Performance assessment of FPGAs as HPC accelerators using the FPGA Empirical Roofline

Enrico Calore

INFN & Univeristy of Ferrara, Italy

25/05/2022

Enrico Calore
INFN & UniFE
enrico.calore@fe.infn.it

Outline

EuroEXA EU H2020 FET-HPC Project

EuroEXA: *Co-designed innovation and system for resilient exascale computing in Europe: from application to silicon*

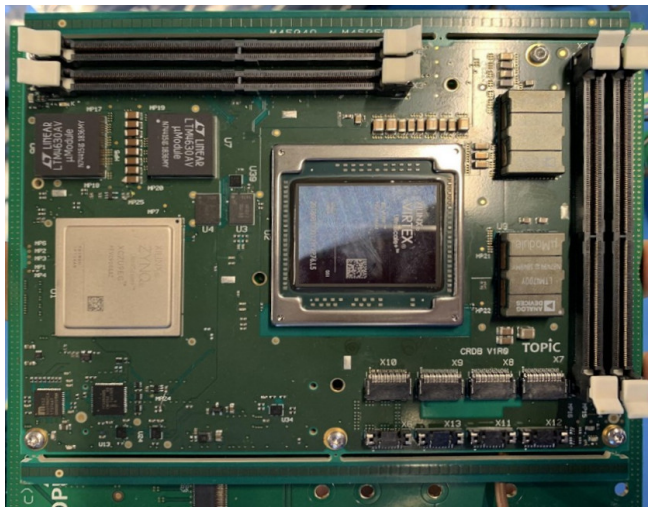
A Co-design HPC Project featuring:

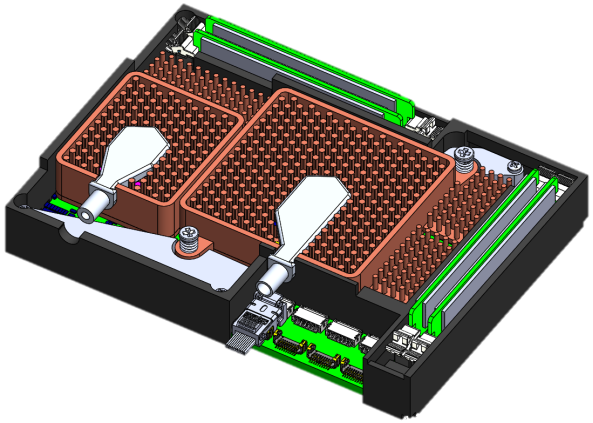
- use of FPGAs as accelerators;
- use of FPGAs to implement custom interconnects;
- co-design a balanced architecture for both compute and data-intensive applications.

EuroEXA Prototype Architecture

Co-design Recommended Daughter Board (CRDB):

- Zinq UltraScale+ ZU9 for interconnect and compute.
- Virtex UltraScale+ VU9 as a compute accelerator.
- Liquid cooled board.

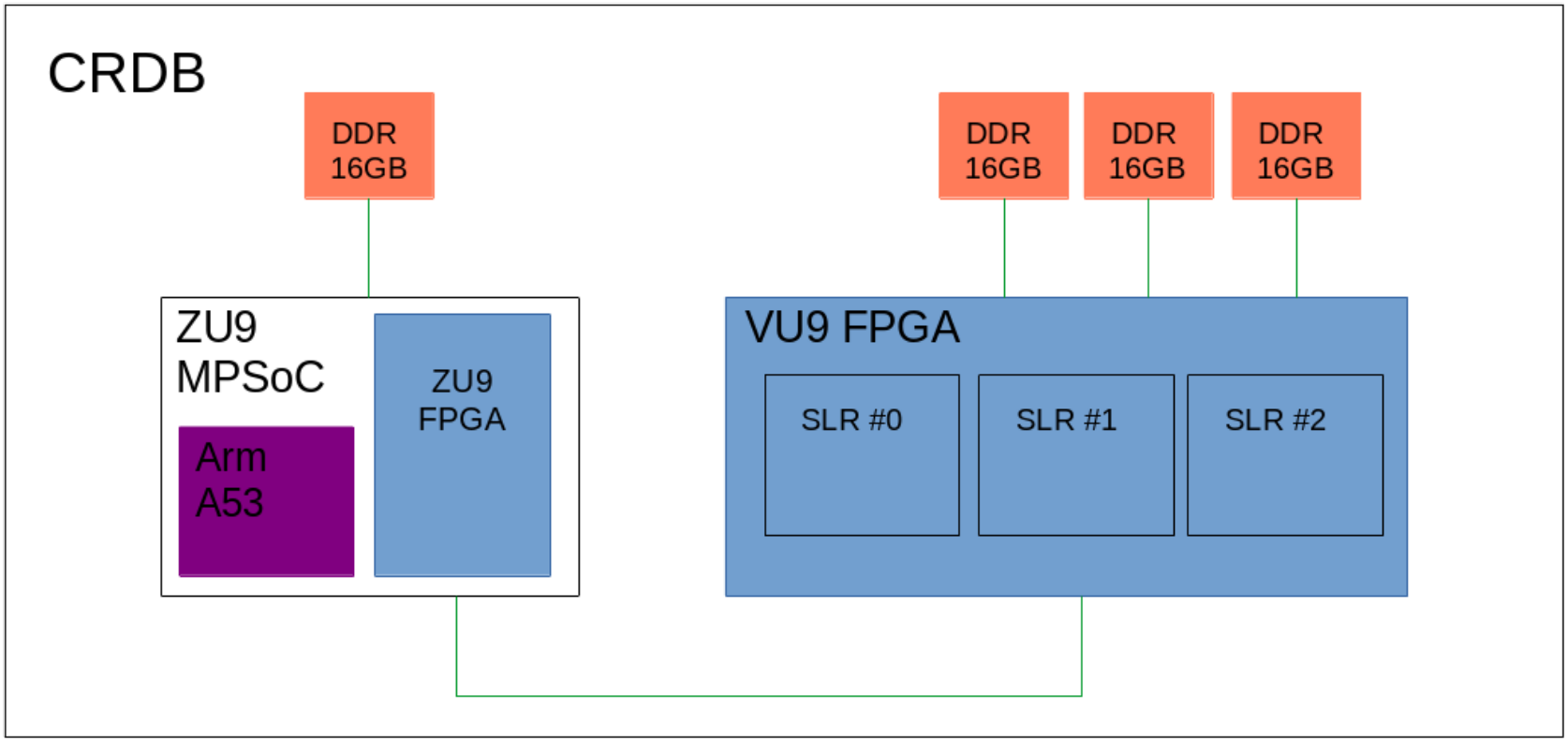




We needed to:

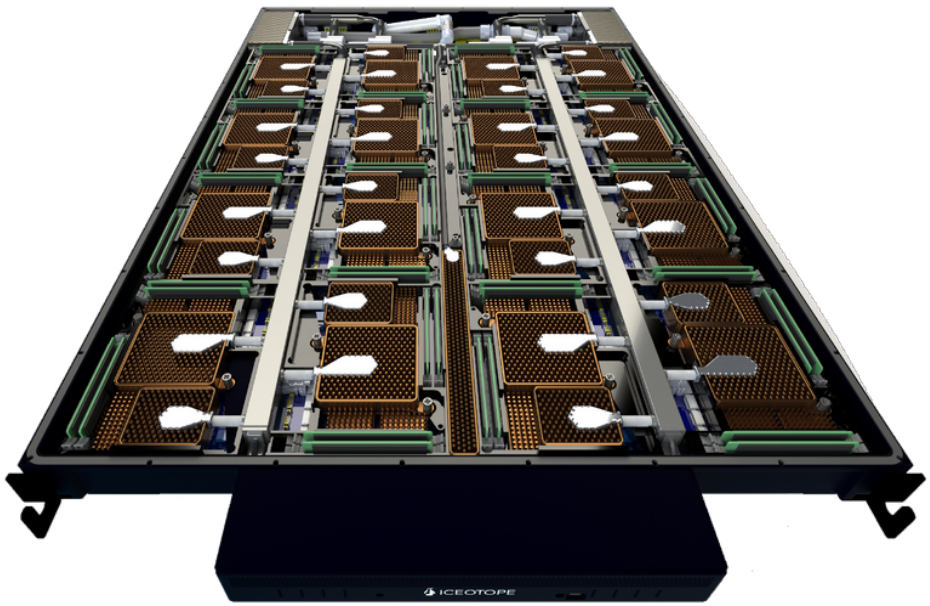
estimate applications expected performance for co-design and evaluation.

EuroEXA Prototype Architecture



Sketch of the Single CRDB board.

First EuroEXA Prototype at scale

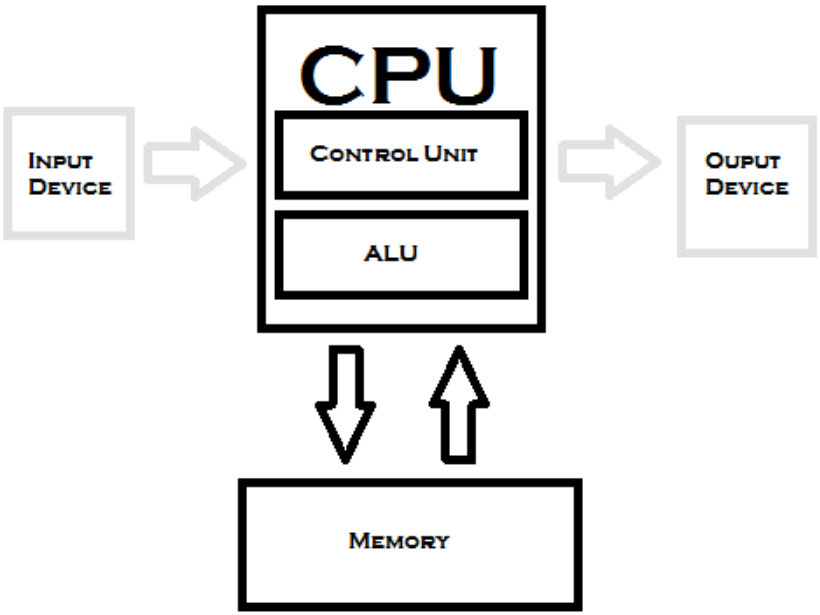


EuroEXA Architecture:

- 16 CRDBs in a Blade.
- 32 EuroEXA Blades in one Rack.
- hierarchical network with hybrid topology: all-to-all at Blade level and torus for inter-Blade level.

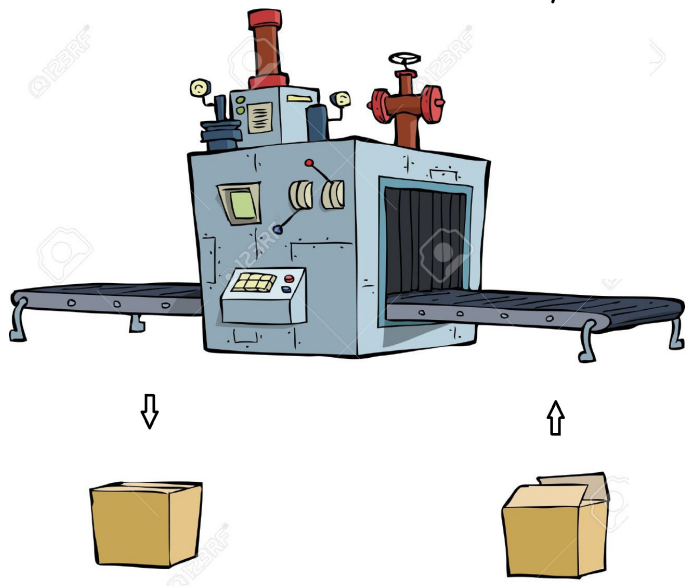
What is limiting the performance?

