Looking into the trigger future

Maastricht University

Department of Advanced Computing Sciences

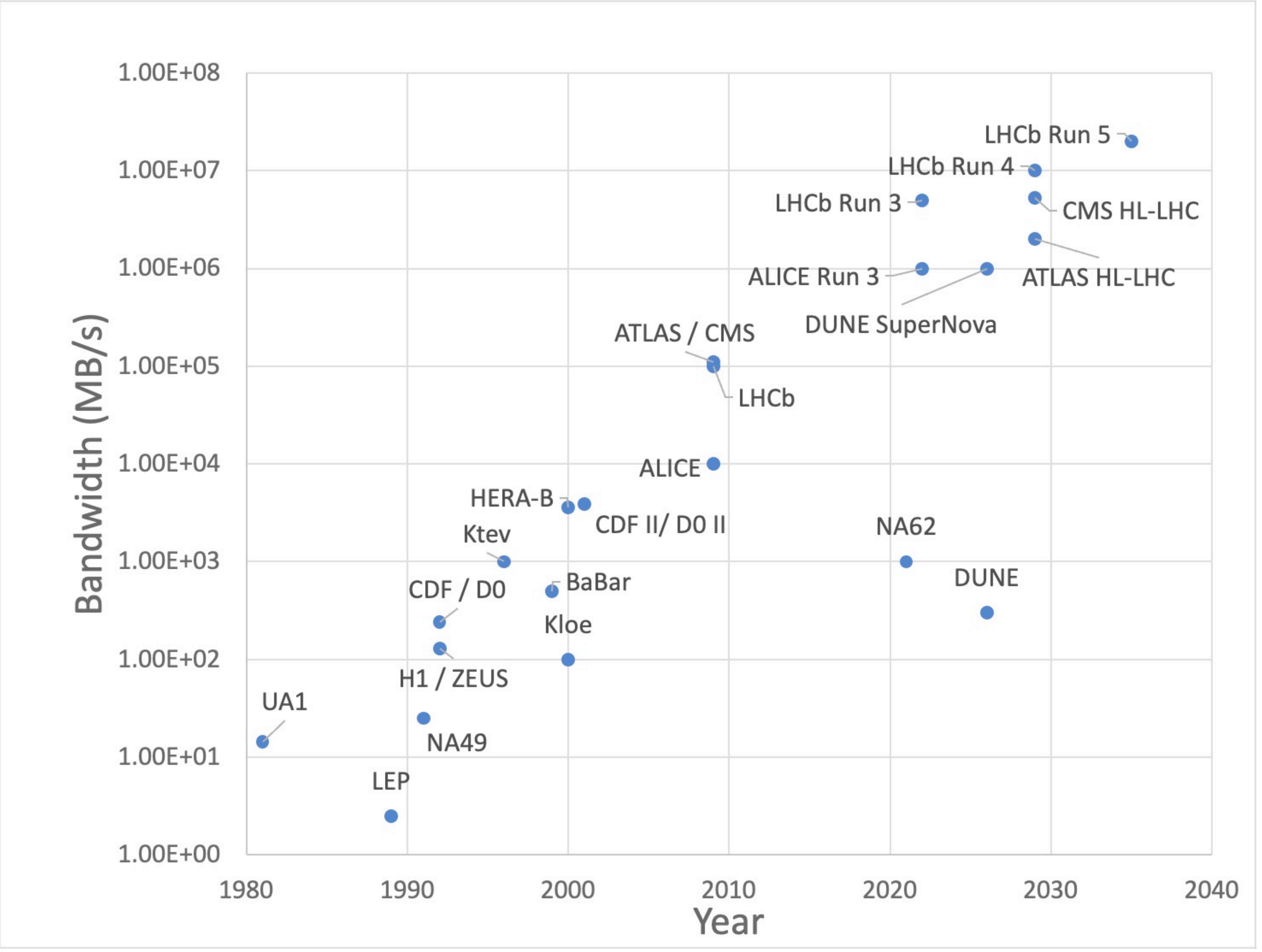
Daniel Cámpora

The HLT, an ever-growing problem

- In all incarnations of LHCb, we have pushed the tech available to get the most out of the detector
- Decisions not always coming from hardware, software demonstrators being particularly relevant
- This has prompted collaboration between hardware and software teams, although not always consistently

The problem ahead of us

- More challenging than ever
- LHCb Run 5 will require processing 5x more data than today
    - The biggest real-time data processing challenge in HEP
- A multi-dimensional challenge
    - Software, hardware, networking
- Hardware and software should encompass the throughput requirement



A. Cerri, University of Sussex

Daniel Cámpora

A software's perspective

- Complexity bounds of the problems show a nontrivial road ahead of us

	Theoretical problem	Simplification
Data sorting	$O(n^2)$	-
Track seeding	$O(2^n)$	$O(n \cdot log(n)^2)$
Track following	$O(2^n)$	$O(n \cdot log(n))$
RICH likelihood minimisation	$O(2^n)$	$O(n^6)$
CALO energy sharing	$O(n^2)$	$O(\|V\| + \|E\|)$
Selections	$O(2^n)$	$O(n^2)$

