TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

OctopuScheduler: On-Chip DNN Scheduling on the SpiNNaker2 Neuromorphic MPSoC

Tim Langer*, Matthias Jobst*†, Chen Liu*, Florian Kelber*, Bernhard Vogginger*, Christian Mayr*†‡
*Chair of Highly-Parallel VLSI Systems and Neuro-Microelectronics, TU Dresden, Germany
†Centre for Tactile Internet with Human-in-the-Loop (CeTI), TU Dresden, Germany ‡ScaDS.AI Dresden/Leipzig, Germany
{tim.langer, matthias.jobst2, chen.liu, florian.kelber, bernhard.vogginger, christian.mayr}@tu-dresden.de

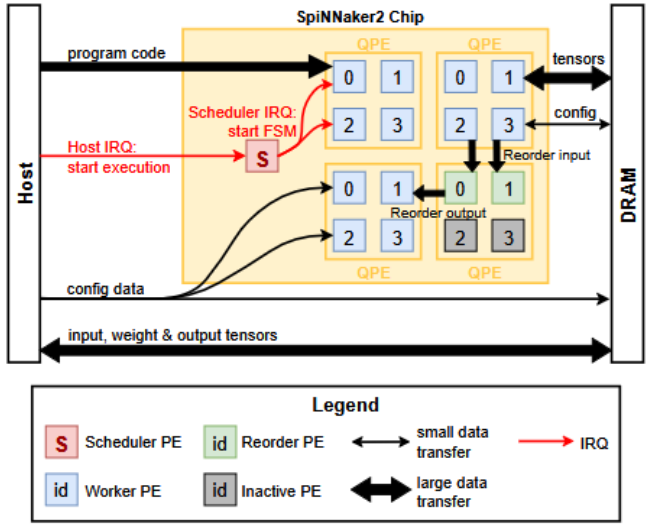
Presenter: Tim Langer

27.03.2025 @ NICE'25 (Heidelberg)

OctopuScheduler
Contributions

What have we developed?

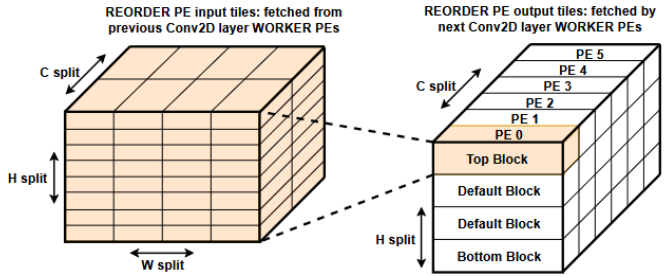
- framework for execution of single DNN layers on SpiNNaker2
- on-chip scheduling for SpiNNaker2:
  - avoiding host-to-chip communication
  - minimizing scheduling overhead
- hardware-aware strategy for automated tiling design space exploration for SpiNNaker2
- on-chip reordering mechanism for Conv2D on SpiNNaker2:
  - avoiding DRAM transfers of intermediate activations



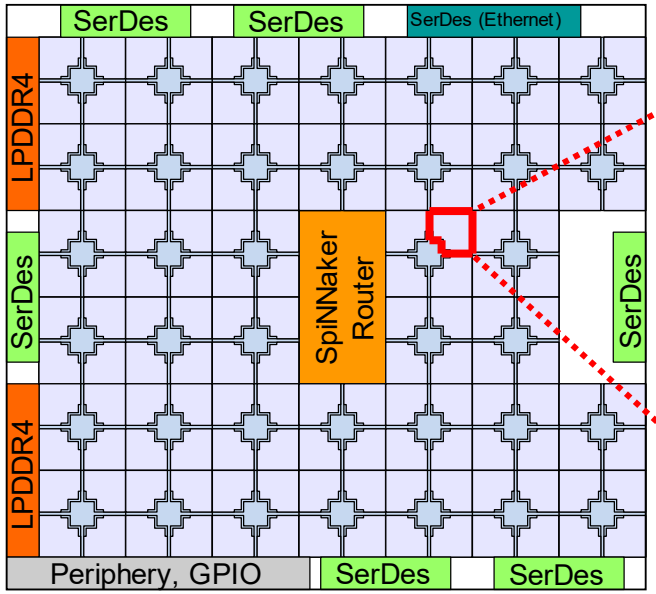
$$\mathbb{T}(M) := \{m : \ (m \mid M) \wedge (m \bmod 4 = 0)\} \subset \mathbb{N}$$
$$\mathbb{T}(P) := \{p : \ (p \mid P) \wedge (p \bmod 16 = 0)\} \subset \mathbb{N}$$
$$\mathbb{W} := \{w : \ (w \mid (M \cdot P)) \ \wedge \ (w \leqslant \#(\mathrm{PE})_{avail})\} \subset \mathbb{N}$$