FLOP/s



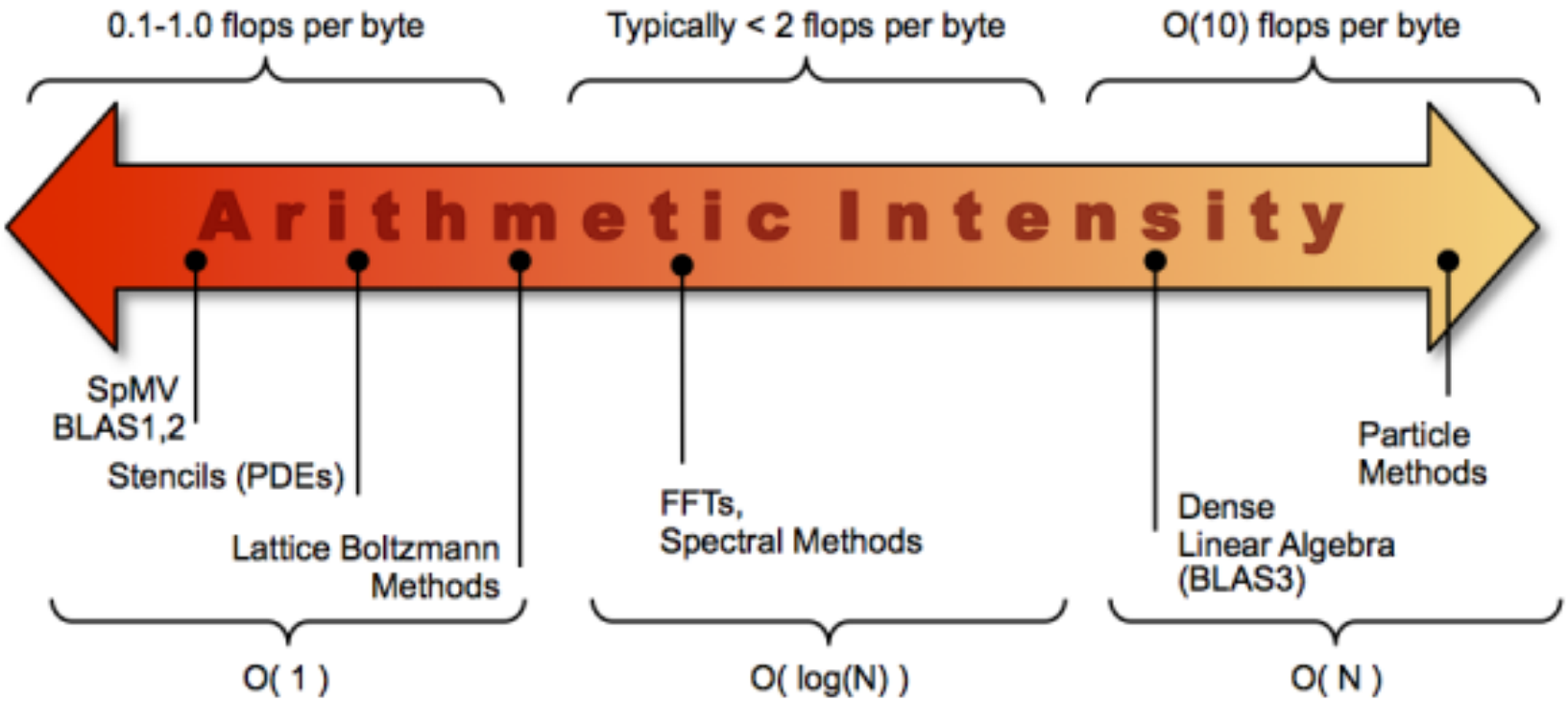
Byte/s

Manufacturing/s



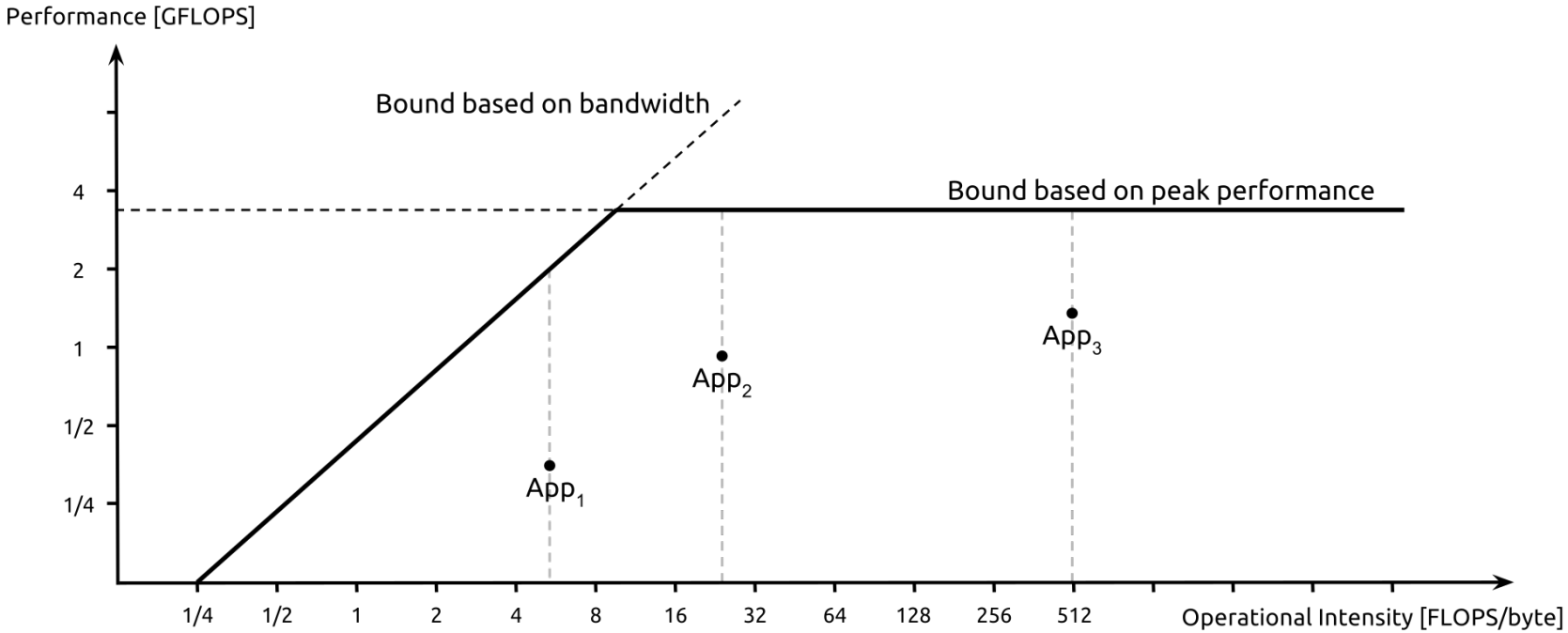
Load&Store/s

Arithmetic/computational Intensity



Roofline Model

The *Roofline Model* is used to provide performance estimates of a given compute kernel running on a given architecture.

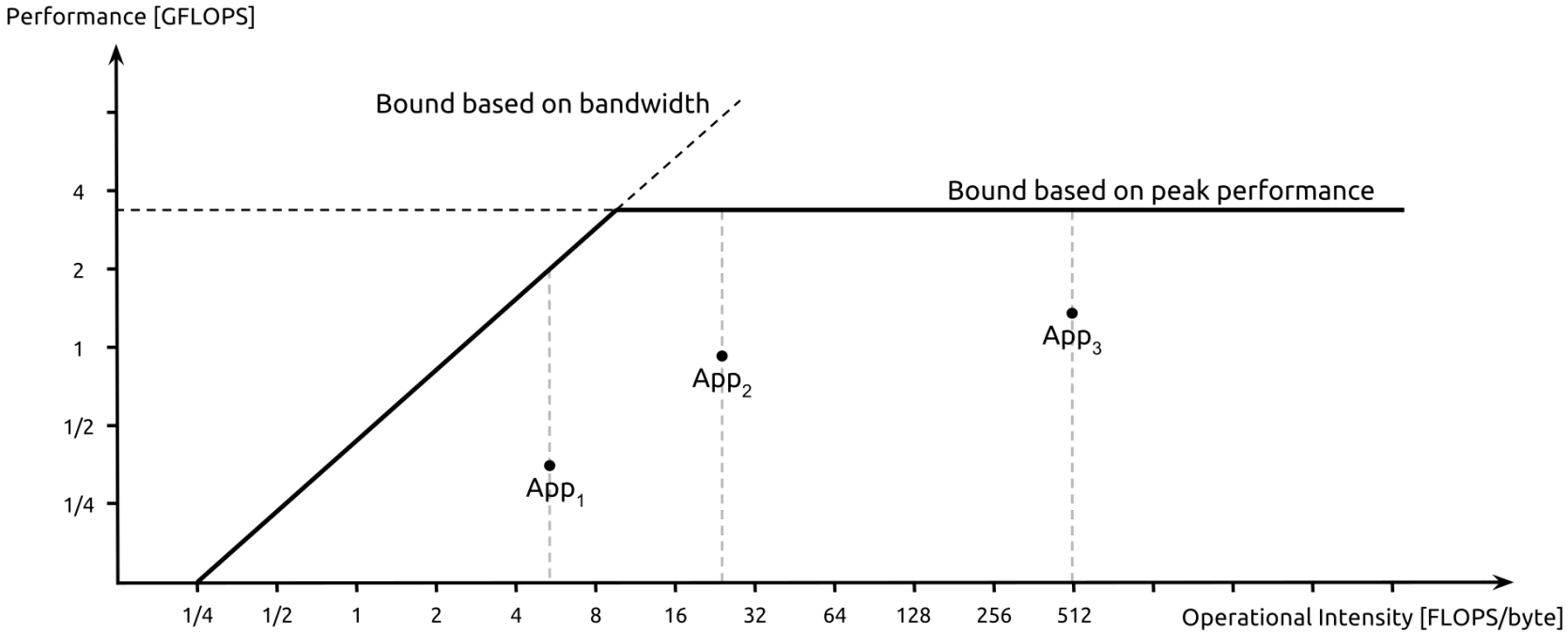


Williams, S., Waterman, A., & Patterson, D. (2009) "Roofline: an insightful visual performance model for multicore architectures" *Communications of the ACM*, 52(4), 65-76.

Peak Bandwidth and Performance could be theoretical ones, or empirically measured...

Roofline Model

The *Roofline Model* is used to provide performance estimates of a given compute kernel running on a given architecture.



Williams, S., Waterman, A., & Patterson, D. (2009) "Roofline: an insightful visual performance model for multicore architectures" *Communications of the ACM*, 52(4), 65-76.

Peak Bandwidth and Performance could be theoretical ones, or empirically measured... **Not trivial on FPGAs.**

FPGA performance model

To estimate the peak performance C of an FPGA in terms of op/s, we can assume that to implement a hardware core performing op, are required R_{op} hardware resources.
If an FPGA contains R_{av} of these resources, the maximum number of implementable hardware cores H_c is:

$$H_c = \frac{R_{av}}{R_{op}}$$

If each core operates at a maximum clock frequency f_{op}, and starts a new operation every clock cycle, the theoretical performance C is:

$$\begin{aligned} C &= f_{op} \times H_c \\ &= f_{op} \left(\frac{R_{av}}{R_{op}} \right) \end{aligned}$$

FPGA performance model

Given that on FPGAs are commonly available Ri_{av} different i types of resources, one of them will limit the number of cores:

$$C = f_{op} \times \min_{i} \left(\frac{Ri_{av}}{Ri_{op}} \right)$$

Such theoretical models, have already been used by FPGA manufacturers to publicize peak performance, but actual applications could be able to reach much lower values.

