Load Balancing and Communication Bottlenecks in SpMV on the Cerebras WSE-3

Saho Orihara, Kouki Sakai, Reo Nagashima and Takaaki Miyajima
Graduate School of Science and Technology, Meiji University, Japan, e-mail: ce255032@meiji.ac.jp

明治大学 MEIJI UNIVERSITY

Overview

SpMV is a fundamental kernel in scientific computing, but its random memory access limits performance. The Cerebras CS-3, with 44 GB of on-chip SRAM, is a promising platform for this problem. That memory, however, is distributed across roughly 900,000 PEs, so how the non-zero elements (nnz) are distributed determines performance. The simple 2D matrix partitioning used in our previous work left most cores underutilized because of skew in the non-zero elements [1]. We therefore introduce load balancing and evaluate both its effectiveness and the execution-time breakdown as the PE count grows. **For Poisson3Da, load balancing achieved a 3.40x speedup, although the benefit diminishes as the PE count increases. For F1, inter-PE communication accounted for 71.2% of execution time.** This work is a first step toward load balancing on the CS-3.

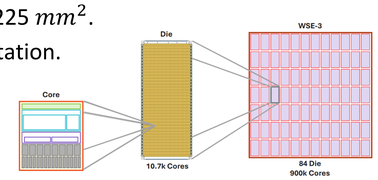
Cerebras Wafer Scale Engine 3 (WSE-3)

The entire 300 mm wafer is used as one chip

- 84 dies are interconnected with a parallel interface called Cross Scribe Line Connections.
- Process rule: TSMC 5 nm , Chip area: 46,225 mm^2.
- Running at 750 MHz for scientific computation.

Specifications

- Total PE counts: **893,064 (762x1172).**
- Total amount of on-chip SRAM: **44 GB.**
- Memory bandwidth: **21 PB/s.**
- SGEMM performance: **349.0 TFlops/s** [2]

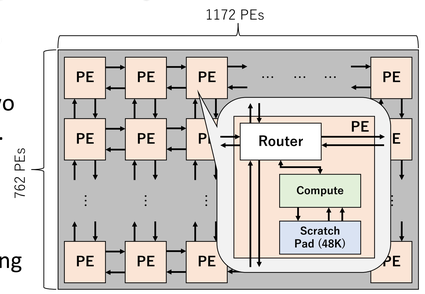


Physical structure of WSE-3

Processing Element (PE): Memory, CE, and Router

PE comes with scratchpad memory, compute element (CE) and router

- CE and memory are connected to two 64-bit read ports and one write port.
- CE has 128-bit SIMD data path and 48KB Scratchpad memory.
- Router is connected to north, south, west, and east routers in neighbouring PE.

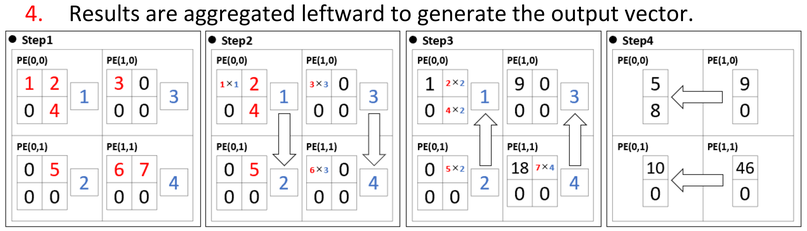


PEs are connected in 2-D mesh.

SpMV Implementation in Cerebras CSL

Algorithm overview

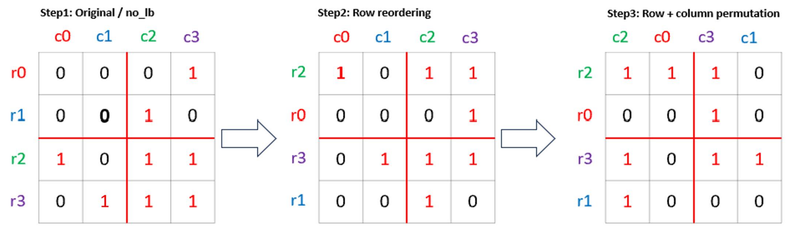
- Our 2-D SpMV on CS-3 applies offline load balancing (row/column permutation; detailed below) before mapping, followed by local computation, Y-axis broadcast, and reduction.
  1. The matrix is 2D-partitioned and mapped onto the PE grid.
  2. The x vector is sequentially distributed, starting from the top-row PEs.
  3. Each PE performs local SpMV.
  4. Results are aggregated leftward to generate the output vector.



Computation Algorithm

Load balancing algorithm

- To balance the load among PEs, we apply a greedy row/column-permutation method that reduces the maximum nnz/PE, based on Cerebras's official load-balancing utility *analyze.cpp* [3]. Specifically,
  1. Count nnz per row/column
  2. Greedily assign heavy rows/columns to lighter PE rows/columns
  3. Iterate until max nnz/PE stops improving.



Load balancing algorithm

Evaluation method

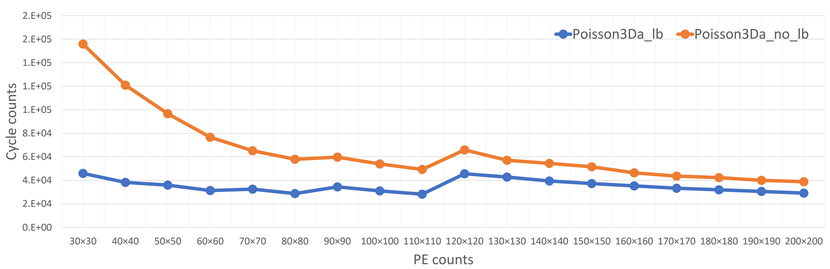
- Performance evaluation is conducted on CS-3 System @Argonne National Laboratory, ALCF AI Testbed w/ SDK v2.10.0.
- Five sparse matrices are used, spanning diverse structural characteristics (FEM/structural, quantum chemistry, circuit simulation), represented in FP32. All are obtained from the SuiteSparse Matrix Collection [4].
- Strong scaling study: measurements were performed using an M x M matrix and an N x N PE configuration. For Poisson3Da, N was varied from 30 to 200 in steps of 10, while for the other matrices, N was varied from 100 to 700 within the executable range.

