ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Accelerating OpenVX Applications on Embedded Many-Core Accelerators

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Giuseppe Tagliavini, DEI (University of Bologna)

Germain Haugou, IIS-ETHZ

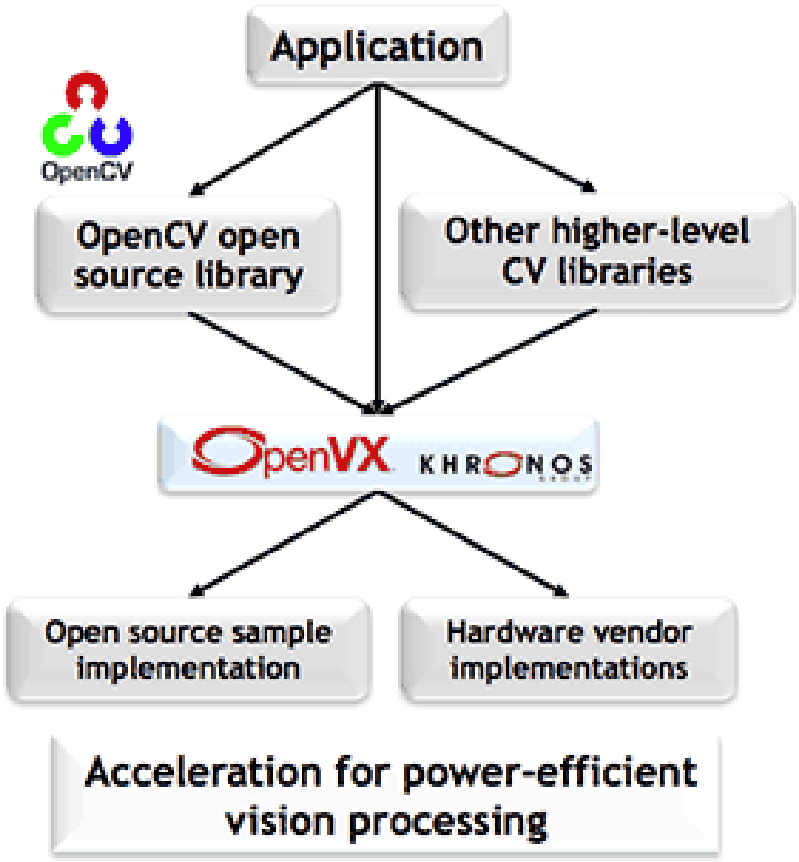
Andrea Marongiu, DEI (University of Bologna) & IIS-ETHZ

Luca Benini, DEI (University of Bologna) & IIS-ETHZ

Outline

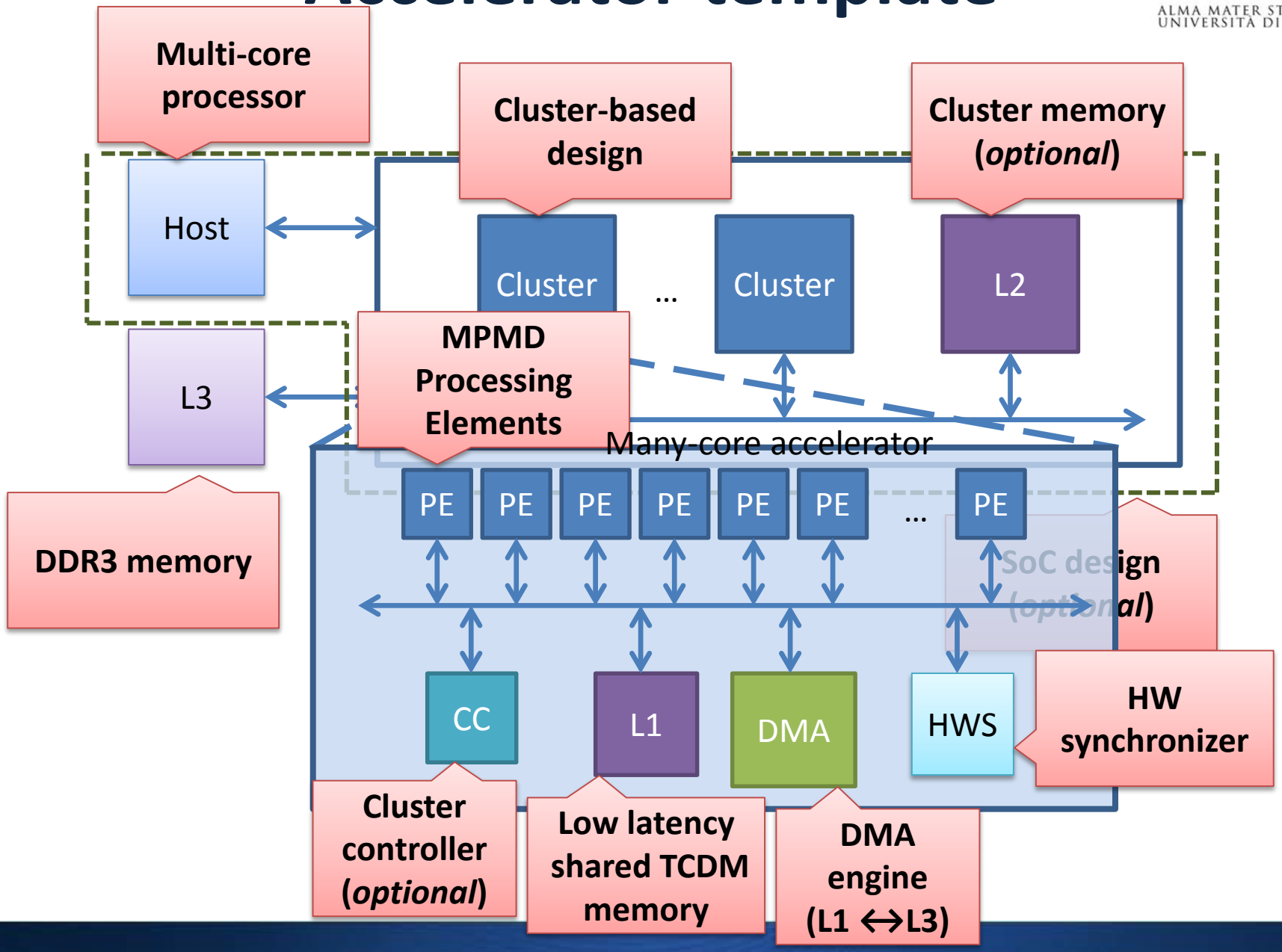
- **Introduction**
- OpenVX acceleration
- Work in progress