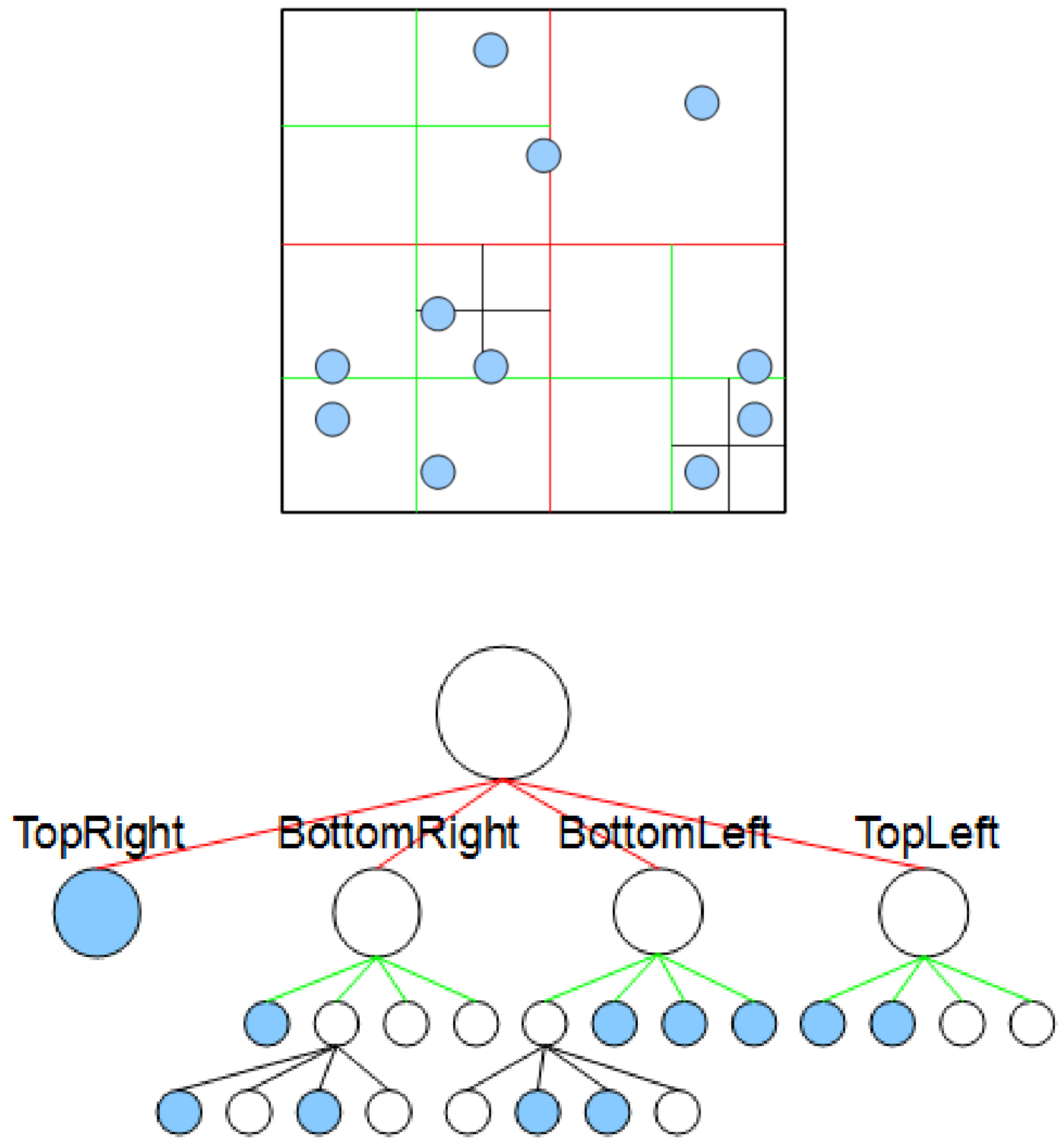
- In spite of clever simplifications, we will have to develop smart data traversing algorithms to account for the increased throughput in the following Runs

A time for collaboration

- Collaborations between teams doing software and hardware will impact significantly the trigger of tomorrow
- It is an opportunity to deliver something truly excellent and broaden the solutions we come up with
- Collaboration between teams requires dedication
    - One of the successes of Allen was productive collaboration between Online, reconstruction experts, trigger and software developers

Advances in FPGAs

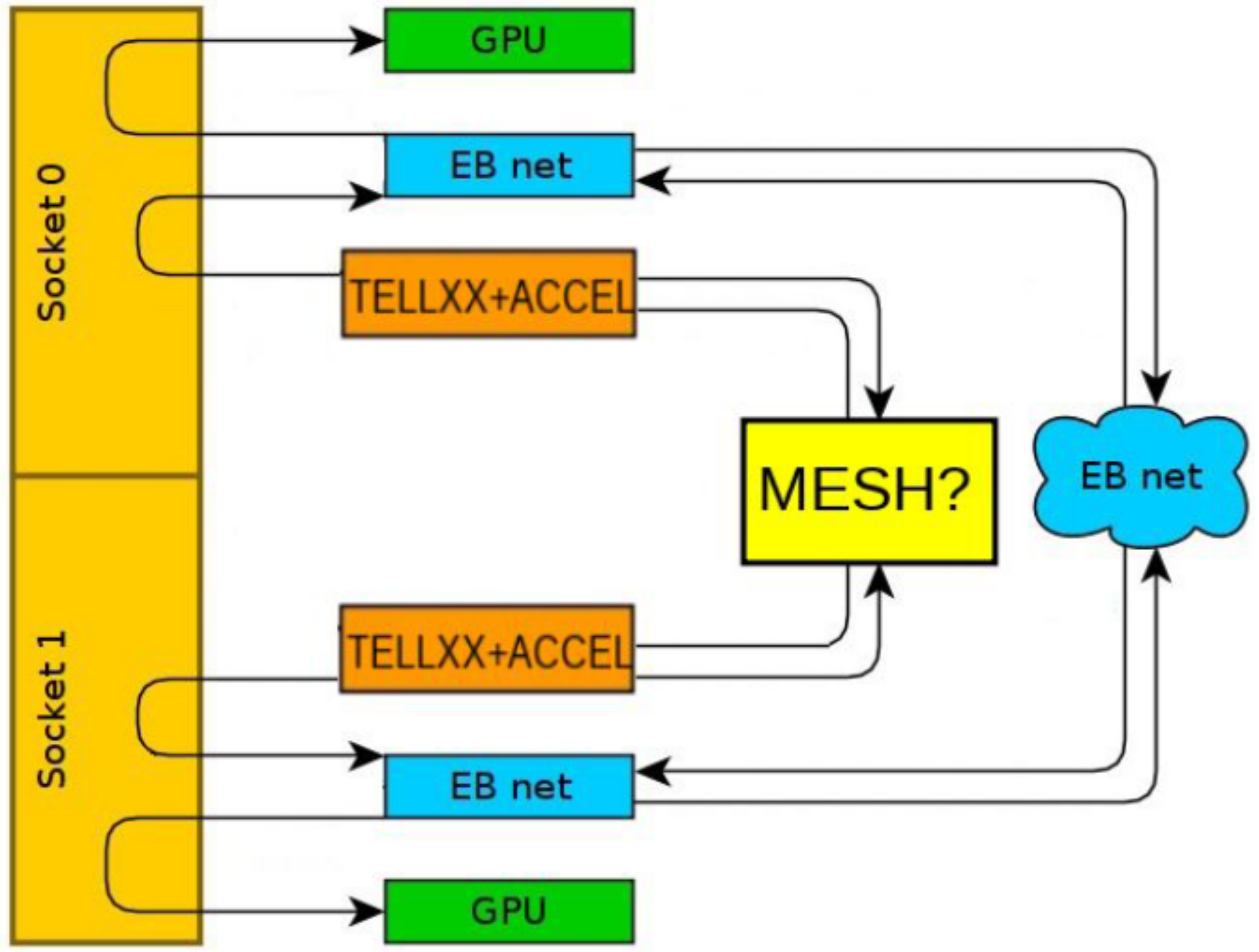
- The implementation of VELO clustering in FPGA is a success story that has sped up the HLT1 sequence by about 10-15%, at no additional hardware cost
  - There are further possible gains mainly relating to efficiently preparing memory / search structures, which have a deep impact in performance
  - "***Pre-processing opens new possibilities*** *of exploiting data that might not be possible or practical with traditional data processing architectures*" (see https://bit.ly/3nb5ah4)
- In more optimistic scenarios, even a partial reconstruction could be done
- The RETINA project will test various avenues in this direction



