

# Evaluating FPGA Acceleration with Intel® oneAPI Toolkit for High-Speed Data Processing

Alberto Perro<sup>1,2</sup>, Paolo Durante<sup>1</sup>, Flavio Pisani<sup>1</sup>, Eleni Xochelli<sup>1,3</sup> <sup>1</sup> CERN, <sup>2</sup> Aix-Marseille Université, <sup>3</sup> University of Thessaly *Corresponding Author: alberto.perro@cern.ch*

## Context

The **LHCb Experiment** uses GPU cards in its **high level trigger** system to efficiently handle a data rate of **32 Tb/s** from the detector.<sup>1,2</sup>

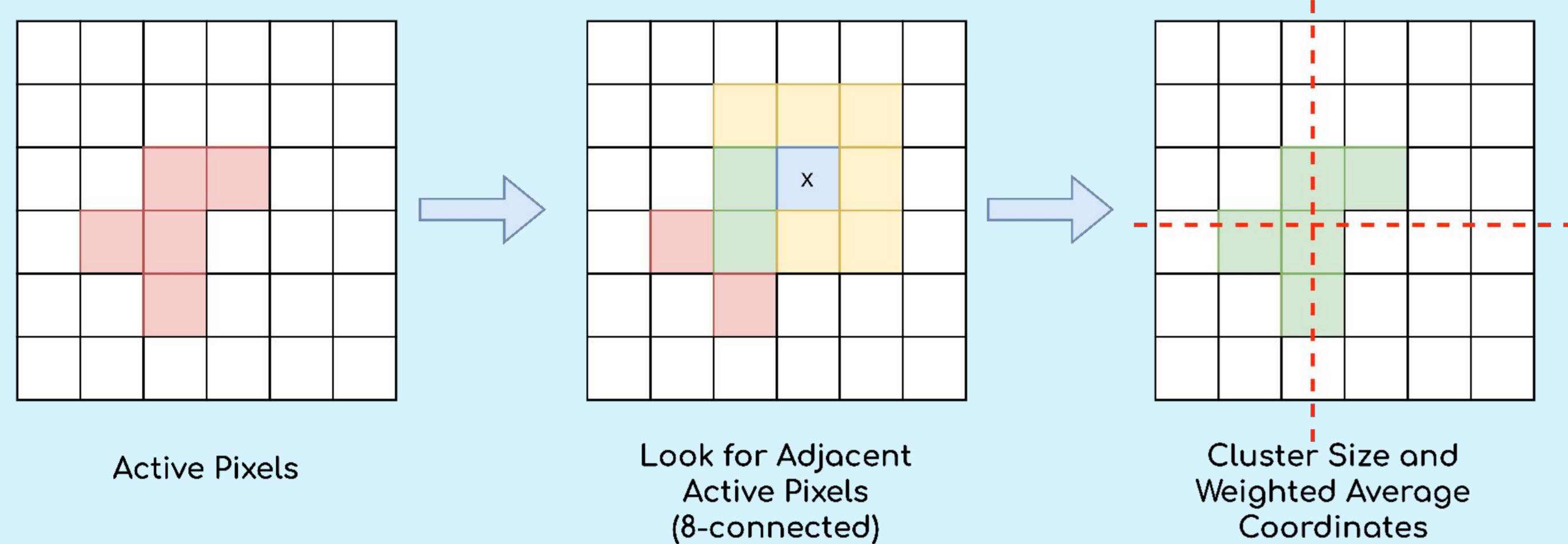
Some trigger tasks, such as decoding, are better suited for **FPGAs** due to their ability to handle bitwise and combinatorial operations.

However, FPGAs are more difficult to program using standard Hardware Description Languages (HDLs) compared to CPUs and GPUs.

Intel OneAPI FPGA Toolkit<sup>3</sup> offers an **GPU-like programming framework** to develop FPGA-accelerated workloads:

- Data Parallel **C++** language with SYCL<sup>3</sup> cross-platform abstraction layer
- **Emulator** compatible with software debugging tools (e.g. GDB)
- High level FPGA integration through a **multi-architecture binary**

## Algorithm



The current **pixel clustering** algorithm<sup>4</sup> of the VeLo detector was chosen for the evaluation of the toolkit.

Active pixels coming from the frontend are grouped in 208 banks corresponding to different independent parts of the detector.

The clustering algorithm starts from an **active pixel** (candidate) and **looks for adjacent active pixels**, updating cluster size and weighted averaged coordinates when it finds one.

When there are no remaining adjacent active pixels, the **definitive cluster** information is returned.

