CHAIR FOR COMPILER CONSTRUCTION

cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Programming abstractions and optimizing compilers for energy-efficient computing

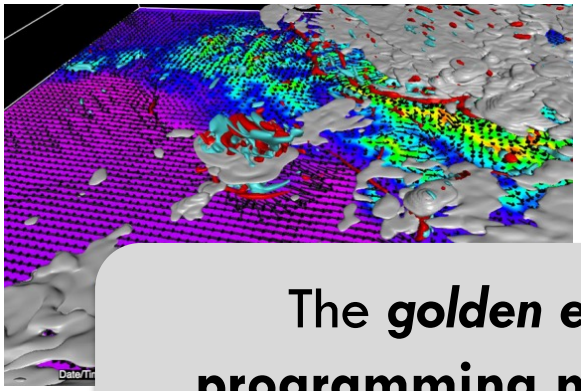
Jeronimo Castrillon

Chair for Compiler Construction (CCC)

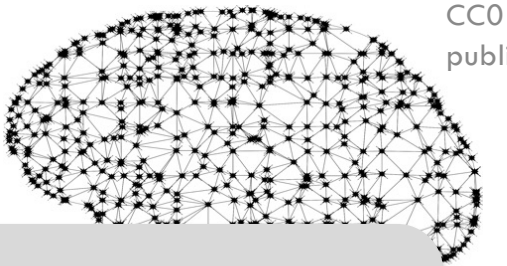
TU Dresden, Germany

1st Workshop on NetZero Carbon Computing (NetZero'23)
Montreal, Canada
February 25, 2023

Programming challenges



CIMA



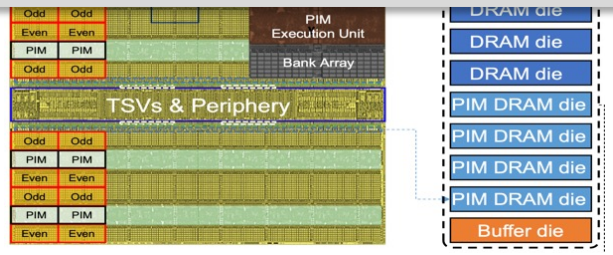
CC0
public domain

ML

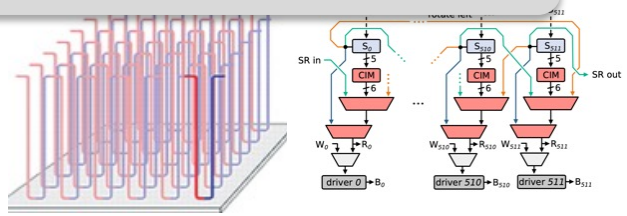
The ***golden era*** in computer architecture: Requires **next-gen programming methodologies** to **democratize** heterogeneous and emerging computing across domains

In this talk: Examples for energy-efficiency & (a bit of) life-time

https://www.hpcwire.com/2017/04/10/nvidia-responds-google-tpu-benchmarking/



Samsung @ ISCA 2021



2023

Semantic gap ➔ performance gap

$$v_{ijk,e} = \sum_{i'=0}^{p} \sum_{j'=0}^{p} \sum_{k'=0}^{p} A_{kk'} A_{jj'} A_{ii'} u_{i'j'k'e}$$

What we want

What we (naively) code

```c
void cfd_kernel(
  double A[restrict 7][7],
  double u[restrict 216][7][7][7],
  double v[restrict 216][7][7][7])
{
  /* element loop: */
  for(int e = 0; e < 216; e++) {
    for(int i0 = 0; i0 < 7; i0++) {
    for(int j0 = 0; j0 < 7; j0++) {
    for(int k0 = 0; k0 < 7; k0++) {
      v[e][i0][j0][k0] = 0.0;
      for(int i1 = 0; i1 < 7; i1++) {
      for(int j1 = 0; j1 < 7; j1++) {
      for(int k1 = 0; k1 < 7; k1++) {
        v[e][i0][j0][k0] += A[i0][i1]
                          * A[j0][j1]
                          * A[k0][k1]
                          * u[e][i1][j1][k1];
    } } } } } }
  } /* end of element loop */
}
```

100X

What performance experts code

```c
void cfd_kernel(
  double A[restrict 7][7],
  double u[restrict 216][7][7][7],
  double v[restrict 216][7][7][7])
{
  /* element loop: */
  #pragma omp for
  for (int e = 0; e < 216; e++) {
    double t6[7][7][7];
    /* 1st contraction: */
    #pragma simd
    for (int i0 = 0; i0 < 7; i0++) {
    for (int i1 = 0; i1 < 7; i1++) {
    /* #pragma simd */
    for (int i2 = 0; i2 < 7; i2++) {
      double t8 = 0.0;
      for (int i3 = 0; i3 < 7; i3++)
        t8 += A[i0][i3] * u[e][i1][i2][i3];
      t6[i0][i1][i2] = t8;
    } } } /* end of 1st contraction */
    double t7[7][7][7];
    /* 2nd contraction: */
    #pragma simd
    for (int i4 = 0; i4 < 7; i4++) {
    for (int i5 = 0; i5 < 7; i5++) {
    /* #pragma simd */
    for (int i6 = 0; i6 < 7; i6++) {
      double t9 = 0.0;
      for (int i7 = 0; i7 < 7; i7++)
        t9 += A[i4][i7] * t6[i5][i6][i7];
      t7[i4][i5][i6] = t9;
    } } } /* end of 2nd contraction */
    /* 3rd contraction: */
    #pragma simd
    for (int i8 = 0; i8 < 7; i8++) {
    for (int i9 = 0; i9 < 7; i9++) {
    /* #pragma simd */
    for (int i10 = 0; i10 < 7; i10++) {
      double t10 = 0.0;
      for (int i11 = 0; i11 < 7; i11++)
        t10 += A[i8][i11] * t7[i9][i10][i11];
      v[e][i8][i9][i10] = t10;
    } } } /* end of third contraction */
  } /* end of element loop */
```

Semantic gap ➔ performance gap

$$v_{ijk,e} = \sum_{i'=0}^{p} \sum_{j'=0}^{p} \sum_{k'=0}^{p} A_{kk'} A_{jj'} A_{ii'} u_{i'j'k'e}$$

What we want

```
void cfd_kernel(
  double A[restrict 7][7],
  double u[restrict 216][7][7][7],
  double v[restrict 216][7][7][7])
{
  /* element loop: */
  for(int e = 0; e < 216; e++) {
    for(int i0 = 0; i0 < 7; i0++) {
    for(int j0 = 0; j0 < 7; j0++) {
    for(int k0 = 0; k0 < 7; k0++) {
      v[e][i0][j0][k0] = 0.0;
      for(int i1 = 0; i1 < 7; i1++)
      for(int j1 = 0; j1 < 7; j1++) {
      for(int k1 = 0; k1 < 7; k1++) {
        v[e][i0][j1][k1] += A[i0][i1]
                          * A[j0][j1]
                          * A[k0][k1]
                          * u[e][i1][j1][k1];
}
```

???

