Welcome!

Thank you for joining us today! As we wait for everyone to get settled, we'd like to bring a few things to your attention:

1. This webinar is being recorded. The recording will be available via the Neocortex webinar webpage. Slides will also be made available online.
2. There will be 50 minutes of presentations followed by Q&A. To maintain a quality experience for everyone, please mute your microphone during the presentations.
3. We hope you will participate in this interactive webinar by:
    - Asking questions to our team via the Q&A Zoom feature.
   - These questions will seed the Q&A session in the final 10 minutes.
4. This webinar abides to the Neocortex code of conduct.

NEOCORTEX

Neocortex Code of Conduct

Neocortex has an external code of conduct which represents our commitment to providing an inclusive and harassment-free environment in all interactions regardless of race, age, ethnicity, national origin, language, gender, gender identity, sexual orientation, disability, physical appearance, political views, military service, health status, or religion. The code of conduct extends to all *Neocortex*-related events, services, and interactions.

Code of Conduct: www.cmu.edu/psc/aibd/neocortex/neocortex-code-of-conduct

Contact:

Neocortex ombudspersons:

- Paola A. Buitrago, PSC, Carnegie Mellon University (paola@psc.edu)
- Sergiu Sanielevici, PSC, Carnegie Mellon University (sergiu@psc.edu)

Neocortex Overview and Spring 2023 Call for Proposals

Paola A. Buitrago

Principal Investigator & Project Director, Neocortex
Director, AI and Big Data, Pittsburgh Supercomputing Center
Carnegie Mellon University
University of Pittsburgh

February 28, 2023

NEOCORTEX
Unlocking Interactive AI for Rapidly Evolving Research

Supported by OAC 2005597

Questions We are Addressing Today

1. What is the Neocortex program and what are its goals?
2. What is the innovative hardware a researcher can get access through Neocortex?
3. What type of applications are supported by the Neocortex system?
4. How can I gain access to the Neocortex program?
    - What is the evaluation criteria for proposals submitted to the Neocortex program?
    - What would be expected from a Neocortex researcher?
    - What can I expect as Neocortex researcher?
5. How to get additional information or support from the Neocortex team?

Speakers

Dirk T. VanEssendelft

SDK researcher, Neocortex

HPC, AI, and Data Scientist, NETL

Leighton Wilson

SDK support collaborator, Neocortex

HPC Solutions Engineer, Cerebras

Claire Zhang

ML support collaborator, Neocortex

ML Solutions Engineer, Cerebras

Paola A. Buitrago

Principal Investigator, Neocortex

Director of AI and Big Data, PSC

Questions We are Addressing Today

1. What is the Neocortex program and what are its goals?
2. What is the innovative hardware a researcher can get access through Neocortex?
3. What type of applications are supported by the Neocortex system?
4. How can I gain access to the Neocortex program?
    - What is the evaluation criteria for proposals submitted to the Neocortex program?
    - What would be expected from a Neocortex researcher?
    - What can I expect as Neocortex researcher?
5. How to get additional information or support from the Neocortex team?

The Neocortex Program

NEOCORTEX

Context – NSF Solicitation

NSF Solicitation – 19-587

Advanced Computing Systems and Services: Adapting to the Rapid Evolution of Science and Engineering Research

“The intent of this solicitation is to request proposals from organizations to serve as service providers … to provide advance cyberinfrastructure (CI) capabilities and/or services … to support the full range of computational- and data-intensive research across all science and engineering (S&E).”

Two categories:

- Category I, Capacity Systems: production computational resources.
- **Category II, Innovative Prototypes/Testbeds: innovative forward-looking capabilities deploying *novel technologies, architectures, usage modes*, etc., and exploring new target applications, methods, and paradigms for S&E discoveries.**

NEOCORTEX

Context – NSF Award

Acquisition and operation of *Bridges, Bridges-AI*, *Bridges-2,* and ***Neocortex*** are made possible by the National Science Foundation:

NSF Award OAC-2005597 ($12.25M awarded to date):
Category II: Unlocking Interactive AI Development for Rapidly Evolving Research