OpenVX overview



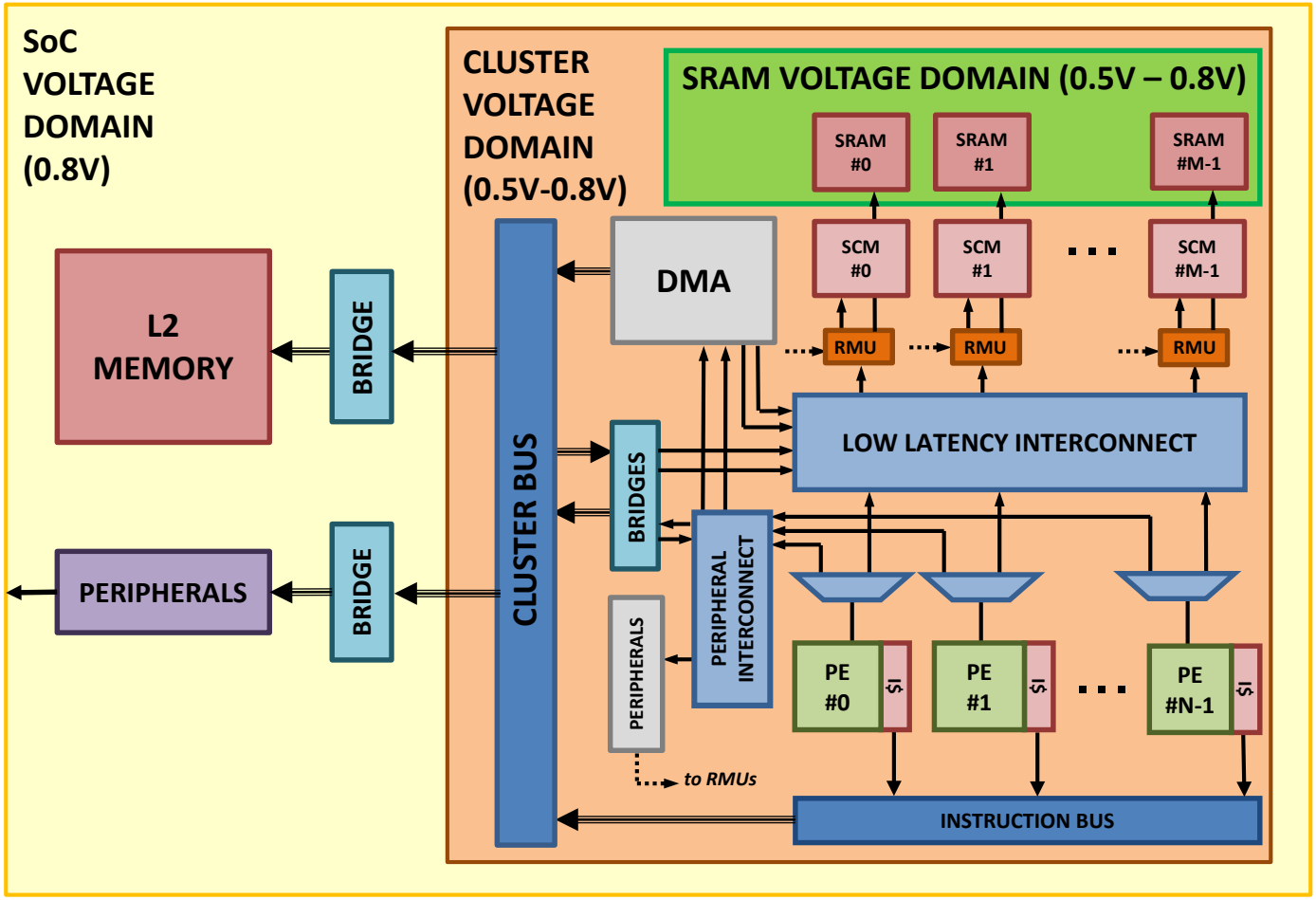
- Foundational API for vision acceleration
  - Focus on ***mobile and embedded systems***
- Stand-alone or complementary to other libraries
- Enable **efficient implementations** on different devices
  - CPUs, GPUs, DSPs, ***many-core accelerators***