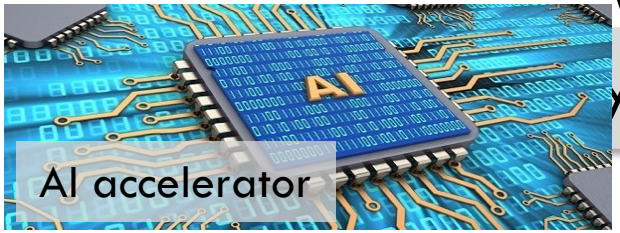
???

???

100X

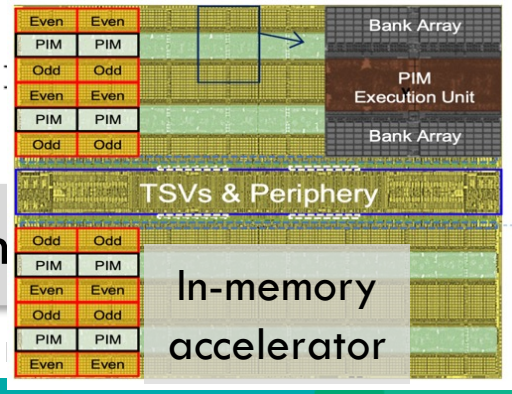
```
void cfd_kernel(
  double A[restrict 7][7],
  double u[restrict 216][7][7][7],
  double v[restrict 216][7][7][7])
{
  /* element loop: */
  #pragma omp for
  for (int e = 0; e < 216; e++) {
    double t6[7][7][7];
    /* 1st contraction: */
    #pragma simd
    for (int i0 = 0; i0 < 7; i0++) {
    for (int i1 = 0; i1 < 7; i1++) {
    /* #pragma simd */
    for (int i2 = 0; i2 < 7; i2++) {
      double t8 = 0.0;
      for (int i3 = 0; i3 < 7; i3++)
        t8 += A[i0][i3] * u[e][i1][i2][i3];
      t6[i0][i1][i2] = t8;
    } } } /* end of 1st contraction */
```

```
    double t10 = 0.0;
    for (int i11 = 0; i11 < 7; i11++)
      t10 += A[i8][i11] * t7[i9][i10][i11];
    v[e][i8][i9][i10] = t10;
  } } } /* end of third contraction */
} /* end of element loop */
```



AI accelerator

https://www.hpcwire.com/2017/04/10/nvidia-responds-google-tpu-benchmarking/



In-memory accelerator

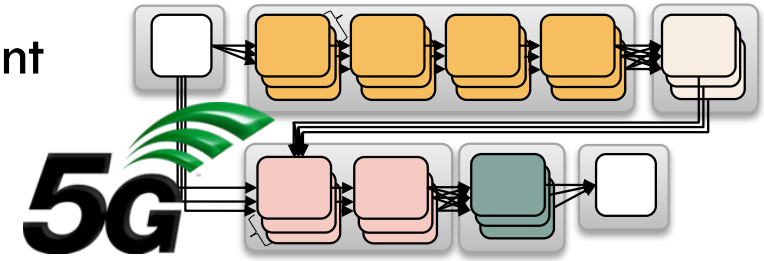
HBM-FPGA

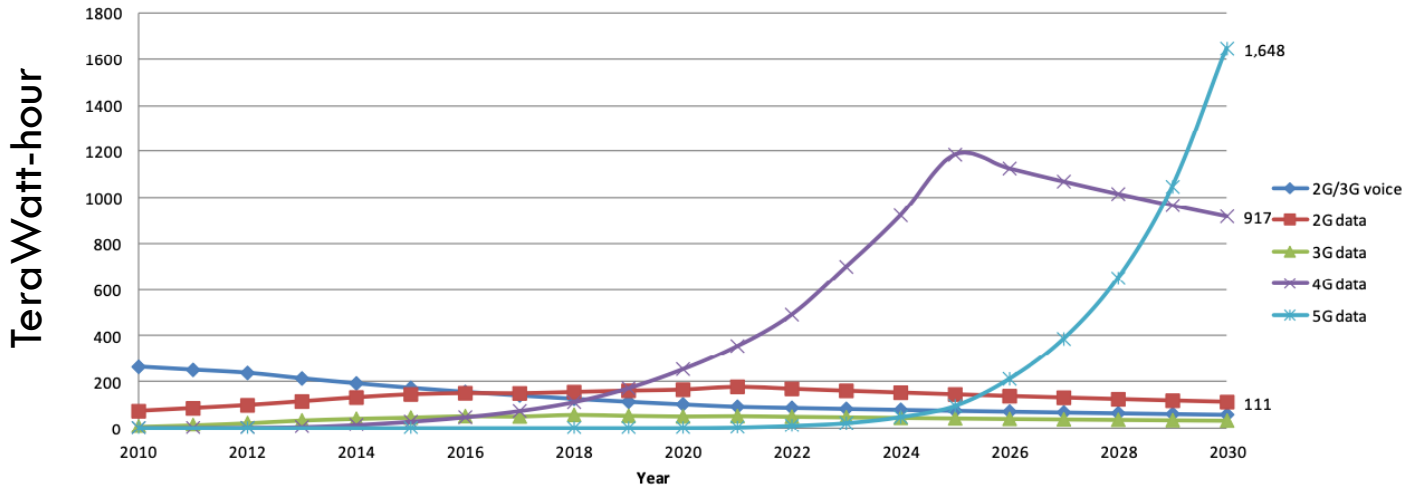
Lee, Sukhan, et al. "Hardware Architecture and Software Stack for PIM Based on Commercial DRAM Technology: Industrial Product." ISCA 2021.

Wh… …rts code

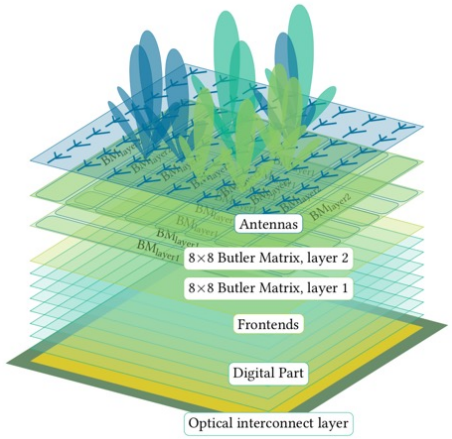
Example 1: 5G baseband processing

- E4C collaborative project: Extreme energy-efficient cloud radio access network
  - **Domain-specific modeling of comm. workload**
  - Novel server architecture (links, PCB, …)