Cerebras and HPE delivered *Neocortex*

Context – Project Goals

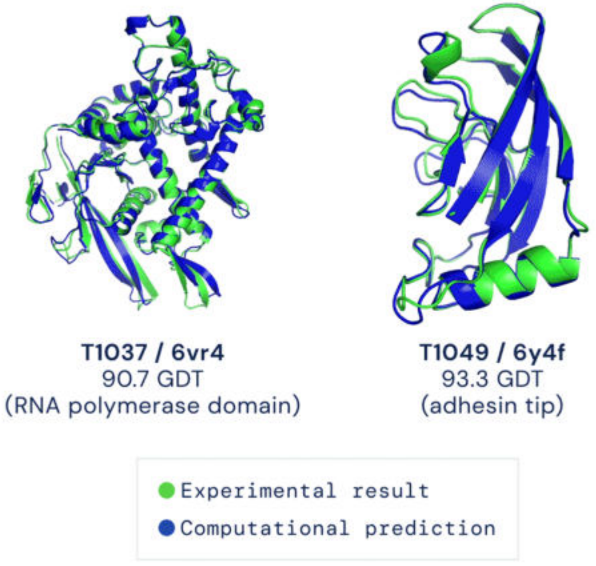
***Neocortex*, Unlocking Interactive AI Development for Rapidly Evolving Research**

A new NSF funded advanced computing project with the following goals:

- Deploy *Neocortex* and offer the national open science community revolutionary hardware technology to accelerate AI training at unprecedented levels.
- Explore, support and operate *Neocortex* for 5 years.
- Engage a wide audience and foster adoption of innovative technologies.

Driving Use-Cases

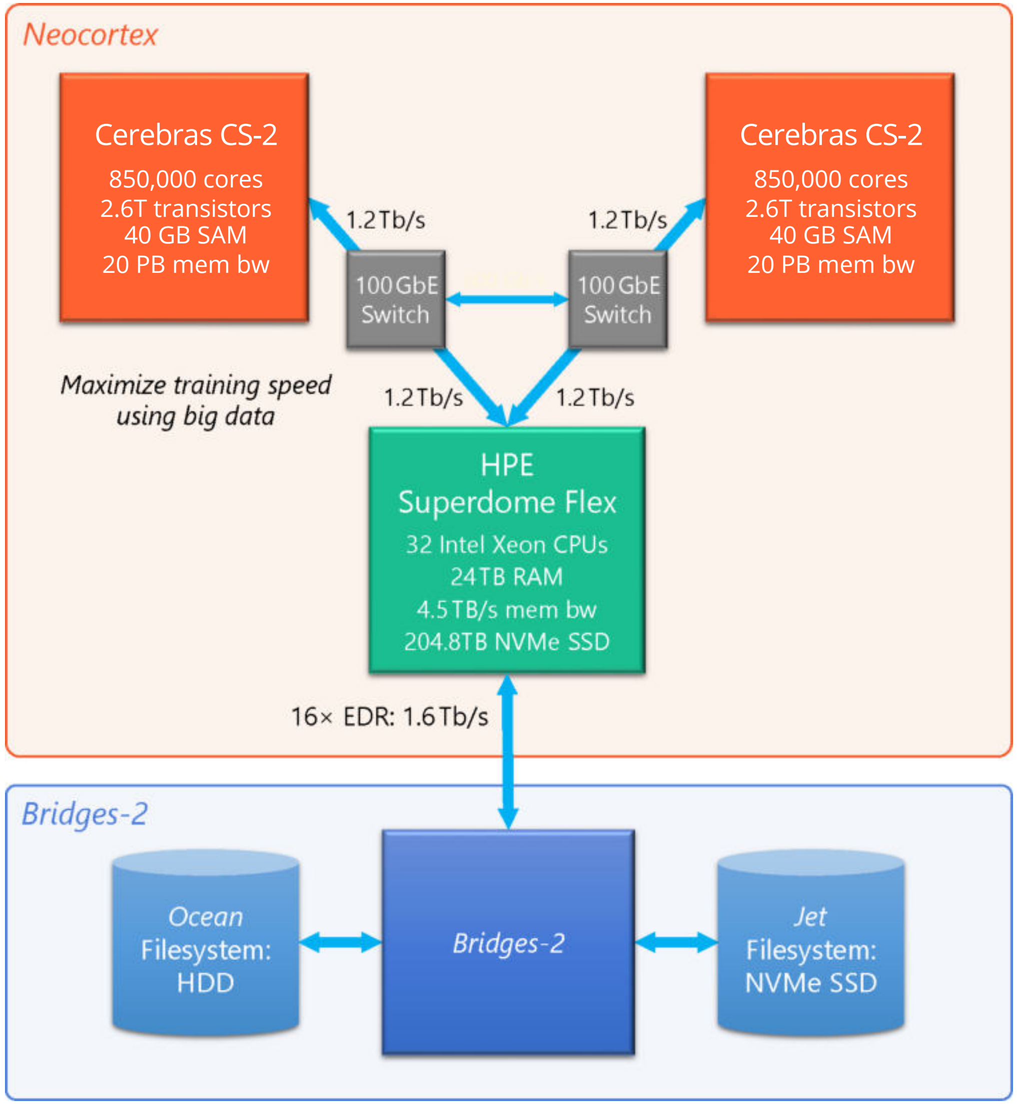
- Transform and accelerate AI-enabled research
- Development of new and more efficient AI algorithms and graph analytics
- Foster greater integration of artificial deep learning with scientific workflows
- Democratize access to game changing compute power
- Explore the potential of a groundbreaking new hardware architecture
- Support research needing large-scale memory (genomics, brain imaging, simulation modeling)
- Augmenting traditional computational science with rapidly-evolving methodologies and technologies
- User-centric and interactive computing modalities



