CMS High Level Trigger performance comparison on CPUs and GPUs

Andrea Bocci, on behalf of the CMS Collaboration
CERN, EP/CMD

E-mail: andrea.bocci@cern.ch

Abstract. At the start of the upcoming LHC Run-3, CMS will deploy a heterogeneous High Level Trigger (HLT) farm composed of x86 CPUs and NVIDIA GPUs. In order to guarantee that the HLT can run on machines without any GPU accelerators - for example as part of the large scale Monte Carlo production running on the grid, or when individual developers need to optimise specific triggers - the HLT reconstruction has been implemented both for NVIDIA GPUs and for traditional CPUs. This contribution will describe how the CMS software used online and offline (CMSSW) can transparently switch between the two implementations, and compare their performance on GPUs and CPUs from different architectures, vendors and generations.

1. Introduction

At the start of Run-3 in 2022, the Compact Muon Solenoid (CMS) experiment at CERN will deploy a High Level Trigger (HLT) farm composed of 200 dual processor servers, each equipped with two AMD EPYC "Milan" 7763 CPUs and two NVIDIA Tesla T4 GPUs. This GPU has a half-width, half-length form factor and a power consumption of 70 W – a low power, compact GPU suitable for deployment in the "1U" servers chosen for the HLT farm.

CMS uses a single simulation, reconstruction and analysis software (CMSSW) for both the online selection at the HLT and the offline data processing and analysis. An overview on the use of GPUs at HLT planned for Run-3 and later can be found in Chapter 12 of the *Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger Technical Design Report* [1]. The solutions that have been developed to make an efficient use of GPUs (or other accelerators) and support the transparent fall-back to CPU-only processing are described in [2]. The algorithms that have been ported to the CUDA platform to run on NVIDIA GPUs are described in [3], [4] and [5]. The exploration of different *performance portability* solutions to automate the CPU implementations of these algorithms and the porting to other GPU architectures are described in more detail in [6] and [7].

Section 2 describes the performance of the HLT configuration of CMSSW running only on CPUs: Section 2.1 examines the scaling of the performance with the number of threads; Section 2.2 shows the evolution of HLT performance across different generations of CPUs; and Section 3 compares it with the results of the HEP-SPEC06 benchmark [8]. Section 3 examines the performance of the HLT when the "heterogeneous" algorithms described in [3], [4] and [5] are offloaded to a GPU: Section 3.1 shows how the performance improvements depend on the combination of host CPUs and GPUs; while Section 3.2 tries to make a more objective

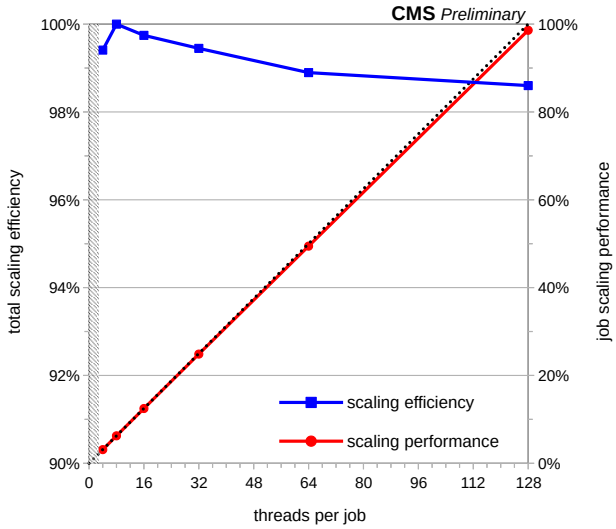

Figure 1. Scaling of the HLT throughput with the number of threads per job. The measurements were performed on a dual AMD EPYC "Milan" 7763 machine with a total of 256 hardware threads, varying the number of threads per job and adapting the number of concurrent jobs to fully utilise the machine, as described in Section 2.1. The blue line (on the primary y axis) shows the total throughput relative to the maximum value, obtained with 32 concurrent jobs each with 8 threads. The red line (on the secondary y axis) shows the performance normalised to the number of threads per job, while the dashed line indicates the ideal scaling.

comparison of the performance of different generation of GPUs. Section 4 gives an overview of the ongoing developments and future work directions, and concludes the paper with a look at the implications of this work on the HLT for the upcoming Run-3 and the future Phase-2 upgrades. All measurements are based on a snapshot of the HLT menu under development for Run-3, running over data from 2018 with a pileup of 50 interactions per event.