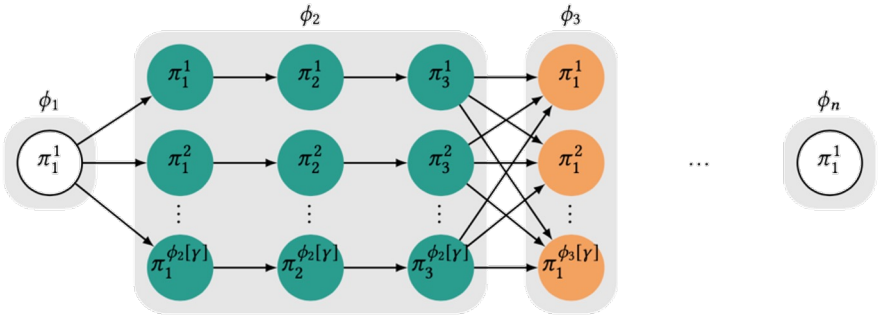


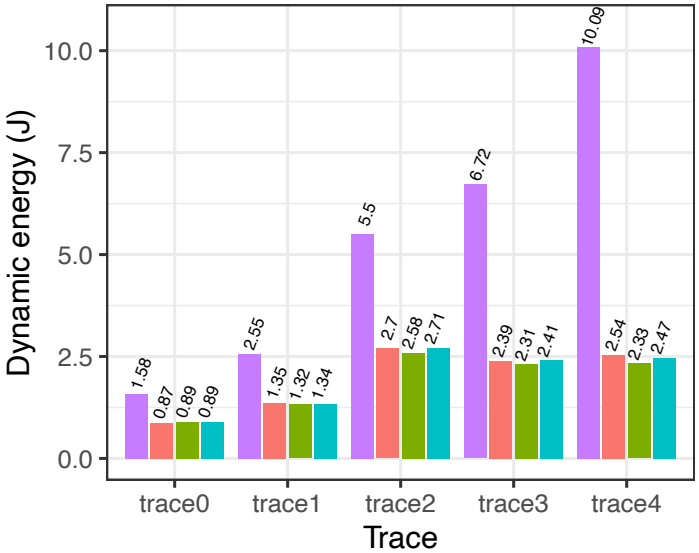
Andrae, et al. On Global Electricity Usage of Communication Technology: Trends to 2030. Challenges 2015



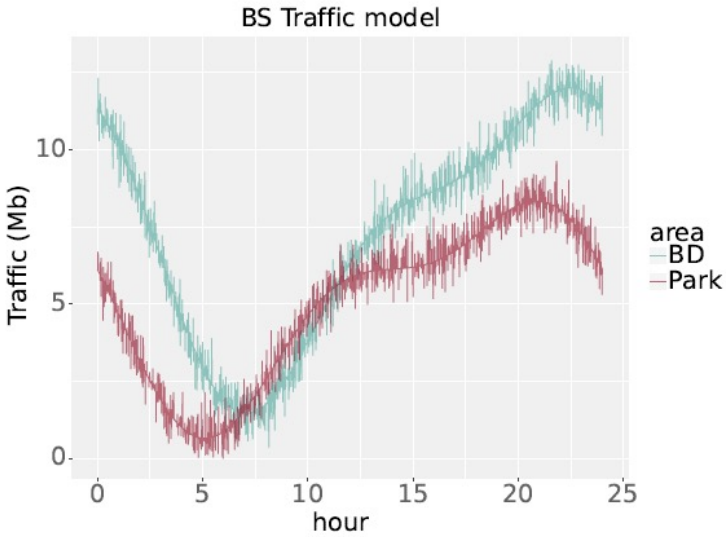
Domain-specific knowledge to scale up

- Dynamic dataflow model of computation
- Domain-specific global optimization
- Traffic models for different scenarios

