Sandia National Laboratories

Exceptional service in the national interest

# An Optimistic View for HPC in the Future

Siva Rajamanickam

April 22, 2024

SAND2022-8365 PE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

NNSA

# Outline

- Context on Exascale and current work

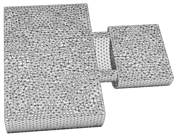
- Two examples from Exascale work
  - Batched Sparse Solvers
  - Multiprecision

- Two examples from post-Exascale work
  - HPCG on data flow hardware
  - Molecular Dynamics on data flow hardware

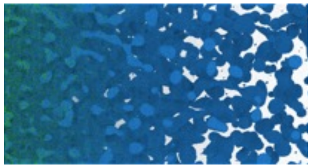
# Exascale Landscape

Applications

SNL ATDM SPARC
Reentry aerodynamics



SNL ATDM Empire
E&M Plasma Physics



SNL LAMMPS
Molecular Dynamics

My sandbox to play with algorithms, programming models, software
Sparse Linear Algebra, Dense Linear Algebra, Linear Solvers, Graph Algorithms, ....
Trilinos, Kokkos, Kokkos Kernels ...

El Capitan (LLNL)
AMD CPUs & GPUs

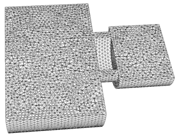
Frontier (ORNL)
AMD CPUs & GPUs

Aurora21 (ANL)
Intel Xeon CPUs,
Xe GPUs

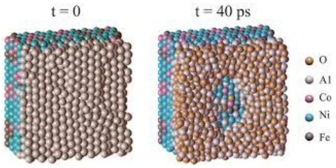
Astra (SNL)
ARM Architecture

# Post-Exascale Landscape

SNL ATDM SPARC
Reentry aerodynamics



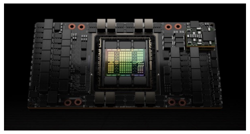
SNL ATDM Empire
E&M Plasma Physics



SNL LAMMPS
Molecular Dynamics

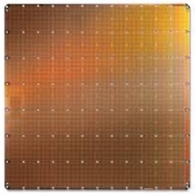
My sandbox to play with algorithms, programming models, software
Sparse Linear Algebra, Dense Linear Algebra, Linear Solvers, Graph Algorithms, ….
Trilinos, Kokkos, Kokkos Kernels …

AMD GPUs



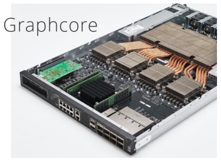
NVIDIA GPUs

ARM CPUs



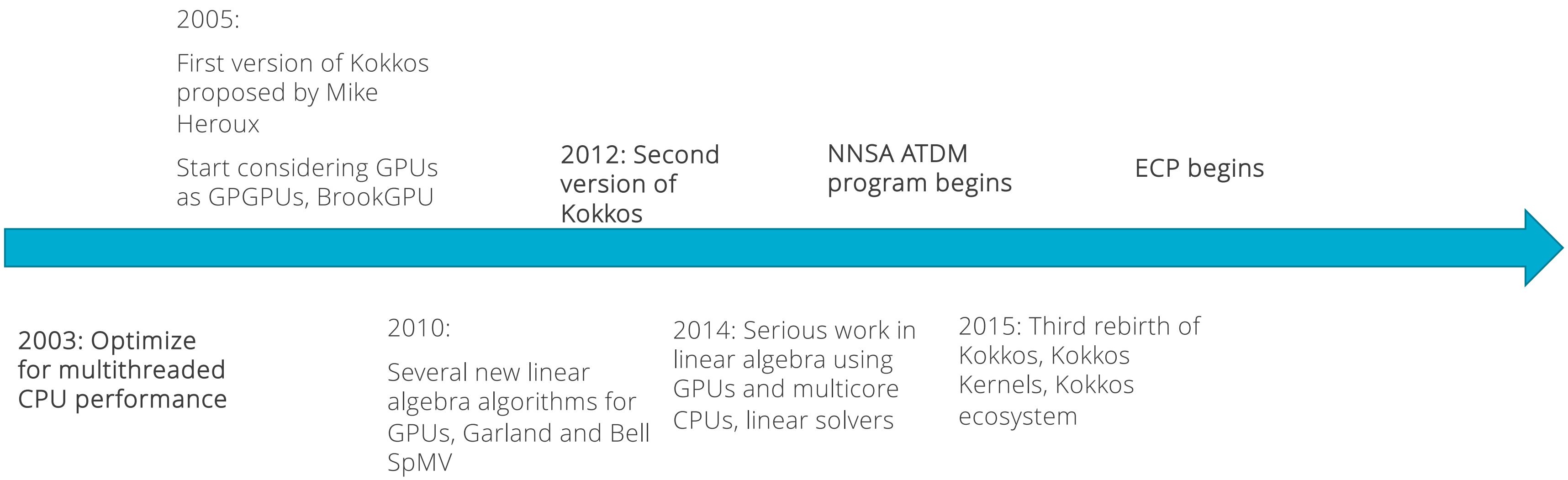
Cerebras WSE

Graphcore



SambaNova

# View of HPC since 2003: Change is the only constant



2003: Optimize for multithreaded CPU performance

2005:

First version of Kokkos proposed by Mike Heroux

Start considering GPUs as GPGPUs, BrookGPU

2010:

Several new linear algebra algorithms for GPUs, Garland and Bell SpMV

2012: Second version of Kokkos

2014: Serious work in linear algebra using GPUs and multicore CPUs, linear solvers

NNSA ATDM program begins

2015: Third rebirth of Kokkos, Kokkos Kernels, Kokkos ecosystem

ECP begins

**Almost two decades since the first set of "general purpose" kernels on HPIs to applications running at scale**

# Algorithm Design for Exascale Hardware

0. Identify performance critical kernels

- Call library based option when possible
  - Allows library developers to optimize the kernels

1. Develop portable algorithms when library based option not available

- Use a portable programming model
- Use architecture independent abstractions
- Pay attention to **memory layouts, hierarchical parallelism, synchronization costs**

2. Expose more parallelism with new algorithms

- **Edge-based, Non-zero based algorithms, Block-based algorithms, Clever use of sparsity** etc.