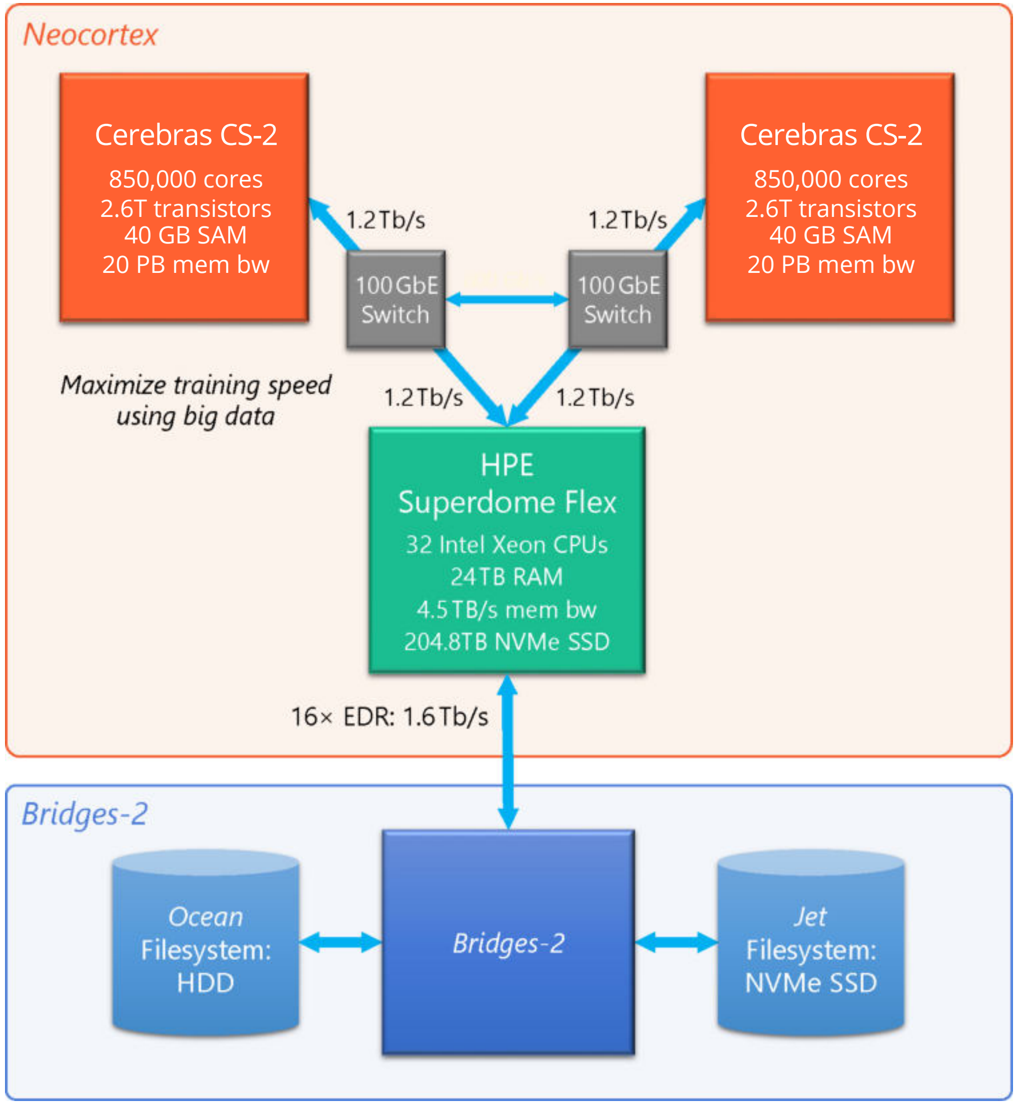
Animation from https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology. Retrieved on August 2021.

NEOCORTEX

Neocortex System Overview



Neocortex System Overview



Neocortex main AI accelerator is the Cerebras WSE, an alternative to other accelerators like GPUs.