Tiling Space:

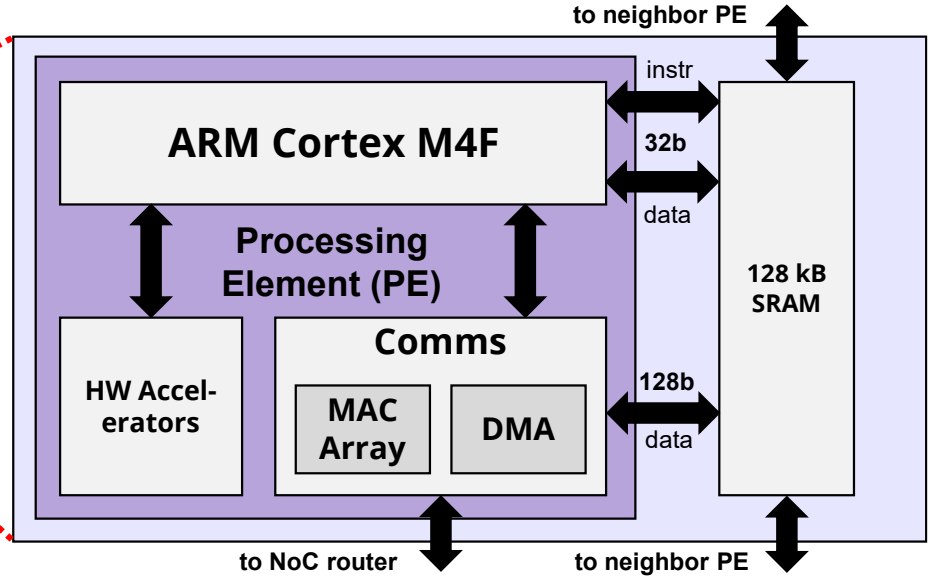
$$\mathbb{S}_{MatMul} = \mathbb{T}(M) \times \mathbb{T}(P) \times \mathbb{W}$$



SpiNNaker2 Hardware Architecture



SpiNNaker2 Chip



Processing Element

Modified after: Gonzalez, H. A. et al. (2023): „SpiNNaker2: A Large-Scale Neuromorphic System for Event-Based and Asynchronous Machine Learning". MLNCP Workshop at NeurIPS 2023, New Orleans (USA).

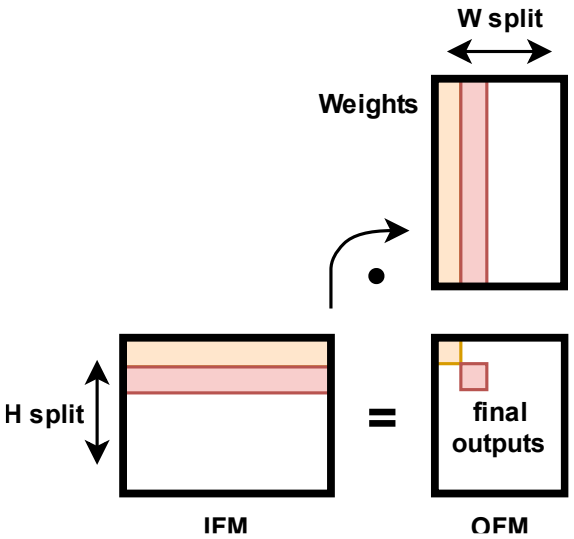
TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

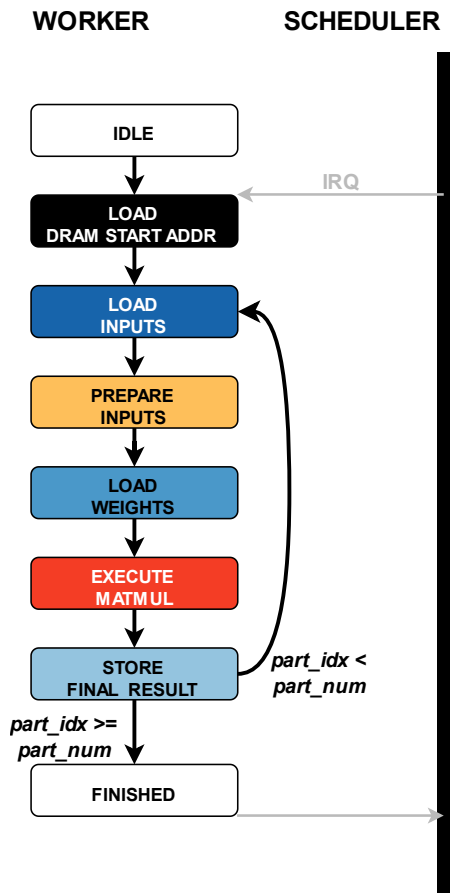
How can we execute DNN layers on SpiNNaker2?

