A Naive D3Q27 Lattice Boltzmann Implementation on the Cerebras CS-3: Comparison with GPUs

Kouki Sakai†, Saho Orihara†, Takashi Shimokawabe††, and Takaaki Miyajima† e-mail:takaaki.miyajima@cs.meiji.ac.jp
†Meiji University (Japan), ††Information Technology Center, The University of Tokyo (Japan)

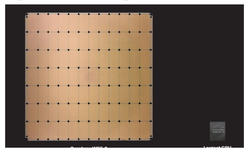
Overview

- LBM is a stencil-based CFD method: the domain is divided into a lattice, each element is updated independently, then exchanges data with its neighbors.
- This pattern maps well onto the Cerebras CS-3, which offers large distributed on-chip memory and high-bandwidth inter-PE communication.
- We implement a naive 3D LBM (D3Q27) on the CS-3 by assigning a $1 \times 1 \times n$ lattice to each PE — to our knowledge, the first 3D LBM implementation and performance study on this platform.
- We measure performance in LUPS and weak scaling: on the largest domain, the CS-3 achieves 83.74 GLUPS at a parallel efficiency of 1.0, outperforming the A100 and GH200 by 17.4x and 8.5x — showing wafer-scale computing is a promising platform for memory-bound stencil computations.

Cerebras Wafer-Scale Engine 3 (WSE-3)

The entire 300 mm wafer is used as one chip

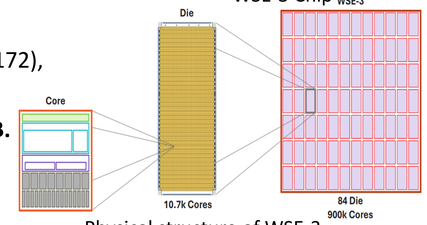
- 84 dies are interconnected with a parallel interface called Cross Scribe Line Connections.
- Process rule: TSMC 5 nm, chip area: 46,225 mm^2.
- Running at 750MHz for scientific computation.



WSE-3 Chip

Specifications

- Total number of PEs: 893,064 (762x1172), **effective: 883,350 (755x1170).** [1]
- Total amount of on-chip SRAM: **44 GB.**
- Memory bandwidth: **21 PB/s.**
- SGEMM performance: **349.0 TFlop/s**

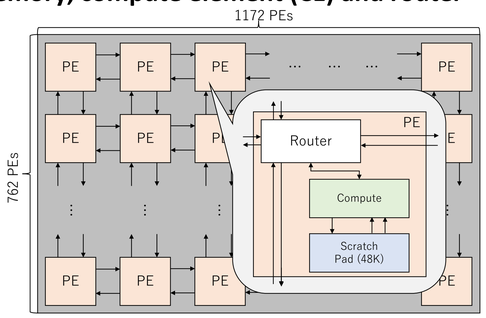


Physical structure of WSE-3

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.
- 893,064 Processing elements (PEs) with 2-D mesh topology.

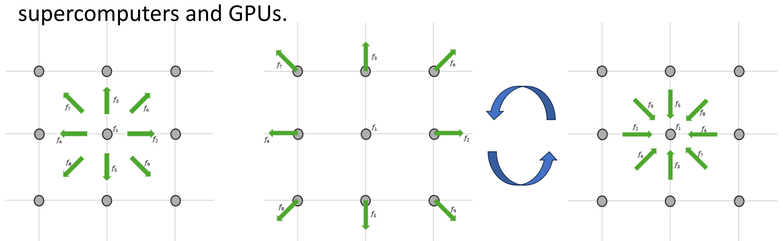


PE Layout and Architecture

Lattice Boltzmann Method

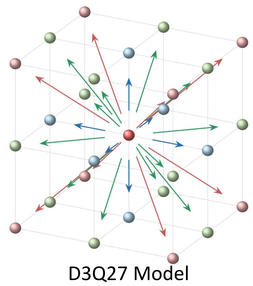
Algorithm overview

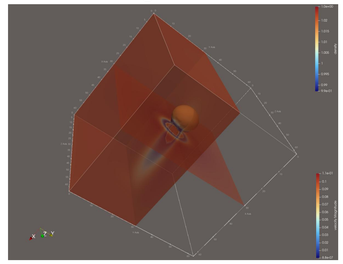
- Fundamental idea: streaming and collision of virtual particles at lattice points[2].
- Its inherent parallelism, where operations at each lattice point are independent, enables high computational efficiency on parallel architectures such as supercomputers and GPUs.



D3Q27 Model

- We employ the D3Q27 model, in which each lattice point holds 27 discrete velocities: a rest particle and 26 directions to its neighbors.

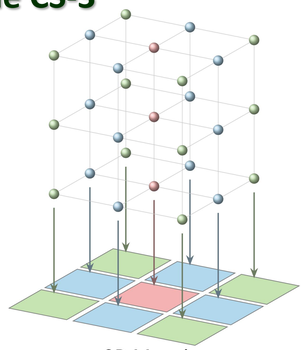

D3Q27 Model


Execution Results

Implementation on the CS-3

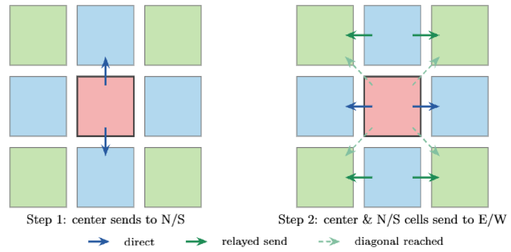
3D mapping

- A design in which a single PE calculates a $1 \times 1 \times n$ region.
- Since lattices adjacent along the z-axis are located within the same PE, communication between PEs is not required.
- Communication is required only within the x-y plane, and each grid point exchanges data with the eight directions: north, south, east, west, and the diagonals.