Accelerator template



PULP

Parallel Ultra-Low-Power platform



OpenVX programming model

- The OpenVX model is based on a **directed acyclic graph** of nodes (***kernels***), with data (***images***) as linkage

```
vx_image imgs[] = {
  vxCreateImage(ctx, width, height, VX_DF_IMAGE_RGB),
  vxCreateVirtualImage(graph, 0, VX_DF_IMAGE_U8),
```

Virtual images are not required to actually reside in main memory
✓They define a data dependency between kernels, but they cannot be read/written

An OpenVX program must be verified to guarantee some mandatory properties:
✓Inputs and outputs compliant to the node interface
✓No cycles in the graph
✓Only a single writer node to any data object is allowed
✓Writes have higher priorities than reads.
✓Virtual image must be resolved into concrete types

```
vxVerifyGraph(graph);
vxProcessGraph(graph);
```

Outline

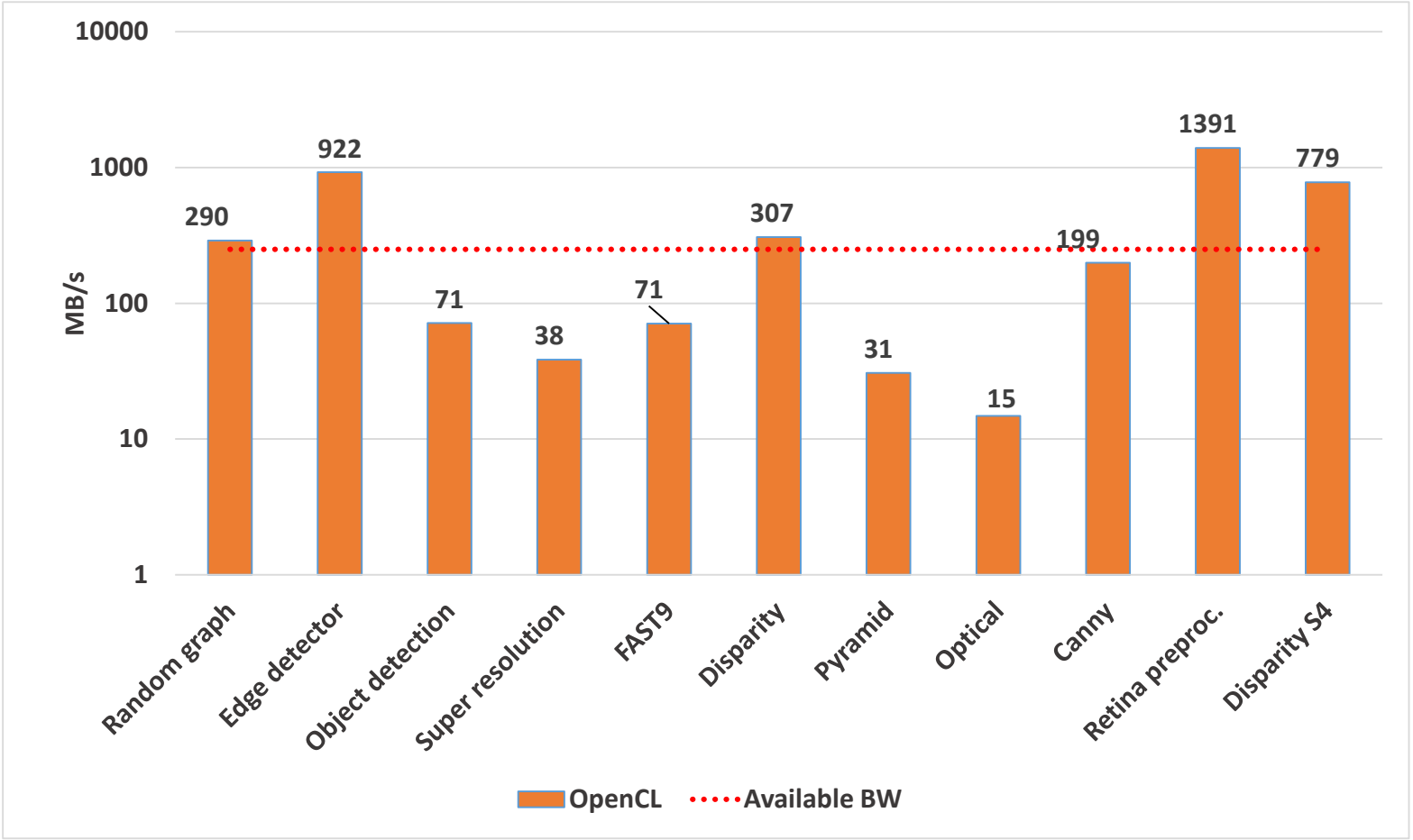
- Introduction
- **OpenVX acceleration**
- Work in progress

