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‡ and Joshua Finkelstein‡

* Department of Computer and Information Science, University of Mississippi, Oxford, MS, USA

† Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA

‡ Theoretical Divison, Los Alamos National Laboratory, Los Alamos, NM, USA

***Abstract*—We present a linear-scaling electronic structure solver for the Cerebras Wafer-Scale Engine (WSE), a dataflow architecture integrating more than 900,000 processors on a single silicon wafer. We combine a novel, WSE-optimized sparse-matrix squaring kernel with a sparse, recursive Fermi-operator expansion to construct the density matrix on this massively parallel architecture. Our approach solves the quantum-mechanical eigenvalue problem for one self-consistent-field iteration with a computational complexity that is linear in the number of atoms. For a bulk water system described at the density-functional tight-binding level of theory, we achieve a $31\times$ speedup for a single density-matrix solve relative to a single-GPU reference calculation running on an Nvidia GH200 Grace Hopper superchip.**

***Index Terms*—wafer-scale computing, electronic structure solvers, density matrix purification, sparse matrix multiplication, high-performance computing**

I. Motivation

Conventional electronic-structure eigensolvers based on dense matrix algebra have cubic computational complexity, making them prohibitively expensive for systems containing more than a few hundred atoms. Linear-scaling methods instead exploit sparse matrix algebra [1], but their computational kernels often underutilize CPUs and GPUs because of irregular memory access patterns, limited memory bandwidth, and costly inter-device communication. The Cerebras Wafer-Scale Engine (WSE) offers a fundamentally different architecture, integrating approximately 900,000 processing elements connected by a two-dimensional mesh and equipped with local SRAM and dataflow-based communication (Figure 1). In this work, we develop a computational framework that, for the first time, enables the solving of a quantum-mechanical eigenvalue problem on the WSE.

Our solution to the quantum-mechanical eigenvalue problem is given by the density matrix D, which describes the electronic structure of a physical system defined by the Hamiltonian H, and is obtained by evaluating the matrix Heaviside step function

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

where μ is the Fermi level, defined as the energy threshold at which the total number of occupied states (or, electrons) equals the desired occupation number, $N_{\rm occ}$. To compute this matrix function on the WSE, we build an approximation to D based on a recursive Fermi-operator expansion method known as the second-order spectral projection (SP2) algorithm [2], [3]. SP2 transforms a shifted and scaled Hamiltonian, X_0, into D through repeated applications of the second order polynomials x^2 and $2x - x^2$, where the chosen branch is based on which moves the matrix trace closer to $N_{\rm occ}$. In its most basic form, the i-th iteration of SP2 is given by

$$X_i = \begin{cases} X_{i-1}^2, & \mathrm{Tr}[X_{i-1}^2] > N_{occ} \\ 2X_{i-1} - X_{i-1}^2, & \mathrm{Tr}[X_{i-1}^2] \leq N_{occ} \end{cases} ,$$

with the algorithm terminating once the target trace is sufficiently close to N_{occ}.

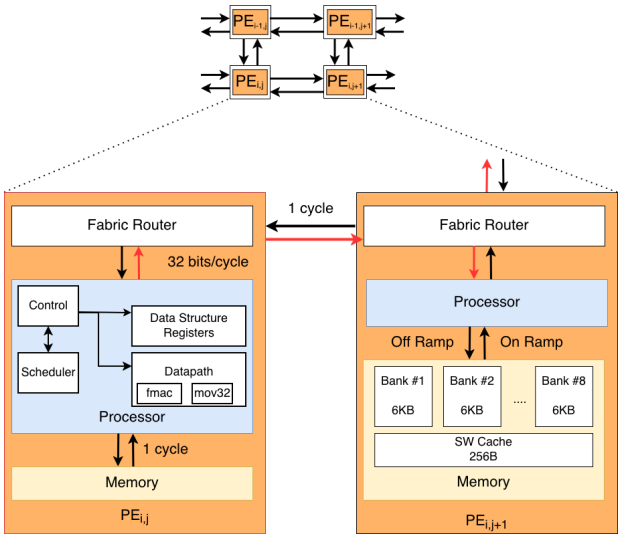


Fig. 1. Cerebras processing-element microarchitecture and inter-PE wavelet communication path

