

# Performance portability for the CMS Reconstruction with Alpaka



Wahid Redjeb, RWTH Aachen University  
on behalf of the CMS collaboration

## Introduction

For CMS, **Heterogeneous Computing** is a powerful tool to face the computational challenges posed by the upgrades of the LHC and will be used in production at the HLT during Run 3. In principle, in order to offload the computational work on non-CPU resources, different implementations of the same code are required. This would introduce code-duplication which is not sustainable in terms of maintainability and testability of the software. **Performance Portability Libraries** allow to write code once and run it on different architectures with close-to-native performance. The CMS experiment is evaluating performance portability libraries for the near term future.

## Heterogeneous Computing in CMS



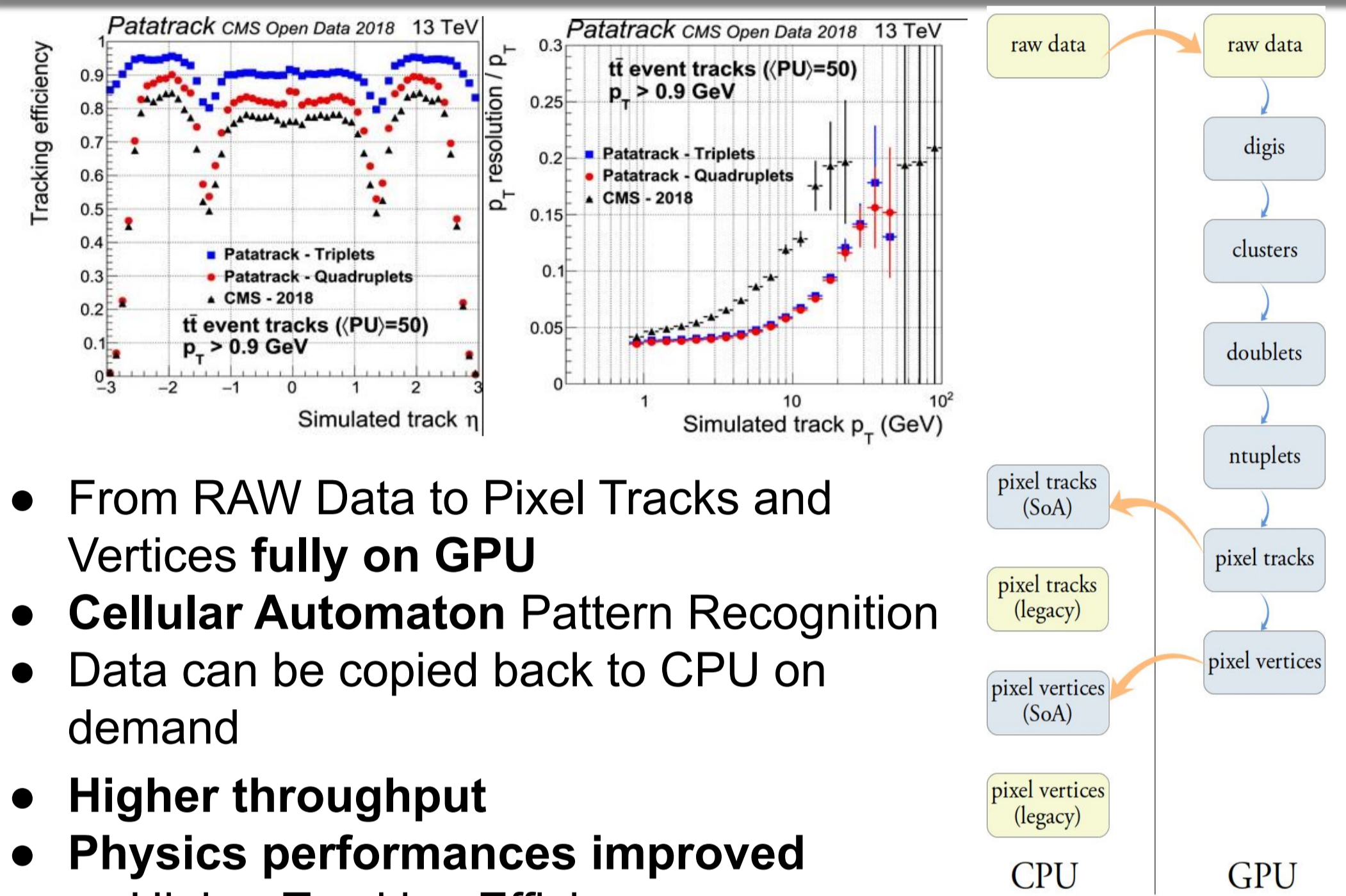
For new the Data Acquisition period (Run-3), CMS will run different modules for the online reconstruction at the HLT on GPUs:

- **Pixel tracks and Vertices Reconstruction**
- HCAL, ECAL local reconstruction
  - HGCAL (Phase-2)
- Particle Flow Clustering

By adopting **Heterogeneous Computing**, CMS demonstrated:

- **Better computing performance**
- **Better physics performance** (reinventing algorithms)
- Adapt to heterogeneous architectures at present and **future HPC and WLCG centres**

## Patatrack Pixel Tracks And Vertices Reconstruction



- From RAW Data to Pixel Tracks and Vertices **fully on GPU**
- **Cellular Automaton Pattern Recognition**
- Data can be copied back to CPU on demand
- **Higher throughput**
- **Physics performances improved**
  - Higher Tracking Efficiency
  - Better  $p_T$  resolution
  - Reduction of Duplicate and Fake rate

## Alpaka in CMS

Alpaka is a header only, open source, C++ library.

Alpaka implements an **abstraction layer**:

- “Hides” the backend implementation
  - Allowing the user to express parallelism on different architectures
- Implement a **Hierarchical Redundant Parallelism Model** (similar to CUDA)
  - Grid, Block, Threads, Elements
- Data-agnostic memory model

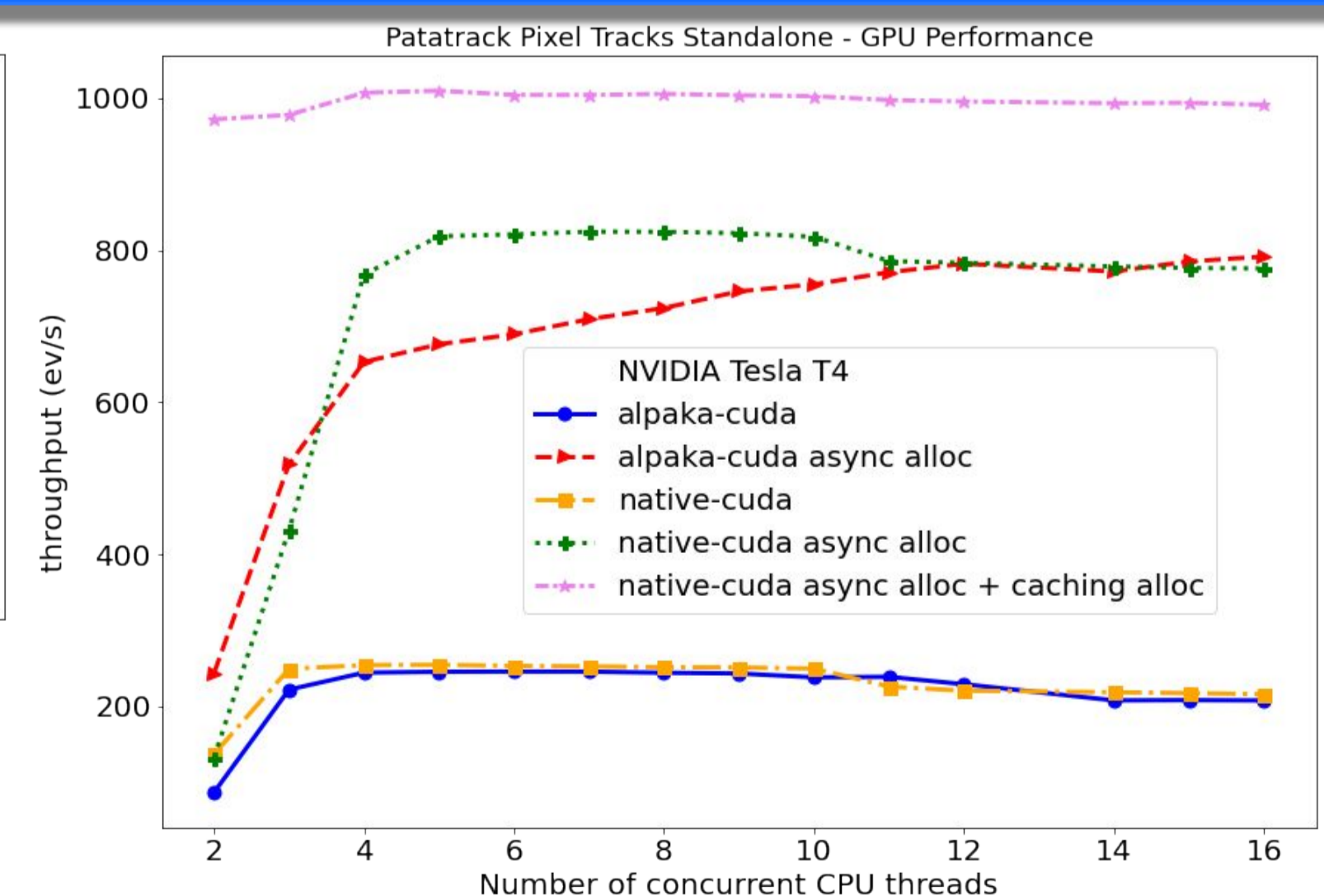
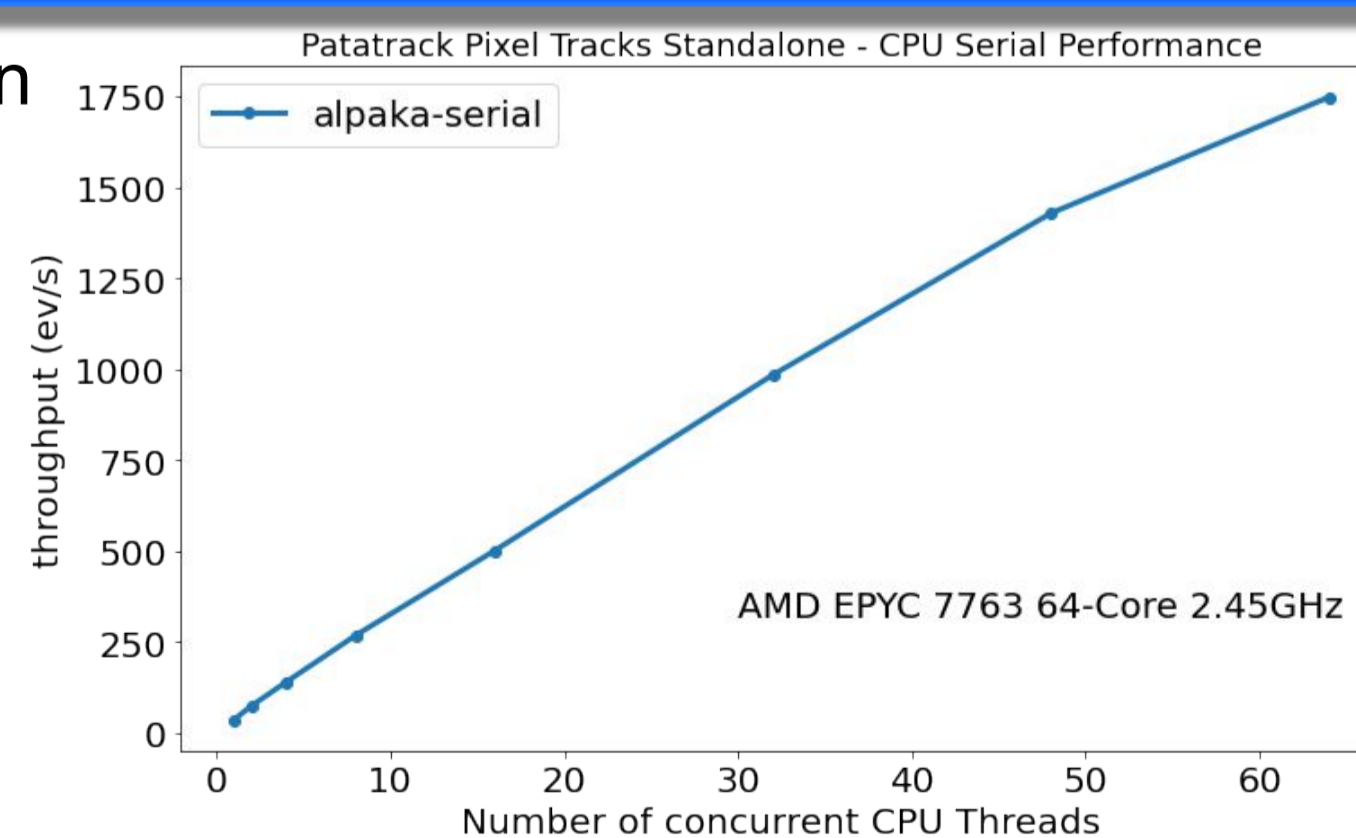


Alpaka is being tested in CMS porting the current Patatrack Pixel Tracks and Vertices Reconstruction. The goal is to obtain a **single source code** that can run on multiple architectures and backend, with computing performance close to the native ones.

## Results and Future Perspectives

Obtained a **single source code** that can be compiled on multiple devices and backends

- Alpaka CPU version scales well with the number of threads
- Alpaka GPU performance are very close to the native ones
- Missing Caching Allocator in the Alpaka version
- Stream Ordered memory allocations
  - In progress



**CMS chose Alpaka as Portability Layer for the deployment at the High Level Trigger in Run-3**

- CMS aims to replace CUDA with Alpaka for Run III data taking
  - Porting all the GPU reconstruction modules to Alpaka
- Interface Alpaka with the CMS Software
- Improve performance with Alpaka-base reconstruction

### Links and References

- [Pixel Tracks Standalone - Alpaka Demonstrator](#)
- [Alpaka at ACAT 2019](#)