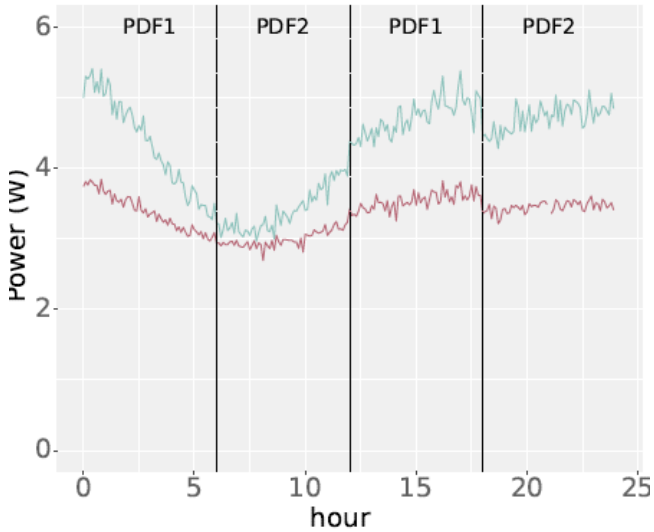




Khasanov, ACM TECS 2021



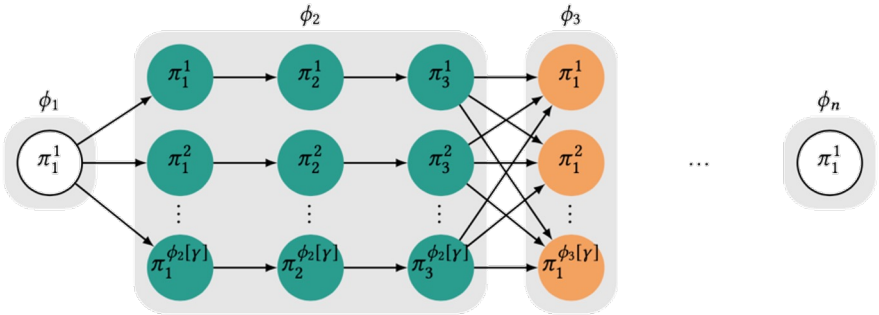
Robledo, MCSoC'22

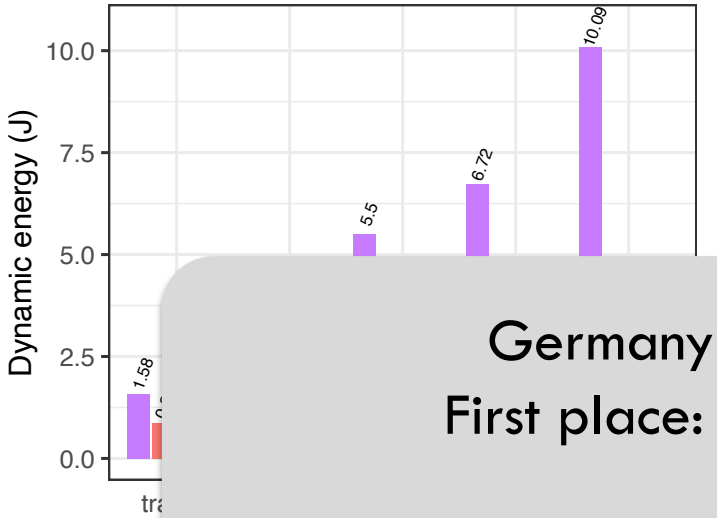


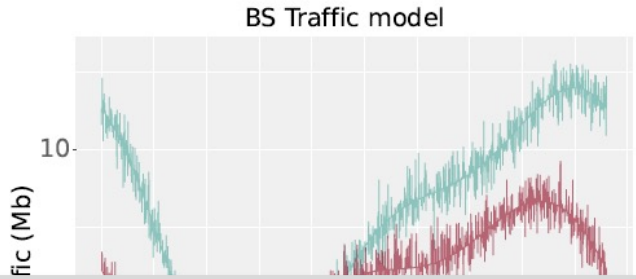
Robledo, MCSoC'22

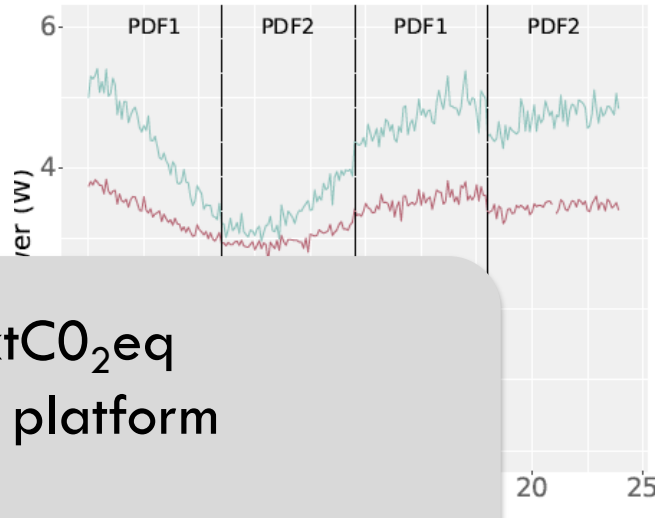
Domain-specific knowledge to scale up

- Dynamic dataflow model of computation
- Domain-specific global optimization
- Traffic models for different scenarios









Germany-wide what-if analysis: ~400 kt$C0_2$eq

First place: BMBF competition in green ICT platform

https://bit.ly/3LXdkSo

Khasanov, ACM TECS 2021

Example 2: HPC physics simulation

- Domain-specific languages (DSLs)
  - Code closer to what we want
  - Suitable abstractions ➔ powerful optimization
- MLIR-based compilation

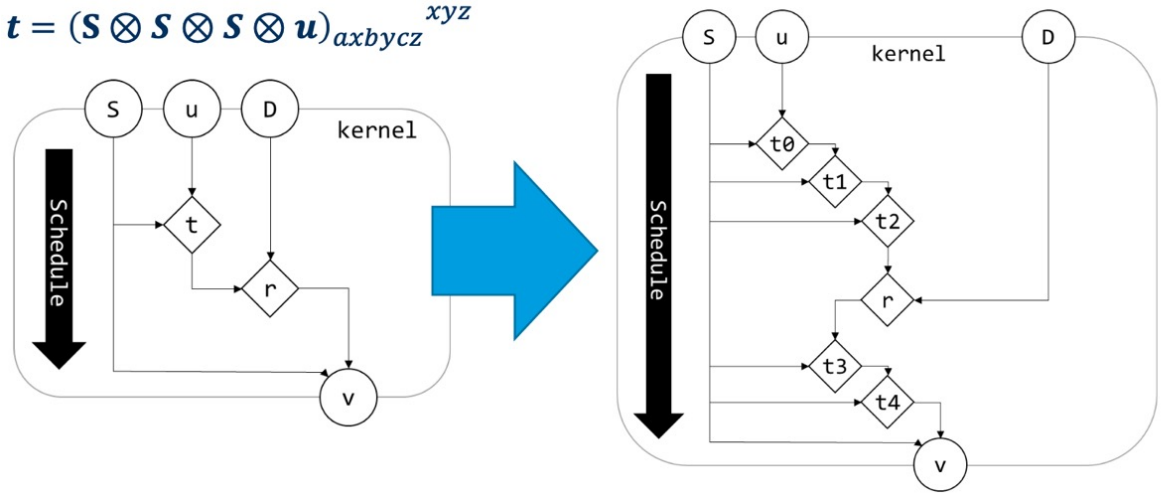
$$v_{ijk,e} = \sum_{i'=0}^{p} \sum_{j'=0}^{p} \sum_{k'=0}^{p} A_{kk'} A_{jj'} A_{ii'} u_{i'j'k'e}$$

What we want

```
source =  ...
var input A    : matrix          &
var input u    : tensorIN        &
var input output v  : tensorOUT  &
var input alpha : []             &
var input beta  : []             &
v = alpha * (A # A # A # u .
    [[5 8] [3 7] [1 6]]) + beta * v
```

$$t = \left(S \otimes \left(S \otimes \left(S \otimes u\right)_{cz}{}^{xyz}\right)_{by}{}^{cxy}\right)_{ax}{}^{bcx}$$

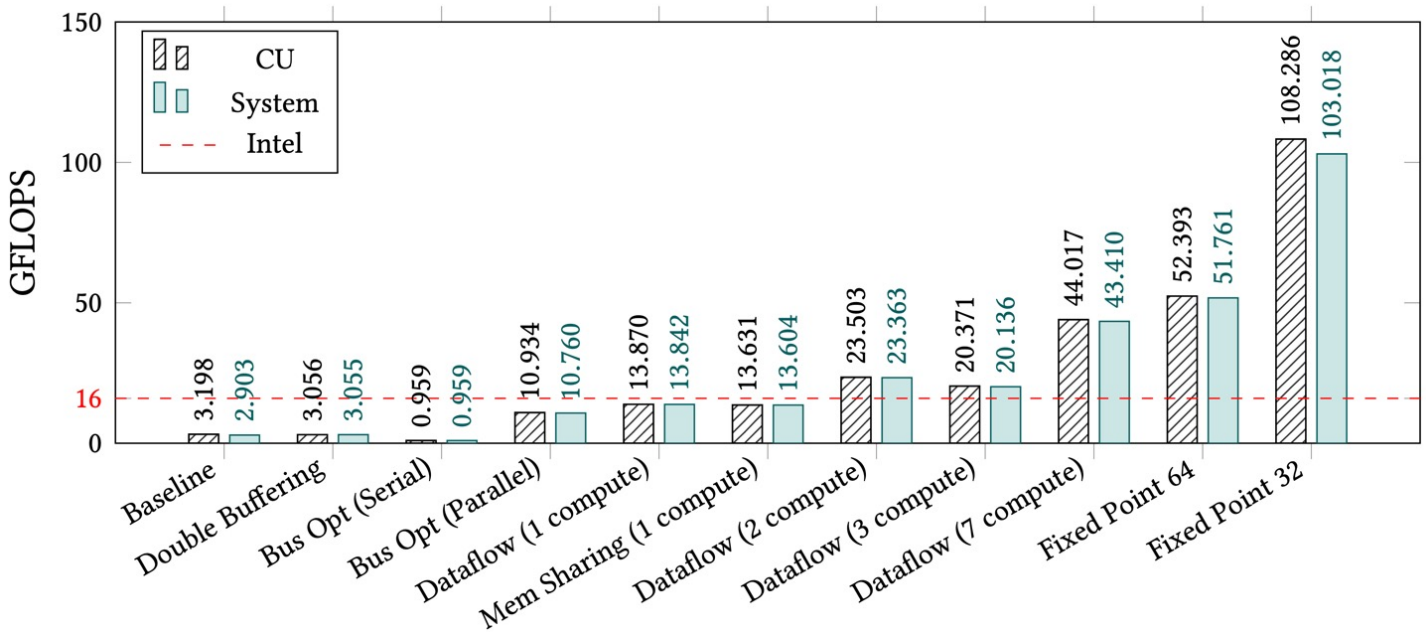
$$t = \left(S \otimes S \otimes S \otimes u\right)_{axbycz}{}^{xyz}$$