FSM Scheduling on SpiNNaker2

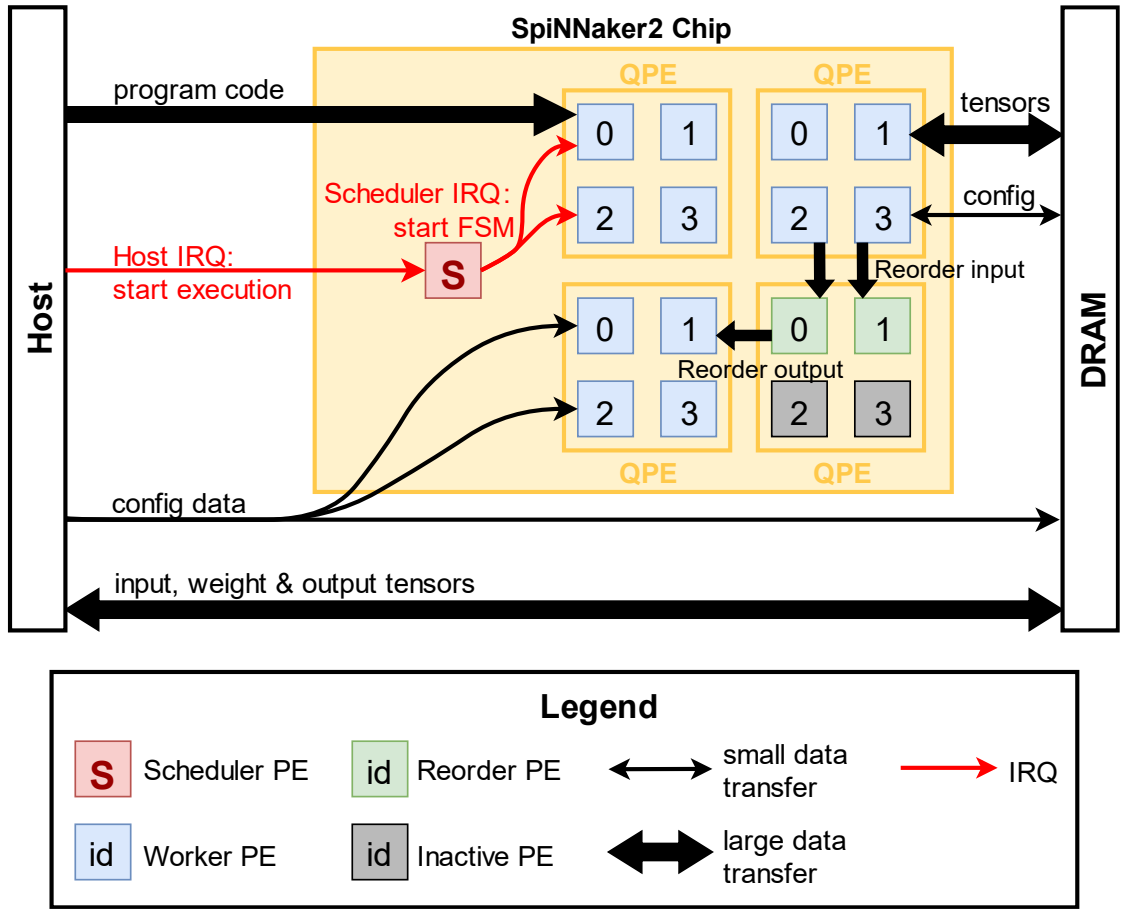
Tiling



FSM



Control & Data Flow

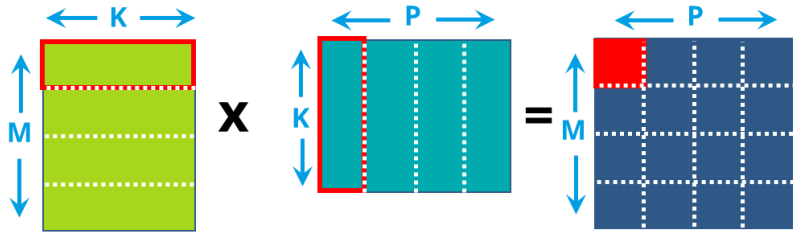


How do we distribute the computation?