II. Sparse Matrix Squaring on the WSE

The dominant operation in linear-scaling SP2 is sparse matrix squaring. Given an $N \times N$ sparse Hamiltonian matrix, we distribute this across a $P \times P$ grid of processing elements (PEs), with each PE storing a sparse $K \times K$ tile, where $N = PK$. We implement the multiplication using

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.

a sparse Scalable Universal Matrix Multiplication Algorithm (SUMMA), which broadcasts one tile across each processor row and one tile along each processor column at every step. The two operands in the multiplication are identical, so each PE stores only one permanent input tile that it uses for each broadcast.

Tiles use a fixed-capacity ELL-like representation (see Figure 2), which is well suited for the WSE's statically allocated local memory. Local products are computed using a Gustavson-style sparse multiplication and are accumulated directly into a sparse ELL-like output structure [4]. Before execution, a symmetric permutation of the input matrix uniformly distributes the nonzeros across the PEs for load balancing.

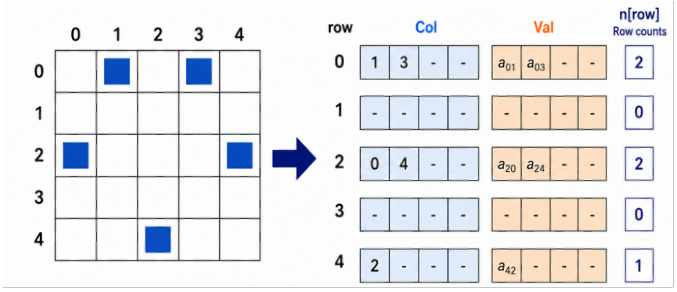


Fig. 2. Sparse matrix ELL-like data structure used in squaring kernel.

We implemented both a synchronous kernel, which serializes communication and multiplication, and an asynchronous kernel, which prefetches the operands for the next step, enabling parallel communication and multiplication but increasing the per-PE memory burden.

III. On-Wafer SP2 Solver

The SP2 algorithm, shown in Figure 3, executes on the wafer. At each iteration, the selected polynomial is formed directly in the accumulators on each PE: the X^2 branch starts from zero and adds products, while the $2X - X^2$ branch starts from $2X$ and subtracts products. Each iteration thresholds small entries, reduces diagonal traces to a control PE, checks for convergence, selects the next polynomial, and broadcasts the decision across the wafer.

Require: Hamiltonian H, its estimated spectral bounds $\epsilon_{\min}, \epsilon_{\max}$, occupation N_{occ}, threshold τ

$\big(X = \epsilon_{\max} I - H\big)\big/\big(\epsilon_{\max} - \epsilon_{\min}\big)$
repeat
  $X^2 = XX$ {*Sparse multiply optimized for WSE*}
  if $|\operatorname{Tr}(X^2) - N_{\text{occ}}| \le |2\operatorname{Tr}(X) - \operatorname{Tr}(X^2) - N_{\text{occ}}|$ **then**
    $X = X^2$
  else
    $X = 2X - X^2$
  end if
  Update $\operatorname{Tr}(X)$ and drop entries of X below τ
until idempotency $\operatorname{Tr}(X - X^2)$ is converged
return X

Fig. 3. SP2 density-matrix solver algorithm.

IV. Results

Figure 4 demonstrates linear scaling for the asynchronous WSE solver on real DFTB water Hamiltonians, with per-SP2-iteration time growing linearly with the system size N at fixed processor grid P. Because the WSE solver has a smaller linear-scaling prefactor than the GPU reference, its advantage widens with system size. The speedup over the single-precision sparse GPU reference grows from **20**× at $N \approx 10{,}000$ to **51**× at $N \approx 50{,}000$. The largest system tested, a $50{,}116 \times 50{,}116$ Hamiltonian, corresponds to roughly 25,000 atoms of water.