Rink, RDWSL'18
Susungi, GPCE'18
Rink, Array'19

FPGA code generation: HBM FPGA

- H2020 EU Project: Convergence HPC, Big Data and ML
- Transformations for a **17x speedup** (same precision)



S. Soldavini, ACM TRETS'22

EVEREST

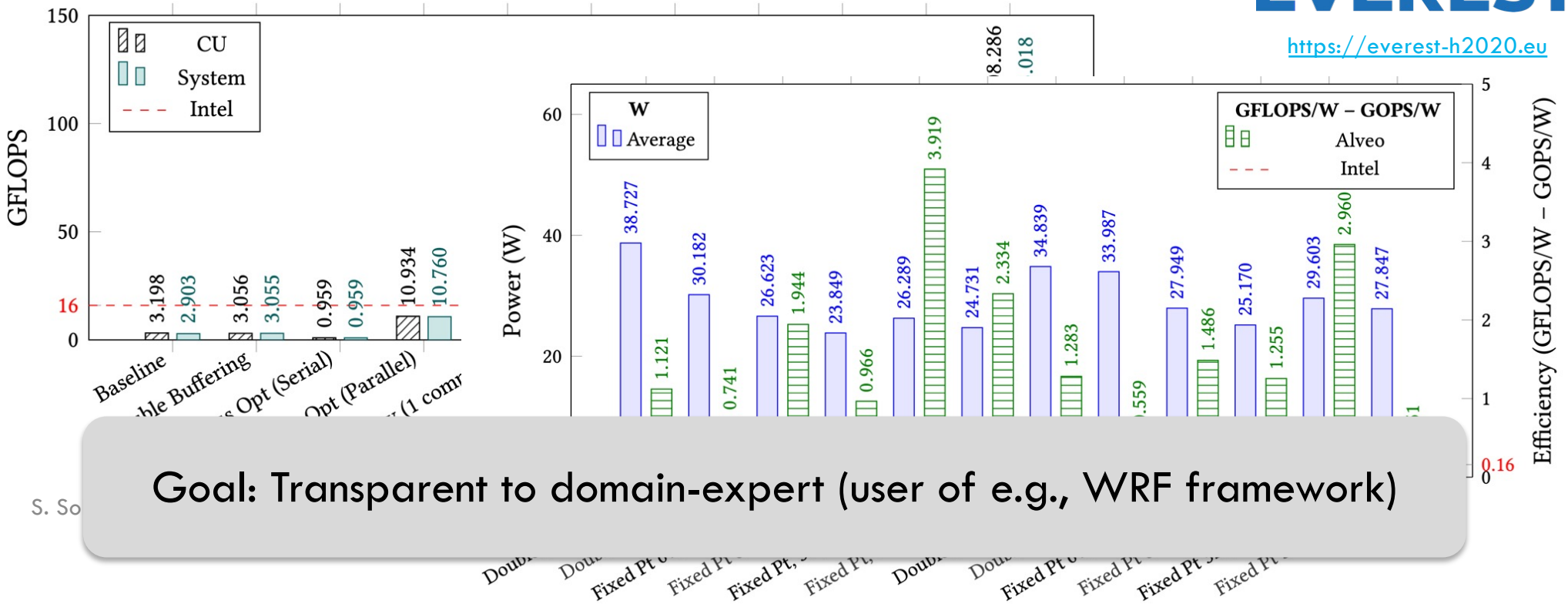
https://everest-h2020.eu

CHAIR FOR COMPILER CONSTRUCTION

FPGA code generation: HBM FPGA

- H2020 EU Project: Convergence HPC, Big Data and ML
- Variants with up to **24x better energy efficiency**

https://everest-h2020.eu

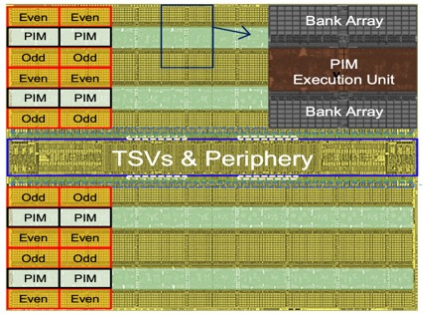


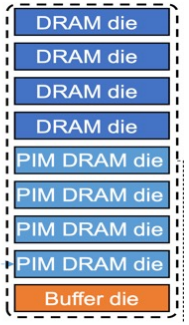
Goal: Transparent to domain-expert (user of e.g., WRF framework)

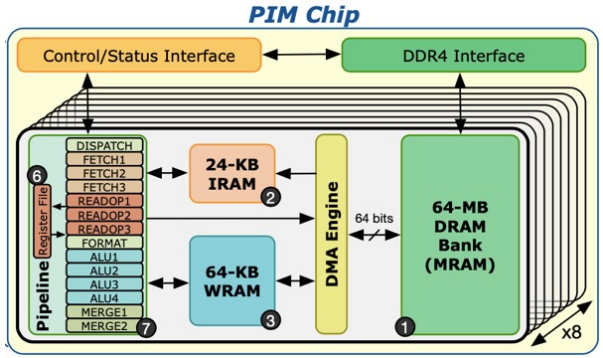
Example 3: Emerging data-centric architectures

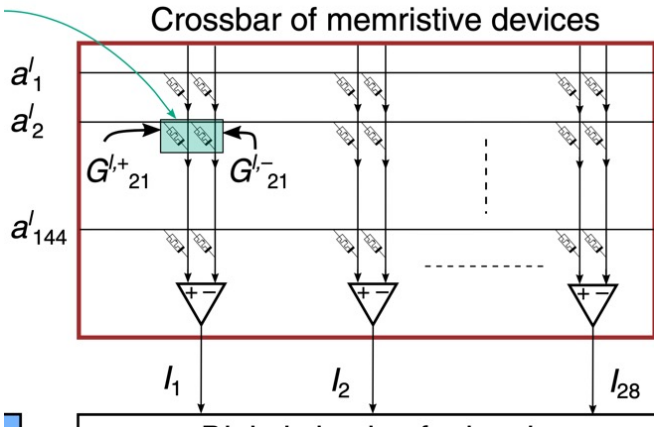
- Compute in-place, avoid data movement, transformations to match primitives
- Novel architectures for near-memory and in-memory computing

Samsung, Lee, Sukhan, et al. ISCA 2021
UPMEM by Gómez-Luna, Juan, et al. arXiv:2105.03814 (2021)
In-PCM Computing: Joshi, Vinay, et al. Nature Communications 11.1 (2020): 1-13.



