodes to dynamically extend the pool of GPUs already available to CMS Offline Computing.
- To demostrate the capability of executing “any type of workflow” on HPC sites, while gaining valuable experience with the operational model to ensure long-term sustainability.

This paper describes the process and the strategy adopted to achieve these goals. After detailing the more technical aspects of the resources integration, the rationale of the workflow selection and the use case adopted to achieve the best out of the grant will be described. Finally, results and future perspectives will be given.

2 VEGA integration via transnational site extension

From a technical perspective, HPC centers differ from regular Grid sites in both hardware configurations and operational policies. The latter are mostly driven by security reasons. Based on the current CMS experience in this context, the major differences include variations in operating systems, management of scratch disk space on compute nodes, core-to-memory ratio, and, most importantly, outbound network connectivity, which is often very limited, or in some cases, totally absent. Finally, local disk storage for handling input and output data is also managed very differently compared to the highly federated model of WLCG [3]. However, as already stated, storage management is not part of the presented work, also because there is not yet the concept of opportunistic storage for managing the CMS data. Following a requirement-driven approach to overcome these limitations, CMS has developed several solutions over the past few years, enabling the exploitation of multiple supercomputers. Overall, the activities can be allocated in the following three categories:

- Overlay batch model: it consists of creating a batch system or extending an existing one on top of the batch provided by the site itself. The execution of the regular CMS pilots creates the connection of the HPC batch system to CMS. Pilots reach the HPC batch via one of the natively supported mechanisms, such as schedd forking, job routing, etc. Currently, the two major implementations of this model are HEPCloud in the US [4] and COBalD/TARDIS [5] in Germany.
- Site Extension: it identifies the HPC resources as belonging to a Grid site, and are located in a distinct physical location (sub-site). This approach relies on a concept already provided by CMS and it is not strictly connected to a specific technical implementation. In

fact, there are several ways to implement this model, with two notable approaches as of the time of writing: the Italian CNAF Tier1 that extends its capacity using CINECA resources [6], and the above-mentioned COBalD/TARDIS system in Germany.

- HTCondor split starter: a mechanism that enables the use of a shared file system as communication path for HTCondor. This is used in cases where nodes are located in a highly segregated network. This is for example the case at the Barcelona Supercomputing Center (BSC), where the only available communication channel with the outside world is through incoming SSH connections into the user login machines [7].

Despite the various technical variations in the approaches, except for the HTCondor split starter, they shared a common trait: they all ultimately start glideins, which then bring job slots into the CMS GlobalPool [8].

2.1 Technical setup at VEGA

The strategy to integrate the VEGA computing resources is based on the site extension to the Italian Tier-1 CNAF. The motivation for this choice is twofold: first, being VEGA a sub-site of the regular Tier-1 at CNAF, it is completely transparent to the CMS operations team. Second, it is extremely lightweight from a technical integration perspective and minimizes the impact on site support. The decision to use the CNAF Tier-1 was primarily due to their geographical proximity (as one of the CMS Tier-1 sites near VEGA) and because CNAF already has experience integrating HPC resources through the concept of elastic site extensions. This has been the case for CINECA resources since 2019 with Marconi KNL, with Marconi100 in 2022, and with the Leonardo pre-exascale machine in 2024.

Figure 1. First time CMS successfully integrated an HPC resource located in a different country from the Grid site, with VEGA transparently integrated as a sub-site extension to the Italian Tier-1 site at CNAF.

The setup was designed with VEGA HPC as a storage-less provider, relying on the following key aspects:

- Manual execution of CMS regular Pilot jobs (glidein wrappers) without any modification to either the pilot or the corresponding entry in the glidein Factory.

- Lightweight mechanism to maintain pressure on VEGA's SLURM queues. Instead of being pushed from the GlideinWMS (CMS pilot factory), the pilots were directly injected into the VEGA batch system as slurm jobs. This is possible thanks to the manual pilot functionality officially supported by the CMS computing system.

- Remote data access via the xrootd federation (AAA) [9]. This was possible as VEGA allows for outbound connectivity.

- Stage out to CNAF storage element. The VEGA sub-site was configured to stage all the produced output files directly to the CNAF Grid storage site via WebDAV [10]. The configuration foresaw a fallback to CERN.

- Site provided services. VEGA provided us the possibility to use Squid caching web proxies supported centrally by the site. Moreover, the Apptainer (formerly known as Singularity) container platform [11] and the CVMFS [12] client were already available as well. All this simplified the whole technical integration process.