3D Mapping

Two step communication

- The WSE-3 can only communicate in the north, south, east and west directions.
- LBM communicates diagonally, so it is divided into two steps.



Data-Driven Task Execution

- Send is asynchronous (.async = true) — the PE issues the transfer and moves on without waiting for it to complete.
- Receive is event-driven: the relay logic inside recv_north fires automatically once a neighbor's data has fully arrived.
- Together, this overlaps fabric communication with local computation.

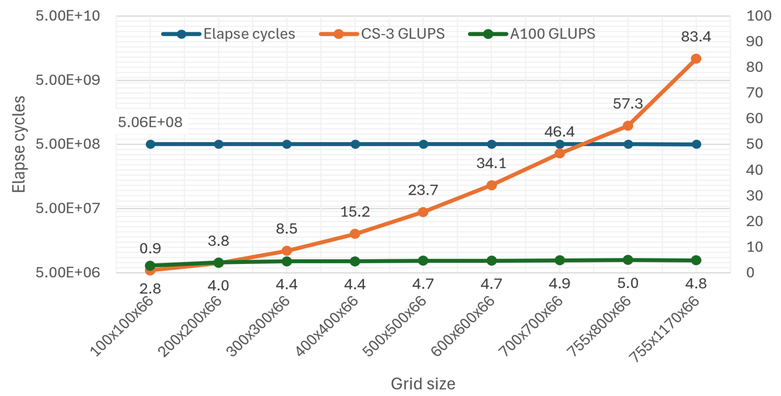
```
task send_north_south() void {
 // outbound channels to north/south neighbor
 const fab_north_dsd = @get_dsd(fabout_dsd, .{ .fabric_color = send_north_color, ... });
 const fab_south_dsd = @get_dsd(fabout_dsd, .{ .fabric_color = send_south_color, ... });
 @fmovs(fab_north_dsd, mem_s_dsd, .{ .async = true, ... });  // send north, non-blocking
 @fmovs(fab_south_dsd, mem_n_dsd, .{ .async = true, ... });  // send south, non-blocking
 @block(send_north_south_task_id);  // suspend until re-activated next step
}
// North-neighbor receive complete -> relay diagonal components to the EW send buffer
task recv_north(val: f32) void {
 rn_vals[n_recv_count] = val; n_recv_count += 1;  // buffer the incoming value
 if (n_recv_count == ns_n_slab_size) {          // all data from the north neighbor has arrived
  send_e_buffer[...] = rn_vals[...];          // NE component -> relay east
  send_w_buffer[...] = rn_vals[...];          // NW component -> relay west
  if (num_recv == num_neighbors)
   @activate(start_next_gen_task_id);         // once all directions are in, launch the next compute task
 }
}
```

send (async) and receive (event-driven relay) tasks

Measurement Results

Weak Scaling

- Using a $1 \times 1 \times 66$ lattice with 1 PE, we increased the number of PEs from 100×100 to 755×1170 while keeping the workload per PE constant.
- The maximum grid size that a single PE could occupy was 66 lattice units.



- Using 100×100 as baseline, efficiency stayed near-ideal (1.0), degrading by at most ~3×10^{-5} (0.003%) at maximum scale.
- Cycle count is scale-independent because LBM's locality matches CS-3's 2D mesh, avoiding added communication overhead.

Performance comparison of CS-3 vs. A100 vs. GH200 (FP32)

- Problem domain size: $755 \times 1170 \times 66$.
- Compared using **LUPS** (Lattice Updates Per Second), a standard metric in the LBM field representing the number of lattice points updated per second.

	CS-3	A100	GH200
Environment	ALCF, SDK 2.10.0	40GB PCIe, nvhpc/24.11	Miyabi (Univ. Tokyo), nv-hpcx/25.9
Build	CSL	nvcc -O3 -std=c++14 --expt-extended-lambda -arch=sm_80	nvcc -O3 -std=c++14 --expt-extended-lambda -arch=sm_90
GLUPS	**83.74**	**4.81**	**9.84**
Speedup of CS-3	(1.0x)	**17.4x**	**8.5x**

- CS-3 achieved **17.4**× the performance of A100 and **8.5**× that of GH200.
- Since the LBM is a memory-bound application, GH200 is about twice as fast as A100.

Future work

- Performance optimization, including communication tuning and the use of SIMD unit on CS-3.

Acknowledgement and references

[1] Sean Lie, Co-founder and CTO, Cerebras Systems Hot Chips 2024
https://hc2024.hotchips.org/assets/program/conference/day2/72_HC2024.Cerebras.Sean.v03.final.pdf
[2] Krüger et al., *The Lattice Boltzmann Method: Principles and Practice*, Springer, 2017

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.
An award of computer time was provided by the U.S. Department of Energy's (DOE) Innovative and Novel Computational Impact on Theory and Experiment (INCITE) Program. This research used supporting resources at the Argonne and the Oak Ridge Leadership Computing Facilities. The Argonne Leadership Computing Facility at Argonne National Laboratory is supported by the Office of Science of the U.S. DOE under Contract No. DE-AC02-06CH11357.