Questions We are Addressing Today

1. What is the Neocortex program and what are its goals?
2. What is the innovative hardware a researcher can get access through Neocortex?
3. What type of applications are supported by the Neocortex system?
4. How can I gain access to the Neocortex program?
    - What is the evaluation criteria for proposals submitted to the Neocortex program?
    - What would be expected from a Neocortex researcher?
    - What can I expect as Neocortex researcher?
5. How to get additional information or support from the Neocortex team?

Applications Supported by Neocortex – as of February 2023

1. Cerebras modelzoo ML models
2. Models similar to the Cerebras modelzoo models
3. General purpose SDK
4. WFA, WSE Field-equation API

Applications Supported by Neocortex – as of February 2023

1

Cerebras modelzoo ML models

PyTorch

TensorFlow

ML only!

15 DL models: BERTx5 (standard, classifier, name entity recognition, summarization, question answering), GPT-2, GPT-3, GPT-J, Linformer, RoBERTa, T5, Transformer, FC-MNIST, 2D Unet.

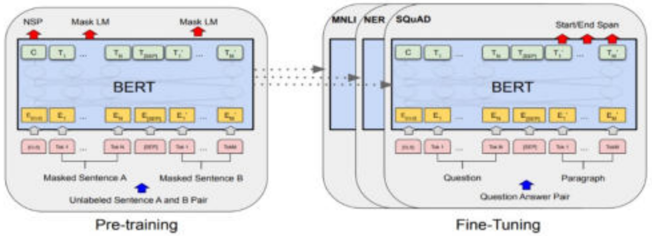
Model	Layer Pipeline mode	Weight Streaming mode
BERT	TensorFlow code PyTorch code	-
BERT (fine-tuning) Classifier	TensorFlow code PyTorch code	-
BERT (fine-tuning) Named Entity Recognition	TensorFlow code PyTorch code	-
BERT (fine-tuning) Summarization	TensorFlow code PyTorch code	-
BERT (fine-tuning) Question Answering	TensorFlow code PyTorch code	-
GPT-2	TensorFlow code PyTorch code	TensorFlow code
GPT-3	-	TensorFlow code
GPT-J	-	TensorFlow code
Linformer	TensorFlow code	-
RoBERTa	TensorFlow code PyTorch code	-
T5	TensorFlow code PyTorch code	-
Transformer	TensorFlow code PyTorch code	-
MNIST (fully connected)	TensorFlow code PyTorch code	-
2D UNet (experimental)	TensorFlow code	-

List of topics

Overview of the model

Bidirectional Transformers for Language Understanding (BERT) is an encoder-only transformer-based model designed for natural language understanding. This directory contains implementations of the BERT model. It uses a stack of transformer blocks with multi-head attention followed by a multi-layer perceptron feed-forward network. We support removing next-sentence-prediction (NSP) loss from BERT training processing with only masked-language-modeling (MLM) loss. The training pipeline has `2` phases. We first train with maximum sequence length of `128` and then train with maximum sequence length of `512`. More details of the model can be found in the appendix.

An overview of the model diagram is here:



We also support the RoBERTa model, which is very similar to BERT in terms of architectural design. In order to improve the results on BERT, some changes are made with objective functions (removing NSP), batch sizes, sequence lengths and masking patterns (dynamic vs. static). Difference between dynamic and static masking is discussed here.

We also support [`PubMedBERT`] model, which is pre-trained from scratch using abstracts from PubMed dataset. `PubMedBERT` model achieves state-of-the-art performance on several biomedical NLP tasks, as shown on the [Biomedical Language Understanding and Reasoning Benchmark].

https://portal.neocortex.psc.edu/docs/models_supported.html

NEOCORTEX

Applications Supported by Neocortex – as of February 2023

ML only!

2

Models similar to the Cerebras modelzoo models

ML models that are a **combination of the building blocks** used by modelzoo models and/or the layers supported by Cerebras as listed in their documentation.

https://docs.cerebras.net/en/latest/tensorflow-docs/api-rst/tf.html

15 Modelzoo DL models: BERTx5 (standard, classifier, name entity recognition, summarization, question answering), GPT-2, GPT-3, GPT-J, Linformer, RoBERTa, T5, Transformer, FC-MNIST, 2D Unet.

https://portal.neocortex.psc.edu/docs/models_supported.html

Applications Supported by Neocortex – as of February 2023

- Pioneering algorithm coding from scratch.
- Analogous to CUDA for Nvidia GPUs.
- ML kernels cannot be integrated with Pytorch and/or TensorFlow.
- Requires significant commitment.
- Will learn more form Leighton in the upcoming presentation.

General purpose!