2. Performance on CPUs

Modern server CPUs have modular, multi-core architecture, and achieve their full potential using Simultaneous Multi Threading (SMT) technologies to run two or more hardware threads on each core. When they are combined in a dual-socket platform, the result is a machine with a large number of hardware threads: for example, the HLT machines deployed in 2018 have two Intel Xeon Gold "Skylake" 6130 processors, resulting in $2 \times 16 \times 2 = 64$ threads, while the machines deployed in 2022 have two AMD EPYC "Milan" 7763 processors, with a total of $2 \times 64 \times 2 = 256$ threads. The use of multithreaded software is essential to exploit the full processing power of these platforms while maintaining a relatively low memory footprint. In addition, using a smaller number of processes with a larger thread count can improve the utilisation of GPUs, reducing the contention from the processes that share them, and the total amount of GPU memory needed.

2.1. Multi-threaded performance

Support for multithreading was introduced in CMSSW [9] during the Long Shutdown 1 period (2013-2014), based on the Intel Threading Building Blocks library [10], and continuously improved over the years. Figure 1 shows the scaling of the performance of CMSSW running the HLT application as a function of the number of threads per job. The measurements were performed on a dual AMD EPYC "Milan" 7763 machine with a total of 256 hardware threads,

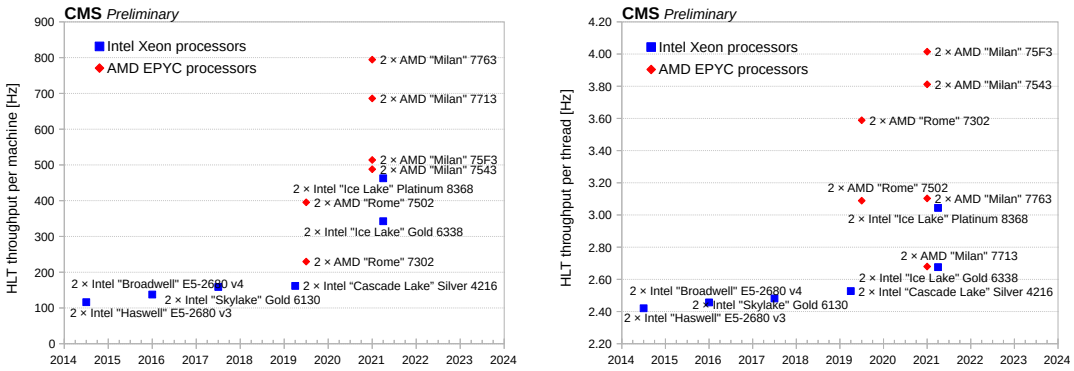

Figure 2. Evolution over time of the performance of x86 CPUs from different generations and vendors, measured by the throughput achieved by the HLT on a full dual-socket machine (left) and scaled by the number of hardware threads (right).

running as many concurrent jobs as needed to utilise all of them: 64 jobs with 4 threads, 32 jobs with 8 threads, *etc.*, up to two jobs with 128 threads; each job was pinned to a specific set of cores from either one of the two CPUs; it was not possible to run 128 jobs with 2 threads or 256 jobs with a single thread due to the amount of memory available on the machine, as indicated by the shaded area; a single job with 256 threads was not considered, to avoid incurring in NUMA effects. The highest performance is obtained by 32 concurrent jobs with 8 threads each, while all other configurations achieve a relative efficiency higher than 98%.

2.2. Performance across CPU generations

The machines used at HLT during Run-2 (2015-2018) are based on the latest (at the time) generations of Intel Xeon processors, and achieve high density with a compact form factor fitting four dual-socket motherboards in a "2U" enclosure. During the Long Shutdown 2 period (2019-2021), after evaluating different solutions based on server CPUs by Intel and AMD, the adoption of the AMD EPYC line of processors led to a different choice for Run-3, with a dual-socket motherboard in a "1U" enclosure. Figure 2 shows the evolution of the performance of these CPUs between 2014 and 2021, using the HLT software as a benchmark. A fixed version of the software has been run on each machine, and the optimal job configuration was chosen after performing a scan similar to the one shown in Figure 1. The right plot shows how the performance per core (or per thread) has been stable across many generations of Intel Xeon CPUs, and received a dramatic increase with the AMD EPYC CPUs. The left plot shows that the overall performance for the whole machines largely followed the small increase in core count for the Intel Xeon CPUs, while it jumped due to the high performance per core and the large core count in the AMD EPYC CPUs.

2.3. Comparison with the HEP-SPEC06 benchmark

While the use of the HLT application itself is the ideal choice to evaluate the performance of different machines, it is not always a practical on. On the other hand, a standardised benchmark is useful only if it reflects the performance of the actual workloads. Over the past 10 years the high energy physics community has used the HEP-SPEC06 benchmark [8] to evaluate the performance of multi-core computing systems. It is thus interesting to see how, despite its age, it is still a useful metric for the evaluation of modern systems. Figure 3 shows the correlation between the HLT performance, measured as described in Section 2.2, and the HEP-SPEC06 score measured [11] on the same machines.