3. Use team level data structures and linear algebra kernels when possible

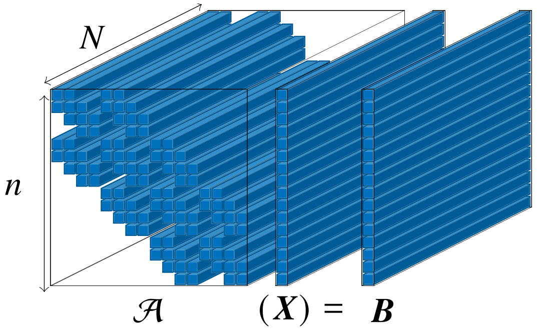
- **Optimize performance at all the hierarchical le**vels

Linear algebra community is very good at adapting to changes in hardware and to application needs

# Performance Portable Batched Sparse Solvers

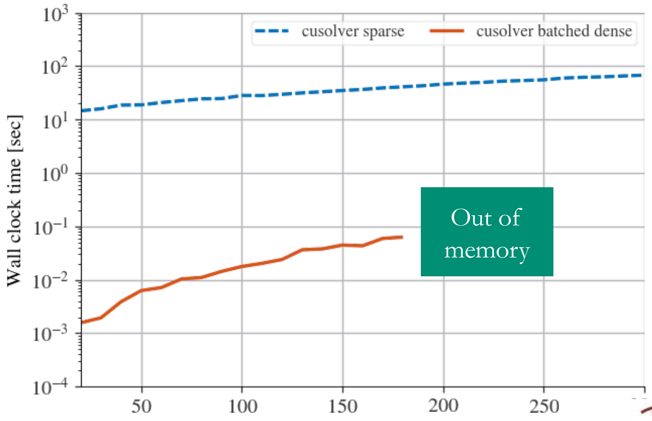
w/ Kim Liegeois, and L. Berger-Vergiat

Motivation: Numerical strategies for solving PDEs can lead **to large number (*N*) of small (*n*) similar sparse linear systems** to be solved ***independently*** (*N* >> *n*)



Two vendor strategies:

1. Loop over *N* systems, solve each with vendor sparse solver (**slow**)
2. Convert all systems to dense and use batched dense solver if available (**high-memory footprint**) -> **This is what the community has been focused on for 6-7 years**

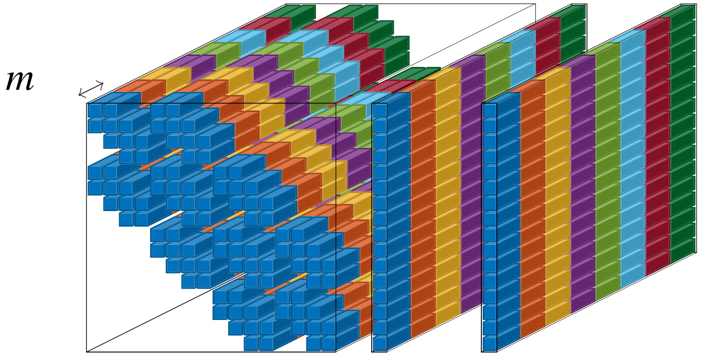


Out of memory

**No good options that can satisfy performance needs and be robust**

# New Strategy for Batched Sparse Krylov

- $N$ **systems gathered into groups of** $m$ **systems** $m << N$
  - E.g., on new Intel CPU architectures, one can use $m$=8 and solvers can use **vectorization to solve the group of systems at the team level**.
- **Hierarchical parallelism to solve** $m$ **systems** using a sparse iterative solver at team level

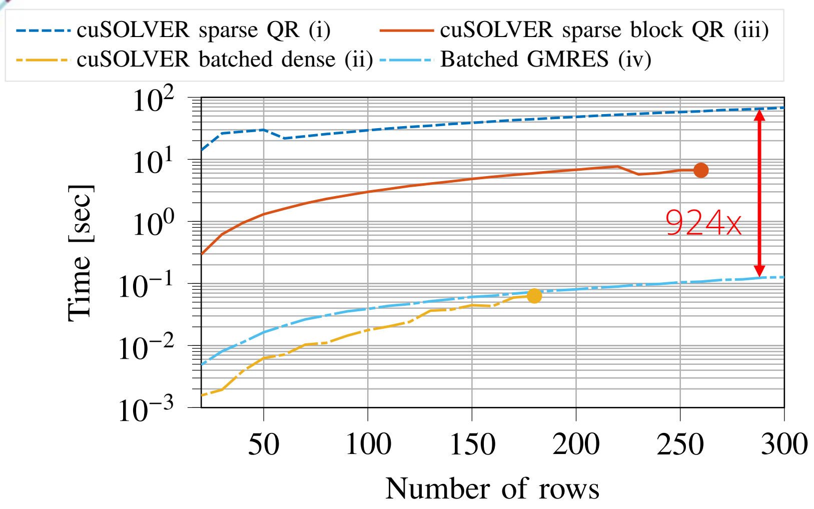


One group/team per color.

