A linear-scaling, sparse matrix electronic structure solver using the Cerebras Wafer Scale Engine

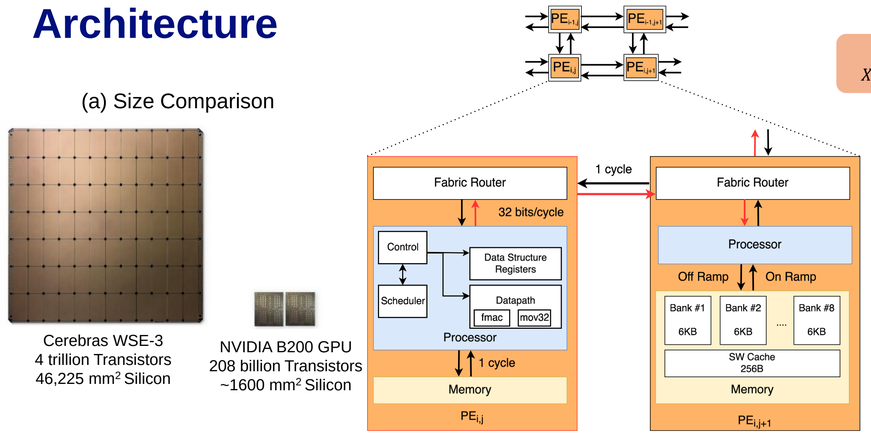
Md Sabbir Hossain Polak†, Anthony Gatti††, Anders M. N. Niklasson*, Benjamin S. Southworth*, Joshua Finkelstein*.
† University of Mississippi, †† University of Pittsburgh, ∗ Los Alamos National Laboratory

Abstract

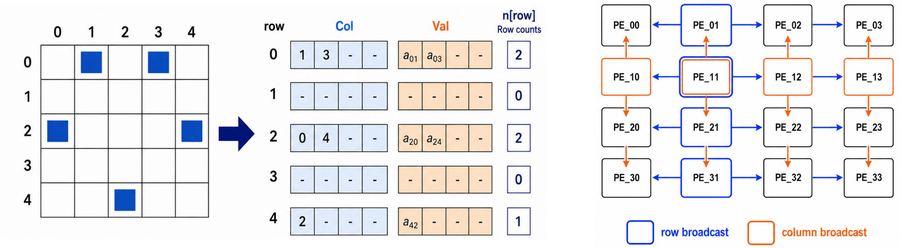
Pioneering ultrafast sparse electronic structure calculations

We present an electronic structure solver developed for the Cerebras Wafer Scale Engine (WSE). We combine a novel WSE-optimized sparse-matrix squaring operation with a Fermi-operator expansion algorithm that maps the electronic structure problem onto the WSE. Our approach constructs the quantum-mechanical density matrix with linear-scaling computational complexity in system size, achieving ***31x speedup*** over a sparse SP2 solver on an NVIDIA GH200 Grace Hopper superchip.

Cerebras Wafer-Scale Engine Architecture



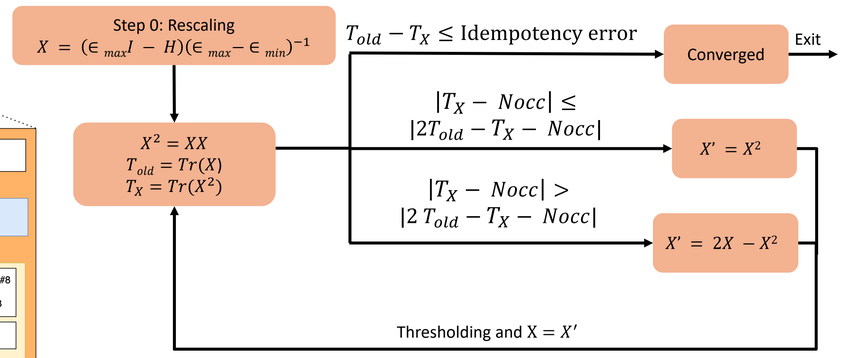
→ Monolithic 300mm fault-tolerant wafer, roughly 56x larger than leading GPU die
→ Roughly 900,000 processing elements (PEs) connected on a 2D mesh topology
→ **Distributed Memory:** 48 KB of local SRAM per processing elements and in total 40 GB on-chip SRAM and 20 PB/s memory bandwidth.
→ **Network-on-Chip:** Nearest four neighbor communication using 32-bit fabric packets (**wavelets**) routed by virtual channels (**colors**).
→ **Data Structure Registers** streamline and accelerate vector memory access operations.



