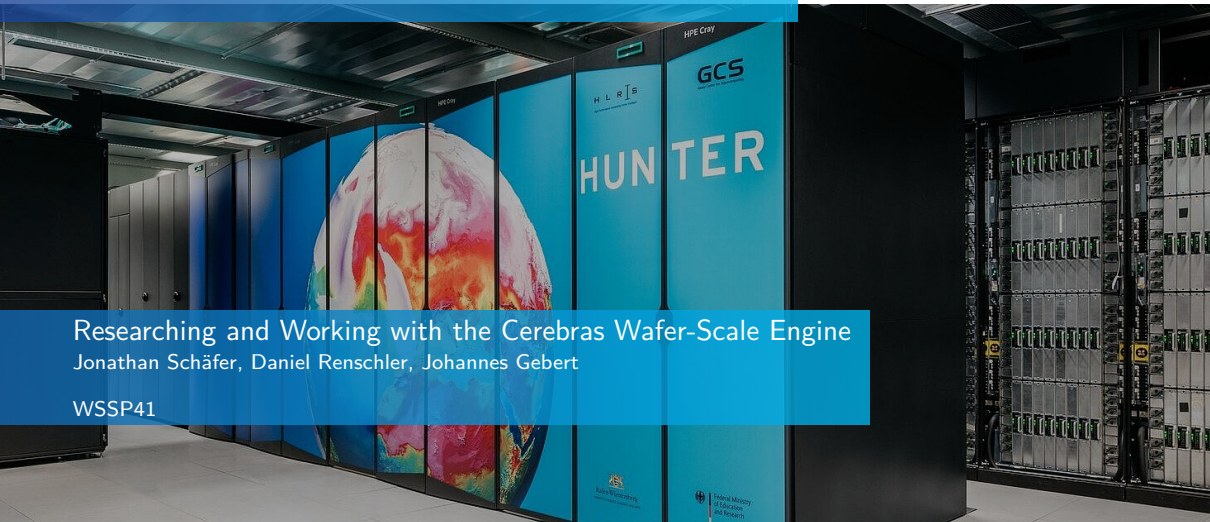
High-Performance Computing Center Stuttgart

HLRS



Researching and Working with the Cerebras Wafer-Scale Engine

Jonathan Schäfer, Daniel Renschler, Johannes Gebert

WSSP41

Motivation

The Cerebras Wafer-Scale Engine in Numbers

The Cerebras Wafer-Scale Engine in Structure

What Have We (FCG@HLRS) Done Lately?

Further Work

Motivation

Technological Limits. For Example: Memory Wall

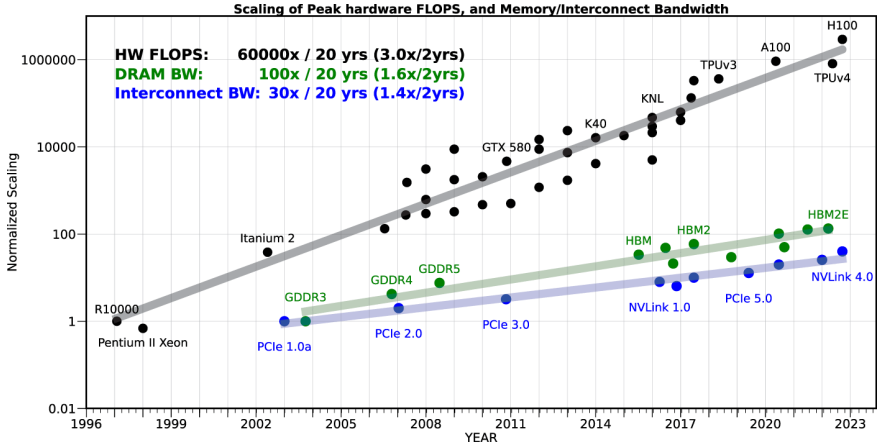


Figure 1: The memory subsystems can not keep up with the development of the floating-point performance. [1]