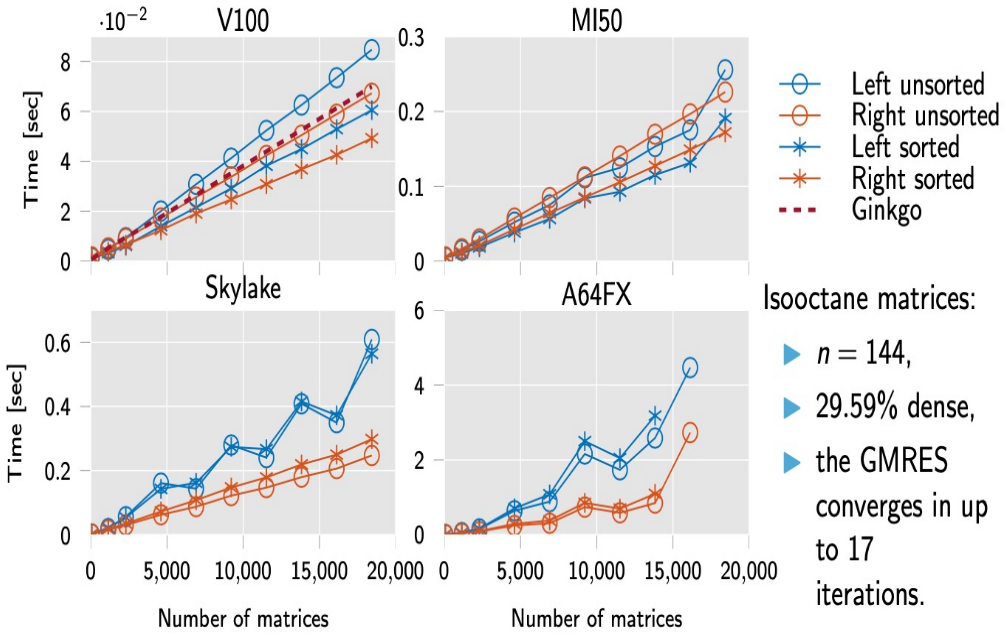
**Batched sparse solvers implemented using performance portable batched kernels at team level**

**Cool but crazy idea: Implementing a sparse Krylov solvers to solve many small systems with ~100 of kB!!**

# Batched Sparse Solver (GMRES) Performance Results



Batched GMRES can solve larger problems than batched dense solver from NVIDIA due to reduced memory



Isooctane matrices:

- $n = 144$,
- 29.59% dense,
- the GMRES converges in up to 17 iterations.

Performance portable to other architecture (same algorithm, different hyperparameters such as $m$)

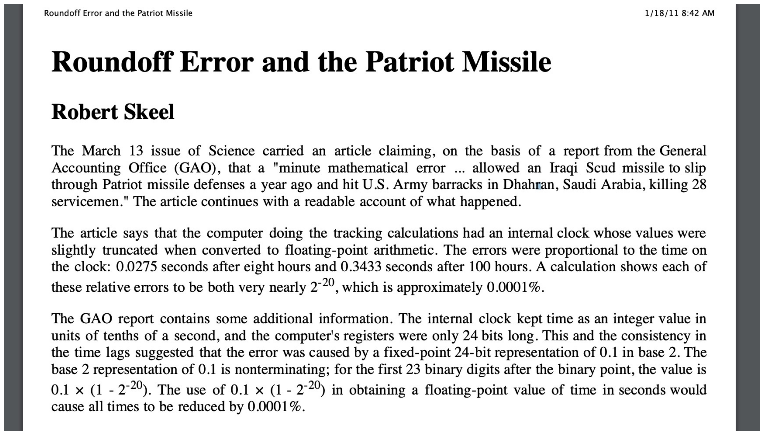
**Team batched sparse solvers efficient use of GPU resources yields two order magnitude speedup versus sequential use of sparse solvers**

# Multiprecision for Linear Solvers

Multiprecision xSDK
PI: S. Rajamanickam
Team: Boman, Loe, Glusa, Yamazaki

- Multiprecision hardware is becoming more common
- Avoid issues such as the 1991 roundoff error on Partiot missile
- **“Hence, with a less accurate truncated time of one radar pulse being subtracted from a more accurate time of another radar pulse, the error no longer cancelled.”**
- Sparse linear solvers: 64-bit accuracy with lower precision or multiprecision
- Option 1: GMRES-IR
  - Inner iterations using lower precision
  - Iterative refinement using higher precision
- Option 2: Lower precision preconditioners and higher precision solvers

**Can we get the higher performance offered by multiprecision hardware without sacrificing accuracy?**


Roundoff Error and the Patriot Missile 1/18/11 8:42 AM

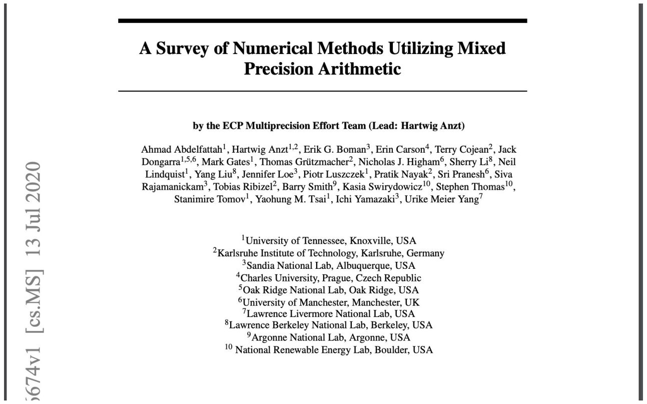
## Roundoff Error and the Patriot Missile

**Robert Skeel**

The March 13 issue of Science carried an article claiming, on the basis of a report from the General Accounting Office (GAO), that a "minute mathematical error ... allowed an Iraqi Scud missile to slip through Patriot missile defenses a year ago and hit U.S. Army barracks in Dhahran, Saudi Arabia, killing 28 servicemen." The article continues with a readable account of what happened.