A first solution: using OpenCL to accelerate OpenVX kernels

- OpenCL is a commonly used programming model for many-core accelerators
- First solution: **OpenVX kernel == OpenCL kernel**
  - When a node is selected for execution, the related OpenCL kernel is enqueued on the device
- Main limiting factor**: memory bandwidth**

OpenCL bandwidth

- Experiments performed on the STHORM evaluation board



Our solution

- We realized an OpenVX framework for many-core accelerators coupling a **tiling approach** with algorithms for **graph partition and scheduling**
- Main goals:
  - **Reducing the memory bandwidth**
  - Maximize the accelerator efficiency
- Several steps are required:
  - **Tile size propagation**
  - **Graph partitioning**
  - **Node scheduling**
  - **Buffer allocation**
  - **Buffer sizing**

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Common access patterns for image processing kernels

(D) GLOBAL OPERATORS
Compute the value of a point in the output image using the whole input image
Support: ***Host exec / Graph partitioning***

(A) POINT OPERATORS
Compute the value of each output point from the corresponding input point

Support: ***Basic tiling***

(E) GEOMETRIC OPERATORS
Compute the value of a point in the output image using a non-rectangular input window
Support: ***Host exec / Graph partitioning***

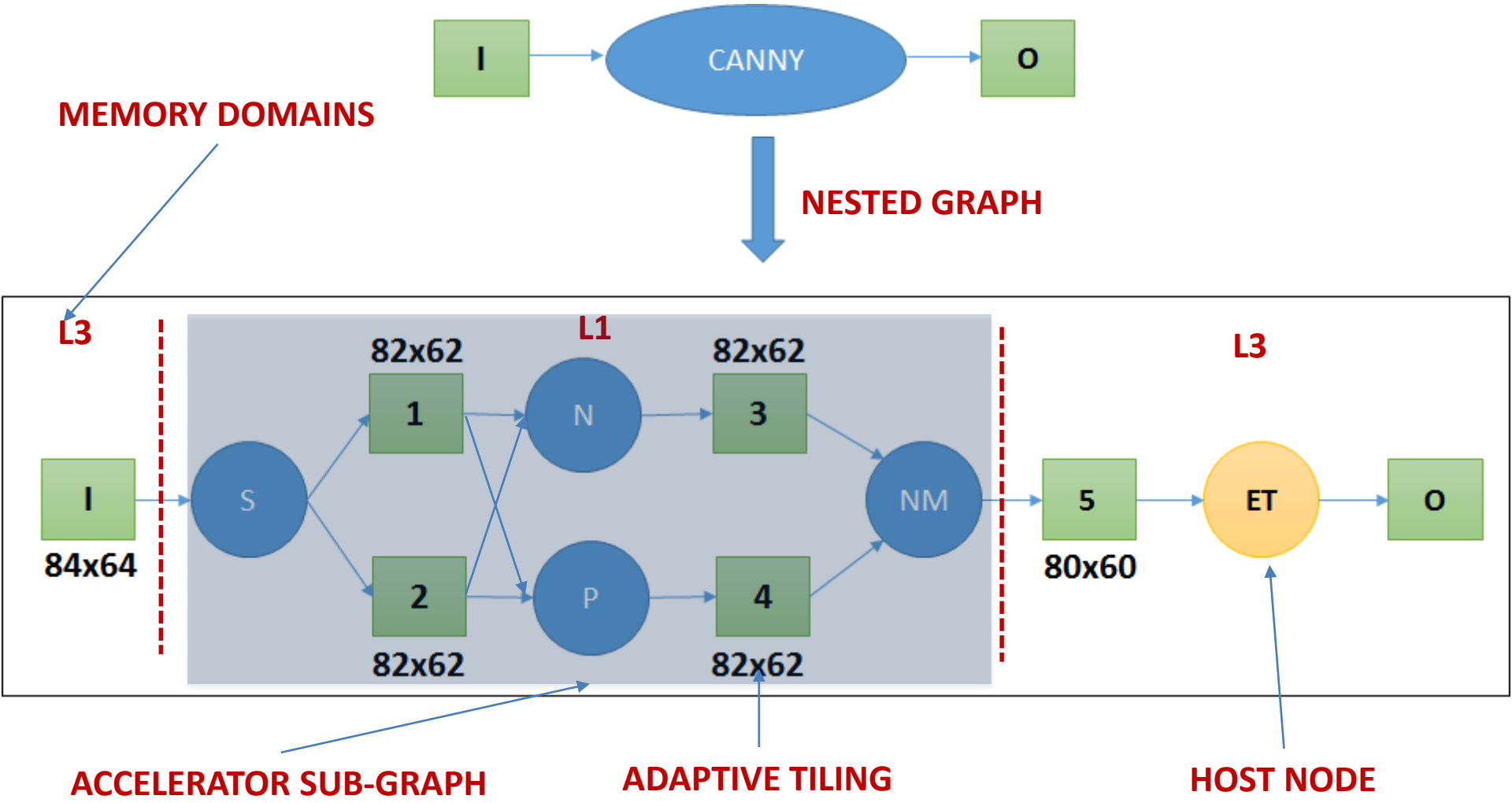
(B) LOCAL NEIGHBOR OPERATORS
Compute the value of a point in the output image that corresponds to the input window
Support: ***Tile overlapping***