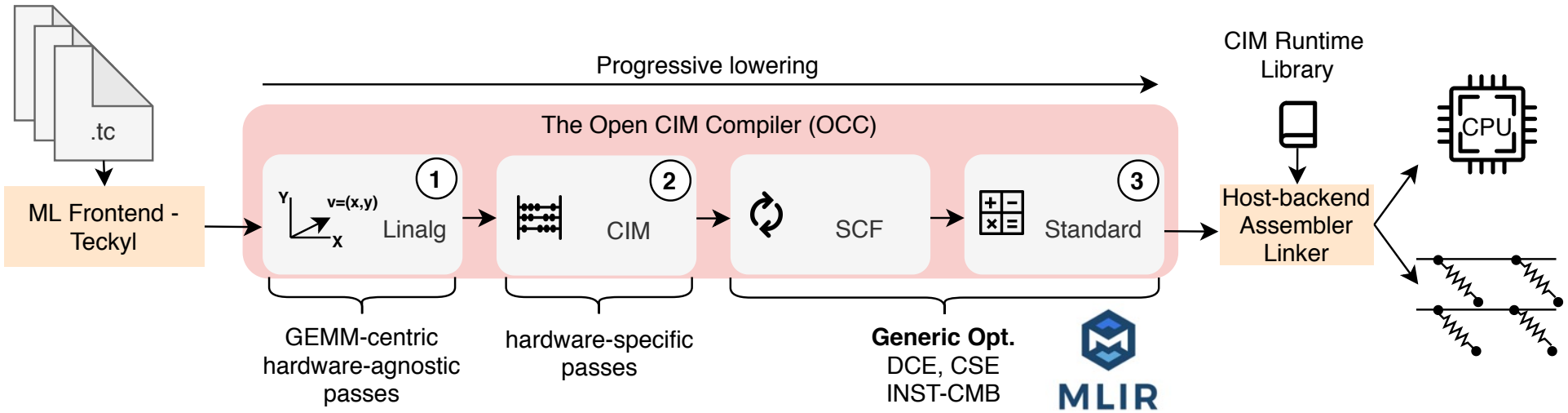






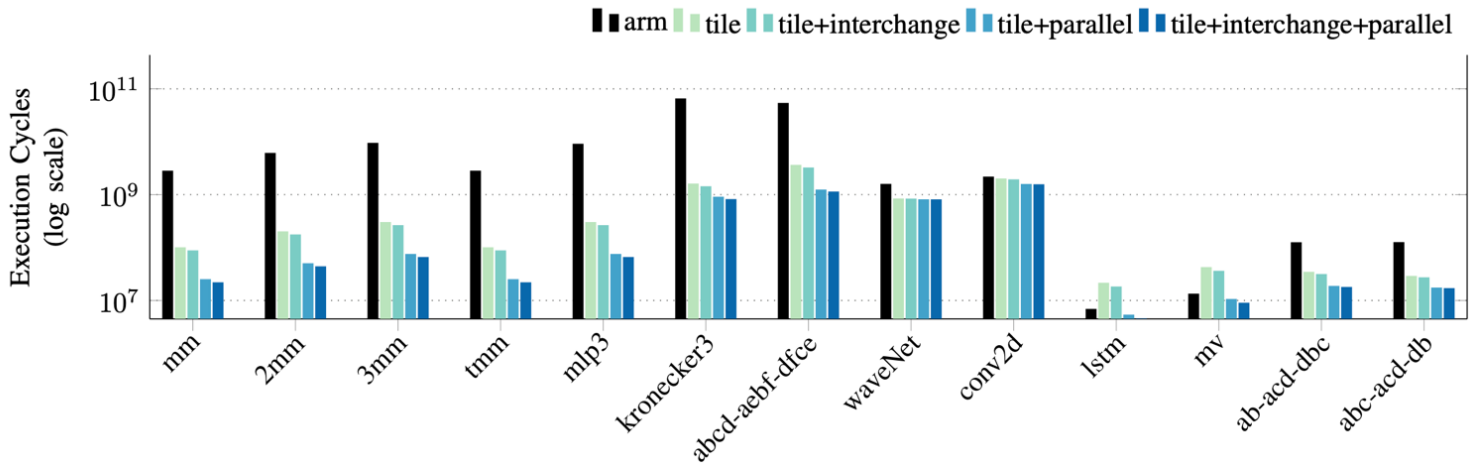
End-to-end compiler for PCM acceleration

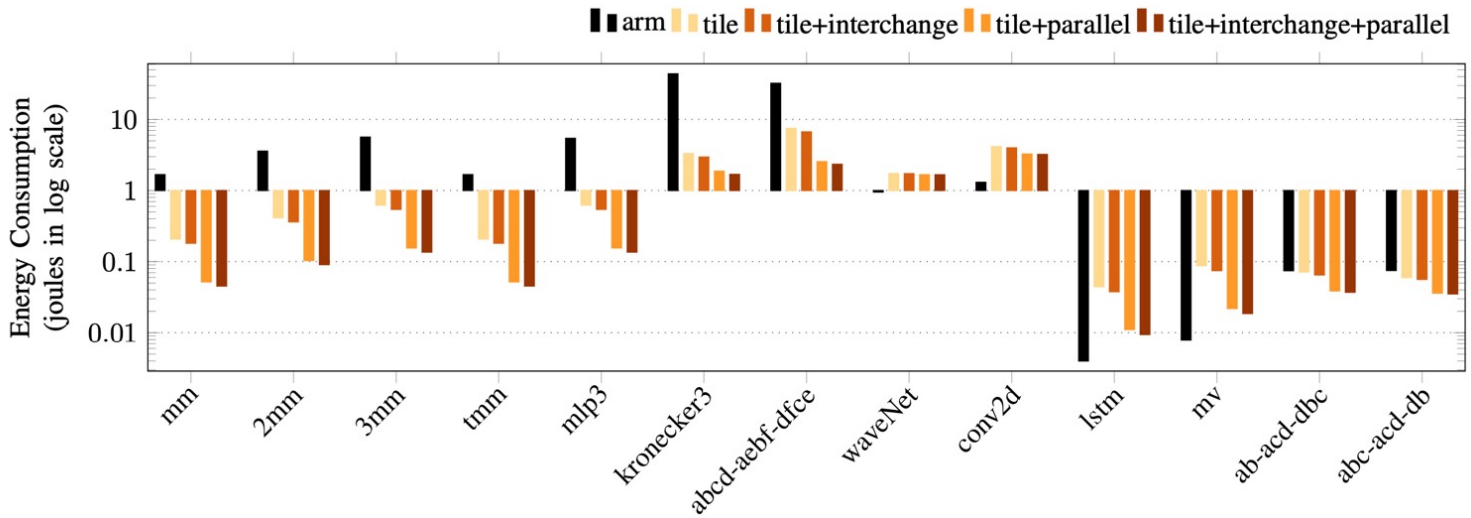
- MLIR frontend for general tensor expressions
  - Reuse GEMM transformations from **linalg** (in MLIR)
  - Transformations for **performance** and **wear-leveling**