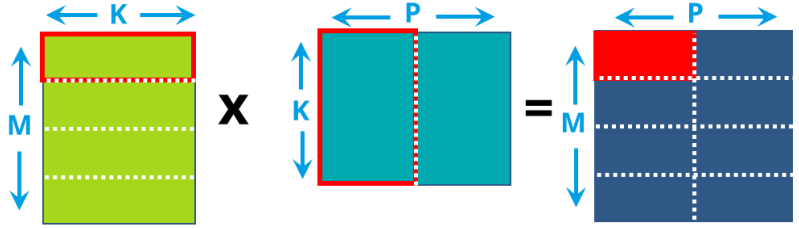
Automated Tiling

How to tile a matrix multiplication?

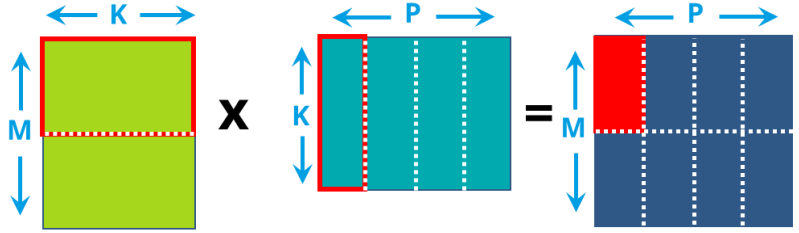
Option 1:



Option 2:



Option 3:



Dimensional Dividers:

$$\mathbb{T}(M) := \{m : \ (m \mid M) \wedge (m \bmod 4 = 0)\} \subset \mathbb{N}$$
$$\mathbb{T}(P) := \{p : \ (p \mid P) \wedge (p \bmod 16 = 0)\} \subset \mathbb{N}$$
$$\mathbb{W} := \{w : \ (w \mid (M \cdot P)) \ \wedge \ (w \leqslant \#(\text{PE})_{avail})\} \subset \mathbb{N}$$

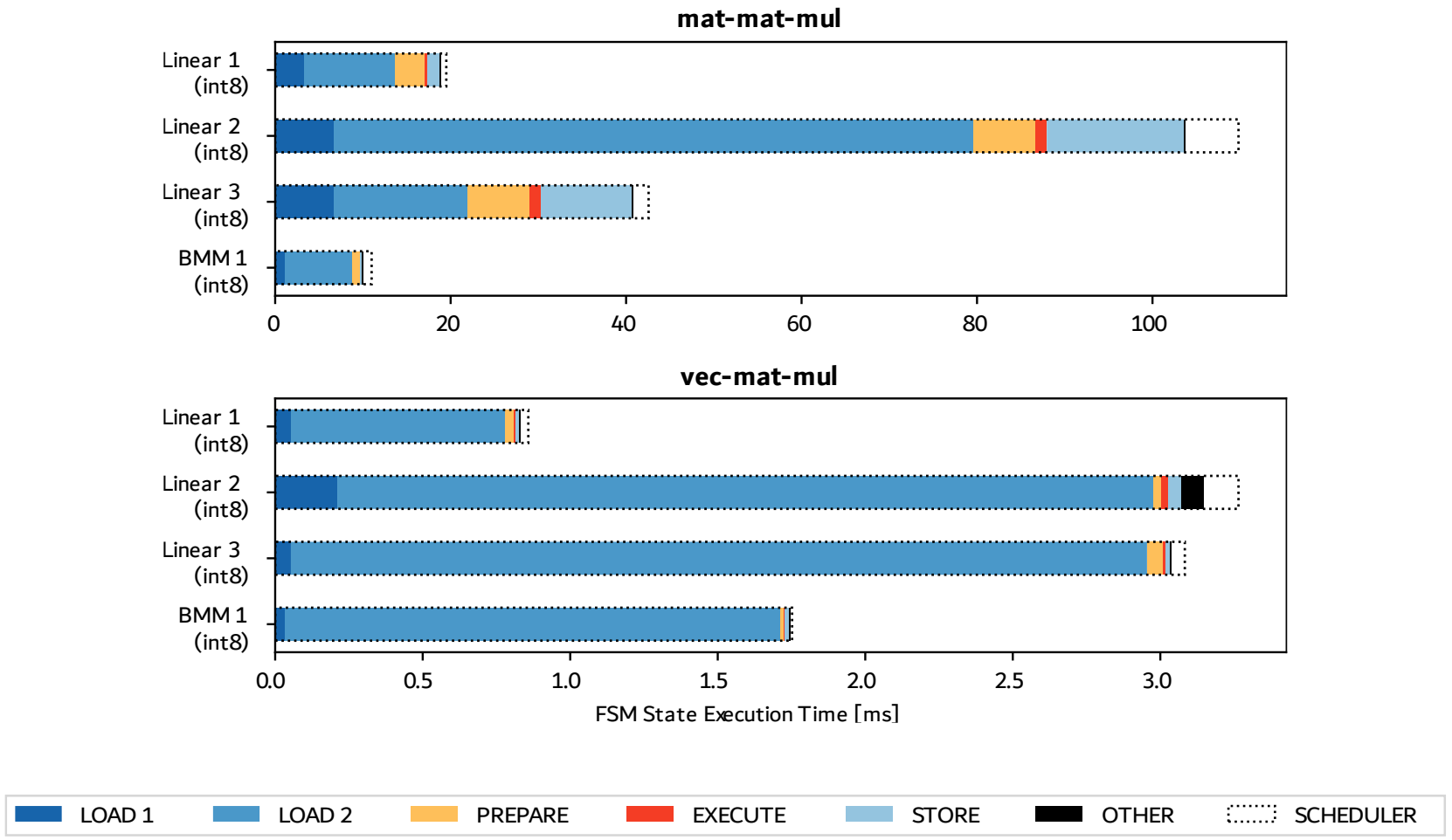
Tiling Space:

$$\mathbb{S}_{MatMul} = \mathbb{T}(M) \times \mathbb{T}(P) \times \mathbb{W}$$

Priority	Type	Parameter	Condition / Sorting
0	hard	$\#(\text{PE})^{(part)}_{WORKER}$	$\in \mathbb{N}$
0	hard	$\#(\text{PE})^{(part)}_{WORKER}$	$\leqslant \#(\text{PE})_{avail}$
0	hard	$\textbf{size}(\text{SRAM})^{(PE)}_{max}$	$\leqslant \textbf{size}(\text{SRAM})^{(PE)}_{avail}$
0	hard	$\#(\text{parts})$	$\in \mathbb{N}$
1	soft	$\#(\text{parts})$	min
2	soft	$\#(\text{PE})^{(part)}_{WORKER}$	max

Hard and soft constraints for finding an optimal hardware-aware partitioning for matrix multiplications.

Matrix Multiplication: Large Impact of Data Transfers!

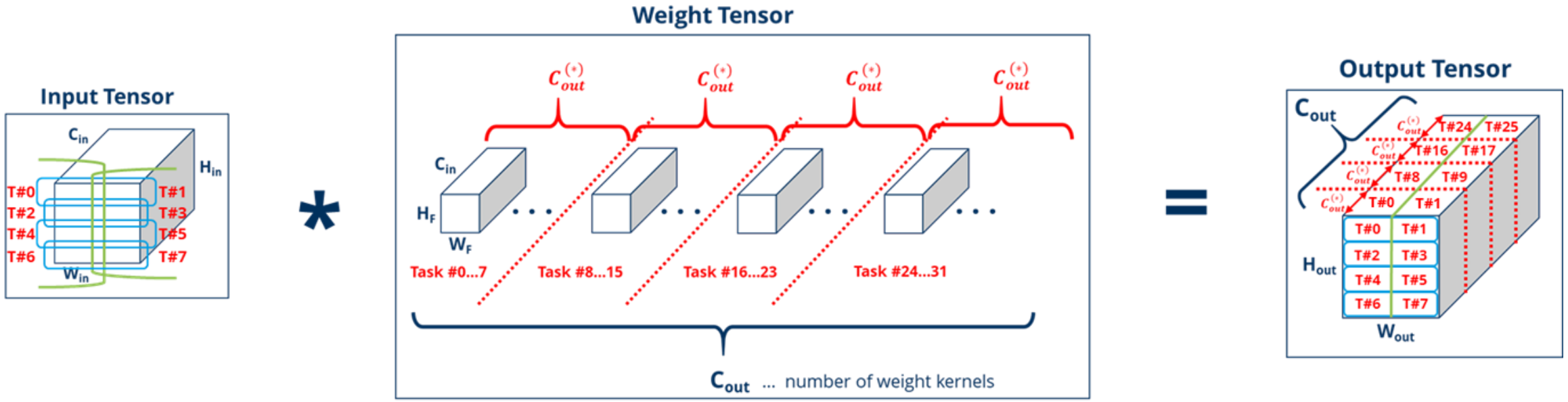


TECHNISCHE UNIVERSITÄT DRESDEN

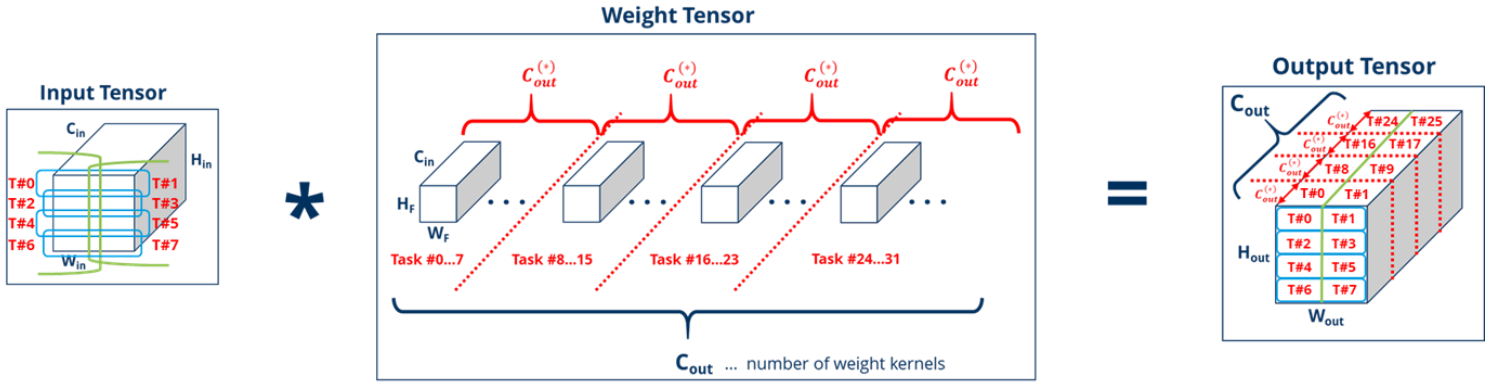
DRESDEN concept

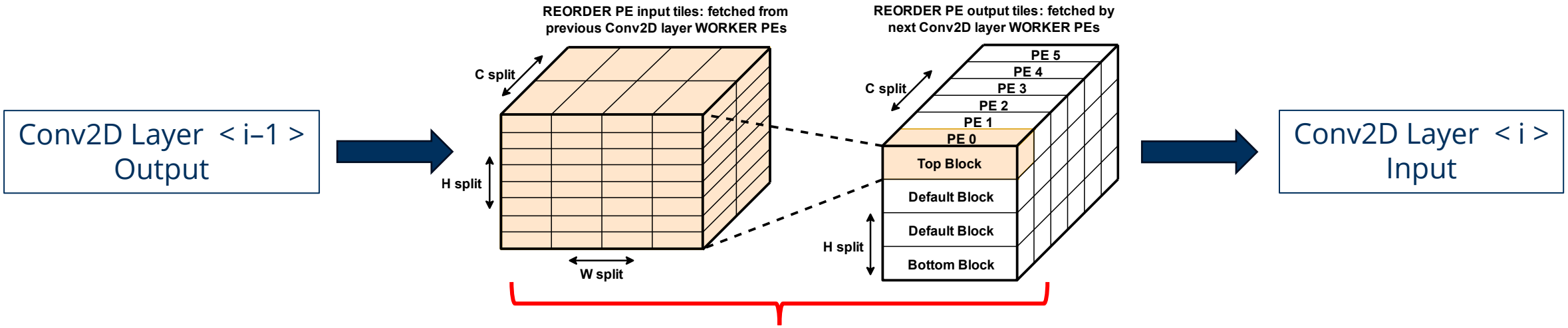
Could we avoid DRAM transfers by handling intermediate activations on-chip?