3

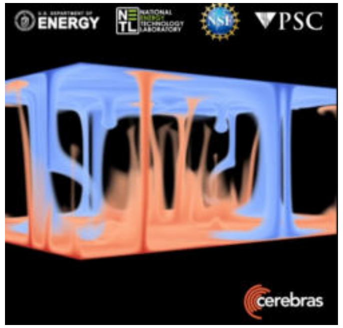
General purpose SDK

Applications Supported by Neocortex – as of February 2023

WFA: API recently used for advancing CFP simulations at unprecedented resolution and speed (more info).

- Pioneering work - Beta testing of the WFA library.
- Only a few groups would be welcomed.
- Close collaboration with Dirk's team, PSC, and Cerebras.
- More details in Dirk's upcoming presentation.

Field equations, includes ML inference



4

WFA, WSE Field-equation API

NEOCORTEX

Questions We are Addressing Today

1. What is the Neocortex program and what are its goals?
2. What is the innovative hardware a researcher can get access through Neocortex?
3. What type of applications are supported by the Neocortex system?
4. How can I gain access to the Neocortex program?
    - What is the evaluation criteria for proposals submitted to the Neocortex program?
    - What would be expected from a Neocortex researcher?
    - What can I expect as Neocortex researcher?
5. How to get additional information or support from the Neocortex team?

Spring 2023 Call for Proposals

- All details available in the official webpage: https://www.cmu.edu/psc/aibd/neocortex/2023-03-cfp-spring-2023.html

Neocortex Spring 2023 Allocation Submissions	
Name	**Date (ET)**
Application begins	**March 15, 2023**
Application ends	April 12, 2023 (Anywhere on Earth time zone)
Response ends	May 10, 2023
Allocation starting date	User access to start mid-May 2023 (rough estimate)

Spring 2023 Call for Proposals

- Open to all U.S.-based university and non-profit researchers.
- Offered at no cost for researchers advancing open-science work.
- Applications welcomed and processed through EasyChair.
- Applications welcomed for a period of 5 weeks.
- Applications will be evaluated as they come in. Apply as soon as convenient!
- Lightweight application via a short form.
- Onboarding meetings will be scheduled to confirm scope of the project and suitability.

Spring 2023 Call for Proposals

- Users expected to be onboarded by mid May.
- Allocations to Neocortex resources and Bridges-2 will be initially granted for a year by default.
- Close collaboration and constant communication between domain projects, PSC, and vendors is expected. Checkpoint sessions every 3 months or so.
- **Feedback and user experiences are expected** to further enrich the project.
- More technical details on the Cerebras servers, the ML frameworks, SDK, WFA, and applications supported, in the second part of the webinar to be presented by Cerebras and NETL collaborators.

Questions We are Addressing Today

1. What is the Neocortex program and what are its goals?
2. What is the innovative hardware a researcher can get access through Neocortex?
3. What type of applications are supported by the Neocortex system?
4. How can I gain access to the Neocortex program?
    - What is the evaluation criteria for proposals submitted to the Neocortex program?
    - What would be expected from a Neocortex researcher?
    - What can I expect as Neocortex researcher?
5. How to get additional information or support from the Neocortex team?

To Learn More and Participate

Watch the Neocortex website for updates!	https://www.cmu.edu/psc/aibd/neocortex/
Join the neocortex-updates list	https://www.cmu.edu/psc/aibd/neocortex/newsletter-sign-up.html
Apply to upcoming CFP	https://www.cmu.edu/psc/aibd/neocortex/2023-03-cfp-spring-2023.html
Contact us with additional questions, input, or requests	neocortex@psc.edu

To Learn More and Participate

Watch the Neocortex website for updates!	https://www.cmu.edu/psc/aibd/neocortex/
Join the neocortex-updates list	https://www.cmu.edu/psc/aibd/neocortex/newsletter-sign-up.html
Apply to upcoming CFP	https://www.cmu.edu/psc/aibd/neocortex/2023-03-cfp-spring-2023.html
Contact us with additional questions, input, or requests	neocortex@psc.edu

NEOCORTEX

Thank you to all those contributing to *Neocortex*!