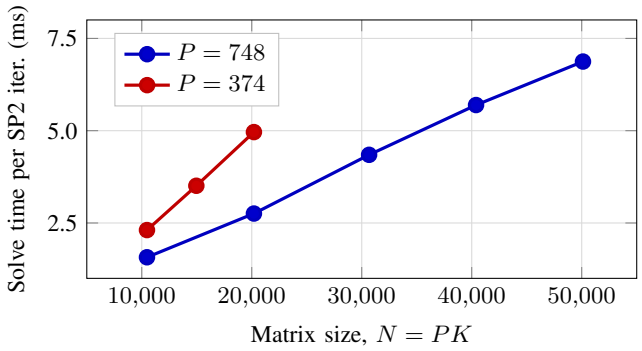


Fig. 4. Linear scaling behavior for increasing system size, N, when P remains fixed and K is increased. A threshold value of $\tau = 10^{-6}$ was used.

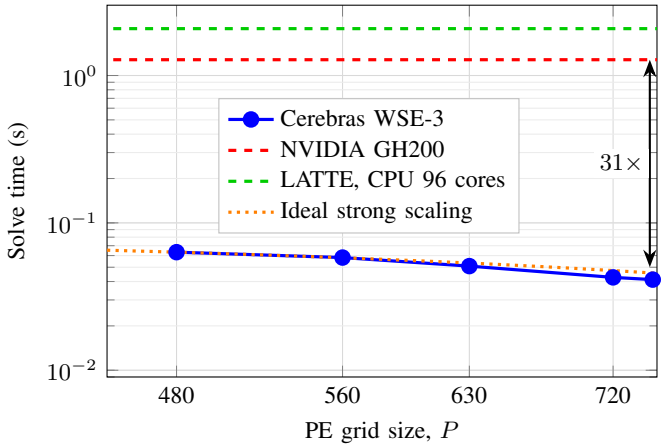


Fig. 5. Time-to-solution to compute density matrix D of a 10,056-atom water system for our sparse CUDA implementation and LATTE reference compared with the WSE asynchronous SP2 solver. With a PE-grid of 747×747, our sparse WSE-3 solver implementation represents a 31x speedup over our sparse CUDA reference. The threshold value was set at $\tau = 10^{-6}$.

Figure 5 shows WSE-3 strong scaling for a real, sparse $20{,}196 \times 20{,}196$ Hamiltonian (a 10,056-atom water system). We fix $N = 20{,}196$ so that P can be swept without the tile $K = N/P$ exceeding the 48 KB per-PE memory at small P. In the strong-scaling limit at $P = 747$, the WSE kernel is **31**× faster than a single-precision sparse GPU reference on an Nvidia GH200 Grace Hopper Superchip, and **45**× faster than double-precision Los Alamos Transferable Tight-Binding for Energetics (LATTE) [5] running on a CPU with 96 OpenMP

threads. The advantage is more pronounced for larger values of P.

V. Conclusion

We demonstrate a complete device-resident electronic structure solver on the Cerebras WSE-3 using a sparse SUMMA algorithm, fixed-capacity sparse storage, dynamic routing, and asynchronous communication overlap. Our measurements show ideal linear-scaling for fixed P and good strong scaling for a real 10,056-atom water system. We achieve a **31x speedup** over a sparse, SP2 solver running on a Nvidia GH200 Grace Hopper superchip in the strong scaling limit.

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/10.1007/978-90-481-2853-2_16

[2] ——, "Expansion algorithm for the density matrix," *Phys. Rev. B*, vol. 66, p. 155115, Oct 2002.

[3] 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

[4] F. G. Gustavson, "Two fast algorithms for sparse matrices: Multiplication and permuted transposition," *ACM Trans. Math. Softw.*, vol. 4, no. 3, p. 250–269, Sep. 1978. [Online]. Available: https://doi.org/10.1145/355791.355796

[5] M. Elstner and G. Seifert, "Density functional tight binding," *Philos. Trans. R. Soc., A*, vol. 372, no. 2011, p. 20120483, 2014.