The article says that the computer doing the tracking calculations had an internal clock whose values were slightly truncated when converted to floating-point arithmetic. The errors were proportional to the time on the clock: 0.0275 seconds after eight hours and 0.3433 seconds after 100 hours. A calculation shows each of these relative errors to be both very nearly $2^{-20}$, which is approximately 0.0001%.

The GAO report contains some additional information. The internal clock kept time as an integer value in units of tenths of a second, and the computer's registers were only 24 bits long. This and the consistency in the time lags suggested that the error was caused by a fixed-point 24-bit representation of 0.1 in base 2. The base 2 representation of 0.1 is nonterminating; for the first 23 binary digits after the binary point, the value is $0.1 \times (1 - 2^{-20})$. The use of $0.1 \times (1 - 2^{-20})$ in obtaining a floating-point value of time in seconds would cause all times to be reduced by 0.0001%.


## A Survey of Numerical Methods Utilizing Mixed Precision Arithmetic

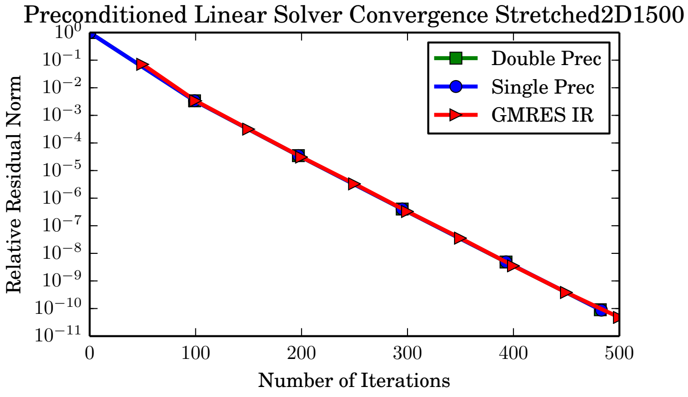
**by the ECP Multiprecision Effort Team (Lead: Hartwig Anzt)**

Ahmad Abdelfattah[1], Hartwig Anzt[1,2], Erik G. Boman[3], Erin Carson[4], Terry Cojean[2], Jack Dongarra[1,5,6], Mark Gates[1], Thomas Grützmacher[2], Nicholas J. Higham[6], Sherry Li[8], Neil Lindquist[1], Yang Liu[8], Jennifer Loe[3], Piotr Luszczek[1], Pratik Nayak[2], Sri Pranesh[6], Siva Rajamanickam[3], Tobias Ribizel[2], Barry Smith[9], Kasia Swirydowicz[10], Stephen Thomas[10], Stanimire Tomov[1], Yaohung M. Tsai[1], Ichi Yamazaki[3], Urike Meier Yang[7]

[1]University of Tennessee, Knoxville, USA
[2]Karlsruhe Institute of Technology, Karlsruhe, Germany
[3]Sandia National Lab, Albuquerque, USA
[4]Charles University, Prague, Czech Republic
[5]Oak Ridge National Lab, Oak Ridge, USA
[6]University of Manchester, Manchester, UK
[7]Lawrence Livermore National Lab, USA
[8]Lawrence Berkeley National Lab, Berkeley, USA
[9]Argonne National Lab, Argonne, USA
[10] National Renewable Energy Lab, Boulder, USA

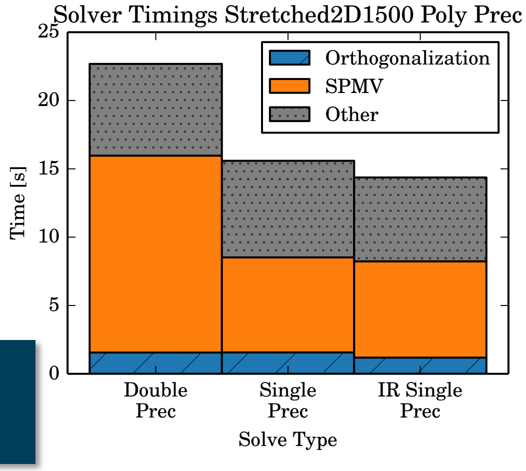
674v1 [cs.MS] 13 Jul 2020

# Multiprecision for Linear Solvers



- GMRES-IR = GMRES with iterative refinement. Run GMRES + preconditioning in FP32, refine in FP64 to get double-precision accuracy.
- **Convergence typically follows double precision GMRES!**
- Can also run GMRES-IR with single precision preconditioning. This example: Polynomial preconditioning of degree 40 for a Laplacian.