- Scratch area on NVMe. In order to maximize the CMS job efficiency as well as to minimize impact on the share file system at VEGA during the scaling tests, some fine tuning was made to better configure the scratch local area for the pilots. The conclusion was to slightly customize the setup to enforce the usage of the NVMe present on the VEGA nodes.

Figure 2. A excerpt of the CMS Submission Infrastructure GPU monitoring. VEGA NVIDIA A100 equipped node appears as CNAF actual resources.

Finally, regarding the integration of the GPU partition, no custom setup was needed, only fine-tuning of the pilot configuration. Indeed, both the glidein and the glidein Factory already provided all the handles to target the GPU-equipped sites.

Thanks to the flexibility of the CMS computing system and the prompt support from the site, the VEGA integration was successful. This has been crucial in expanding the pool of GPU slots available through the global pool. The integration has been completely transparent, as demonstrated in figure 2, where CMS systems continue to recognize only the Tier-1 CNAF resources, even though they are actually provided by VEGA. With the VEGA resources, the NVIDIA A100 graphics card became available to CMS, which was crucial for fully leveraging the potential of these resources, as explained in Sec. 3.

3 Workflows selection

CMS envisions HPC integrations to be as seamless as possible. A key motivation for this approach is to minimize the operational costs associated with managing diverse providers and selecting workflows on a case-by-case basis. Ideally, nearly all production payloads should be able to run on any HPC system. However, this is not yet fully achievable. As a result, to maximize resource utilization, CMS must carefully select the most suitable workflows for each specific center. For example, a common approach is to prioritize workflows with

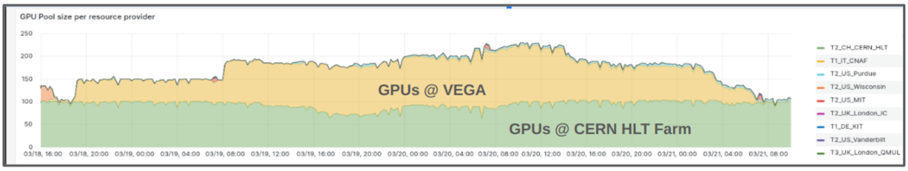

Figure 3. The plot shows the pool of GPU-equipped nodes available to CMS. During the considered time interval, the two main contributors are the HLT farm at CERN and the VEGA nodes.

minimal I/O demands to mitigate integration challenges, particularly those related to network constraints at the site.

Since one of the technical goals of the presented work was to demonstrate the feasibility of running even complex workflows at VEGA, CMS used Release Validation [13] (RelVal). They are composed of 55 distinct and chained tasks. In this work, the RelVal workflows were used both to commission the system and also to generate valuable data for an Alpaka [14] validation campaign, which is essential for the CMS High Level Trigger (HLT) in preparation for the 2024 data-taking period. Indeed, with the start of Run3, CMS has successfully offloaded part of the online reconstruction to NVIDIA GPUs (via CUDA). This includes the reconstruction of HLT pixel tracks and vertices. From 2024 data taking, CMS has chosen Alpaka as performance portability library to have the possibility to target different CPUs and GPUs with a single code base. Alpaka is a header-only library that provides portability of code for various backends by adding an abstraction layer. Also, the performance with respect to the native CUDA was found basically untouched [15] [16].

The actual exploitation of VEGA happened in two main steps. The first was dedicated to the commissioning of the system and running clones of regular CMS workflow (CPU only). Both standard Monte Carlo workflows with remote read of the pre-mixed pileup and data reprocessing with remote read of primary input were tested. The second step, the production phase, was focused in the actual execution of the dedicated release validation (RelVal) workflows executing the Alpaka-based version of the HLT code to produce official samples for the physics validation. In this case, the VEGA GPU-equipped nodes (offline resources) were used to execute HLT-like workflows running simulations. Overall more than 0.6M CoreHrs were used by CMS at VEGA. As shown in figure 3, during the final phase of the processing, CMS successfully managed to exploit, concurrently, more than half of the total GPUs available (124 out of 240 in total).

4 Validation strategy and results

The production campaign had as its objective the validation of the port of the inner tracker reconstruction from native CUDA to Alpaka. This has been done in two ways.