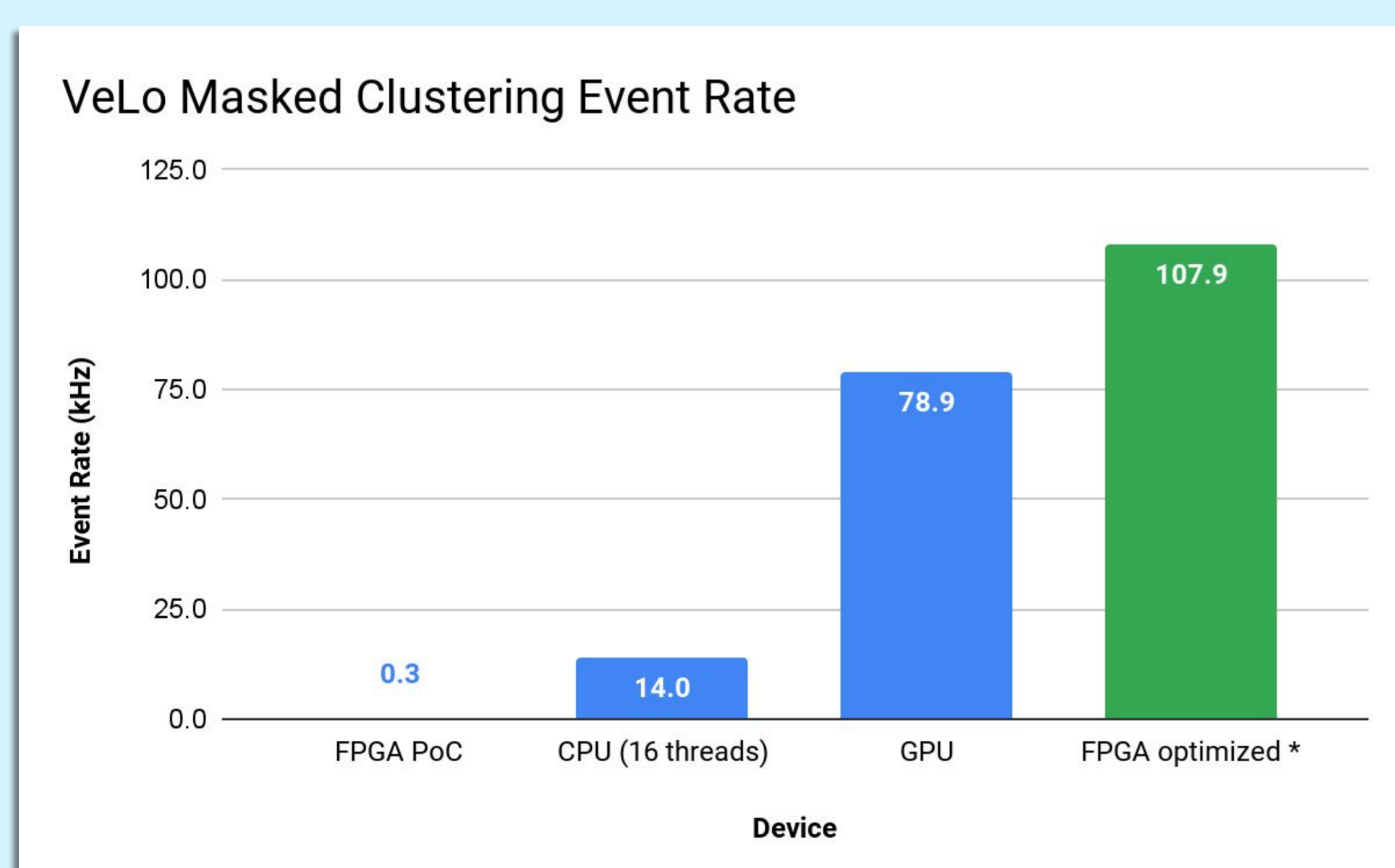
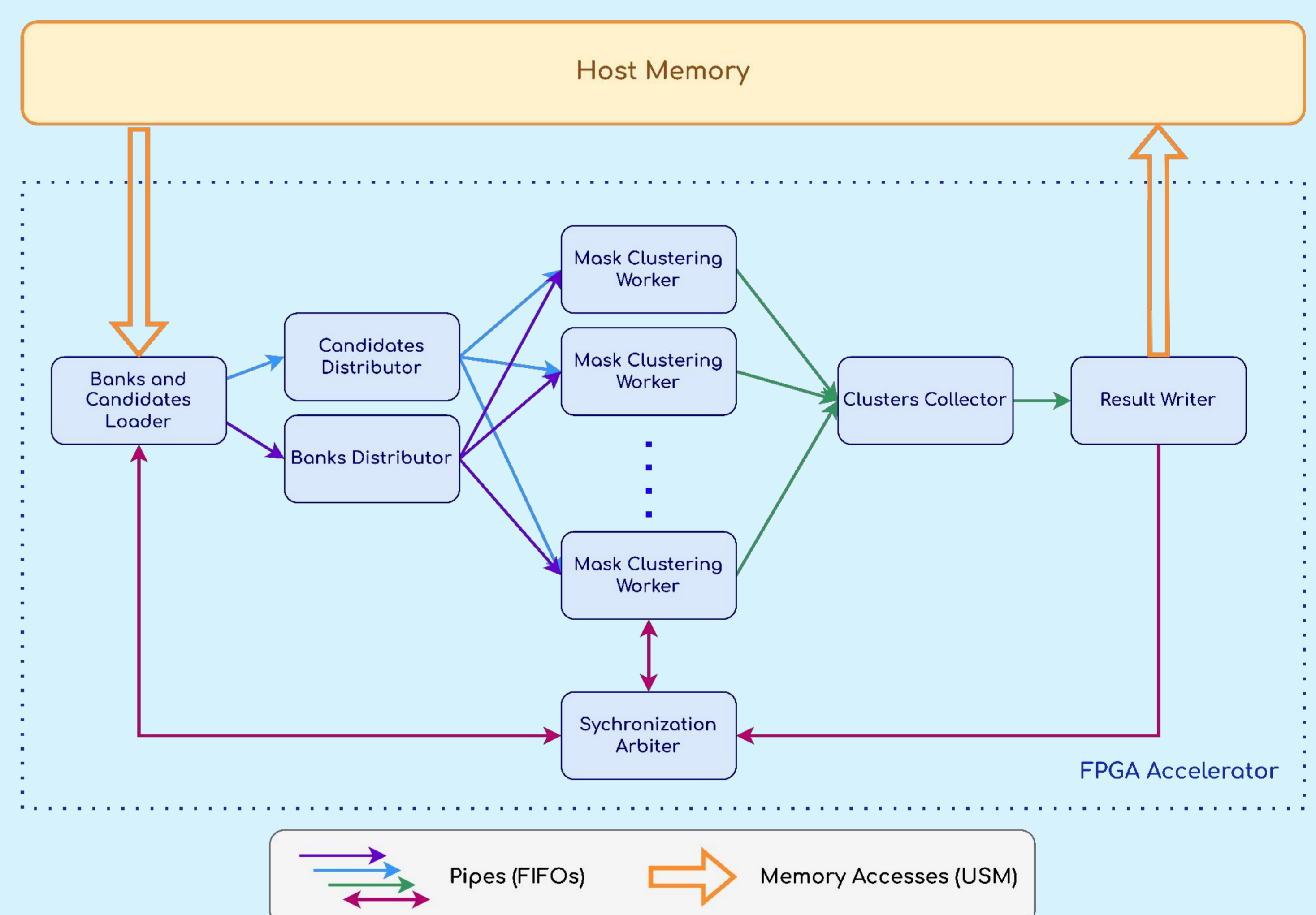
## Architecture