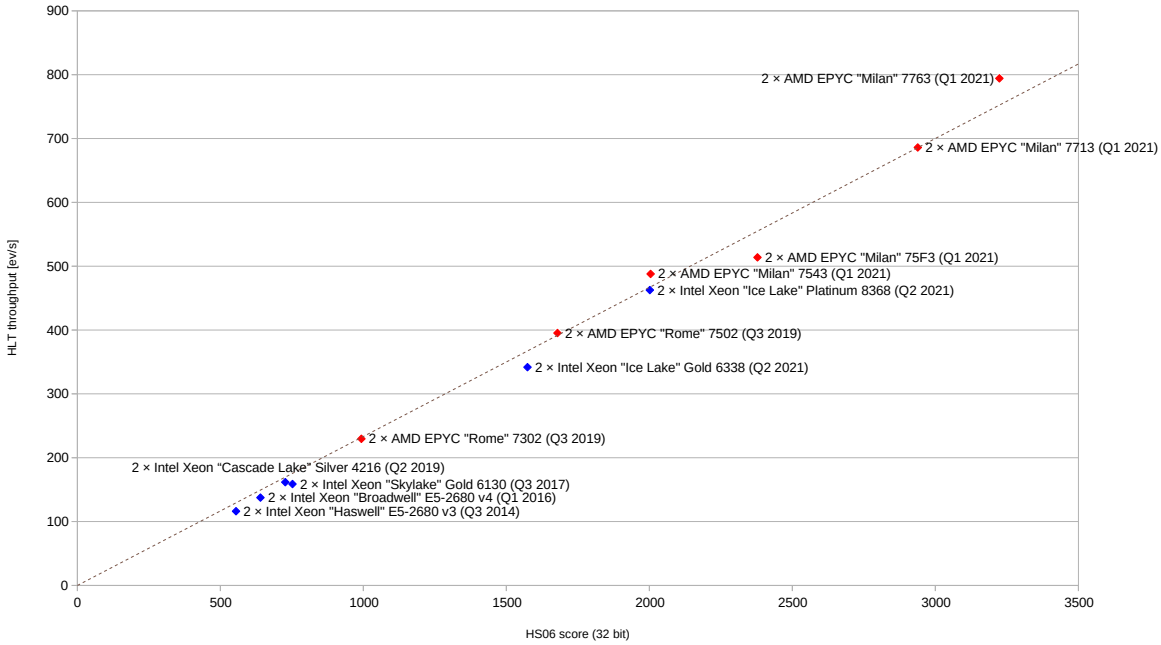

Figure 3. Comparison of the event processing throughput for a reference HLT configuration and the results of the HEP-SPEC06 (HS06) benchmark, measured on nodes equipped with CPUs from different generations (from 2014 to 2021) and different vendors (Intel and AMD).

3. Performance on GPUs

The introduction of computing accelerators, like GPUs, makes measuring the performance of a hardware system or a software application significantly more complex, as the results may depend on multiple factors. For example:

- the performance of the CPU and of the GPU (or other accelerator);
- the application's efficiency on the CPU and GPU (or other accelerator);
- the "heterogeneous" fraction of the application that can be offloaded.

A recent snapshot of the HLT under development for Run-3 has been used to measure the performance of different combinations of CPUs and GPUs, evaluate the improvement obtained offloading the "heterogeneous" algorithms to GPUs, and estimate the relative performance of different GPUs from multiple generations.

3.1. Performance of the full HLT using GPUs

The overall performance of the full HLT application was measured on various combinations of CPUs and GPUs. These measurements highlight how strongly the results depend on the relative performance of the CPUs and GPUs, and suggest that the best results can be obtained when the system is designed around a specific application or use case.

The host systems are dual socket machines equipped with Intel Xeon Gold "Skylake" 6130 CPUs, or AMD EPYC "Milan" 7543, 7713 and 7763 CPUs; the GPUs used in these measurements are one or two NVIDIA Tesla T4 GPUs, or (only in the dual AMD 7763 case) one NVIDIA A10 GPU. The optimal job configuration for each system was chosen performing a scan similar to the one shown in Figure 1; each job was pinned to a specific set of cores from either one of the two CPUs; the GPUs were efficiently shared among the multiple jobs using the NVIDIA Multi-Process Service [12].