[1] https://doi.ieeecomputersociety.org/10.1109/MM.2024.3373763/

Physical Limits. For Example: Power Draw

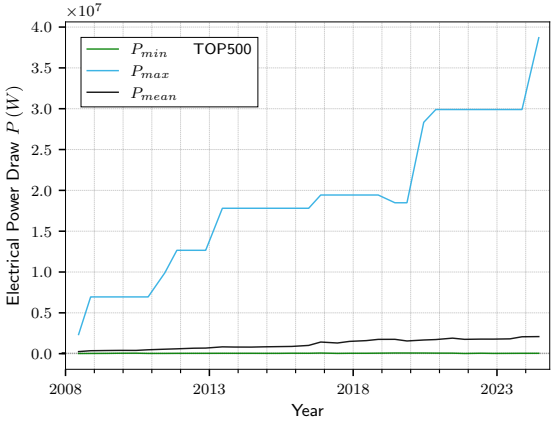


Figure 2: The power draw of the TOP500 increased significantly.

- TOP500 likely won't decrease the maximum power usage
- Especially, smaller centers can't keep up
- We can not reach zettascale without one or a few paradigm shifts

Manufacturing Limits. For Example: Feature Sizes

HLRS

Table 1: Semiconductor manufacturers with a high-end logic fab.

180 nm	130 nm	90 nm	65 nm	45 nm/40 nm	32 nm/28 nm	22 nm/20 nm	16 nm/14 nm	10 nm/7 nm/5 nm
SilTerra								
X-Fab								
Dongbu HiTek								
ADI	ADI							
Atmel	Atmel							
Rohm	Rohm							
Sanyo	Sanyo							
Mitsubishi	Mitsubishi							
ON	ON							
Hitachi	Hitachi							
Cypress	Cypress	Cypress						
Sony	Sony	Sony						
Infineon	Infineon	Infineon						
Sharp	Sharp	Sharp						
Freescale	Freescale	Freescale						
Renesas (NEC)	Renesas	Renesas	Renesas					
Toshiba	Toshiba	Toshiba	Toshiba					
Fujitsu	Fujitsu	Fujitsu	Fujitsu					
TI	TI	TI	TI					
Panasonic	Panasonic	Panasonic	Panasonic	Panasonic				
STMicroelectronics	STM	STM	STM	STM				
HLMC	HLMC	HLMC	HLMC	HLMC				
UMC	UMC	UMC	UMC	UMC				
IBM	IBM	IBM	IBM	IBM	IBM			
SMIC	SMIC	SMIC	SMIC	SMIC	SMIC	SMIC		
AMD	AMD	GlobalFoundries	GF	GF	GF	GF		
Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung
TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	**TSMC**
Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel

Manufacturing Limits. For Example: Suppliers

High-NA-EUV-Lithography:

Taiwan Semiconductor Manufacturing Company (**TSMC**) relies on
Advanced Semiconductor Materials Lithography (**ASML**) relies on
Zeiss optics and
Trumpf lasers

Post-Moore and the Crystal Ball

- CPUs and GPUs will be there for many years.
- Certain algorithms will benefit from specialized accelerators.

The Cerebras Wafer-Scale Engine in Numbers

Cerebras WSE-3

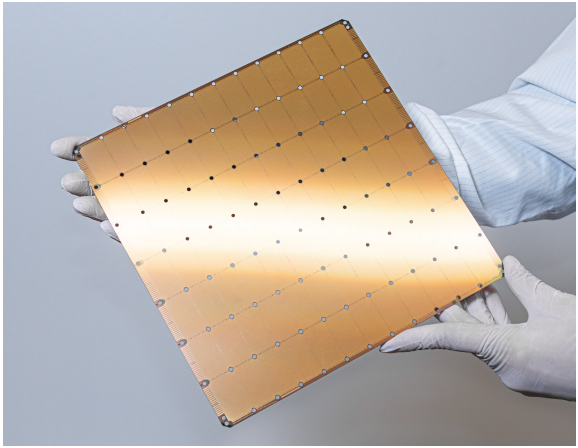


Figure 3: Cerebras Wafer [1]

Important numbers [2]:

- 900 000 processing elements (PEs)
- 125 PetaFLOPs (AI Performance)
- 44 GB on-chip SRAM
- 1,2 Tbit/s system I/O
- 46 250 mm^2 silicon area

Table 2: Performance Comparison for CS-3, H100, and B200. The FP8 and FP16 numbers are peak FLOPS in PetaFLOPS [3].