Second-order Spectral Projection (SP2) Solver On WSE

The quantum-mechanical density matrix describing the electronic structure within theories such as Density Functional Theory or Hartree-Fock, can be constructed without explicit diagonalization on the WSE. By rescaling the system Hamiltonian H such that the spectrum lies inside [0,1], we iteratively apply polynomials X^2 or $2X - X^2$ to approximate the Heaviside matrix step function [1, 2]:

$$D = \theta\,(\mu I - H)$$



SP2 Loop

The trace determines which polynomial is applied. If $Tr(X^2)$ is closer to the occupation number, N_{occ}, than $2Tr(X) - Tr(X^2)$, the chosen polynomial is X^2. Otherwise, the complementary branch $2X - X^2$ is selected. Finally, entries below a threshold value are dropped and the remaining ones are repacked into sparse format.

Sparse Matrix Squaring on the WSE

Sparse Message Format

→ Each $N \times N$ matrix, with $N = PK$, is partitioned over a $P \times P$ grid of processing elements (PEs) with every PE storing a permanent $K \times K$ tile. Each PE receives several operand tiles from other PEs throughout the operation.
→ Tiles are stored in a fixed-size ELLPACK-like data structures, consisting of values, column indices, and entry counts to enable fast accesses and traversals.

SUMMA Communication Model

→ In each step, a row and column broadcast their input tiles to all other PEs.
→ Each PE multiplies their received operand tiles and accumulate the product.
→ The algorithm ends once all rows and columns have acted as the broadcast root.

Local Multiplication and Accumulation

→ Tile multiplication is performed using the Gustavson algorithm.
→ Products are accumulated directly into a persistent sparse data structure.

Synchronous and Asynchronous Operation

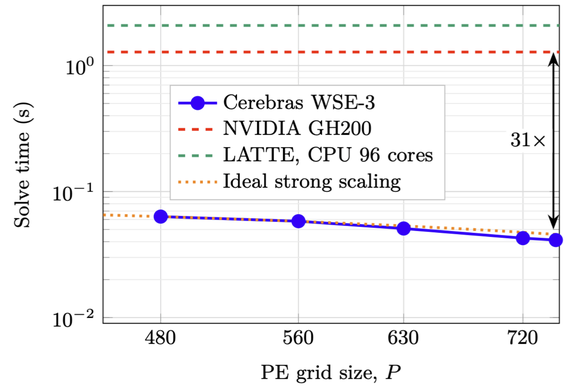
→ *Synchronous:* Communication and local multiplication are sequential. The communication for step r occurs directly before the computation for step r+1.
→ *Asynchronous:* Communication and local multiplication are overlapped. The operands for step r+1 are prefetched during the computation of step r.

Performance evaluation on Cerebras WSE-3 Chip

→ Benchmarked on Sandia's Kingfisher CS-3 system. SP2 and sparse matrix-square kernel written in CSL (v1.4)

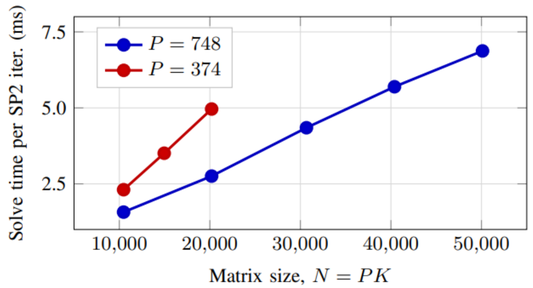
Real Sparse Hamiltonian Matrix

→ We benchmark a DFTB Hamiltonian for a 10,056-atom water system, a sparse **20,196 x 20,196 matrix**. In the strong scaling limit (P = 747), the FP32 WSE solver reaches a **31x speedup** over a single precision sparse GPU reference (custom kernel running on NVIDIA GH200), and **51x** over double-precision LATTE [3] running on a CPU with 96 OpenMP threads.



Linear Scaling With System Size

→ At fixed P, per-iteration time scales linearly with N as expected for our solver on a real matrix.



References

[1] A. M. N. Niklasson, Density Matrix Methods in Linear Scaling Electronic Structure Theory. Dordrecht: Springer Netherlands, 2011, pp. 439–473.[Online]. Available: https://doi.org/10.1007/978-90-481-2853-216
[2] J. Finkelstein, J. S. Smith, S. M. Mniszewski, K. Barros, C. F. A. Negre, E. H. Rubensson, and A. M. N. Niklasson, "Mixed precision fermi-operator expansion on tensor cores from a machine learning perspective," Journal of Chemical Theory and Computation, vol. 17, no. 4, pp. 2256–2265, 04 2021.[Online]. Available: https://doi.org/10.1021/acs.jctc.1c00057
[3] M. Elstner and G. Seifert, "Density functional tight binding," Philos.Trans. R. Soc., A, vol. 372, no. 2011, p. 20120483, 2014.

Acknowledgement

This work is supported by the Laboratory-Directed Research and Development Program (20260113ER) at Los Alamos National Laboratory (LANL). LANL is operated by Triad National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy Contract No. 892333218NCA000001.