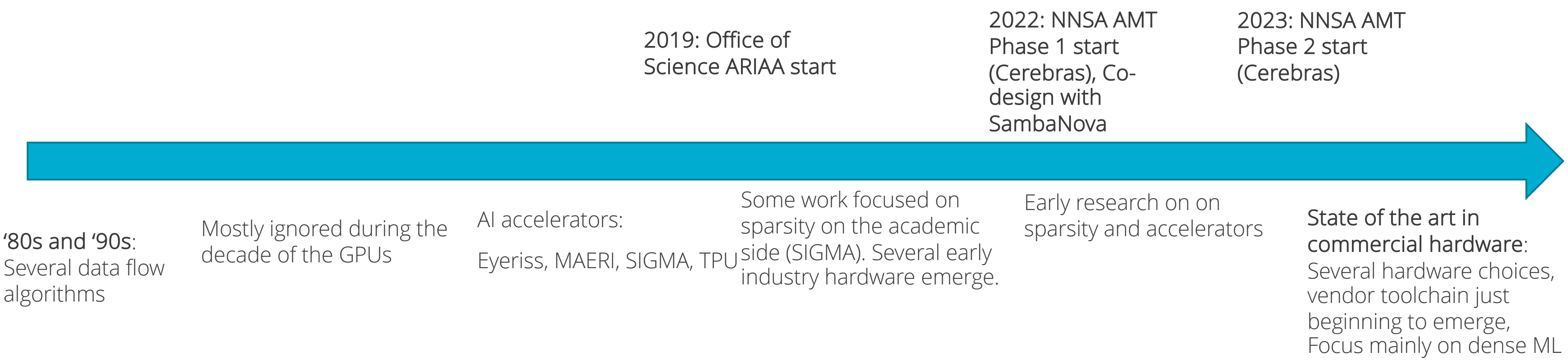
- **About 30% speedup over all-double precision.**
- Trilinos options to use single precision preconditioning with double precision solver.



**Iterative solver benefits for multiprecision is marginal.**

# Post-Exascale: Data flow path feels the same way it felt in 2005

- Data flow is not a new idea
  - Data flow hardware as it exists today is different from what has been considered in the past



**'80s and '90s**: Several data flow algorithms

Mostly ignored during the decade of the GPUs

AI accelerators:

Eyeriss, MAERI, SIGMA, TPU

2019: Office of Science ARIAA start

Some work focused on sparsity on the academic side (SIGMA). Several early industry hardware emerge.

2022: NNSA AMT Phase 1 start (Cerebras), Co-design with SambaNova

Early research on on sparsity and accelerators

2023: NNSA AMT Phase 2 start (Cerebras)

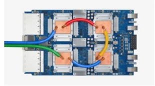
**State of the art in commercial hardware**: Several hardware choices, vendor toolchain just beginning to emerge, Focus mainly on dense ML

Recent work focuses primarily on use cases for machine learning, however promising path for traditional science simulations

# Turning the Titanic with a Leaf-blower: Influencing future hardware design

**Motivation : How can DOE/SNL use developments in industry and computer architectures to accelerate AI/HPC workflows?**

- There are a large number of AI/HPC companies producing specialized hardware.
- Computer architecture community has been focused on data flow acceleration for several years.
- Quantify the benefits of data flow hardware for DOE / SNL mission using simulations and mini-applications on early hardware
- Three parallel efforts
  - **Advanced Memory Technologies – Cerebras**
  - **ASC CSSE Co-design – SambaNova (not part of this talk)**
  - **Vanguard – II / Spectra – NextSilicon (not part of this talk)**



Cloud TPU v3

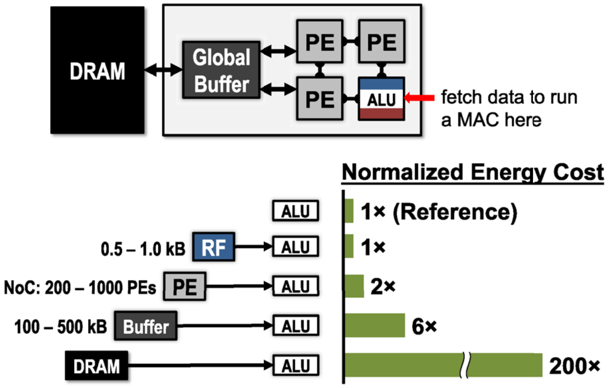
420 teraflops

128 GB HBM

Successful co-design with industry could lead to accelerating new classes of applications and low-energy accelerator alternative for other classes of applications

# Data flow Architectures are emerging as an alternative to traditional accelerators

- **The primary architectural features that distinguish “spatial” or “data flow” accelerators for ML from CPUs and GPUs are**
  - parallelism using hundreds to thousands of processing elements (**PEs**)
  - a fast network-on-chip connecting these and
  - use of private/shared scratchpad buffers for data reuse.
- **Differences from GPUs**
  - PEs communicate using Network-On-Chip (NOCs) without register file
  - Spatial/ Temporal /Spatio-temporal data reuse



Abstract data flow hardware and the potential benefits in energy cost

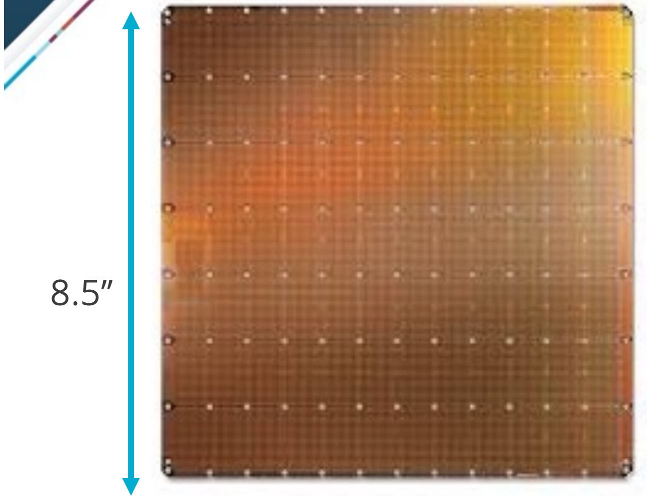
Data flow architectures enable a high-risk, high-reward path to accelerate HPC and ML applications

Need a rethink of algorithms, applications, programming models, compiler stack and the hardware features. Co-design is critical.