Wikipedia

Room for more

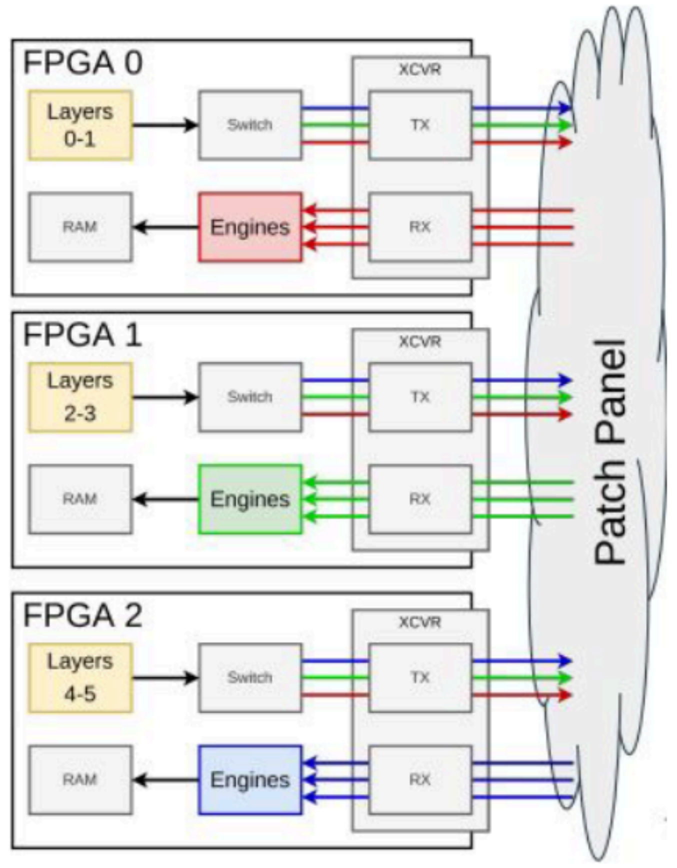
- Several ongoing projects for tracking use-cases: VELO tracking stubs, SciFi seeding primitives, CALO reconstruction on readout boards
- The RETINA architecture can be implemented either in a separate mesh from the DAQ or within the EB servers
  - Demonstrators are in development for Run 4
  - For Run 5, aim is to integrate the accelerator within the TELLXX, providing a more integrated and efficient system



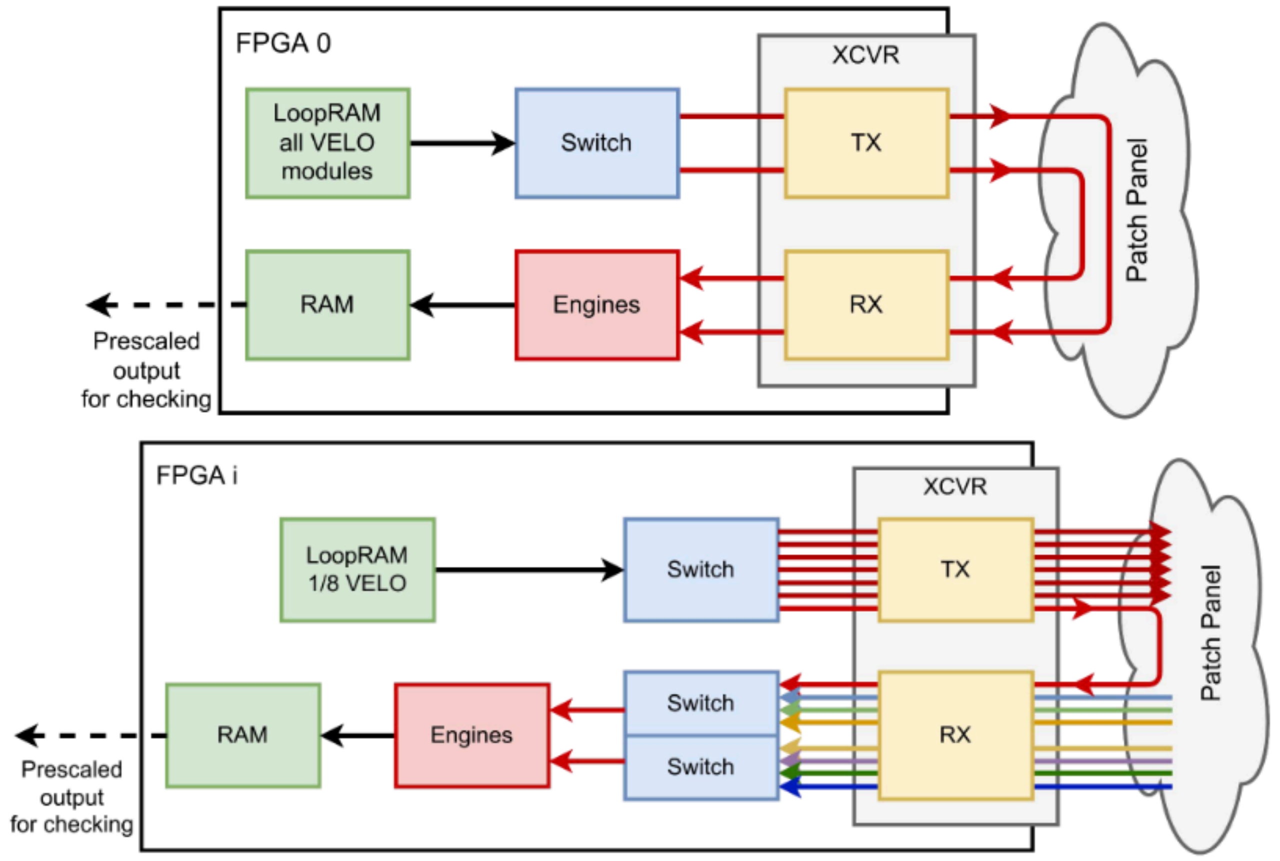
For more on RETINA see: https://indico.cern.ch/event/1252444/