NSF

cerebras

Hewlett Packard Enterprise

NEOCORTEX
Unlocking Interactive AI for Rapidly Evolving Research

Neocortex Team

Cerebras CS-2: the AI Compute Engine for Neocortex

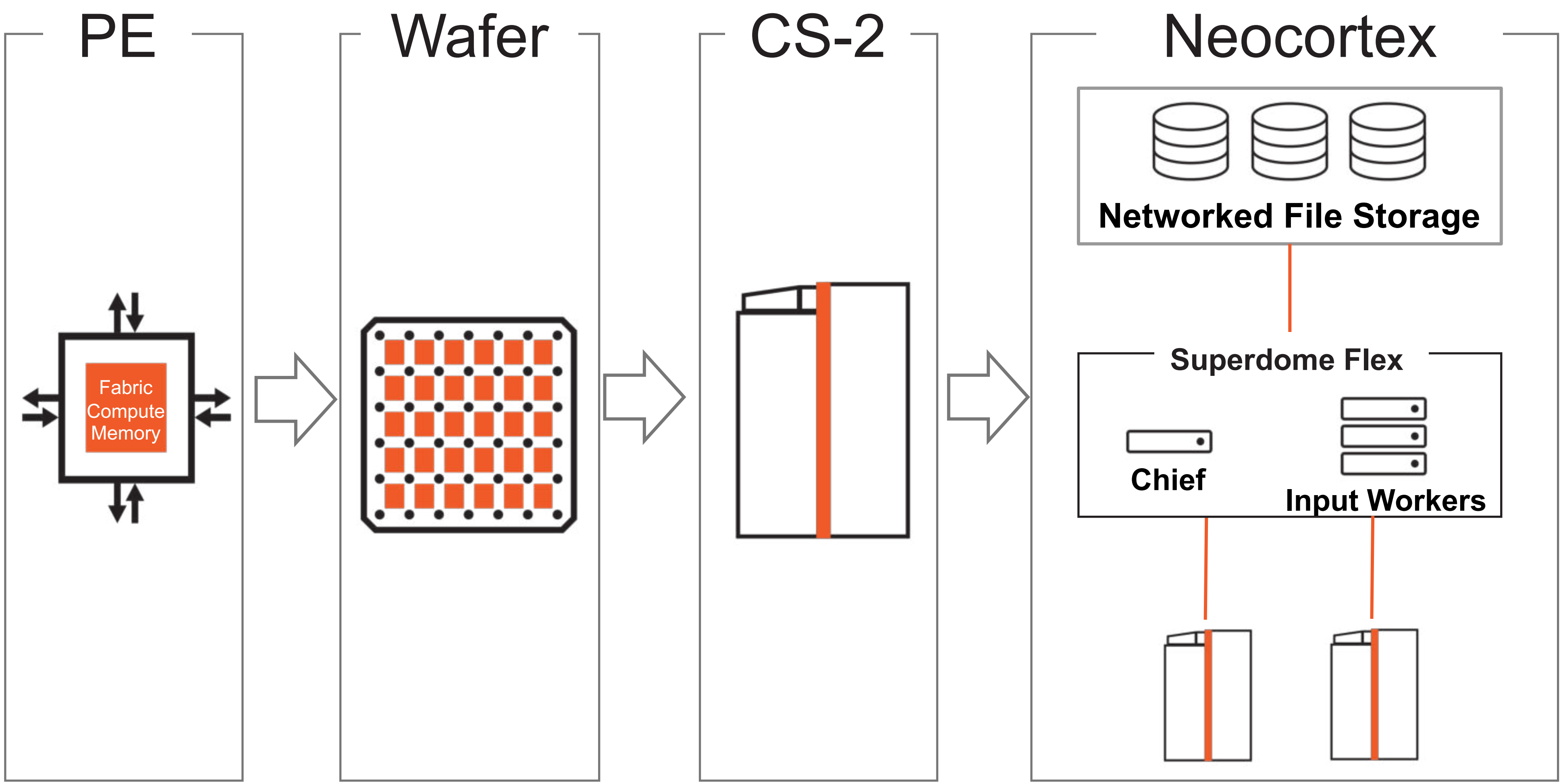
February 28, 2023

Outline

- CS-2 Overview
- CS-2 for Deep Learning
  - PyTorch and TensorFlow integration
  - Reference implementations, docs, supported layers
  - DL projects of interest with examples
- CS-2 for HPC using the SDK
  - Programming with the Software Development Kit (SDK) and Cerebras Software Language (CSL)
  - Examples of successful HPC projects
  - HPC topics of interest

CS-2 Overview

System Hierarchy



Cerebras Wafer Scale Engine (WSE-2)

