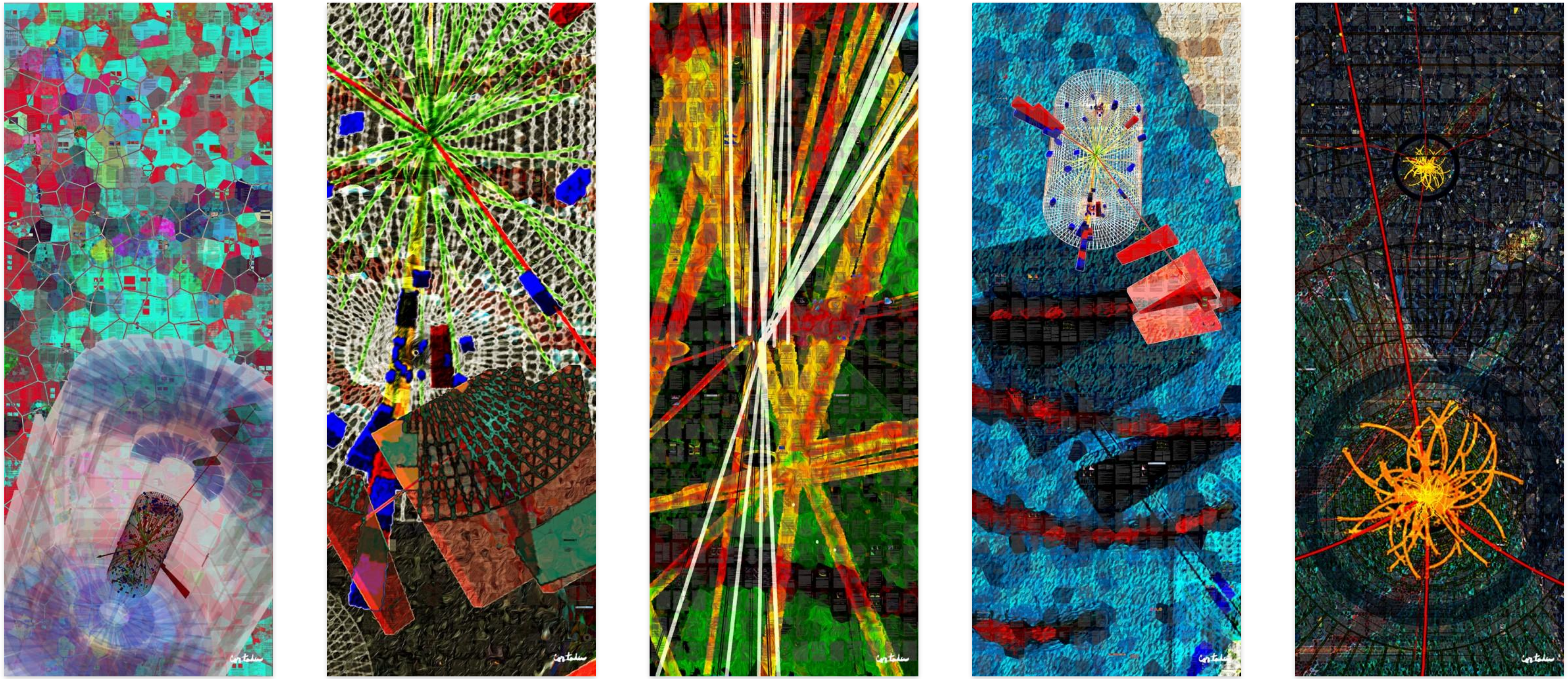
U.S. CMS: Feedback and priorities for year 2

Oliver Gutsche for the U.S. CMS S&C Operations Program

HEP-CCE All-Hands Meeting
06. November 2020



U.S. CMS S&C operations program: HL-LHC strategic plan

- Defines 4 Grand Challenges for HL-LHC Software & Computing
  - Grand Challenge (1): Modernizing Physics Software and Improving Algorithms
  - Grand Challenge (2): Advancing High-throughput Distributed Data Processing
  - Grand Challenge (3): Building Exabyte-size Active Data Storage
  - Grand Challenge (4): Transforming the Scientific Data Analysis Process