RETINA demonstrators

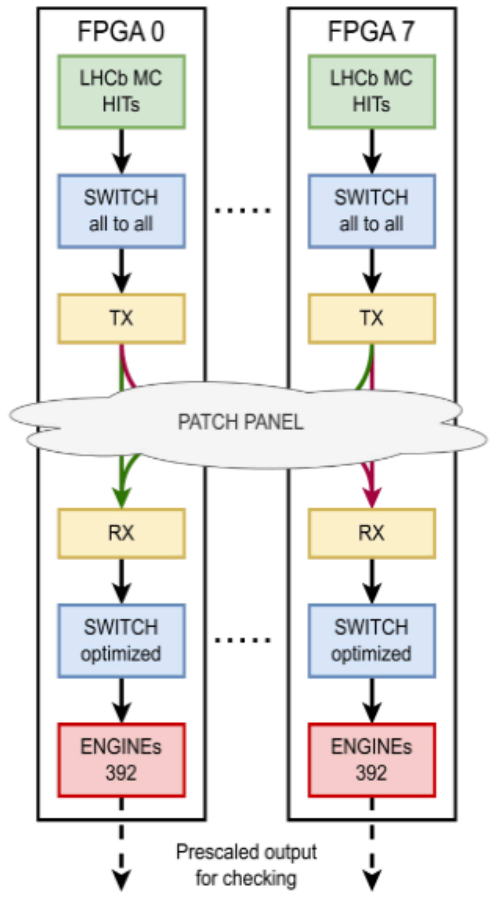
- In the last year the realization status of the Demonstrator evolved rapidly:
  - 104th LHCb week (Jun): 3-boards prototype with dummy engines.
  - 106th LHCb week (Dec): single board and 8-boards demonstrator with VELO engines.
  - Retina for U2 workshop (Feb): 8-boards demonstrator of a VELO quadrant at full-luminosity.