(F) STATISTICAL OPERATORS
Compute any statistical functions of the image points

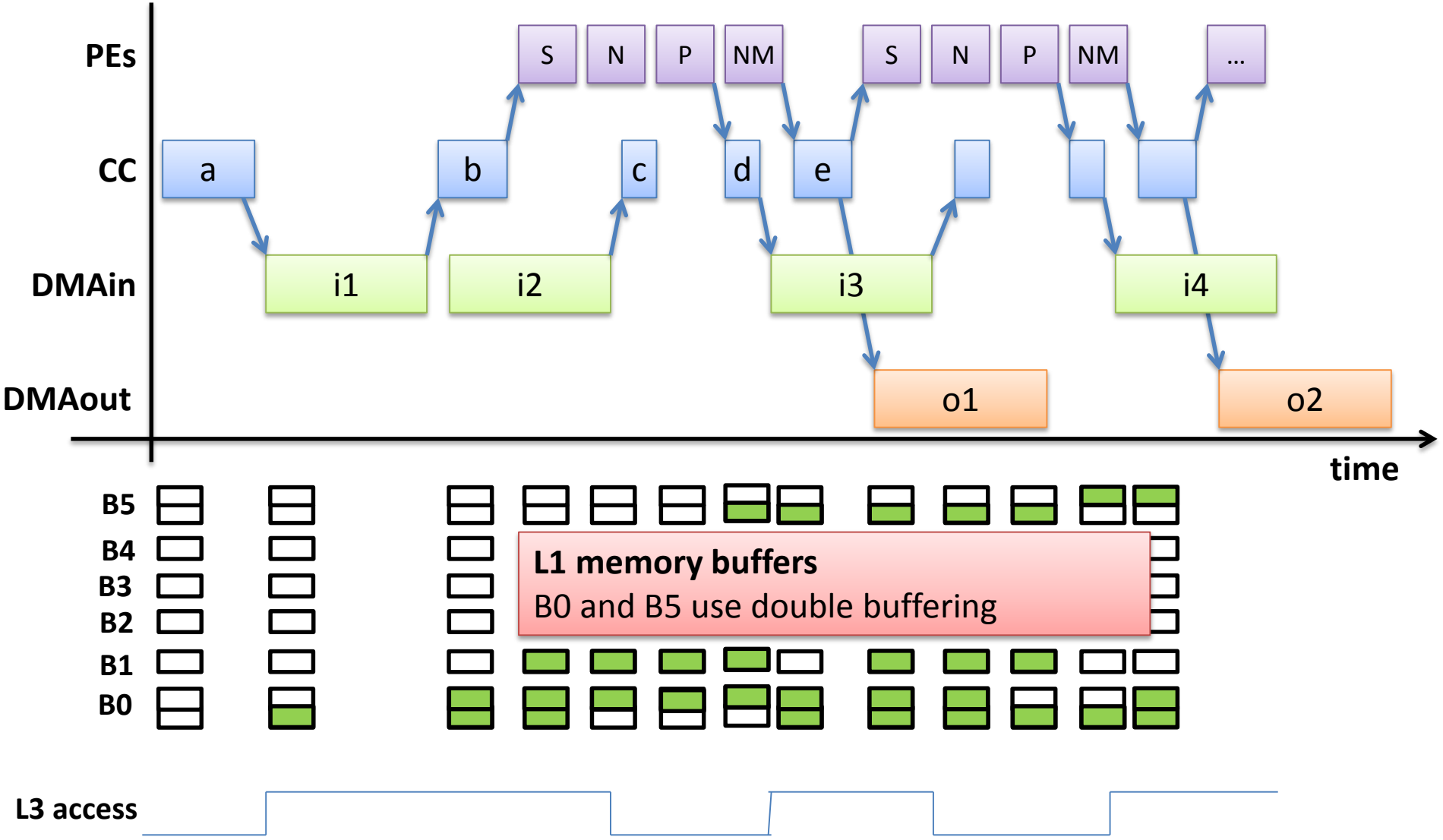
Support: ***Graph partitioning***

(C) RECURSIVE NEIGHBOR OPERATORS
Like the previous ones, but also consider the previously computed values in the output window
