A practical introduction to programming the Cerebras CS-2 for HPC workloads

Nick Brown
EPCC University
of Edinburgh

Joseph Lee
EPCC University
of Edinburgh

Justs Zarins
EPCC University
of Edinburgh

David Kacs
EPCC University
of Edinburgh

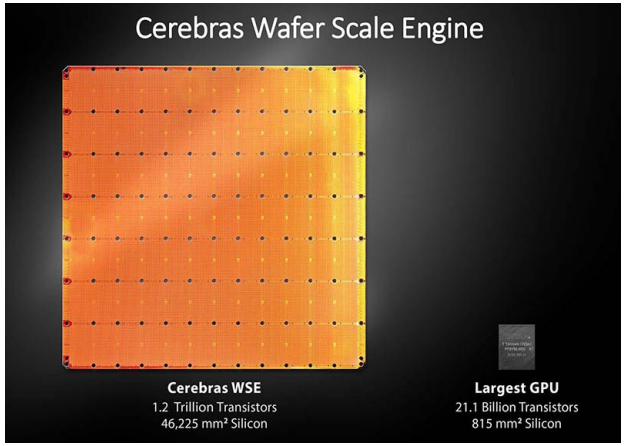
ExCALIBUR

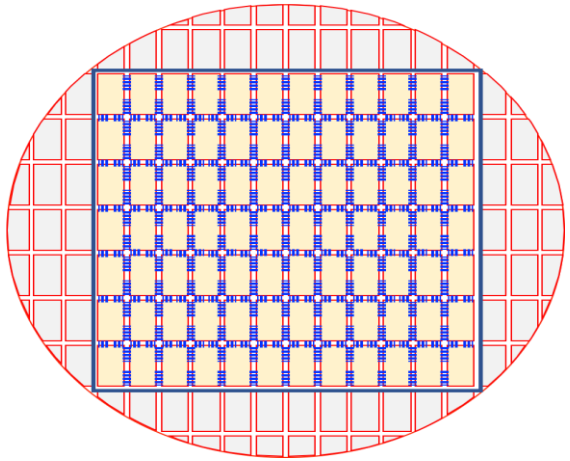
epcc

cerebras

Motivation

- Many HPC codes scale poorly across nodes
  - Such as FFT-based solvers, particle simulators, non-linear problems with iterative solvers
  - The CS-2 has a fabric that is **high bandwidth** and **low-latency**, allowing for excellent parallel efficiency for non-linear and highly communicative codes
    - The CS-2 system has **850k cores** and can fit problems on an individual chip that take tens to hundreds of traditional small compute nodes.
    - Each core is individually programmable





- Many HPC codes are constrained by data accesses
  - Such as Stencil based PDE solvers, linear algebra solvers, signal processing, sparse tensor math, big data analysis
  - The CS-2 system has **40 GB of SRAM** uniformly distributed across the wafer that is **1 cycle** away from the processing element (PE)
    - This speeds up memory access by orders of magnitude
  - The CS-2 system is capable of **1.2 Tb/s bandwidth** onto the chip
    - Streaming data onto the chip as required

Example: Accelerating seismic modelling

The Ask: Deliver improved performance on seismic processing workload by leveraging Cerebras System's unique architecture and massive on-chip memory.

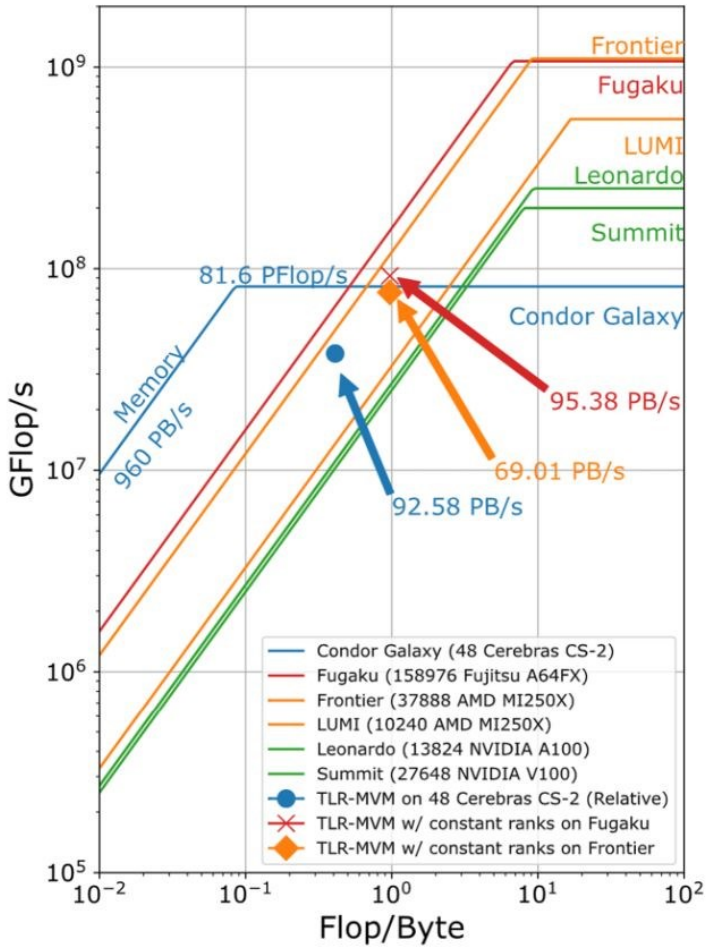
Challenge: Seismic processing algorithms are typically memory-bound problems, limited by the memory access speeds of other architectures. Researchers were challenged to re-design an algorithm to take advantage of Cerebras hardware to improve performance.