3-boards RETINA tracker prototype



Single and 8-boards Demonstrator

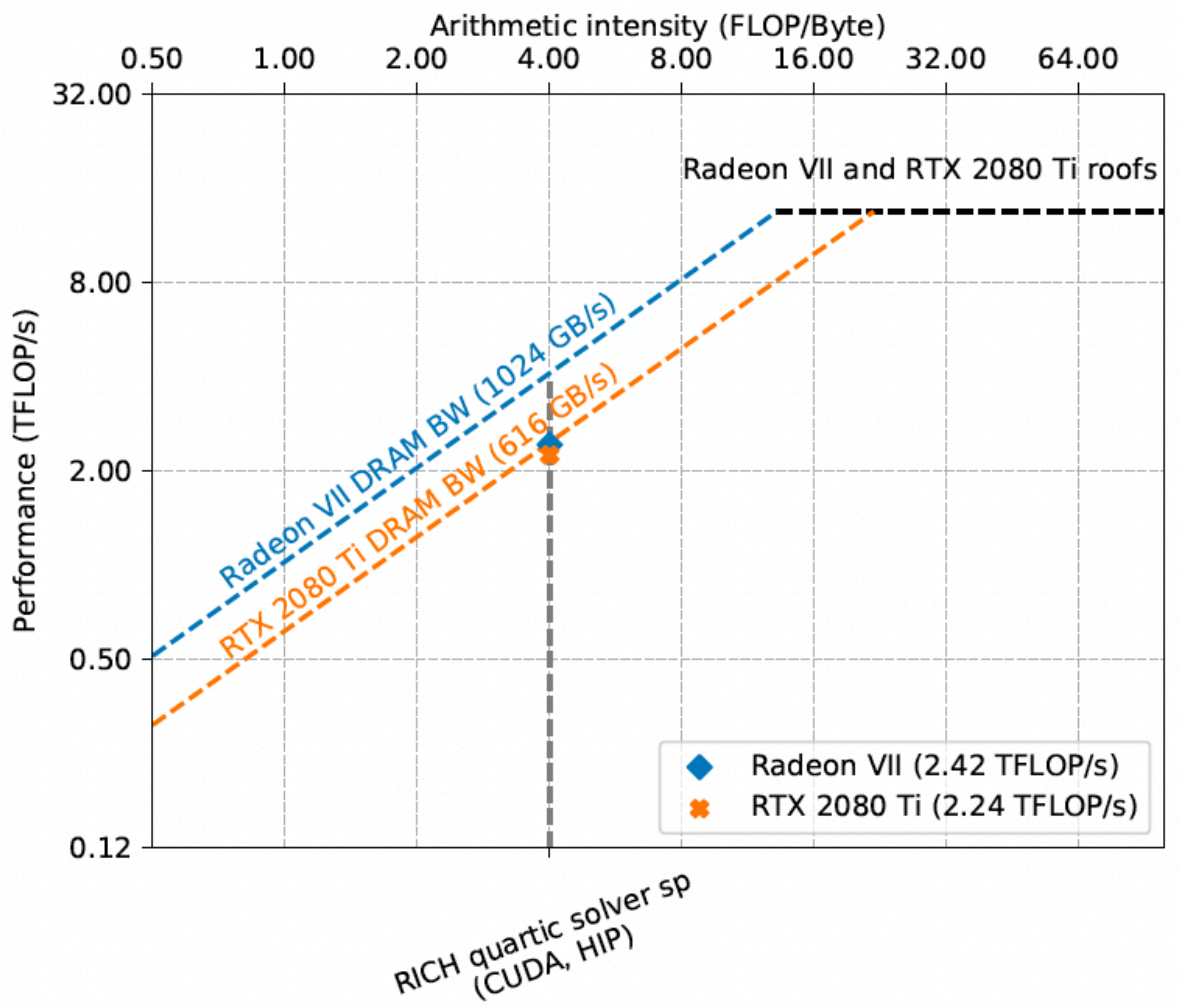


VELO quadrant Demonstrator

See https://indico.cern.ch/event/1267320/contributions/5322089/attachments/2616710/4522972/TB_RETINA_230323.pdf

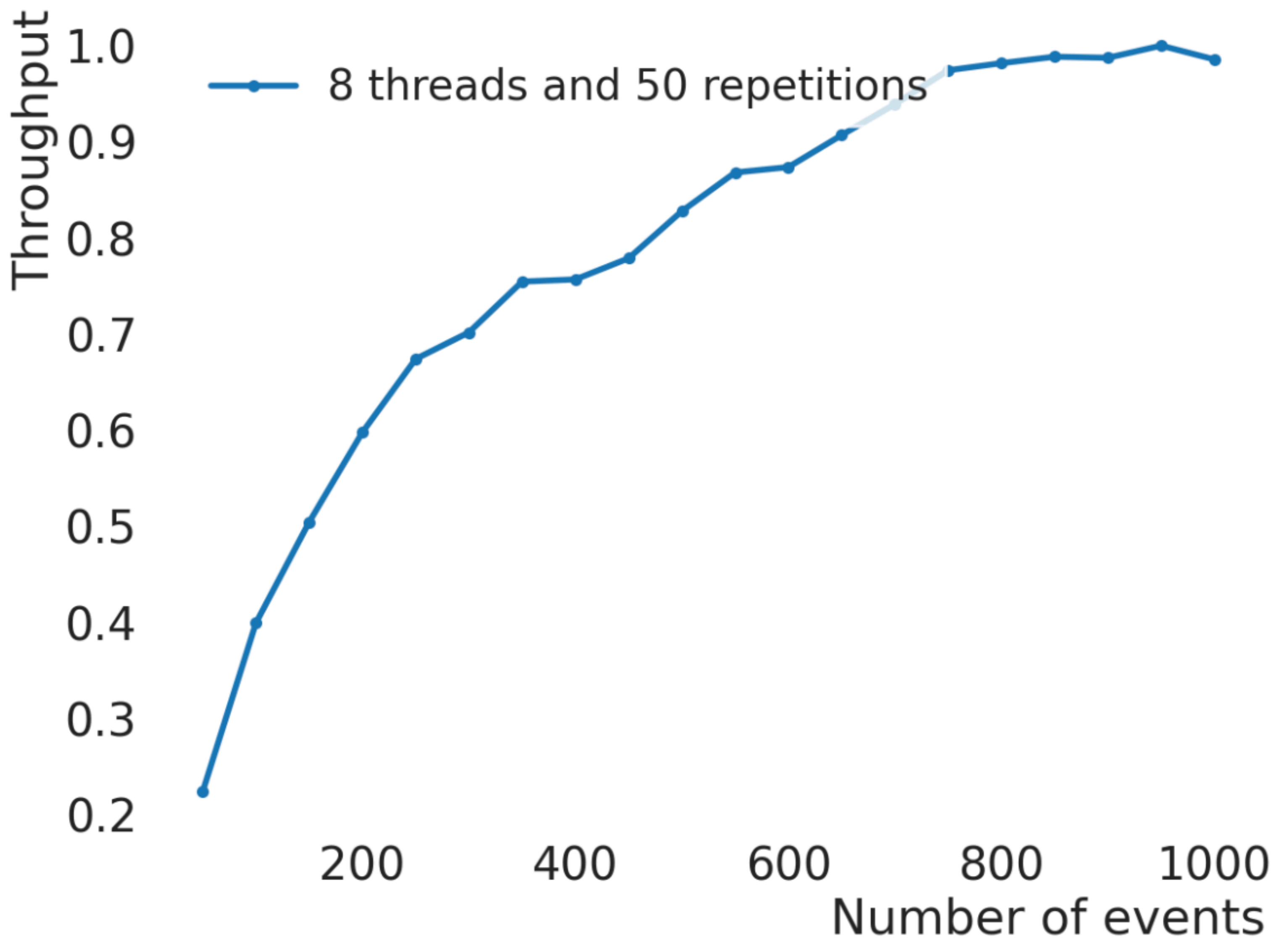
Advances in trigger software

- The installation of an additional GPU stimulates new algorithmic developments. Some recent examples
  - Downstream tracking (presentation, MR)
  - Best track creator (presentation, MR)
  - RICH decoding (MR produced in collaboration with Costa Rican group!)
- Wider HLT1 and HLT2 programmes in GPU and CPU will be possible by Run 4
- Clear indications that full reconstruction on GPUs is possible by Run 5



Other specialized hardware

- IPUs an exciting architecture, language specialization a bit too drastic
- Intel GPUs are being tested alongside a SYCL / OneAPI incarnation of Allen (see this presentation)