Intel says:

"For FPGAs lacking hard floating-point circuits, using the vendor-calculated theoretical GFLOPS numbers is quite unreliable. Any FPGA floating-point claims based on a logic implementation at over 500 GFLOPS should be viewed with a high level of skepticism. In this case, a representative benchmark design implementation is essential to make a comparative judgment."

Intel White Paper: *Understanding Peak Floating-Point Performance Claims* (2017)

Too optimistic theoretical estimations

It is actually too optimistic:

- to assume to be able to exploit all of the available resources of one specific type;
- to assume to reach the maximum clock frequency declared for a single *op* core, when a large fraction of resources is used.

Need for empirical parameters: f_{imp} and u_{Ri}

$$C = f_{imp} \times \min_i \left(\frac{Ri_{av}}{Ri_{op}} \times u_{Ri} \right), \quad u_{Ri} < 1$$

The FPGA Empirical Roofline (FER)

Empirical Roofline Tool (ERT)

Berkeley Lab

- Empirically find the max FLOPs and Bandwidth
- Kernel with tunable arithmetic complexity
- Targeting CPUs/GPUs
  (OpenCL kernel can target also FPGAs)

`https://crd.lbl.gov/departments/computer-science/PAR/research/roofline/software/ert/`

FPGA Empirical Roofline (FER)

INFN & Univ. of Ferrara

- Based on the same principles of ERT
- Written using HLS directives
- Targeting FPGA devices

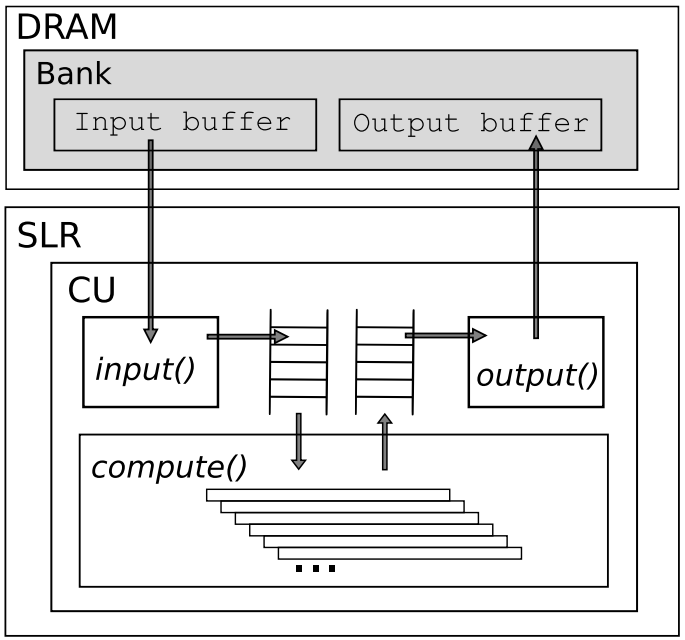
`https://baltig.infn.it/EuroEXA/FER`

Main FER C/HLS kernel function

Implements a task level pipeline (dataflow):

reading elements from an input array, applying to each a given *op*, for O_e times, and storing the result in an output array.

```
1  void fer( const data_v *input,
2                  data_v *output ) {
3
4    hls::stream<data_v> inFifo;
5    hls::stream<data_v> outFifo;
6
7    #pragma HLS dataflow
8
9    readInput(input, inFifo);
10   compute(inFifo, outFifo);
11   writeOutput(output, outFifo);
12 }
```



OmpSs@FPGA for EuroEXA and Xilinx Vitis for Alveo boards.

FER *compute()* function

FER tuning knobs:

$$C = f \times \frac{V \times O_e}{II_c}$$

II_c: Initiation Interval

V: SIMD vector width

O_e: Ops per element

Hardware limit:

$$\frac{V \times O_e}{II_c} < \min_i \left(\frac{Ri_{av}}{Ri_{op}} \times u_{Ri} \right)$$

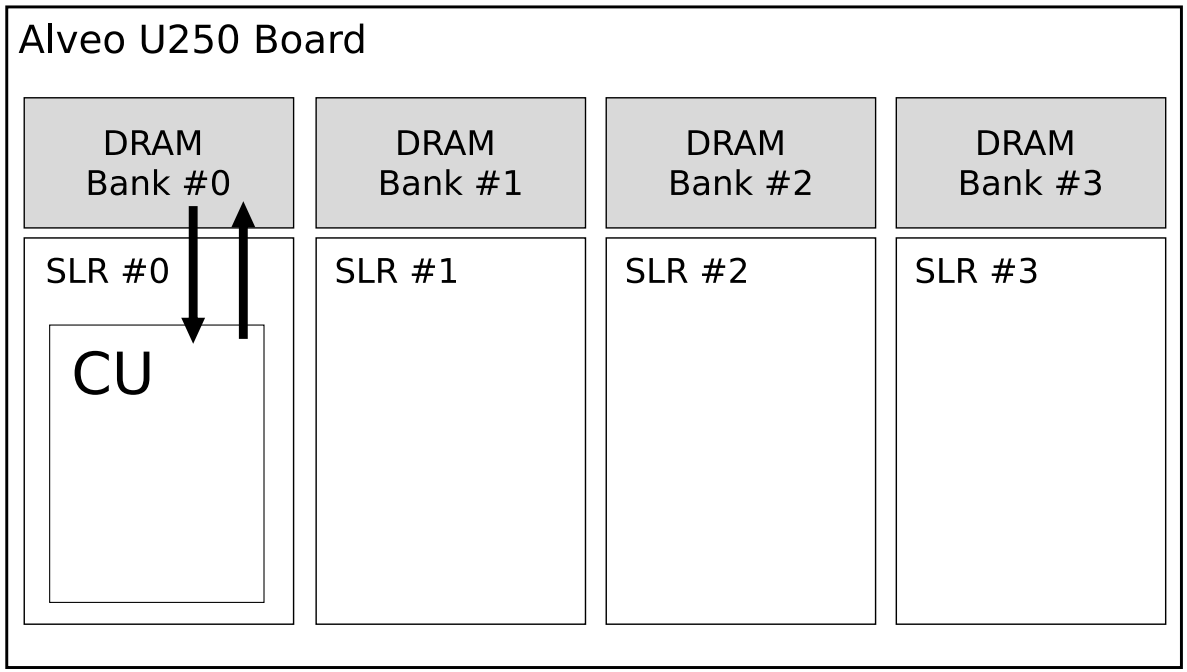
```
1  void compute(hls::stream<data_v> &inFifo,
2               hls::stream<data_v> &outFifo)
3
4    for (i = 0; i < DIM; i++) {
5      #pragma HLS pipeline II=IIc
6
7      data_v in = inFifo.read();
8
9      for (e = 0; e < V; e++) {
10       #pragma HLS unroll
11       data_t elem = in.elem[e];
12       for (o = 0; o < Oe; o++) {
13         #pragma HLS unroll
14         elem = op(elem);
15       }
16       out.elem[v] = elem;
17     }
18
19     outFifo.write(out);
20
21   }
```