HEP-CCE

- Defines work packages for each challenge, these are the challenges for Grand Challenge (1): Modernizing Physics Software and Improving Algorithms
  - Main work package tracking the CCE PPS and IOS activities
    - WP 1.1 Core Software Framework and Software Portability
  - Following work packages significantly influenced and depending on CCE activities
    - WP 1.2 Establish Performance Metric and Performance Baseline for Physics Software
    - WP 1.3 U.S. Contributions to the Charged Particle Tracking Software
    - WP 1.4 U.S. Contributions to Software for the High Granularity Endcap Calorimeter
    - WP 1.5 U.S. Contributions to CMS Advanced Algorithms Work

PPS

- PPS in the context of U.S. CMS
  - Framework (CMSSW) is able to efficiently schedule work concurrently on CPUs and accelerators
  - CMS is planning to utilize accelerators (NVidia GPUs) in the High Level Trigger (HLT) in Run 3
    - Portability under investigation in CMS (European groups currently looking into Alpaka and SYCL (very early studies)) and CCE-PPS
  - Timeline:
    - Run 3:
      - CMS HLT would like to achieve single source portability at least between CUDA and CPU
      - CMS Offline & Computing coordination agreed to contribute to look for a solution, CCE studies will be useful already here
    - Run 4:
      - CMS Offline & Computing coordination will revisit portability solutions to achieve maximum portability (“as portable as feasible”) in time for HL-LHC code development
      - CCE-PPS milestone of 2023Q1 fits well in CMS plans, CCE recommendation integral part of process
- PPS project feedback
  - Overall happy with plans and progress
  - Patatrack: complicated enough code base that we learn most of what we need to make a recommendation
  - Would like to add a math-heavy small use case to investigate performance scaling
    - mkFit has several smaller parts (for example propagation-to-r) that seem suitable
  - Question that came up during year 1 activities: Explore performance impact of CUDA Unified Memory to understand the necessity of managing the host-device memory transfers explicitly

IOS

- CMS is very interested in the IOS project
  - Single node behavior
    - High thread-count jobs are significantly limited by I/O synchronization bottlenecks
  - Multi-node behavior
    - Concern about potential to overload HPC shared filesystem
  - Workflow behavior
    - Multi-step workflows use (local or shared) storage for intermediate output ➜ Can we optimize for different storage backends at HPC sites compared to traditional Grid sites

- IOS project feedback
  - Overall happy with plans and progress
  - Continue/expand the scaling investigations that Chris is performing
    - I/O test framework integral part of these studies
  - Workflow behavior studies have not been discussed and plans have not been made, would like to start talking about this point
  - Need help running scaling tests

Generators and Workflow

- Generators project
  - Limited use
  - We more need HEP generators to be rewritten from the ground up
- Workflows
  - Processing different parts of our workflow on different installations or in different HPC passes is not very promising for CMS
  - HL-LHC:
    - We expect reconstruction to be the major part of the CPU needs
    - Offloading parts of the reconstruction would mean storing and then transferring part of the reconstruction output (multiple tens of MB per event) plus the RAW data (~7 MB per event) to somewhere else
    - Too much data movement would make this not feasible