What was done: Researchers used the Cerebras SDK to re-design a Tile Low-Rank Matrix-Vector Multiplication (TLR-MVM) algorithm to be optimized for Cerebras CS-2.

Outcome: Achieved a record sustained memory bandwidth of 92.58 Petabytes per second (PB/s) through the implementation of a TLR-MVM kernel that is uniquely tailored to exploit the architecture of the CS-2 systems.

More details at https://sc23.supercomputing.org/2023/08/a-look-at-the-2023-gordon-bell-prize-finalists/

Cerebras CS-2 achieves real memory bandwidth performance that rivals best-case performance on world-leading supercomputers



Example: Accelerating CFD

- Cerebras and NETL demonstrate significant performance on a Cerebras CS-2 compared to CPU or GPU
- CS-2 470x faster than Joule 2.0 supercomputer

Disruptive Changes in Field Equation Modeling
A Simple Interface for Wafer Scale Engines

Mino Woo[1,2], Terry Jordan[1], Robert Schreiber[3], Ilya Sharapov[3], Shaheer Muhammad[3], Abhishek Koneru[3], Michael James[3*] and Dirk Van Essendelft[1*]

[1] National Energy Technology Laboratory, Morgantown, 26505, WV, United States.
[2]Oak Ridge Institute for Science and Education, Oak Ridge, 37830, TN, United States.
[3]Cerebras Systems Inc, Sunnyvale, 94085, CA, United States.

*Corresponding author(s). E-mail(s): michael@cerebras.net; dirk.vanessendelft@netl.doe.gov;

Abstract

We present a high-level and accessible Application Programming Interface (API) for the solution of field equations on the Cerebras Systems Wafer-Scale Engine (WSE) with over two orders of magnitude performance gain relative to traditional distributed computing

single wafer-scale system for the solution by BiCGStab of a

tation on a
ssor

chreiber*, Michael Morrison*,
dhava Syamlal† and Michael James*

nia, USA
et

, West Virginia, USA
ov

nsylvania, USA

bandwidth and high communication
ary performance limiters.
and communication systems struggle
processing performance. In 2016 the
atios for both memory and interconnect
in the hundreds, and the flops needed
mory or network latencies were in the
10,000 to 100,000 range, with the trend going higher;
see Figure 1.

Photo: Christian Kuhna, CC BY 3.0

Tutorial learning objectives

- This tutorial is open to everybody, regardless of experience with HPC and accelerators
  - Is practically driven, where we will walk-through key concepts on the machine itself, and then you can explore the concepts more independently via a series of walk-throughs

1. Understand the CS-2 architecture & core concepts
   - We will explore the hardware, how it is designed the and key terminology
2. Get started with the Cerebras SDK
   - Exploring key concepts for writing HPC codes for the CS-2 and understanding how to build these
3. Writing multi-PE codes for the CS-2
   - Exploring how we can run over multiple PEs, have these communicate together and leverage the Cerebras collective communications library
4. Write programs for the CS-2 based on the hands-on activity
   - Bringing the concepts together and using these to develop a real code for the CS-2 and optimise it
5. Running on the CS-2 simulator
   - Much of development is done with a simulator, we will explore how to use this and leverage it when writing code
6. Running on a real CS-2 machine
   - Ultimately we want to accelerate our HPC codes on the CS-2, we will run our hands-on exercises on a real CS-2 machine

We are also happy to discuss your own applications and how these might be ported to the architecture

Session plan

Time	Title	Type
9:30 – 9:35	Introduction, welcome and objectives	Presentation
9:35 – 9:55	An Overview of the CS-2 architecture	Presentation
9:55 – 10:00	Logging onto the CS-2 machine	Practical
10:00 – 11:00	Practical walkthrough: Intro to SDK (CSL + Host Runtime)	Walk through
11:00 – 11:25	Hands-on practical activity one & coffee	Practical
11:25 – 11:45	Practical walkthrough: Using multiple PEs and communication	Walk through
11:45 – 12:15	Hands-on practical activities two and three	Practical
12:15 – 12:25	Wash up from practical activities	Presentation
12:25 – 12:30	Conclusions and audience next steps to continue working with the technologies	Presentation

We will be doing the hands on and walk throughs sdf-cs1, more information about how to connect to that a little later

Materials and the CS-2 community

- We will remind people as we progress through the session
- All materials for this tutorial are open source and can be found at
  - https://github.com/EPCCed/cs2-sdk-training
- More generally if you wish to continue exploring this after the tutorial finishes
  - https://sdk.cerebras.net/csl
- There is a CS-2 developer community that you can join
  - Roughly monthly meetings
  - Forums: discourse.cerebras.net