Results for a Xilinx Alveo U250

Xilinx Alveo U250 Data Center Accelerator Card

Results for a Xilinx Alveo U250

We use as *op* a double-precision floating-point FMA, to allow for cross-architectural comparison, with other HPC processors.



Best performance using 4 Compute Units (CUs), one for each SLR.

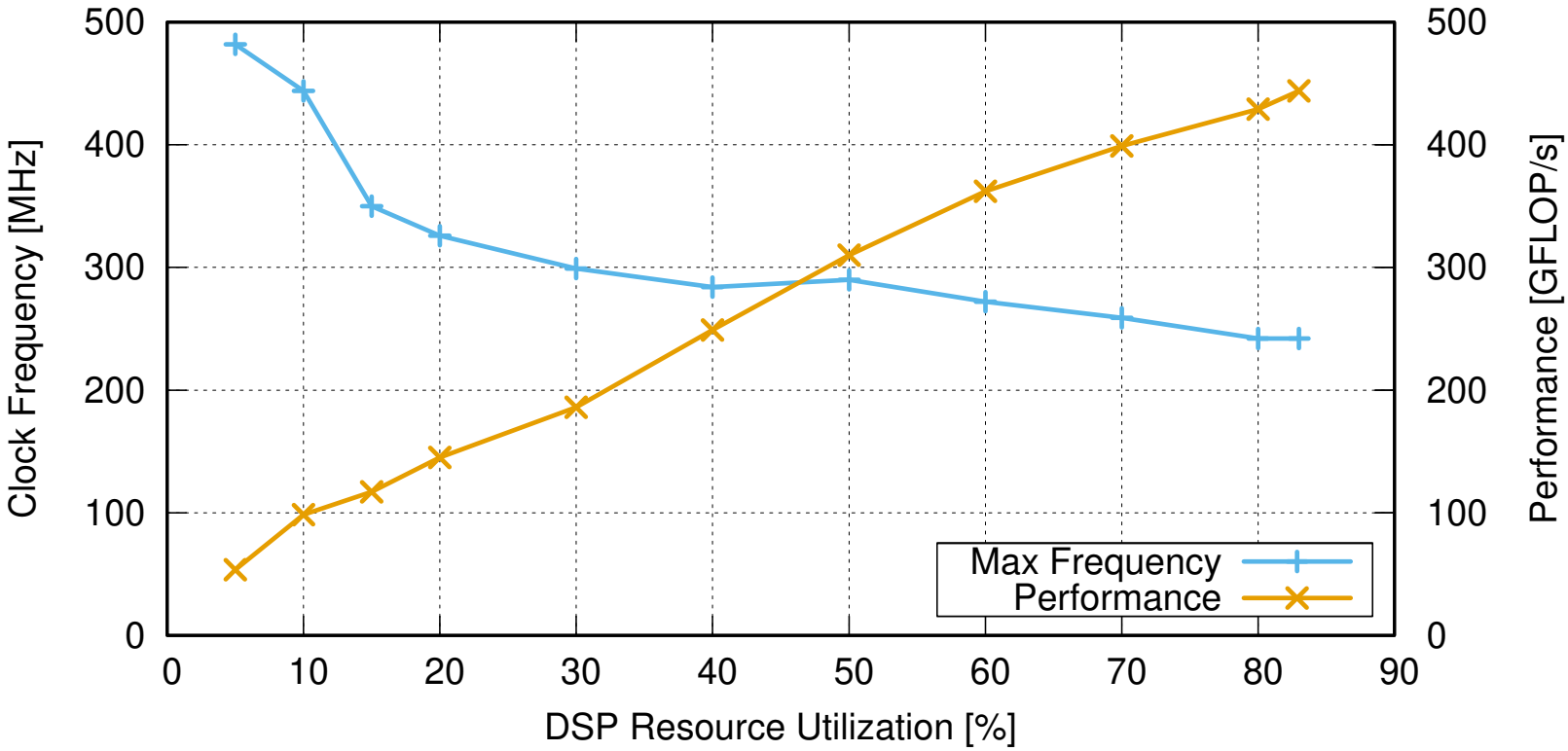
U250 theoretical performance

The theoretical performance of this FPGA should be:

$$
\begin{aligned}
C &= 694\text{MHz} \times \min\left(\frac{1.380 \cdot 10^6}{616 + 172}\text{LUT},\ \frac{11508}{8 + 3}\text{DSP} \right) \\
&= 726 \cdot 10^9\ \text{FMA/s} \\
&= 1.45\ \text{TFLOP/s}
\end{aligned}
$$

Max empirically reachable f_{imp} and C

Synthesized and run FER for different H_c, keeping the arithmetic intensity in the *compute-bound* region.