Image Courtesy: Hardware for Machine Learning: Challenges and Opportunities, Sze et al. CICC 2017

# Advanced Memory Technologies - Cerebras

Cerebras and NNSA Trilabs Team
**Overall AMT Lead**: Jim Laros,
**Cerebras AMT Lead**: Siva Rajamanickam



**Cerebras Wafer scale engine** is the largest chip ever built

- 2.6 trillion transistors
- 850,000 AI optimized cores
- 40 Gigabytes of On-chip Memory
- 20 PByte/s memory bandwidth

AI workloads are the primary target for the architecture

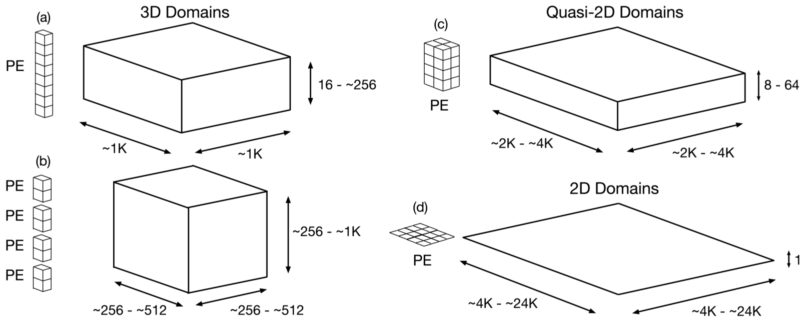
**Key questions for our co-design efforts**:

- How can we map scientific codes (Implicit solvers, Explicit codes, Monte Carlo, Molecular Dynamics, Unstructured multi-resolution problems)
- Is a 64-bit WSE possible?
- Implement two selected codes (implicit solvers, molecular dynamics)
- Sandia-led effort with LLNL, LANL and Cerebras

Explore co-design of algorithms for science simulations with a data flow hardware

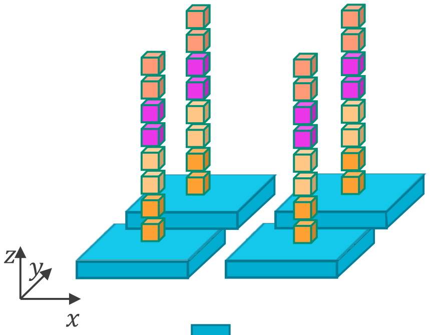
# Cerebras – HPCG Mapping Co-design

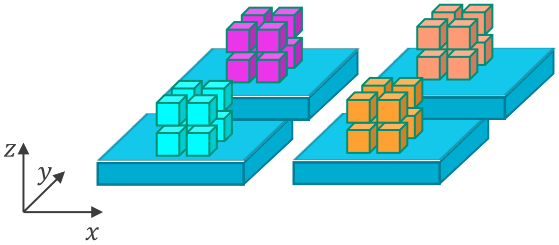
High Performance Conjugate Gradient (HPCG) benchmark as proxy for implicit solvers



Careful mapping of different domain aspect ratios to the 2D PE grid

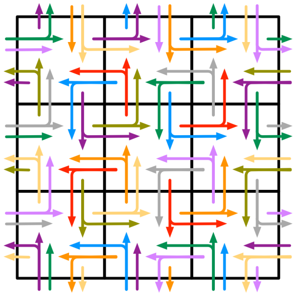
1×1×$Z$ gridpoints per tile



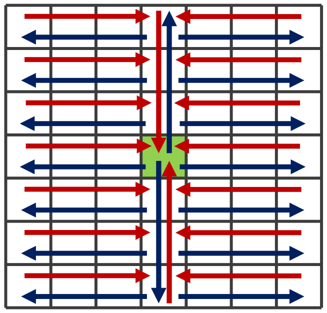


2×2×($^{Z}/_{4}$) gridpoints per tile

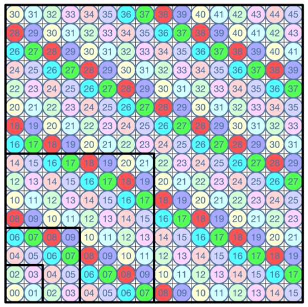
Careful mapping of computational kernels on the grid



SpMV



Dot



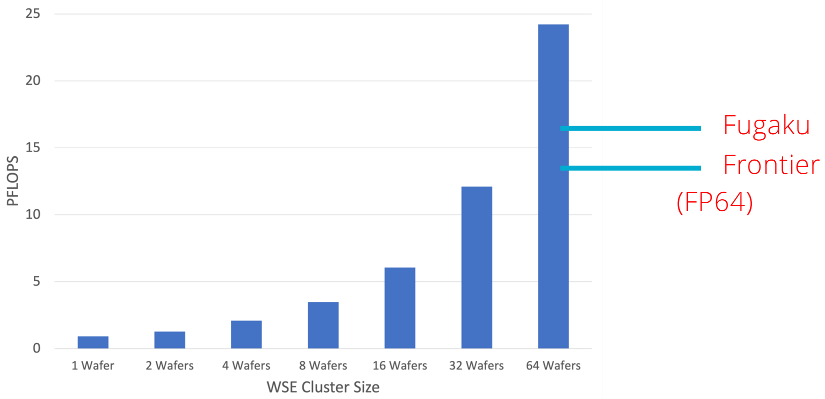
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
| 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 |
| 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 |