System	FP8	FP16	Power (kW)	FP8/W	FP8/W/$	FP16/W	FP16/W/$
CS-3	250	250	46	5.43	1.09	5.43	1.09
H100	64	32	41.6	1.54	1.10	0.77	0.55
B200	216	108	42.9	5.03	3.35	2.52	1.68
CS-3 Advantage							
vs. H100	3.9x	7.8x	-	3.52x	1.00x	7.05x	2.00x
vs. B200	1.16x	2.31x	-	1.08x	0.32x	2.15x	0.65x

The Cerebras Wafer-Scale Engine in Structure

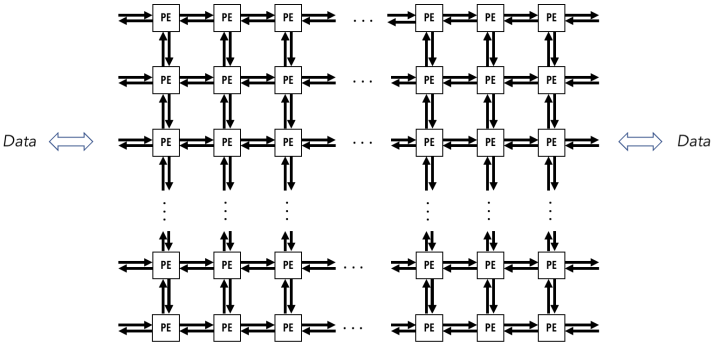


Figure 4: Schematic Wafer Layout (Image by Cerebras)

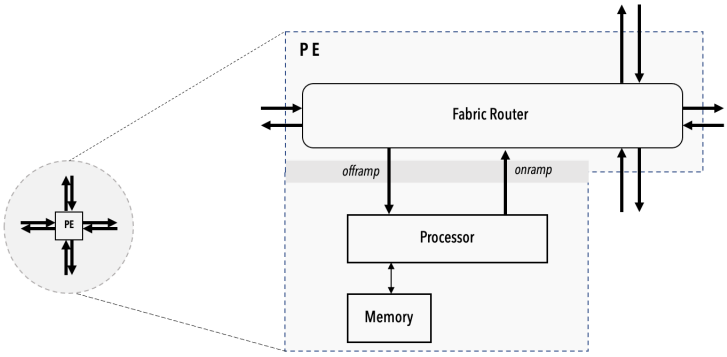


Figure 5: Single PE (Image by Cerebras)

Algorithmic Design Choices?

- Exploit the core features of the WSE:
  - Massive parallelism
  - Extremely fast on-chip nearest-neighbor communication (non-uniform access time)

$\Rightarrow$ What problem (sizes) can be effectively computed with these constraints?
$\Rightarrow$ We need to rethink algorithms with these features in mind!

What Have We (FCG@HLRS) Done Lately?

Our Research: Algorithm (Naive)

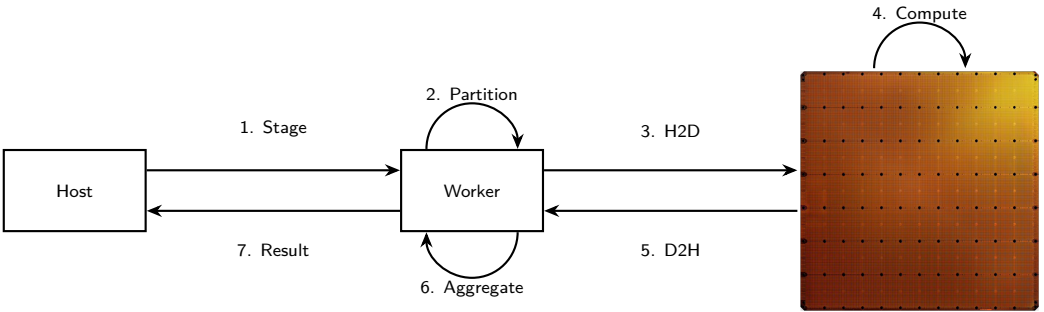


Figure 6: SpMV execution pipeline with the WSE.