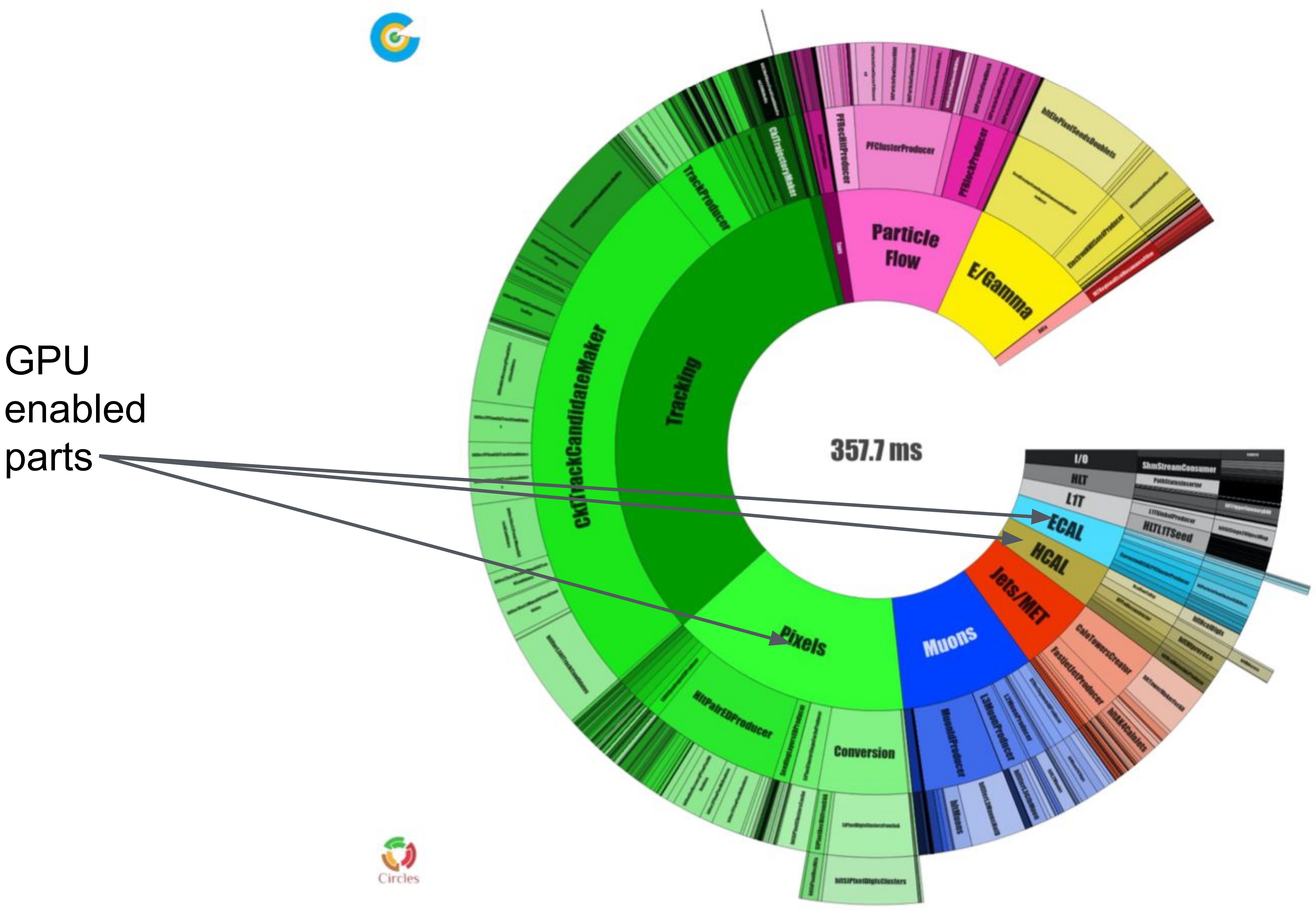
Efficient use of accelerators

- Additional investigations needed for optimal accelerator usage
  - Different parts of the investigation
    - Data formats: How can we optimize the data layout on storage and in memory to make it more efficient for processing on accelerators (SoA anyone?)
    - Event batching: Algorithms executed on single events mostly result in low GPU utilization ➜ want to look into event batching to increase GPU utilization
    - Direct vs. remote accelerator access: optimize data formats for case that the accelerator is not accessed directly from an application (SONIC)

Summary

- Overall happy with plans and progress
- Would like to add math-intensive use case to PPS
- Would like to propose: Efficient use of accelerators

Backup: CMS HLT application using CPU and GPU



GPU enabled parts

The timing is measured on pileup 50 events from Run2018D on a full HLT node (2x Intel Skylake Gold 6130) with HT enabled, running 16 jobs in parallel, with 4 threads each - equipped with an NVIDIA T4 GPU.

Current status of heterogeneous HLT:

- CPU usage reduced by 21%
- Throughput increased by 26%

Includes:

- Memory copies
- Conversion of copied results to legacy formats

From: CERN-EP Software Seminar: Real-time heterogeneous event reconstruction with GPUs at CMS and LHCb during LHC Run-3: https://indico.cern.ch/event/927838/