**Gauss-Seidel**

Mapping a sparse scientific code to the WSE is possible through co-design

# Cerebras – HPCG Performance Projections

## WSE Cluster Performance



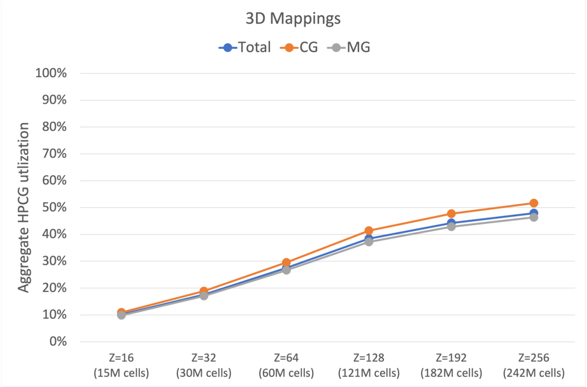
Fugaku

Frontier

(FP64)

Assumes non-overlapped IO
with CS-3 bandwidth levels

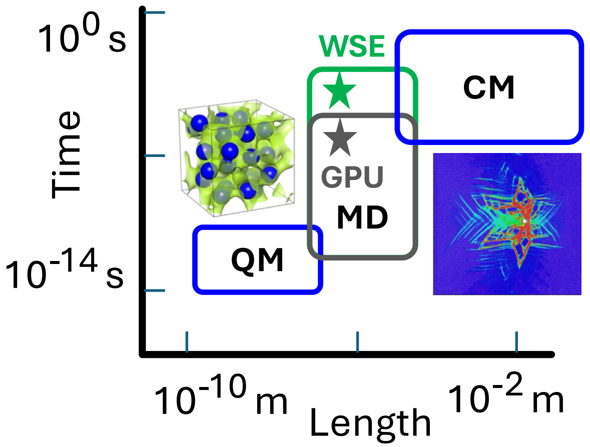
## Utilization per Iteration (3D: 1x1xZ with Rebalance)



Compliant with HPCG <8:1 aspect ratio condition

Simulation studies show potential to achieve ~50% utilization and reach exascale system performance with small number of wafers.

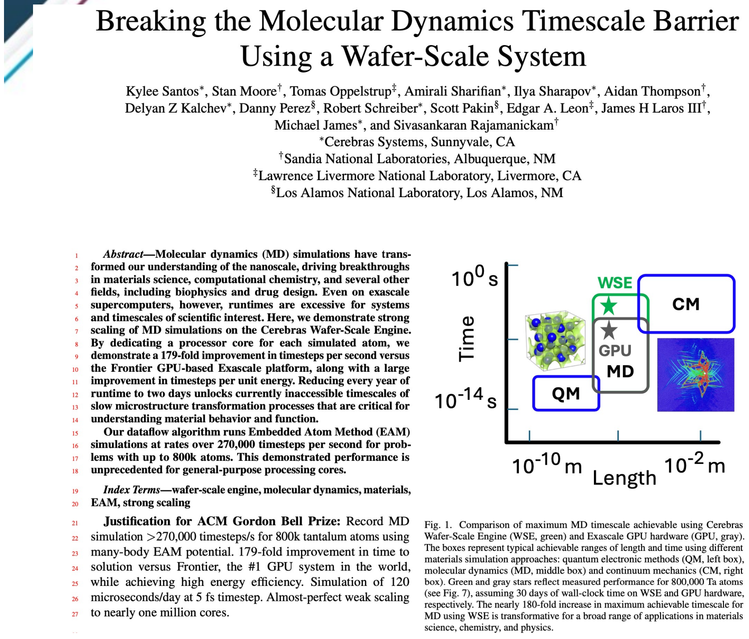
# Molecular Dynamics on Wafer Scale Engine



Molecular Dynamics simulations are limited by the timescales we can simulate using even the entire exascale system

- **Molecular Dynamics on Exascale Systems**
  - Weak Scaling simulations filling the GPUs with as many atoms as we can to simulate several billion or trillion atom systems
- **Grand Challenge: Timescale limitation**
- MD requires femtosecond timestepping whereas as important physical phenomena happen in 100 microseconds
- Month long exascale runs for few microseconds of simulated time
- **Latency limitations on exascale systems**
- **What we need?**
  - Communication bandwidth comparable to compute throughput
  - Communication latency comparable to clock frequency

# Molecular Dynamics on Wafer Scale Engine



## Breaking the Molecular Dynamics Timescale Barrier Using a Wafer-Scale System

Kylee Santos*, Stan Moore†, Tomas Oppelstrup‡, Amirali Sharifian*, Ilya Sharapov*, Aidan Thompson†,
Delyan Z Kalchev*, Danny Perez§, Robert Schreiber*, Scott Pakin§, Edgar A. Leon‡, James H Laros III†,
Michael James*, and Sivasankaran Rajamanickam†
*Cerebras Systems, Sunnyvale, CA
†Sandia National Laboratories, Albuquerque, NM
‡Lawrence Livermore National Laboratory, Livermore, CA
§Los Alamos National Laboratory, Los Alamos, NM

***Abstract*—Molecular dynamics (MD) simulations have transformed our understanding of the nanoscale, driving breakthroughs in materials science, computational chemistry, and several other fields, including biophysics and drug design. Even on exascale supercomputers, however, runtimes are excessive for systems and timescales of scientific interest. Here, we demonstrate strong scaling of MD simulations on the Cerebras Wafer-Scale Engine. By dedicating a processor core for each simulated atom, we demonstrate a 179-fold improvement in timesteps per second versus the Frontier GPU-based Exascale platform, along with a large improvement in timesteps per unit energy. Reducing every year of runtime to two days unlocks currently inaccessible timescales of slow microstructure transformation processes that are critical for understanding material behavior and function.**

**Our dataflow algorithm runs Embedded Atom Method (EAM) simulations at rates over 270,000 timesteps per second for problems with up to 800k atoms. This demonstrated performance is unprecedented for general-purpose processing cores.**

***Index Terms*—wafer-scale engine, molecular dynamics, materials, EAM, strong scaling**

**Justification for ACM Gordon Bell Prize:** Record MD simulation >270,000 timesteps/s for 800k tantalum atoms using many-body EAM potential. 179-fold improvement in time to solution versus Frontier, the #1 GPU system in the world, while achieving high energy efficiency. Simulation of 120 microseconds/day at 5 fs timestep. Almost-perfect weak scaling to nearly one million cores.