Support: ***Persistent buffer***

Example (1)


I
CANNY
O
MEMORY DOMAINS
NESTED GRAPH
L3
L1
L3
82x62
82x62
1
N
3
I
S
NM
5
ET
O
84x64
2
P
4
80x60
82x62
82x62
ACCELERATOR SUB-GRAPH
ADAPTIVE TILING
HOST NODE

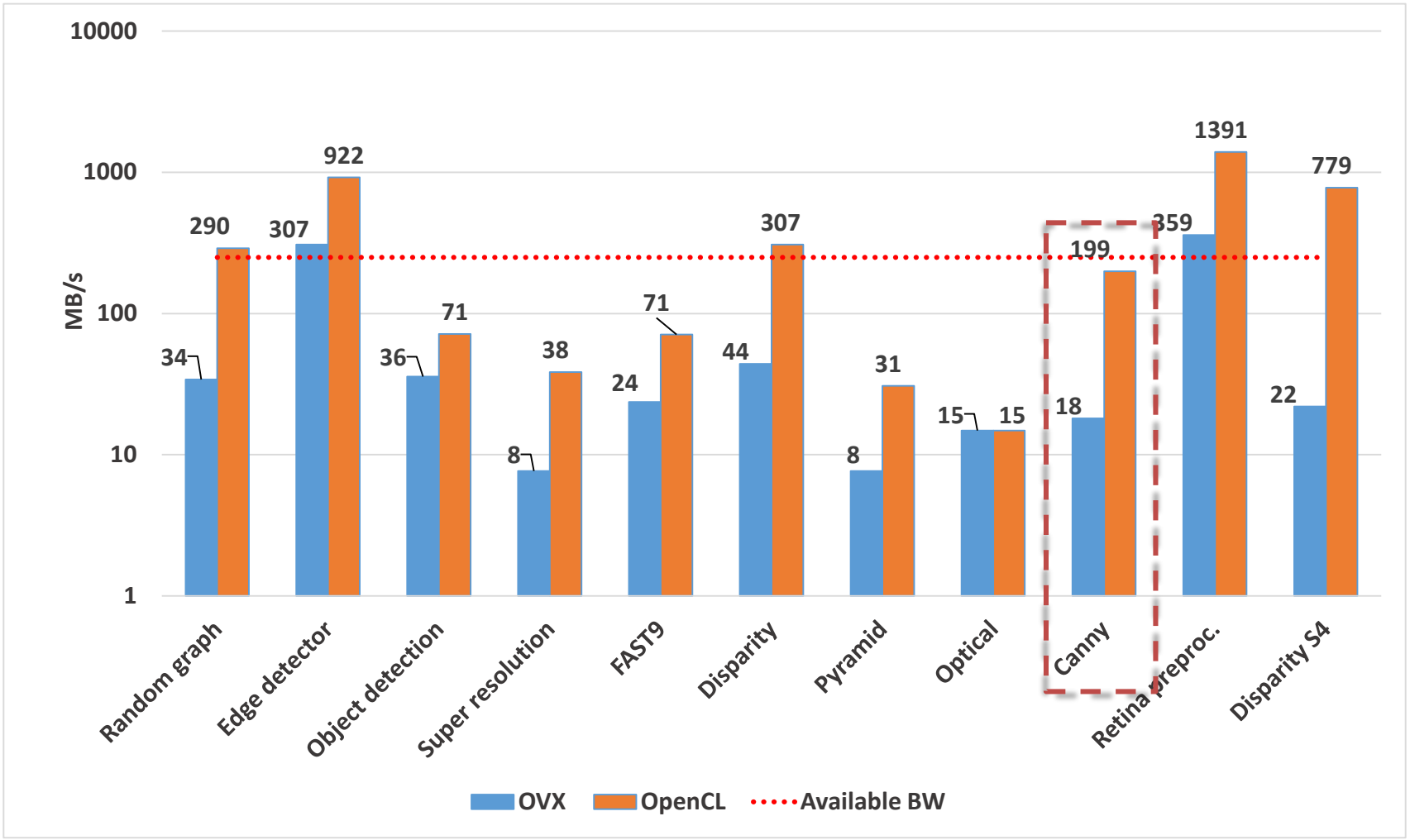
Example (2)