The GPU approach uses a block and thread approach, spawning multiple workers for sets of candidates, banks, and events.

The FPGA approach is a **pipelined distributed design**:

- A single kernel accesses the host memory buffer **contiguously** and decodes banks and candidates.
- Two kernels, one for banks and one for candidates, **distribute** the results of the first kernel to the workers.
- Each **worker** applies the clustering algorithm to a specific bank, since they are independent.
- Clusters are then **collected** by another kernel that forwards them to the writer kernel, which writes the results back to the host memory.

A **synchronization** kernel is in charge of synchronizing the whole chain, keeping track of event boundaries to **avoid corruption**.



\* the FPGA optimized benchmark does not include the writing back to host memory due to a stability issue with the Board Support Package

## Results

Porting the algorithm to the FPGA didn't require major modifications. The first proof of concept was running on the FPGA in **less than a month**.

However, the FPGA design required **rethinking the architecture** of the kernel with respect to the GPU one to fully utilize the computing power available.

Host memory accesses required **specific design choices** both in the kernel and in the host software to reach the advertised PCIe throughput (single pointer access, contiguous data).

Since this algorithm is **compute intensive**, the multiple workers approach was the **most performant**.

Benchmarks were run on a Bittware IA-840F FPGA Accelerator Card, an NVidia RTX A5000 GPU, and an AMD Epyc 7502 CPU.

## References and Acknowledgements

1. LHCb Collaboration, "LHCb Trigger and Online Upgrade Technical Design Report." 2014. <https://cds.cern.ch/record/1701361>.
2. Aaij, R. et al. 2020. "Allen: A High-Level Trigger on GPUs for LHCb." Computing and Software for Big Science 4 (1): 7. <https://doi.org/10.1007/s41781-020-00039-7>.
3. "oneAPI: A New Era of Heterogeneous Computing." Intel. Accessed September 16, 2024. <https://www.intel.com/content/www/us/en/developer/tools/oneapi/overview.html>.
4. Campora Perez, Daniel Hugo. 2019. "Optimization of High-Throughput Real-Time Processes in Physics Reconstruction." Seville U. <https://cds.cern.ch/record/2718278>.

Special thanks to Christian Färber from Altera for the support.