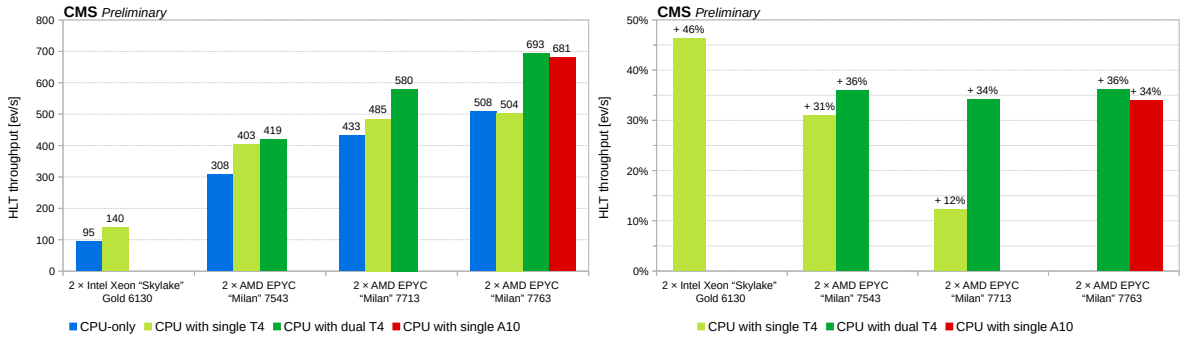

Figure 4. Absolute (left) and relative (right) performance of the HLT on different machines, running only on the CPU (blue bars) and offloading to different GPUs: a single NVIDIA T4 (light green), a pair of NVIDIA T4 (dark green), and a single NVIDIA A10 (red).

Figure 4 shows the performance obtained running the HLT on different combinations of CPUs and GPUs. Adding a second T4 to the dual 7543 gives only a small improvement, suggesting that these CPUs are evenly matched to a single T4. As the performance of the CPUs increases this GPU becomes the limiting factor in the system performance, requiring a second GPU or a single more powerful one. When enough processing power is available from the GPUs, the performance improvement is stable around 35% for all AMD CPUs: the application is limited by the performance of the CPUs and the fraction of the algorithms that can be offloaded.

3.2. Performance of the offloadable algorithms

The HLT configuration can be modified to run only the "heterogeneous" algorithms that can be offloaded to GPUs, and avoid writing any output to disk. These changes ensure that the CPUs or GPUs running the algorithms are not limited by external factors, leading to a more objective comparison of their performance.

Figure 5 shows the relative performance of two CPU-only systems – a dual processor machine with two Intel Xeon "Skylake" 6130 CPUs (2 × 125 W, from 2017), and one with two AMD EPYC "Rome" 7502 CPUs (2 × 180 W, from 2019) – and of six NVIDIA datacentre GPUs from four different generations: a Tesla K40 (250 W, from 2013), a Tesla P100 (250 W, from 2017), a Tesla V100 (250 W, from 2018) and a Tesla T4 (70 W, from 2019), an A100 SXM4 (400 W, from 2020) and an A10 (150 W, from 2021). The CPU-only measurements were performed as described in the previous sections. Most of the GPU measurements have been performed with a single job with 8 and 16 threads, using in each case the configuration that gives the best performance; only in the case of the V100 multiple concurrent jobs and the use of the MPS server were necessary to obtain the best throughput. The comparison of the throughput achieved by the Tesla K40, Tesla P100, Tesla V100 and A100 shows the improvements in performance over different GPU generations. The Tesla T4 is low-power GPU from the same generation as the Tesla V100: their comparison shows how GPUs can be optimised for higher performance or for better power efficiency. Finally, the A10 shows an other interesting working point, with a different trade-off between performance and power consumption.

4. Future work and conclusions

Over the past five years CMS has brought the use of GPUs for physics reconstruction from the R&D phase to the production stage, with the deployment of a fully GPU-equipped HLT farm and the use of GPUs at grid sites. The use of GPUs is expected to grow during and after Run-3, as CMS aims to leverage GPUs for at least 50% of the HLT capacity during Run-4 and 80% during Run-5, in order to reduce the costs and improve the efficiency of the HLT farm during the