Our Research: Results

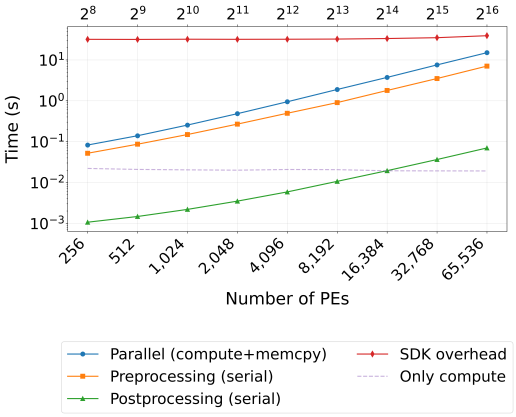


Figure 7: Time breakdown

Problems:

- Host communication is extremely expensive compared to on-chip communication
- Data locality on the chip is not exploited

$\Rightarrow$ Constraints on the algorithm and problem

Our Research: Results

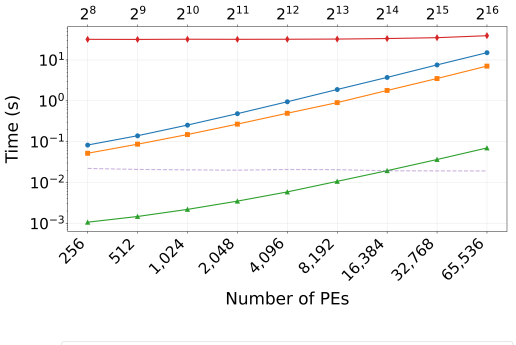
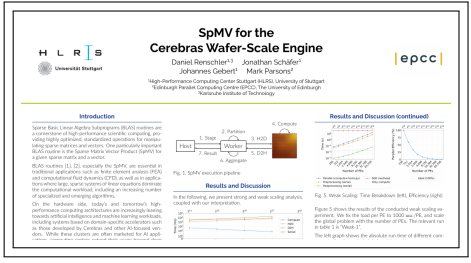


Figure 7: Time breakdown

Problems:

- Host communication is extremely expensive compared to on-chip communication
- Data locality on the chip is not exploited

⇒ Constraints on the algorithm and problem



Our Research: Results

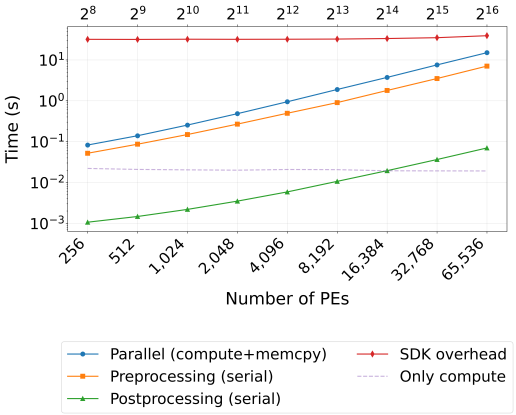
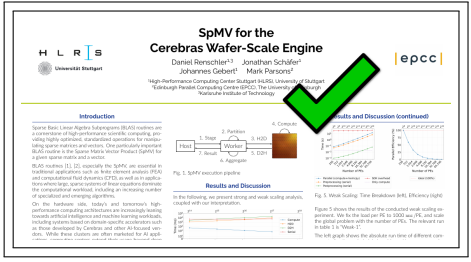


Figure 7: Time breakdown

Problems:

- Host communication is extremely expensive compared to on-chip communication
- Data locality on the chip is not exploited

⇒ Constraints on the algorithm and problem



Further Work

Further Work

Problem constraint: Massive computing power per memory unit ⇒ Algorithms that reuse data multiple times

Algorithm constraint: As much structured data as possible; how to deal with unstructured data?