For one set of the production, four different setups for the HLT pixel track reconstruction have been run on the same set of simulated events:

- the pre-Alpaka migration pixel tracking running on CPU;
- the pre-Alpaka migration pixel tracking running on NVIDIA GPUs with native CUDA;
- the new Alpaka pixel tracking running on CPU;
- the new Alpaka pixel tracking running on NVIDIA GPUs;

For each of the setups the pixel tracking physics performance has been measured using mainly two metrics: the *efficiency* and the *fake rate*. The *efficiency* is defined as the percentage

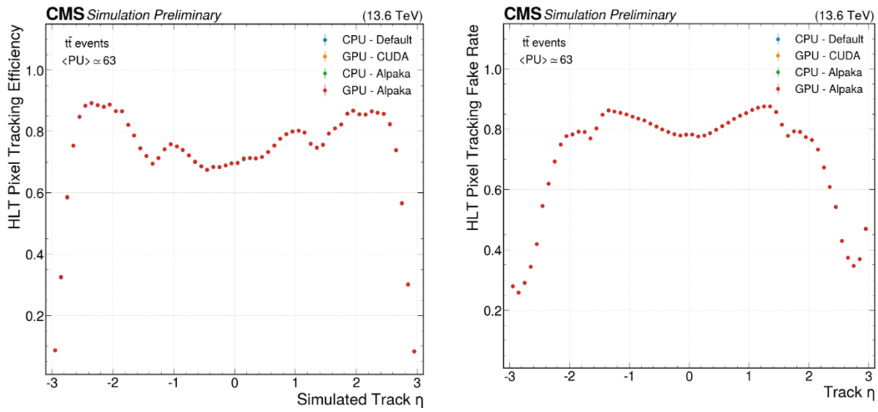

Figure 4. On the left plot the HLT pixel tracking efficiency is shown as a function of the simulated track pseudorapidity η. On the right plot pixel tracking fake rate is shown as a function of the reconstructed track η. Four collections are compared: the pre-Alpaka migration HLT pixel tracks running on CPU (in blue), the pre-Alpaka migration HLT pixel tracks running on GPU with native CUDA (orange), the Alpaka HLT pixel tracks running on CPU (green), the Alpaka HLT pixel tracks running on GPU (red)

of simulated tracks (N_{sim}) that have been associated with at least one reconstructed track ($N_{ass(reco \to sim)}$). The fake rate is defined as the fraction of all the reconstructed tracks (N_{rec}) that are not uniquely associated to a simulated track ($N_{ass(sim \to reco)}$). The results for this first round of comparisons are shown in figure 4. The four distributions are in almost perfect agreement.

A second part of the validation campaign was devoted to monitoring the possible discrepancies arising from running the same algorithms alternatively on NVIDIA GPUs or CPUs. In order to achieve this, the pixel track reconstruction was run twice on CPU and on GPU in a single job and for each event. This allowed us to compare event by event the pixel tracks collection properties, such as the number of reconstructed tracks and the track parameters

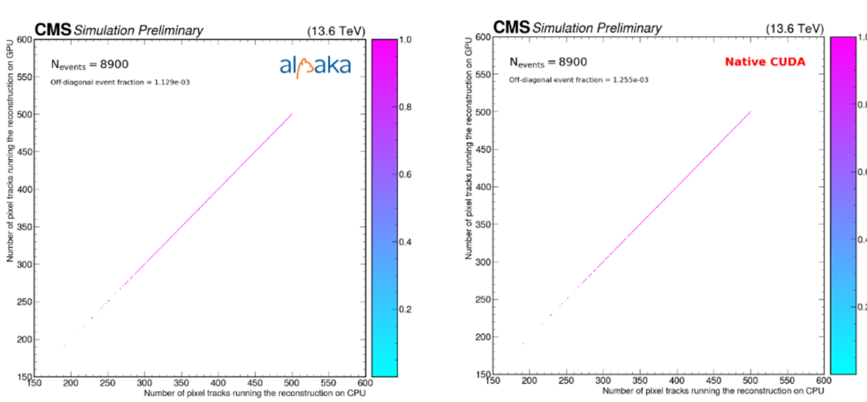

Figure 5. The two plots show the comparison of the number of pixel tracks per event for Alpaka (left) and native CUDA (right). The x-axis(y-axis) represents the number of pixel tracks running the reconstruction on CPU(GPU). Each column is normalized to one.

(e.g. η, ϕ, p_T, n_{hits}). Such a workflow has been run for both the Alpaka and the pre-Alpaka (native CUDA) migration setup. As an example, see in figure 5, two two-dimensional plots comparing the number of reconstructed tracks on GPU and CPU for the two setups. The two reconstructions are in very good agreement. In addition, the native CUDA distribution and the Alpaka one show a comparable off-diagonal deviation. In both cases, the common simulated sample consisted of $t\bar{t}$ events with a superimposed set of parasitic collisions, collectively referred to as *pileup* (PU), with $< PU > \simeq 63$.

4.1 Summary and next steps

CMS received a benchmark grant at VEGA, the Slovenian EuroHPC site. Although the allocated resources were not extensive, the system provided a valuable opportunity to challenge CMS from both software and computing perspectives. On the technical side, it was instrumental in fine-tuning the dynamic resource integration process.