Sparse Matrices Used in Evaluation

Matrix	Dimension	nnz	Kind
Poisson3Da	$13{,}514^2$	352,762	FEM
SiO2	$155{,}331^2$	11,283,503	Quantum Chem
F1	$343{,}791^2$	13,590,452	Structural
inline_1	$503{,}712^2$	36,816,170	Structural
ASIC_680ks	$682{,}712^2$	2,329,176	Circuit

Load balancing effect:

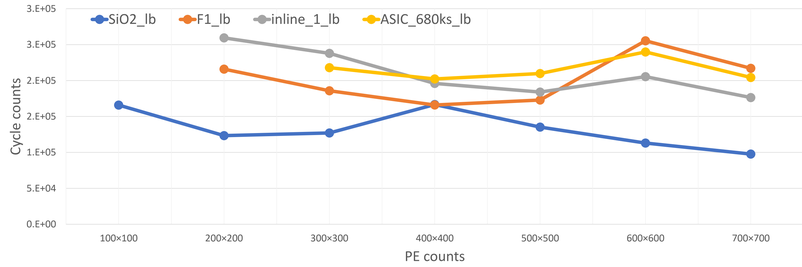
- For Poisson3Da, LB achieved a 3.40x speedup at 30×30 PEs
- The benefit decreased with more PEs, reflecting the relatively smaller impact of load imbalance at larger PE counts.



Poisson3Da: Load-Balanced vs. Non-Load-Balanced

Strong scaling:

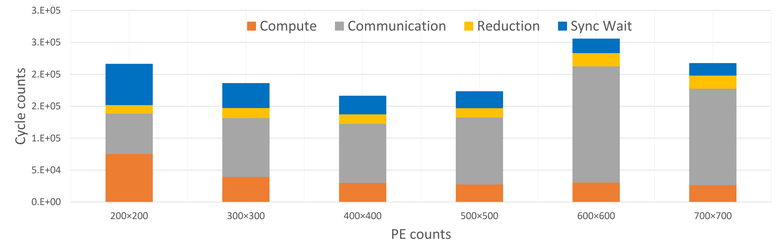
- The cycle count did not decrease monotonically as the PE count increased.
- We found a trade-off between reduced computation and increased communication around $N \approx \sqrt{M}$ within the evaluated range.



PE Scaling of Larger-Scale Matrices under Load-Balancing

Bottleneck Analysis:

- As communication becomes dominant (71.2% at N=600), the behavior is governed by broadcast rounds Y: each PE column holds M/N vector elements.
- Due to the limited local memory capacity of each PE on the CS-3, the input vector is divided into chunks, with each PE storing a portion of the vector and communicating it across all PEs.
- At N ≈ √M (N² ≈ M), the chunk size reaches 1 and broadcast rounds peak, so total cycles also peak there (N ≈ 586 for F1).
- As N increased from 200 to 700, the compute share fell from 35.0% to 12.4% while communication rose from 29.3% to 69.6%, showing a shift toward communication-bound execution.
- As future work, we will design an optimal communication scheme based on communication count and per-round data volume.



Execution-Time Breakdown for F1 on the Critical-Path PE

Time Breakdown by PE Configuration for F1

N	Vector Chunk	Broadcast Rounds	Compute [%]	Comm [%]	Reduction [%]	Sync [%]
200	9	191	35.0	29.3	6.2	29.5
300	4	287	21.4	49.6	8.4	20.6
400	3	287	18.3	55.7	8.9	17.1
500	2	344	16.2	60.6	8.4	14.9
600	1	573	12.1	71.2	8.2	8.5
700	1	492	12.4	69.6	9.4	8.6

Acknowledgement and references

[1] S. Orihara, K. Sakai, and T. Miyajima, "Towards Acceleration of Sparse Matrix-Vector Multiplication on the Cerebras CS-2," Poster Session, SCA/HPCAsia 2026: The International Conference on High Performance Computing in Asia-Pacific Region, Osaka, Japan, 2026, Poster P-254.
[2] S. Lie, "Wafer-Scale AI: GPU Impossible Performance," presented at the 36th IEEE Hot Chips Symposium (Hot Chips 2024), Aug. 2024.
[3] Cerebras Systems, "Hypersparse SpMV," Cerebras SDK Documentation, ver. 1.3.0, accessed Aug. 5, 2026.
[4] T. A. Davis and Y. Hu, "The University of Florida Sparse Matrix Collection," ACM Trans. Math. Softw., vol. 38, no. 1, Art. no. 1, pp. 1–25, 2011, doi: 10.1145/2049662.2049663.
This work was supported by Japan Science and Technology Agency (JST) as part of Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE), Grant Number JPMJAP2341. This work was supported by JSPS KAKENHI Grant Number 24K14972. This research used resources of the Argonne Leadership Computing Facility, a U.S. Department of Energy (DOE) Office of Science user facility at Argonne National Laboratory and is based on research supported by the U.S. DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357.