PEs
CC
DMAin
DMAout
S
N
P
NM
S
N
P
NM
...
a
b
c
d
e
i1
i2
i3
i4
o1
o2
time
B5
B4
B3
B2
B1
B0
L1 memory buffers
B0 and B5 use double buffering
L3 access

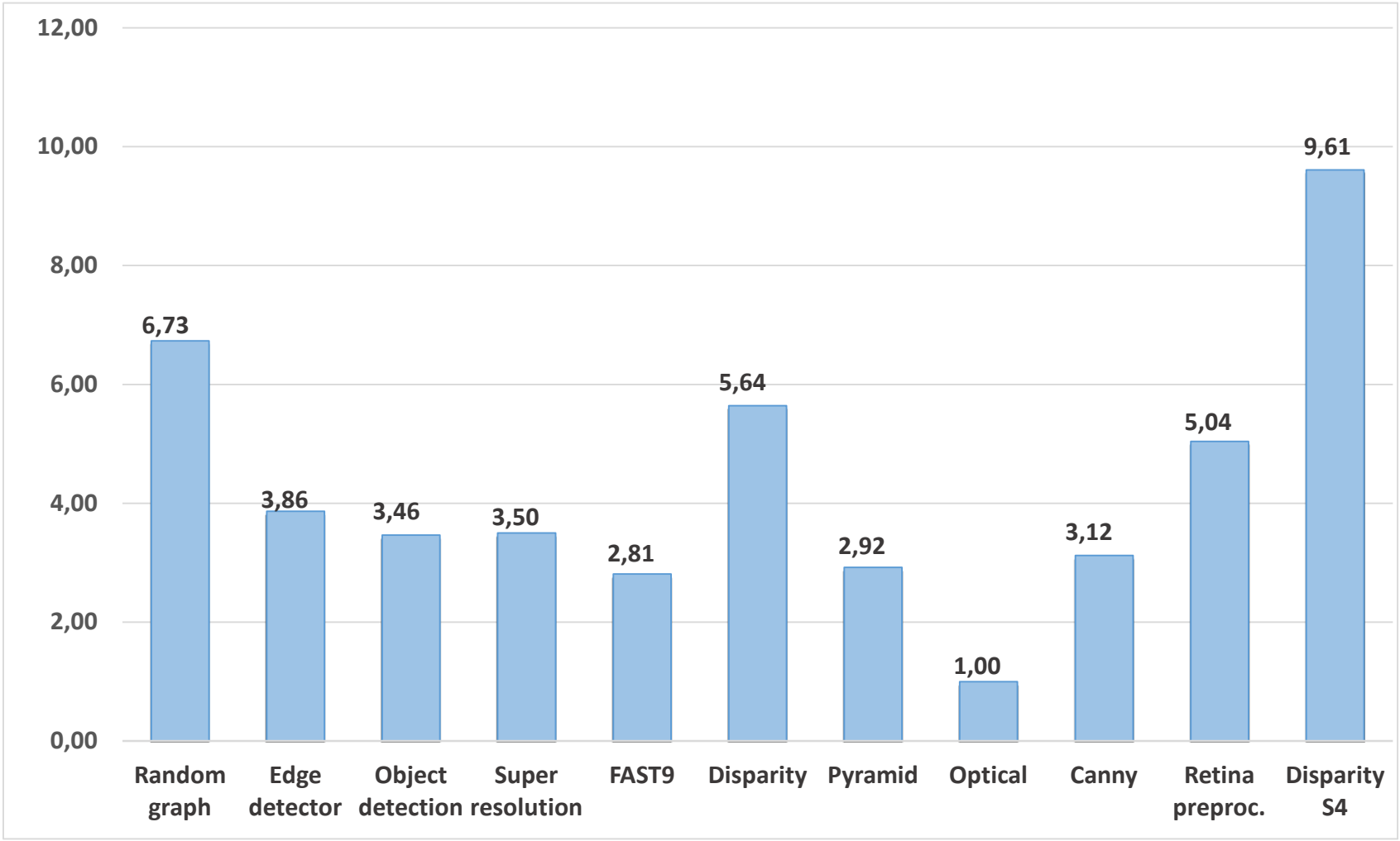
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Bandwidth reduction