A more realistic estimation for Alveos

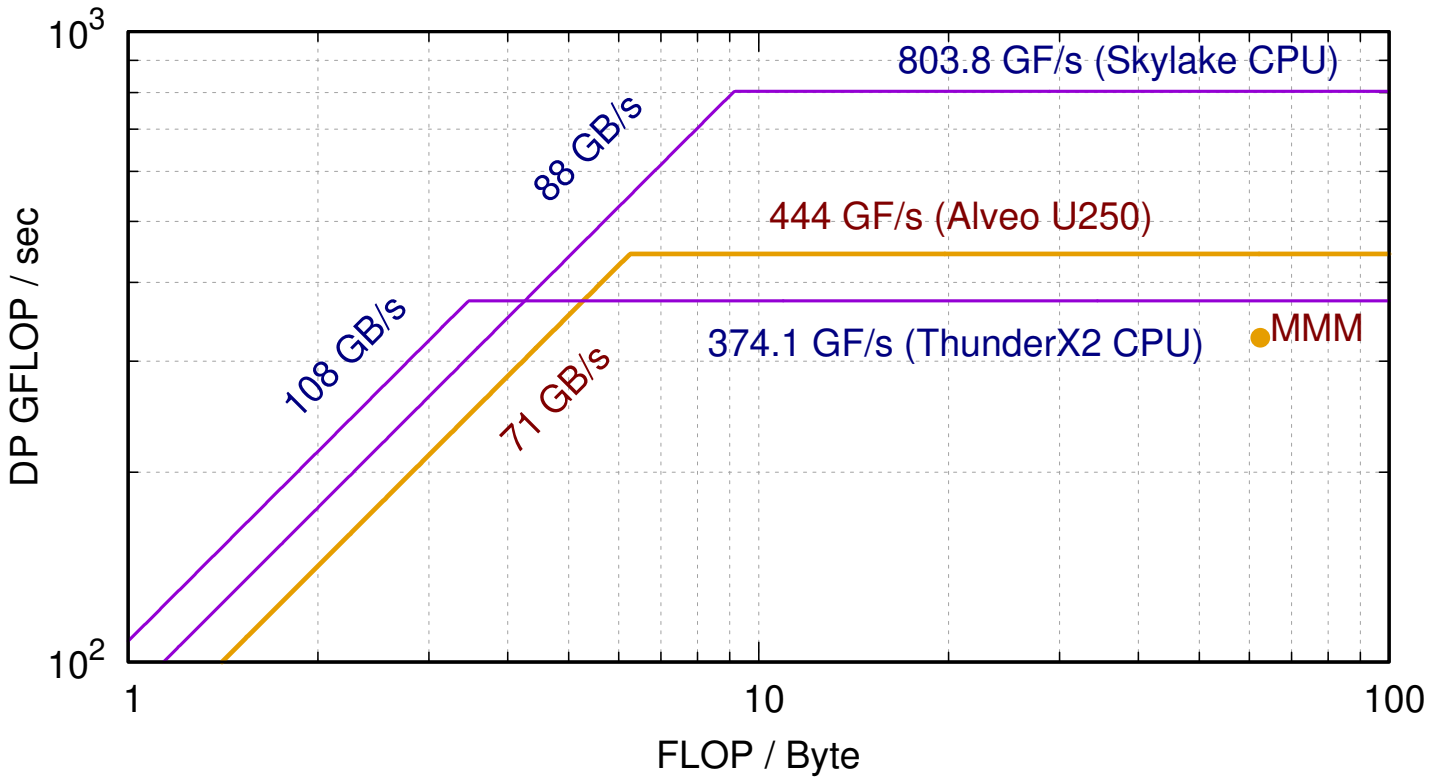
In the Xilinx documentation realistic f_{imp} and u_{Ri} values to be used for performance estimation, could be selected as:

- default clock frequency of 300MHz (provided in the Alveo *Platforms* documentation);
- suggested maximum resources utilization (published in the Vitis Unified Software Platform Documentation as "Timing closure considerations"): i.e., 70% for LUTs and 80% for DSPs.

These would give an estimated performance C of:

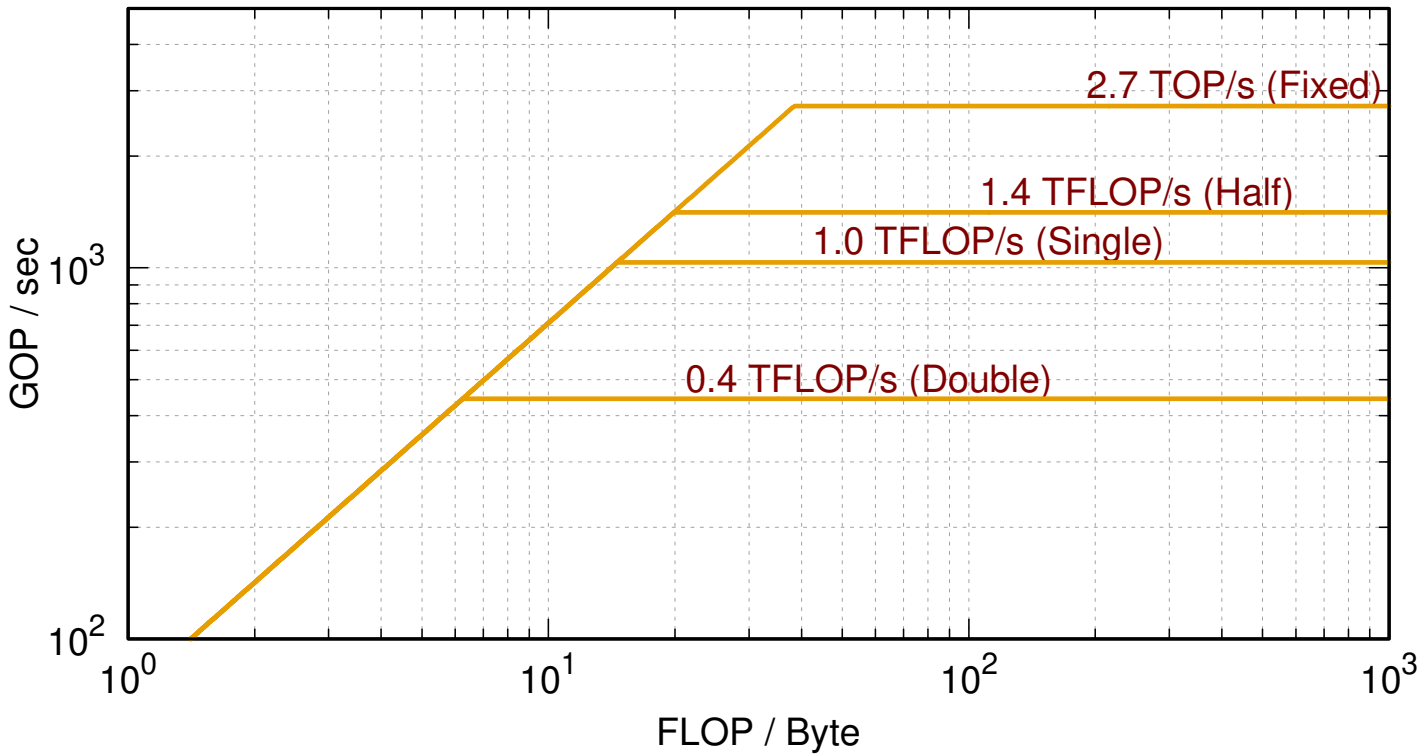
$$
\begin{aligned}
C &= 300 \times \min\left(\frac{1.380 \cdot 10^6}{616 + 172}\text{LUT} \times 0.7,\ \frac{11508}{8 + 3}\text{DSP} \times 0.8\right) \\
&= 251 \cdot 10^9 \text{ FMA/s} \\
&= 502 \text{ GFLOP/s}
\end{aligned}
$$