A. Siemieniuk, IEEE TCAD 2021

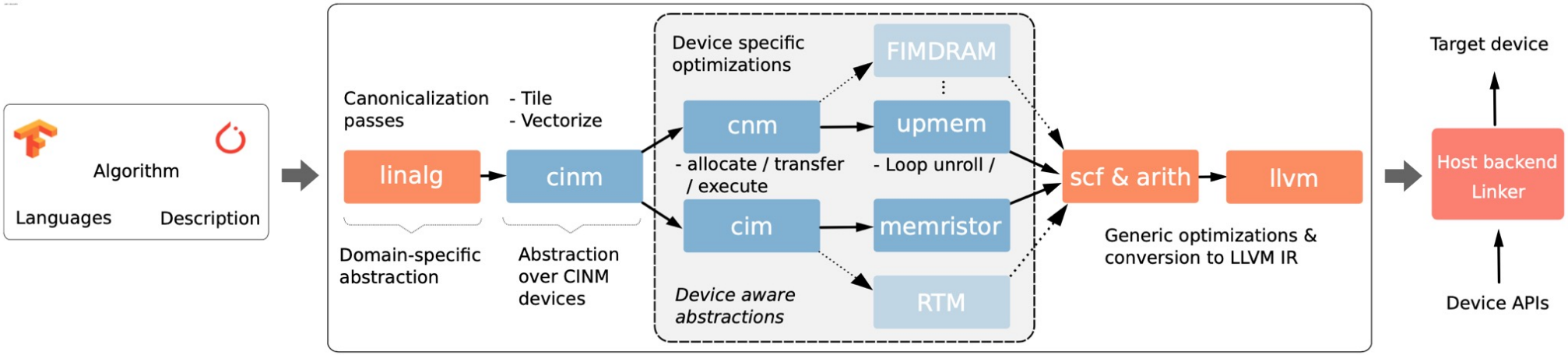
Optimization results: Beyond Matmult





Generalized MLIR infrastructure

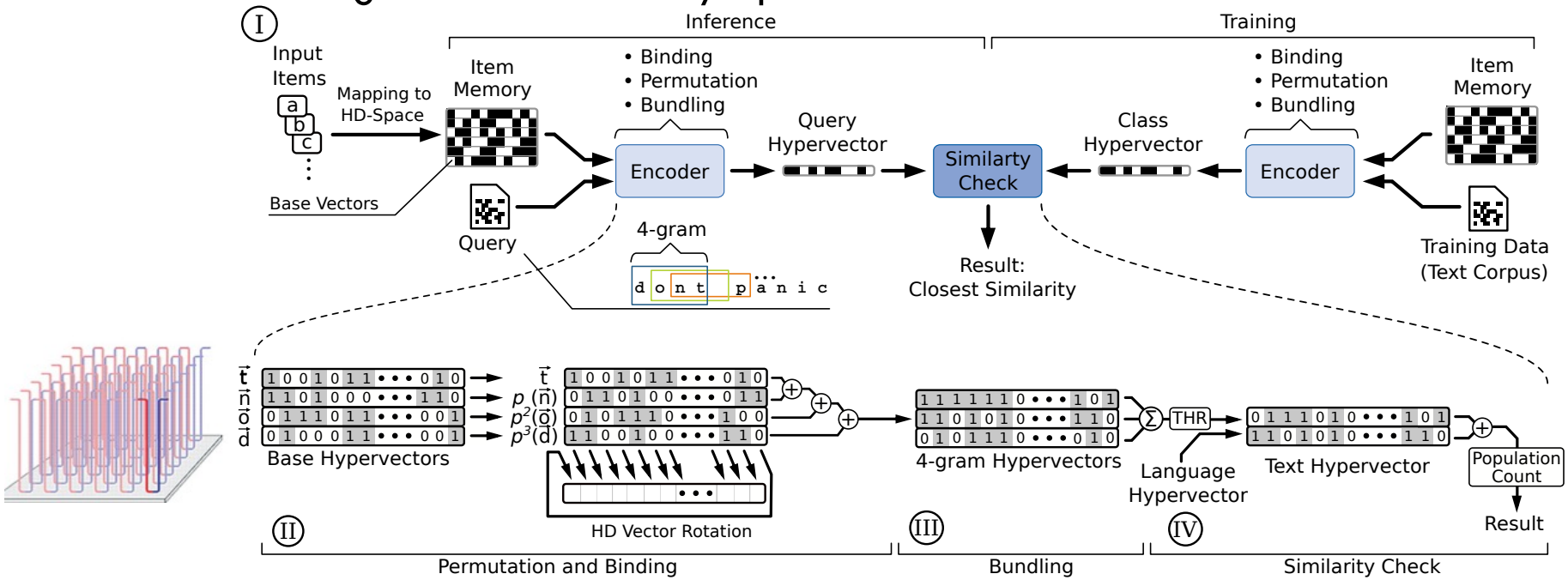
- Entry: linear algebra abstraction (common to ML frameworks and beyond)
- Intermediate languages for **in and near memory computing**
- Target-specific models and optimizations