Maximum throughout: 8820 events/s

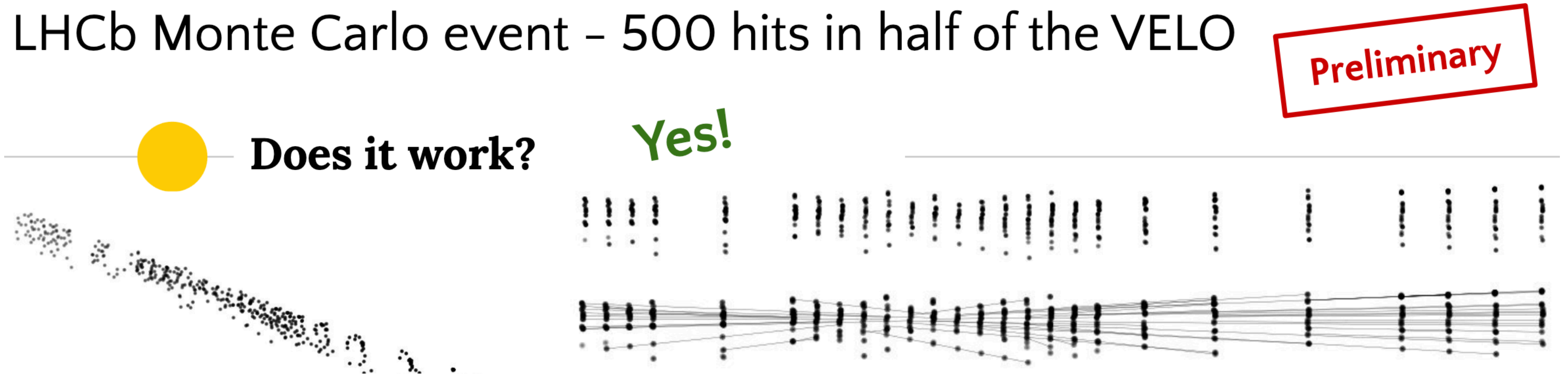
Plateau could be due to:

- Memory bandwidth limitations.
- Thread contention.

Daniel Cámpora

Other specialized hardware (2)

- Quantum computing slowly becoming a reality, LHCb is leading the way (see presentations on this meeting). Still likely 10-20 years before public adoption probably as an accelerator specialized for certain tasks



LHCb Monte Carlo event – 500 hits in half of the VELO

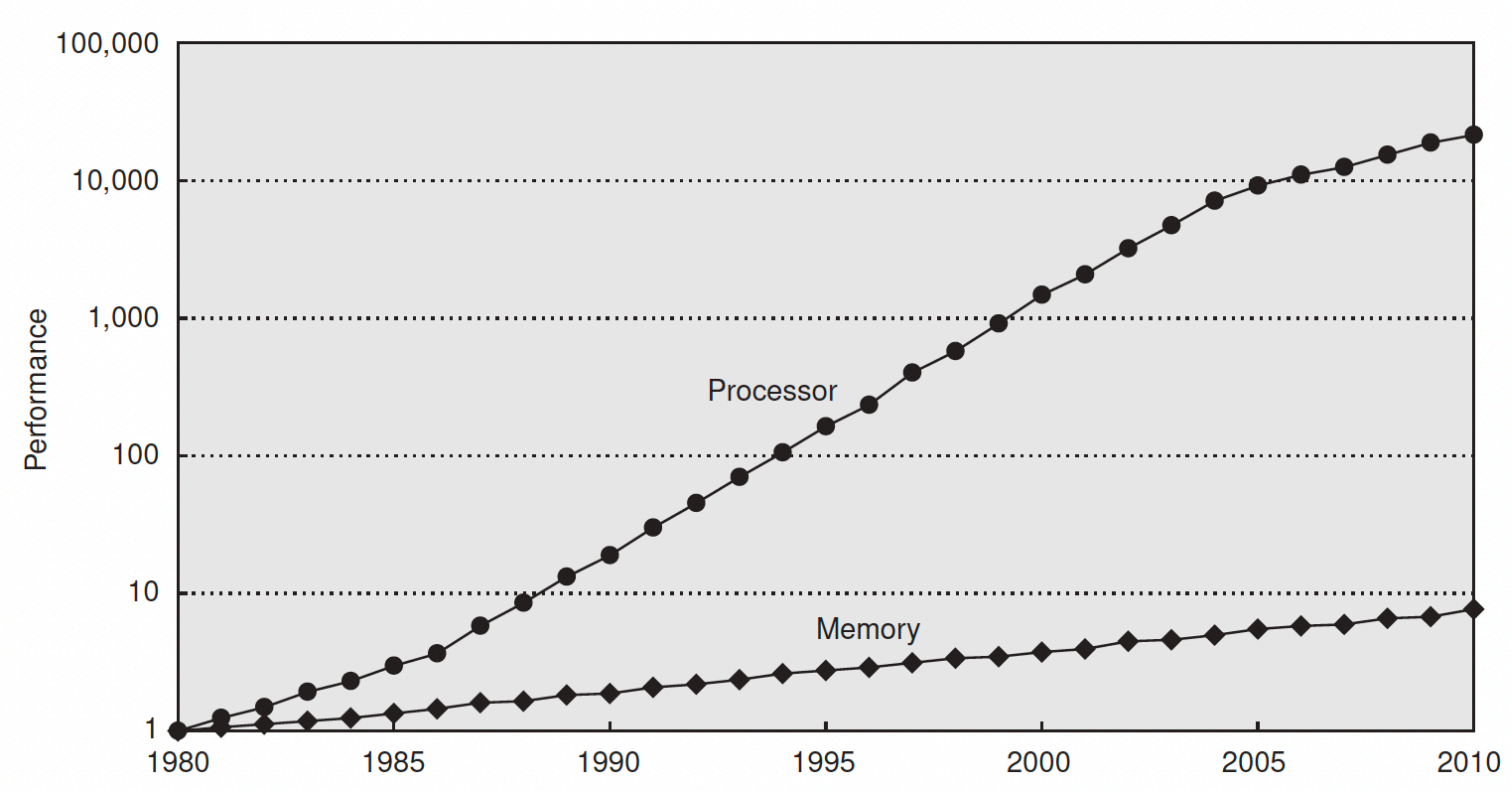
- Intel FPGA inside the CPU package didn't materialize as a competitive product

What are the constraints?