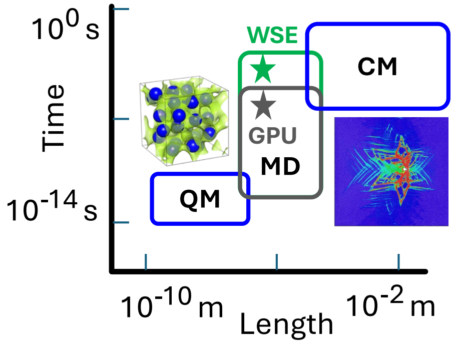
Fig. 1. Comparison of maximum MD timescale achievable using Cerebras Wafer-Scale Engine (WSE, green) and Exascale GPU hardware (GPU, gray). The boxes represent typical achievable ranges of length and time using different materials simulation approaches: quantum electronic methods (QM, left box), molecular dynamics (MD, middle box) and continuum mechanics (CM, right box). Green and gray stars reflect measured performance for 800,000 Ta atoms (see Fig. 7), assuming 30 days of wall-clock time on WSE and GPU hardware, respectively. The nearly 180-fold increase in maximum achievable timescale for MD using WSE is transformative for a broad range of applications in materials science, chemistry, and physics.

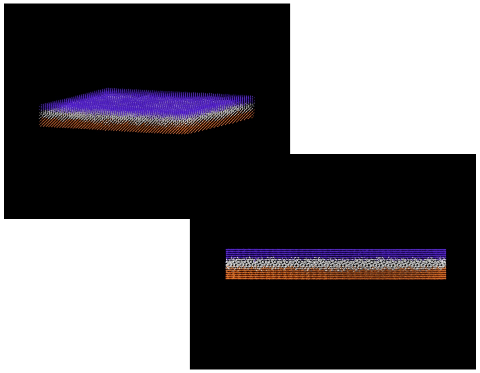
Gordon Bell Submission 2024 under review

**Can we accelerate Molecular Dynamics using Cerebras WSE?**

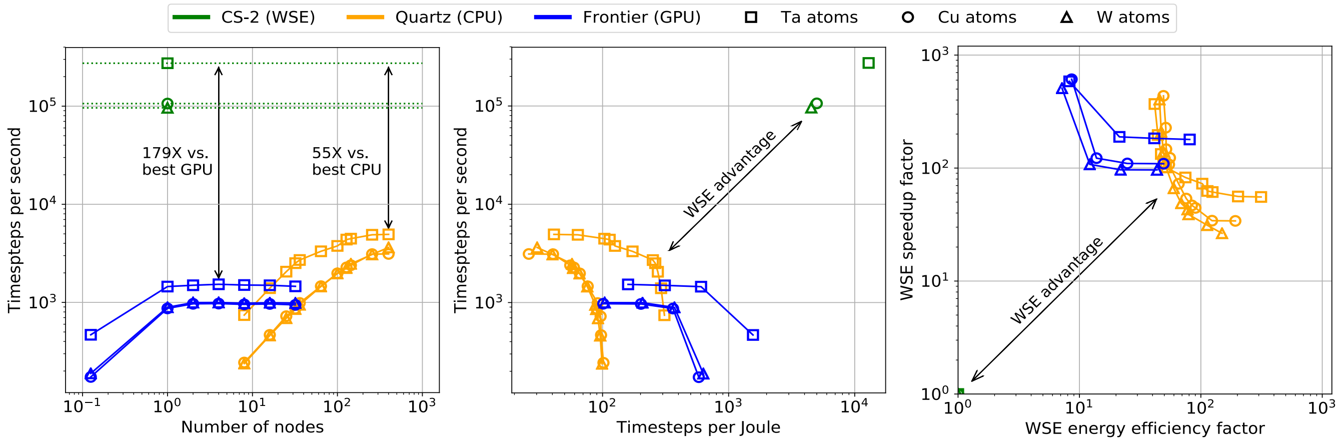
## Key questions for our co-design efforts:

- Can we do strong scaling can we accelerate timesteps per second to what can be achieved on an exascale system?
- Can we study and understand the performance on a data flow hardware?
- What are the energy benefits for using a WSE?





# Early Results: Molecular Dynamics on Wafer Scale Engine



## Key results:

- 179x speedup compared to best time achieved on Frontier simulations and 55x faster than best CPU times
- **One-to-two orders of magnitude energy efficiency** compared to CPUs and GPUs
- **Extreme strong scaling** of one atom per-core used. Could go further down to one atom for many cores
- Simulation of 120 microseconds/day at 5 fs timestep
- A simple performance model allows us to project performance up to 3% accuracy

Exciting Results using the Wafer Scale Engine on a traditional scientific simulation

# Summary

- Two examples from the Exascale efforts
  - Two orders of magnitude improvement with new algorithms that map well to hierarchical parallelism, new memory layouts, and application needs
  - Not so much performance benefit using multiprecision if we want to retain the accuracy

- Two examples from the post-exascale co-design efforts
  - Huge upside for HPCG performance based on performance projections
  - 179x speedup on molecular dynamics compared to Exascale systems

- HPC community is very good at using whatever the hardware architects give us
  - We will join the AI hardware swim lane instead of beating them with the custom hardware swim lane

# Additional information

- Contact: srajama@sandia.gov