Convolutional Layers
Tiling



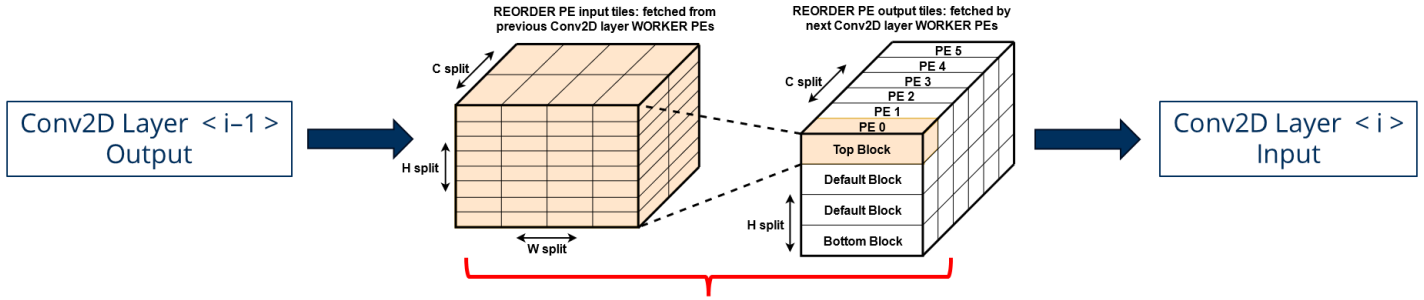
Convolutional Layers
Tiling & Transfer of Intermediate Activations



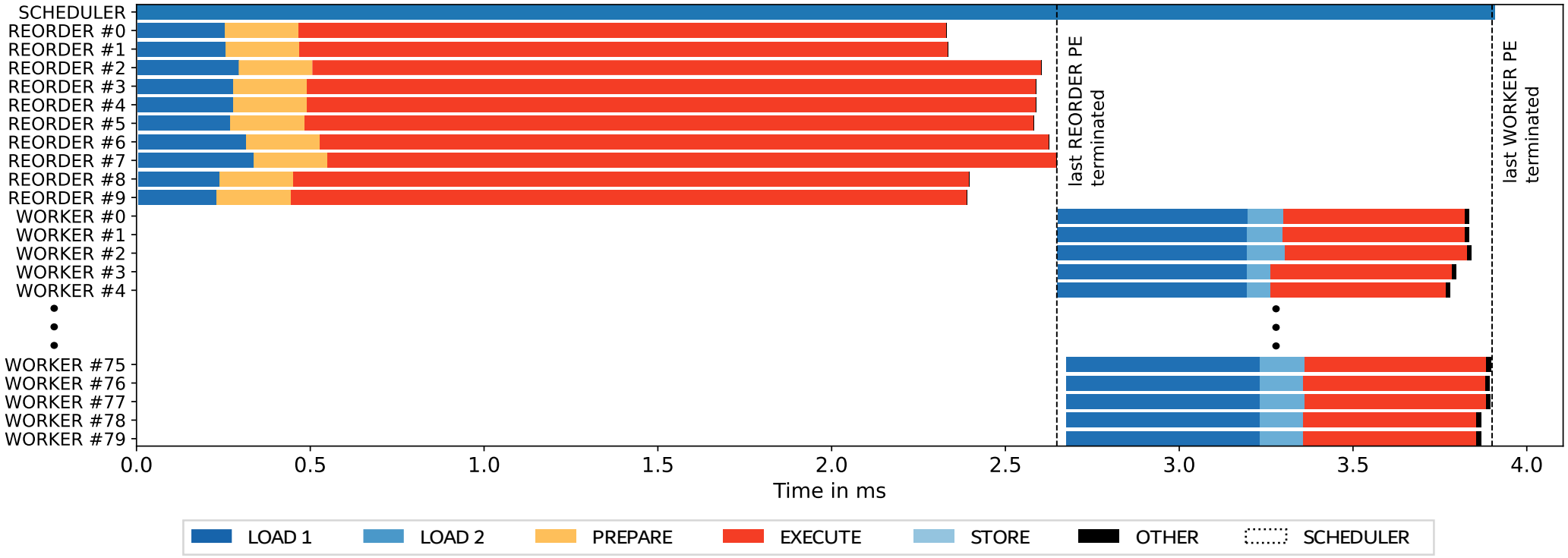


intermediate activations: reordering in on-chip SRAM → no DRAM transfers!