- Data-rates will go up. Data transfer patterns still relevant
  - Some form of double buffer of MEPs in memory will remain the go-to solution, as it is more efficient to transmit a big chunk of data instead of many small pieces
- Bandwidth is a limiting factor that must be taken into account
- Hard-Drive write scaling is not great, reconsider how to do deferred triggering most efficiently

Where does this lead us?

- Architecture-aware programming is a must. In particular, memory will play a major role to designing our future algorithms

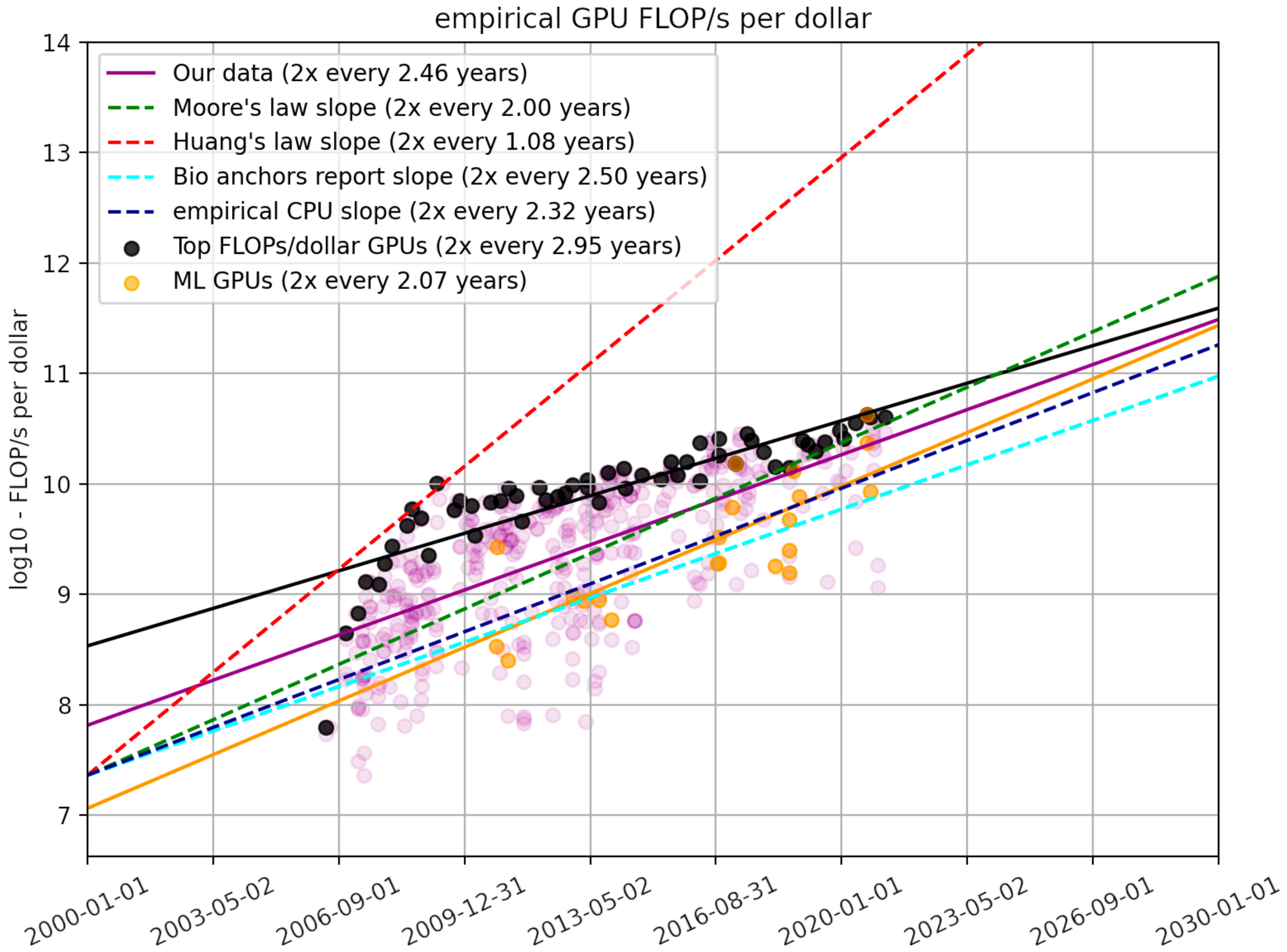


From "Computer Architecture"

Daniel Cámpora

Where does this lead us? (2)

- Many-core is a reality. The lead in GPU performance is likely going to stay as the better throughput oriented processor, with no hard latency requirements driving their design