For the first time, CMS successfully implemented a transparent site extension, integrating a world-class HPC resource located in a different country from the extended Grid site. CMS managed to exploit more than 0.6M CoreHrs. The VEGA GPUs successfully extended the CMS pool of accelerators and their exploitation allowed the generation of official samples needed for the validation campaign of the CMSSW software for the HLT in preparation for the 2024 data-taking period. This work reaffirms the flexibility and agility of the CMS computing system in accommodating unconventional resources as well as heterogeneous architectures.

5 Acknowledgements

The authors and the CMS collaboration as a whole would like to thank the VEGA HPC Center and in particular A. Filipcic and T. Prica for their collaboration, support, and interest in the computing activities of CMS. The VEGA integration is funded via a EuroHPC Project Benchmark Grant (2023B12-002).

References

[1] S. Chatrchyan, et al, The cms experiment at the cern lhc, 2EPJ Web Conf. (2008). 210.1088/1748-0221/3/08/S08004
[2] J. Adelman-McCarthy, T. Boccali, R. Caspart, A. Delgado Peris, M. Fischer, J. Flix Molina, M. Giffels, J.M. Hernández, D. Hufnagel, E. Kühn et al., Extending the distributed computing infrastructure of the cms experiment with hpc resources, Journal of Physics: Conference Series **2438**, 012039 (2023). 10.1088/1742-6596/2438/1/012039
[3] I. Bird, et al., Architecture and prototype of a wlcg data lake for hl-lhc, EPJ Web Conf.214 p. 04024 (2019). 10.1051/epjconf/201921404024
[4] P. Mhashilkar, M. Altunay, E. Berman, D. Dagenhart, S. Fuess1, B. Holzman, J. Kowalkowski, D. Litvintsev, Q. Lu, A. Moibenko et al., Hepcloud, an elastic hybrid hep facility using an intelligent decision support system, EPJ Web of Conferences 214 (2019). 10.1051/epjconf/201921403060
[5] M. Fischer, E. Kuehn, M. Giffels, M.S. abd A. Petzold, A. Heiss, Lightweight dynamic integration of opportunistic resources, EPJ Web of Conferences 245 (2020).
[6] T.Boccali, S. Pra, D.Spiga, et al., Extension of the infn tier-1 on a hpc system, EPJ Web of Conferences 245 (2020). 10.1051/epjconf/202024509009

[7] C. Acosta-Silva, A.D. Peris, J.F. Molina, J. Frey, J. Hernández, M. Livny, A.P.C. Yzquierdo, T. Tannenbaum, Exploiting network restricted compute re-sources with htcondor: a cms experiment experience, EPJ Web of Conferences 245 (2020). 10.1051/epjconf/202024509009

[8] J. Balcas, et al., Using the glideinWMS System as a Common Resource Provisioning Layer in CMS, J. Phys.: Conf. Ser. 664 062031 (2025).

[9] K. Bloom, T. Boccali, B. Bockelman, D. Bradley, S. Dasu, J. Dost, F. Fanzago, I. Sfiligoi, A. Tadel, M. Tadel et al., Any data, any time, anywhere: Global data access for science, 2EPJ Web Conf. (2015).

[10] B. Bockelman, A. Ceccanti, F. Furano, P. Millar, D. Litvintsev, A. Forti, Third-party transfers in WLCG using HTTP, EPJ Web Conf. **245**, 04031 (2020), `2007.03490`. 10.1051/epjconf/202024504031

[11] M.B. G.M. Kurtzer, V. Sochat, Singularity: Scientific containers for mobility of compute, PLoS ONE 12(5): e0177459 (2015). 10.1371/journal.pone.0177459

[12] J. Blomer, et al, Cernvm-fs: delivering scientific software to globally distributed computing resources, Proceedings of the first international workshop on Network-aware data management (2011). 10.1145/2110217.2110225

[13] T.C. Collaboration, Validation of software releases for cms, J. Phys. Conf. Ser, 219 (2010).

[14] E. Zenker, B. Worpitz, R. Widera, A. Huebl, G. Juckeland, A. Knüpfer, W.E. Nagel, M. Bussmann, Alpaka – An Abstraction Library for Parallel Kernel Acceleration, in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (2016), pp. 631–640

[15] A. Bocci, Adoption of the alpaka performance portability library in the CMS software, in *26th International Conference on Computing in High Energy & Nuclear Physics* (2023)

[16] T.C. Collaboration, 2024 HLT timing and throughput results (2024), `https://cds.cern.ch/record/2914421`