Cross-architectural comparison (not fair for FPGA)



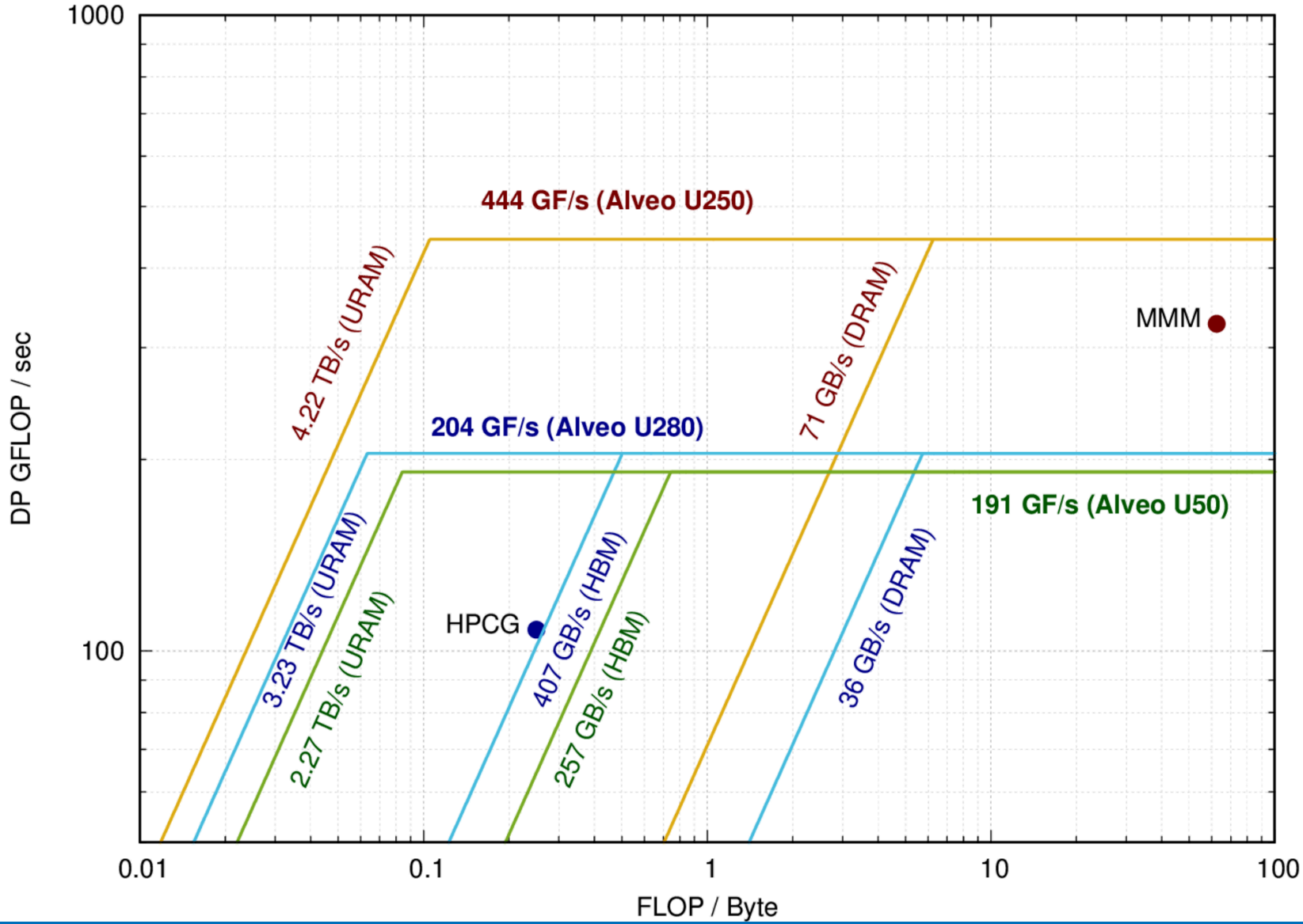
Roofline obtained by FER on the Alveo U250, compared with the ones obtained by ERT on an Intel Skylake CPU and by our custom Arm optimized ERT version on a Marvell ThunderX2 CPU.

Empirical Roofline for lower precision



Much more competitive performance can be obtained using lower precision operations.

Empirical Roofline for other Alveos



Many Thanks

Please Connect at:

https://euroexa.eu/

https://twitter.com/euroexa

Enrico Calore
INFN & UniFE
enrico.calore@fe.infn.it