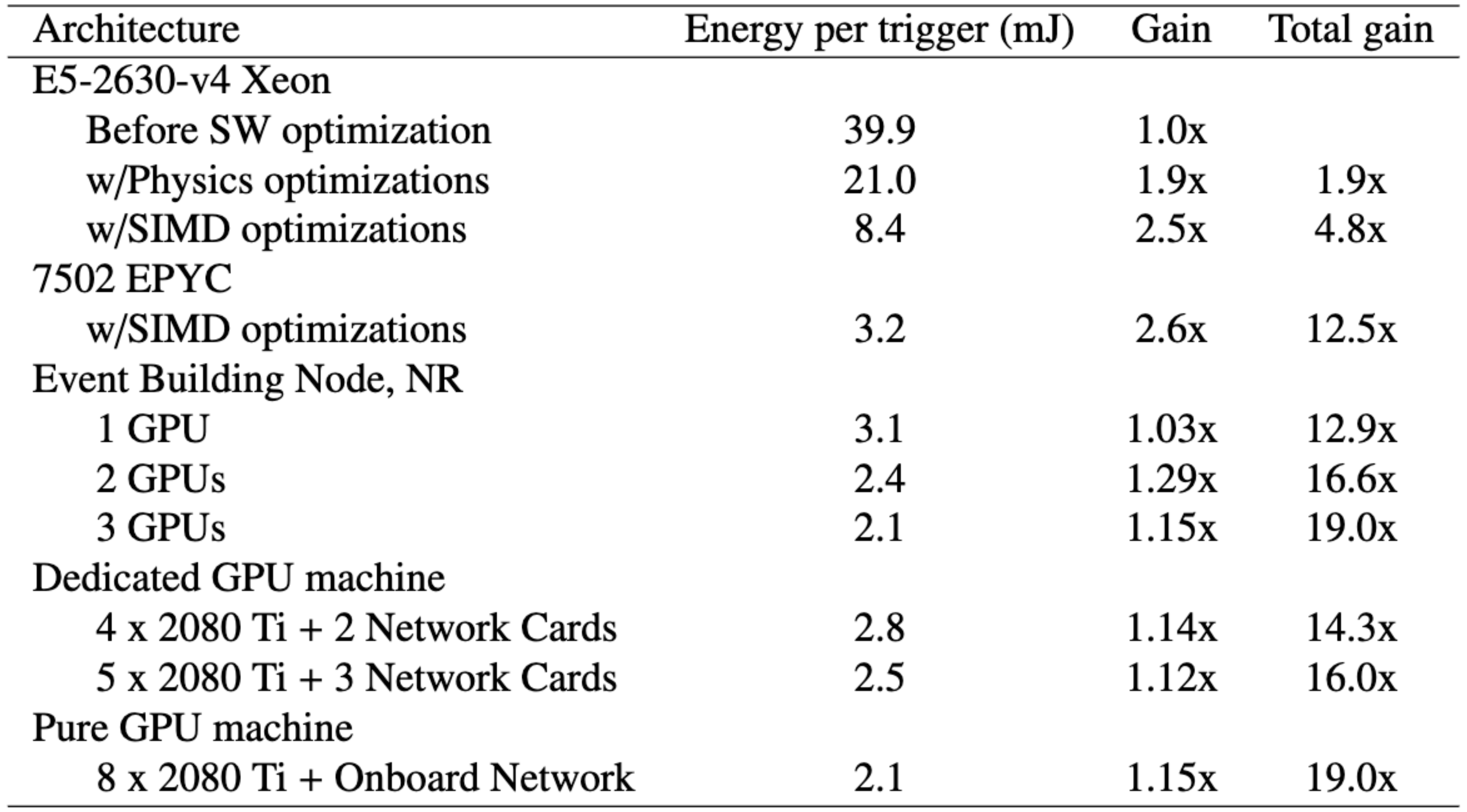
Source: https://epochai.org/blog/trends-in-gpu-price-performance

Where does this lead us? (3)

- GPUs are designed specifically as a C++ architecture (see https://www.youtube.com/watch?v=86seb-iZCnI), we are probably at the right language already
- In particular, choice of accelerator specialization should be driven by performance, as conversion layers are less hard work
- The system must scale - every test so far shows good scalability
    - In particular, there is no evidence that performance for U2 will be *framework bound*
- FPGAs will become more relevant. Tandem between FPGA-software can lead to big gains

A word on energy efficiency

- We should pay attention to energy efficiency, as it is becoming increasingly important
- Take into consideration throughput, energy efficiency and cost
  - Develop methodologies to measure reliably energy contribution of each process
  - Note that *Thermal Design Power (TDP) is not a good metric*, but rather one needs to measure power consumption



Architecture	Energy per trigger (mJ)	Gain	Total gain
E5-2630-v4 Xeon			
Before SW optimization	39.9	1.0x	
w/Physics optimizations	21.0	1.9x	1.9x
w/SIMD optimizations	8.4	2.5x	4.8x
7502 EPYC			
w/SIMD optimizations	3.2	2.6x	12.5x
Event Building Node, NR			
1 GPU	3.1	1.03x	12.9x
2 GPUs	2.4	1.29x	16.6x
3 GPUs	2.1	1.15x	19.0x
Dedicated GPU machine			
4 x 2080 Ti + 2 Network Cards	2.8	1.14x	14.3x
5 x 2080 Ti + 3 Network Cards	2.5	1.12x	16.0x
Pure GPU machine			
8 x 2080 Ti + Onboard Network	2.1	1.15x	19.0x

See https://arxiv.org/abs/2106.07701

A word on specialized hardware

- A.I. is making big advances, and it is affecting how hardware evolves
    - Tensor cores, TPUs, IPU
    - In some cases, this hardware can be utilized by smart software, a form of low-level optimization
    - In most cases however, we can benefit from this trend in the (relatively few) use-cases we have in the HLT
- Other hardware specializations such as Ray Tracing may be usable, but no concrete demonstrator developed yet for the trigger

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Daniel Cámpora

Some A.I. use cases

- Track seeding (see this presentation)
- Fake track removal
- Calorimeter reconstruction (see this proceeding)
- Adoption or demonstrators across experiments mostly in classification and calibration (see this presentation)
- We should have support in our frameworks for these use-cases
  - Hardware-accelerated support for pre-trained Machine Learning models is in the works (Allen, to be presented at CHEP'23)

Reproducibility

- Results should be reproducible offline
  - With a well-defined metric, taking into account different hardware produces slightly different results. Ie. different architectures, different low-level features (FMAs, vector width), different compilers
  - Performance is not a key aspect offline
- The ideal scenario is to write software once, execute it everywhere, as we already do
  - We should remove duplicity
  - Architecture-aware optimizations in hot sections of our code can and should still happen
- Reproducibility of FPGA processing so far accomplished via emulation

Conclusions

- HLT will remain a hot topic in Run 5. Architecture choice will be driven by throughput
    - Performance first, portability as needed
    - This is particularly the case in an online environment, where our physics case is mostly limited by available resources: No cycles are ever spare
- FPGAs are a key player that will improve trigger efficiency
    - Let's strengthen the collaboration between FPGA-software teams
- Other hardware alternatives are not promising at the moment - but we should keep an open mind
- LHCb trigger software is driving innovation and attracting talent
- R&D is fundamental to take informed and benchmarked decisions in the future