Convolutional Layers
Reordering

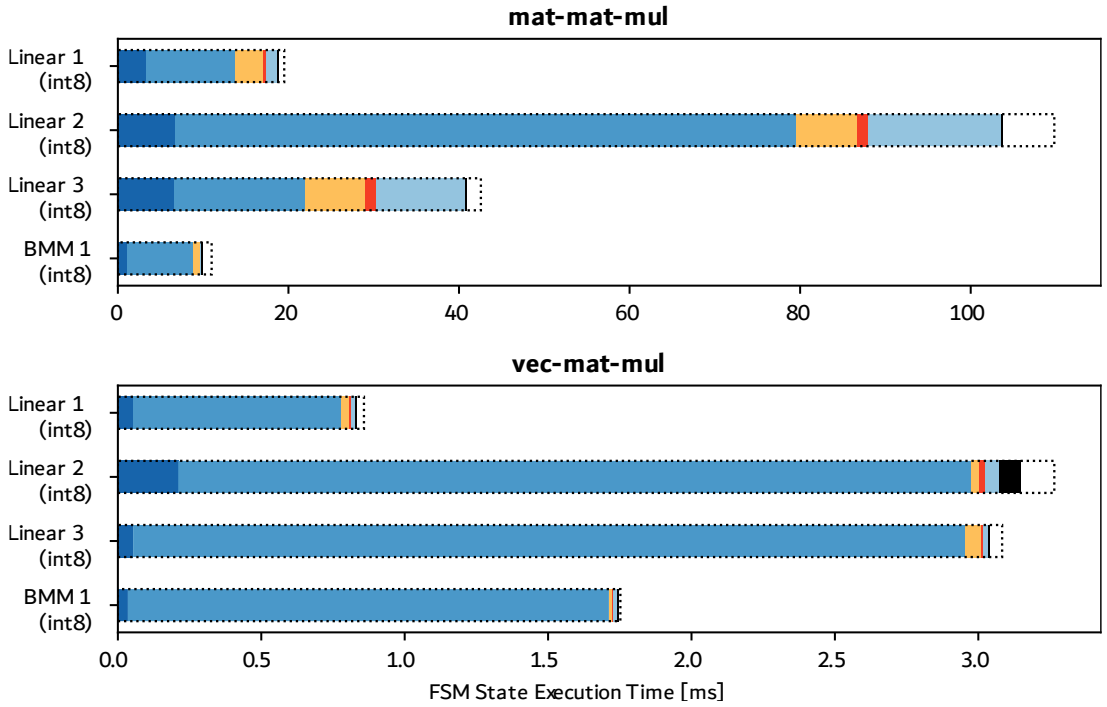


intermediate activations: reordering in on-chip SRAM → no DRAM transfers!

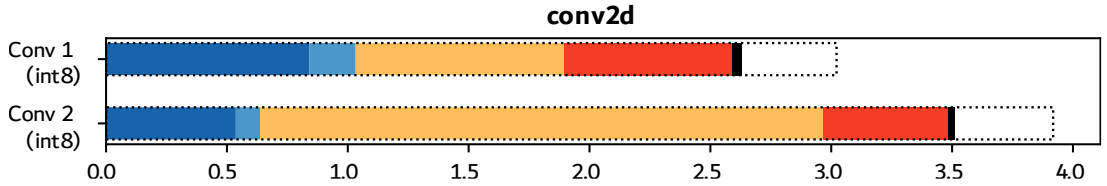


Conv2D: Much Less Impact of Data Transfers!

Matrix Multiplication



Conv2D



OctopuScheduler
Summary

What have we developed?

- framework for execution of single DNN layers on SpiNNaker2
- on-chip scheduling for SpiNNaker2
- hardware-aware strategy for automated tiling design space exploration for SpiNNaker2
- on-chip reordering mechanism for Conv2D on SpiNNaker2

Work in progress:

- multi-layer scheduling
- application to CNN & transformer architectures
- optimizations
- pipelining

What has not been covered:

- different other layer types (Softmax, Add, LayerNorm)
- tiling optimization for matrix multiplications
- detailed principle of Conv2D on-chip reordering & padding

➔ **You can find it in our paper!**

TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

OctopuScheduler: On-Chip DNN Scheduling on the SpiNNaker2 Neuromorphic MPSoC

Tim Langer*, Matthias Jobst*†, Chen Liu*, Florian Kelber*, Bernhard Vogginger*, Christian Mayr*†‡
*Chair of Highly-Parallel VLSI Systems and Neuro-Microelectronics, TU Dresden, Germany
†Centre for Tactile Internet with Human-in-the-Loop (CeTI), TU Dresden, Germany ‡ScaDS.AI Dresden/Leipzig, Germany
{tim.langer, matthias.jobst2, chen.liu, florian.kelber, bernhard.vogginger, christian.mayr}@tu-dresden.de

We would like to thank:

Federal Ministry for Economic Affairs and Climate Action

Federal Ministry of Education and Research

Funded by DFG Deutsche Forschungsgemeinschaft German Research Foundation

SECAI

ScaDS.AI DRESDEN LEIPZIG

CeTI

SpiNNcloud