More Features to Exploit

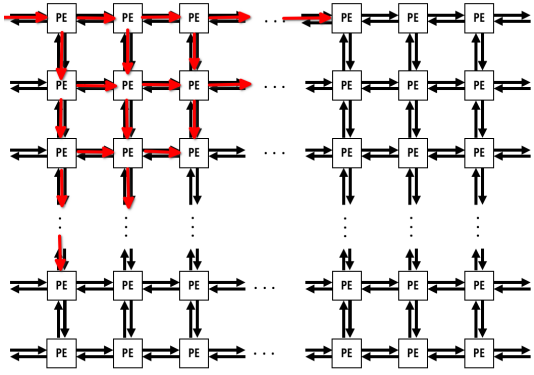


Figure 8: Streaming on the WSE

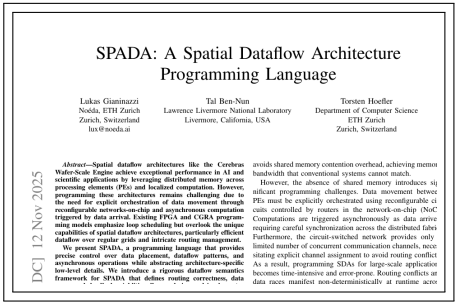
More Features to Exploit

Using higher-level languages to ease development!

More Features to Exploit

Using higher-level languages to ease development!

For example: SPADA [4]



SPADA: A Spatial Dataflow Architecture Programming Language

Lukas Gianinazzi
Noeda, ETH Zurich
Zurich, Switzerland
lux@noeda.ai

Tal Ben-Nun
Lawrence Livermore National Laboratory
Livermore, California, USA

Torsten Hoefler
Department of Computer Science
ETH Zurich
Zurich, Switzerland

***Abstract*—Spatial dataflow architectures like the Cerebras Wafer-Scale Engine achieve exceptional performance in AI and scientific applications by leveraging distributed memory across processing elements (PEs) and localized computation. However, programming these architectures remains challenging due to the need for explicit orchestration of data movement through reconfigurable networks-on-chip and asynchronous computation triggered by data arrival. Existing FPGA and CGRA programming models emphasize loop scheduling but overlook the unique capabilities of spatial dataflow architectures, particularly efficient dataflow over regular grids and intricate routing management.**

We present SPADA, a programming language that provides precise control over data placement, dataflow patterns, and asynchronous operations while abstracting architecture-specific low-level details. We introduce a rigorous dataflow semantics framework for SPADA that defines routing correctness, data

avoids shared memory contention overhead, achieving memor
bandwidth that conventional systems cannot match.

However, the absence of shared memory introduces si
nificant programming challenges. Data movement betwee
PEs must be explicitly orchestrated using reconfigurable ci
cuits controlled by routers in the network-on-chip (NoC
Computations are triggered asynchronously as data arrive
requiring careful synchronization across the distributed fabri
Furthermore, the circuit-switched network provides only
limited number of concurrent communication channels, nece
sitating explicit channel assignment to avoid routing conflict
As a result, programming SDAs for large-scale application
becomes time-intensive and error-prone. Routing conflicts an
data races manifest non-deterministically at runtime acro

Promising Approaches

- Simulation of physical systems, e.g., engineering, based in Euclidean space. [5]
- Exploiting data locality, for example, by compiler features. [6]

References I

[1] Cerebras Systems. *Press Kit*. https://www.cerebras.ai/company/press-kit. 2024.

[2] S Lie. *Cerebras Wafer-Scale AI*. Hot Chips 36, https://hc2024.hotchips.org/assets/program/conference/day2/72_HC2024.Cerebras.Sean.v03.final.pdf. 2024.

[3] Y Kundu, M Kaur, T Wig, K Kumar, P Kumari, V Puri, and M Arora. "A comparison of the cerebras wafer-scale integration technology with nvidia gpu-based systems for artificial intelligence". In: *arXiv preprint arXiv:2503.11698* (2025).

[4] L Gianinazzi, T Ben-Nun, and T Hoefler. "SPADA: A Spatial Dataflow Architecture Programming Language". In: *arXiv preprint arXiv:2511.09447* (2025).

[5] T Oppelstrup, N Giamblanco, DZ Kalchev, I Sharapov, M Taylor, D Van Essendelft, S Rajamanickam, and M James. "Beyond Exascale: Dataflow Domain Translation on a Cerebras Cluster". In: (Nov. 2025). DOI: 10.48550/ARXIV.2511.11542. arXiv: 2511.11542 [cs.DC].

[6] N Stawinoga, D Katz, A Lydike, J Zarins, N Brown, G Bisbas, and T Grosser. "An MLIR Lowering Pipeline for Stencils at Wafer-Scale". In: (Jan. 2026). DOI: 10.1145/3779212.3790124. arXiv: 2601.17754 [cs.DC].