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Speed-up w.r.t. OpenCL



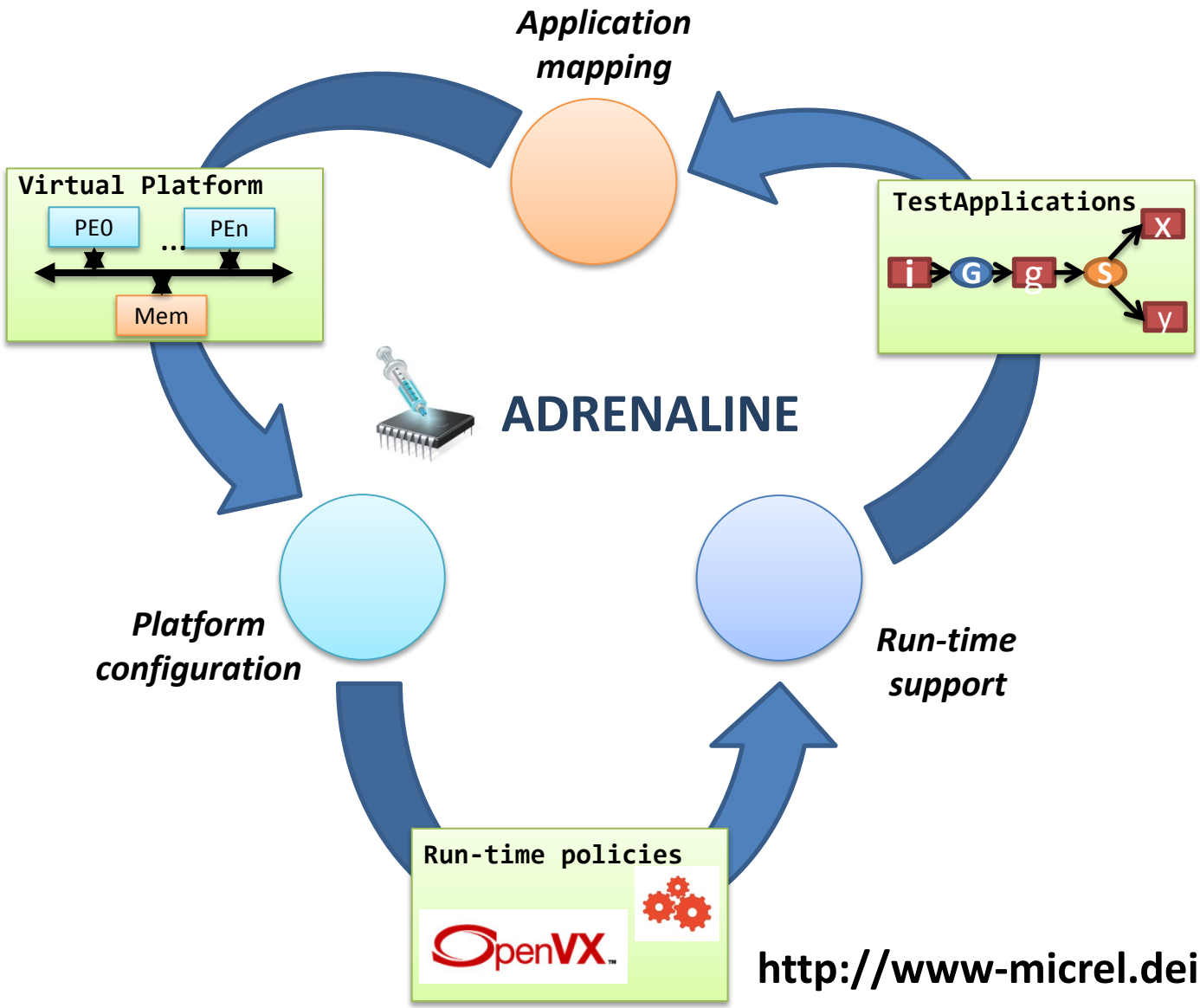
Outline

- Introduction
- OpenVX acceleration
- **Work in progress**

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OpenVX + Virtual Platform

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



http://www-micrel.deis.unibo.it/adrenaline/

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

THANKS!!!

Work supported by EU-funded research projects

PHIDIAS

P-SOCRATES