A. Khan, Arxiv, 2023

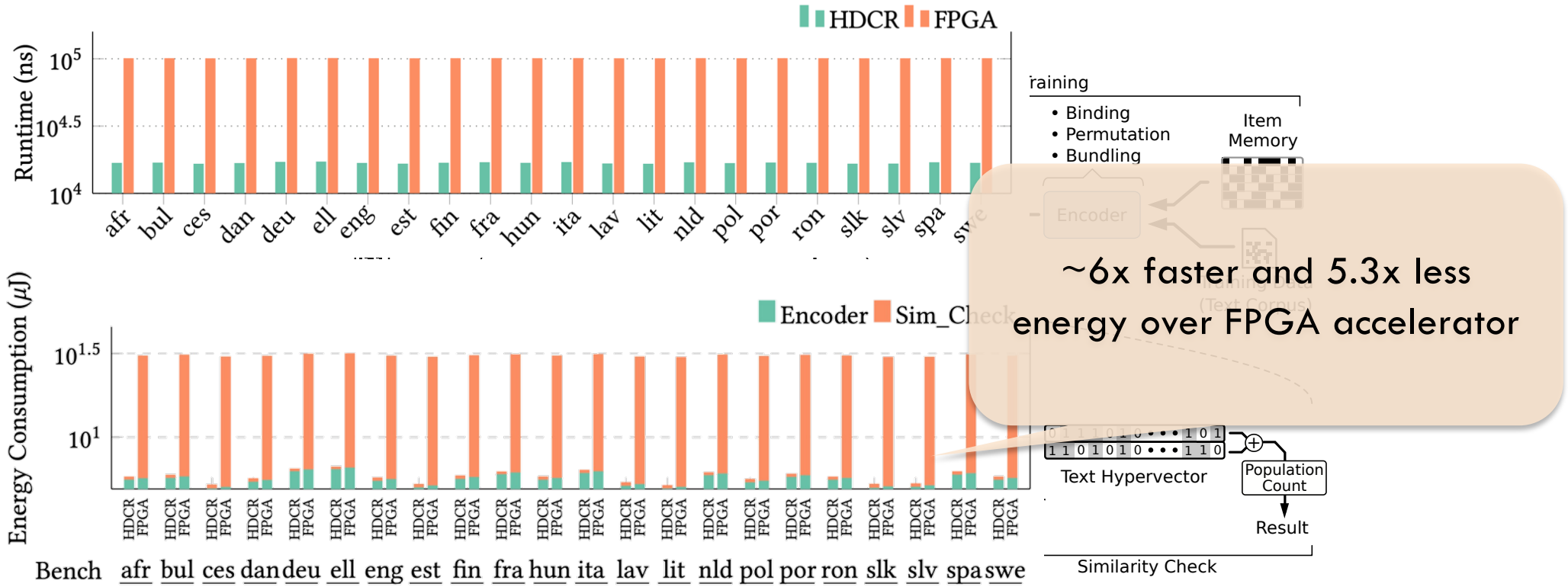
Hyper dimensional computing (HDC)

- HDC: Embed data in 10 k-dimensions – Von-Neuman Bottleneck!
  - Leverage bulk-wise binary operations



Khan, A. A., et al. ACM TECS 2022

Hyper dimensional computing (HDC)

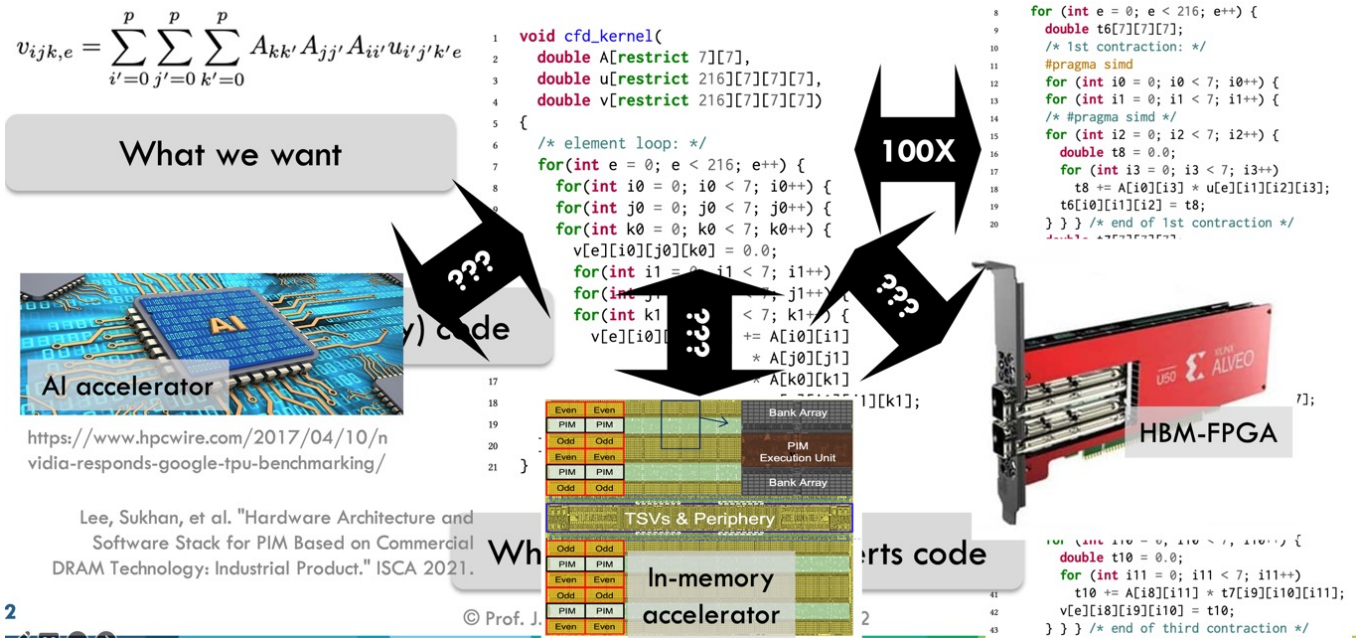


~6x faster and 5.3x less energy over FPGA accelerator

Khan, A. A., et al. ACM TECS 2022

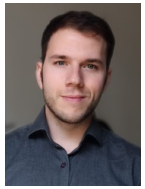
Summary

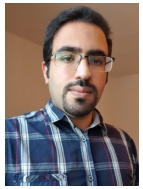
- Challenging & exciting computing landscape!
- Tools automation: Need better understanding of full-life-cycle sustainability aspects, beyond energy-efficiency

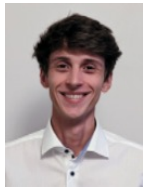


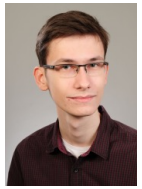
Thanks! & Acknowledgements


Hasna Bouraoui


João P. de Lima


Hamid Farzaneh


Clément Fournier


Karl Friebel


Dr. Asif Khan


Robert Khasanov


Alexander Brauckmann


Nesrine Khouzami


Dr. Steffen Köhler


Christian Menard


Julian Robledo


Lars Schütze


Felix Wittwer


Dr. Fazal Hameed

Funded by

DFG Deutsche Forschungsgemeinschaft

Bundesministerium für Bildung und Forschung

STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS

Freistaat SACHSEN

..., and previous members of the group (Andres Goens, Norman Rink, Sven Karol, Sebastian Ertel), and collaborators (J. Fröhlich, I. Sbalzarini, Alex K. Jones, A. Cohen, T. Grosser, T. Hoefler, H. Härtig, H. Corporaal, C. Pilato, S. Parkin, P. Jääskeläinen, K-H Chen)

CHAIR FOR COMPILER CONSTRUCTION