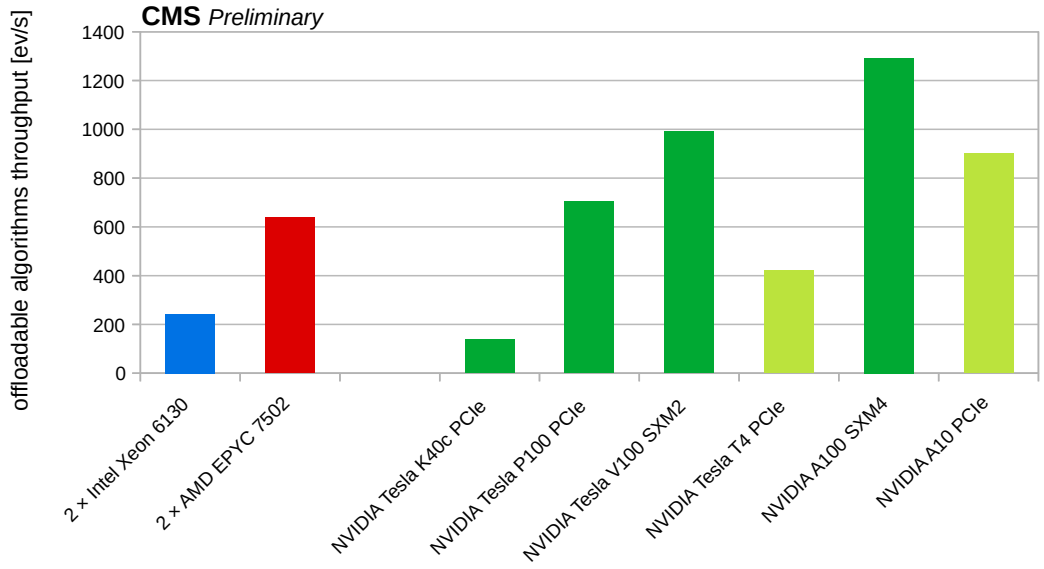

Figure 5. Performance or the offloadable part of the Run-3 HLT reconstruction running on different hardware: two dual-processors machines, and four generations of NVIDIA GPUs.

High-Luminosity LHC [1]. At the same time, multiple R&D activities are starting or continuing: the investigation of various *performance portability* solutions like alpaka [6] (chosen for Run-3), Kokkos [7] or SYCL; the development of new "heterogenous" algorithms both for Run-3 and for Phase-2, like the Particle Flow and TICL [13]; and use of GPUs over high-speed network fabric.

References

[1] CMS Collaboration, "The Phase-2 upgrade of the CMS Data Acquisition and High Level Trigger", Technical Report CERN-LHCC-2021-007, CMS-TDR-022, CERN, Geneva, 2021.
[2] A. Bocci et al., "Bringing heterogeneity to the CMS software framework", *EPJ Web Conf.* **245** (2020) 05009, doi:10.1051/epjconf/202024505009.
[3] A. Bocci et al., "Heterogeneous reconstruction of tracks and primary vertices with the CMS pixel tracker", *Frontiers in Big Data* **3** (2020), no. 10, 49, doi:10.3389/fdata.2020.601728.
[4] T. Reis, "Developing GPU-compliant algorithms for CMS ECAL local reconstruction during LHC Run 3 and Phase 2", *J. Phys.: Conf. Ser.* (2022). Submitted.
[5] V. Khristenko and M. Girone, "High Energy Physics with CMSSW", volume 48 of *Schriften des Forschungszentrums Jülich IAS Series*, pp. 167–185. Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag Jülich, Jülich, 2021.
[6] W. Redjeb et al., "Performance portability for the CMS reconstruction with alpaka", *J. Phys.: Conf. Ser.* (2022). Submitted.
[7] M. Kortelainen et al., "Porting CMS heterogeneous pixel reconstruction to Kokkos", *EPJ Web Conf.* **251** (2021) 03034, doi:10.1051/epjconf/202125103034.
[8] M. Michelotto et al., "A comparison of HEP code with SPEC benchmarks on multi-core worker nodes", *Journal of Physics: Conference Series* **219** (apr, 2010) 052009, doi:10.1088/1742-6596/219/5/052009.
[9] C. Jones et al., "Using the CMS threaded framework in a production environment", *Journal of Physics: Conference Series* **664** (dec, 2015) 072026, doi:10.1088/1742-6596/664/7/072026.
[10] J. Reinders, "Intel Threading Building Blocks: outfitting C++ for multi-core processor parallelism". O'Reilly Media, Inc., 2007.
[11] D. Giordano et al., "HEPiX benchmarking solution for WLCG computing resources", *Computing and Software for Big Science* **5** (Dec, 2021) 28, doi:10.1007/s41781-021-00074-y.
[12] NVIDIA Corp., "NVIDIA Multi-Process Service", October, 2021. https://docs.nvidia.com/deploy/mps/index.html.
[13] B. Alves, "Clustering in the heterogeneous reconstruction chain of the CMS HGCAL detector", *J. Phys.: Conf. Ser.* (2022). Submitted.