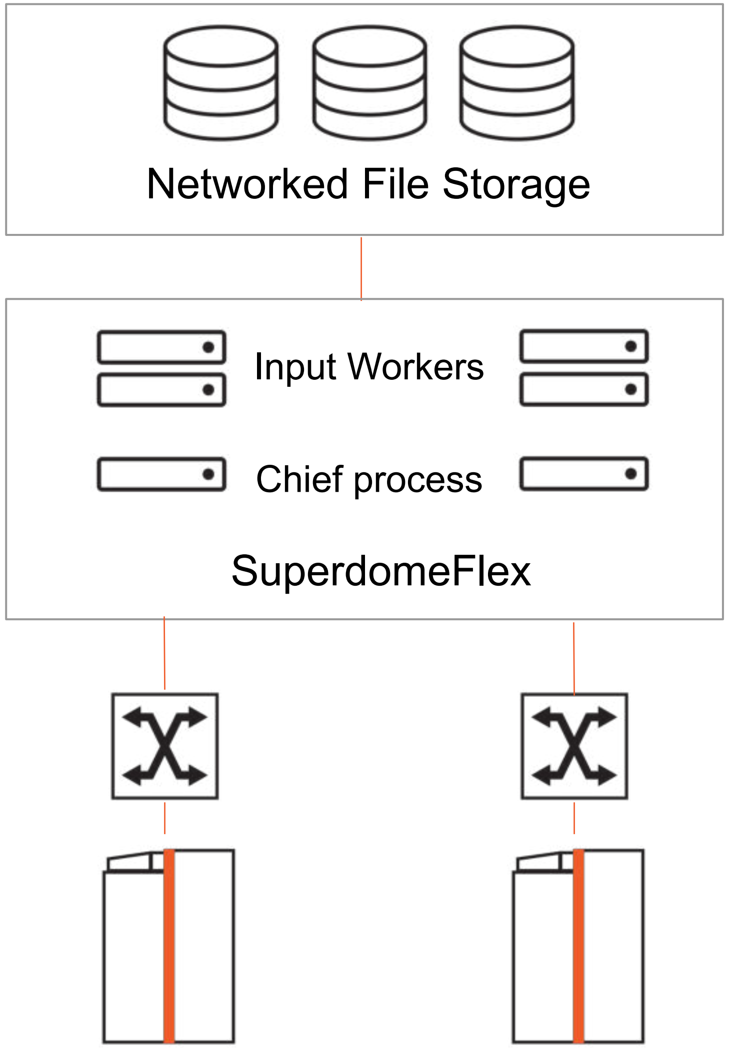
The Most Powerful Processor for AI & HPC

850,000	cores optimized for sparse linear algebra
46,225 mm^2	silicon
2.6 trillion	transistors
40 Gigabytes	of on-chip memory
20 PByte/s	memory bandwidth
220 Pbit/s	fabric bandwidth
7nm	process technology

Cluster-scale performance in a single chip

Deployment and job execution within Neocortex

- CS-2 is a **network-attached accelerator**
- Cerebras software runs on CS-2 and on the SuperdomeFlex
  - Chief compiles the code and manages CS-2
  - Input workers read data, run the input pipeline, and stream data to CS-2
- Loss output, summaries, checkpoints are streamed from CS-2 to SuperdomeFlex
- Jobs are managed by slurm



Developer Resources

*please refer to the *Original Cerebras Installation* version/sections for both the Model Zoo repository and Software Documentation

For ML developers

Cerebras Model Zoo repository (https://github.com/Cerebras/modelzoo/tree/original_cerebras_installation)

Cerebras ML Software Documentation (Original Cerebras Installation sections of https://docs.cerebras.net/en/latest/index.html)

For HPC developers

Cerebras SDK

SDK

CS-2 for Deep Learning

Frameworks supported

TensorFlow

Class:

CerebrasEstimator

- Based on TF Estimator, takes over executions after XLA compilation
- TensorFlow 2.2

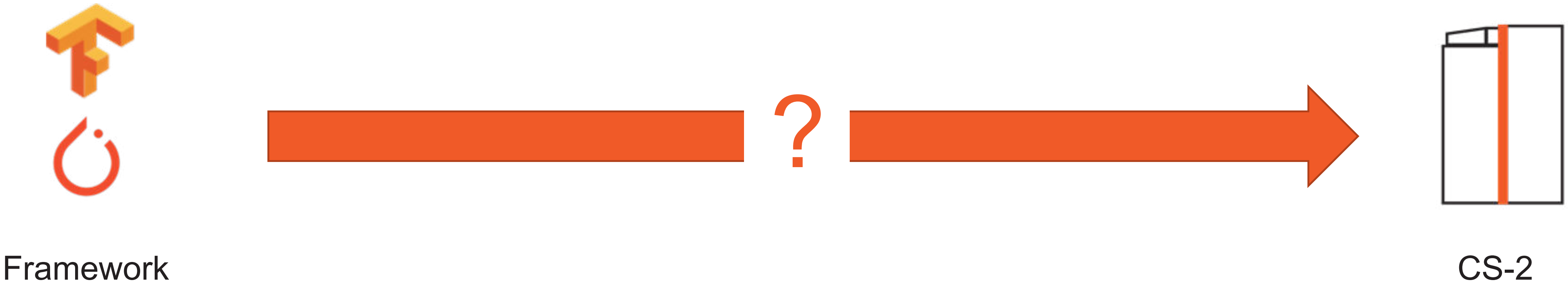
PyTorch

Python Module:

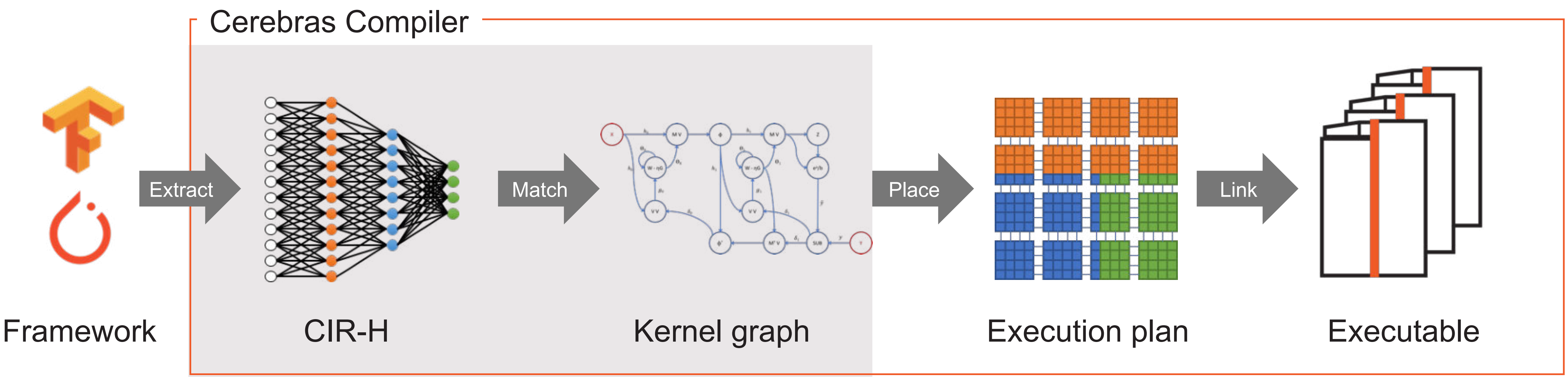
cerebras.framework.torch

- Based on PyTorch XLA
- Wrappers for Dataloader, Module, Session
- PyTorch 1.11

How do we translate a model into a CS executable?



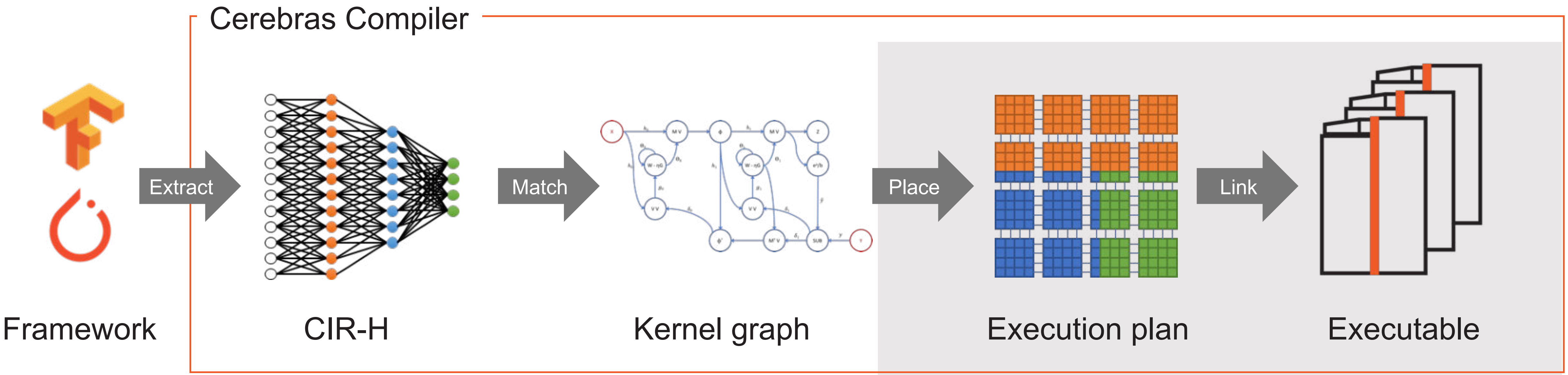
The Cerebras Software Platform



<u>Feature Support</u>

Extract **graph representation** of model from framework and match operations with **kernel library**

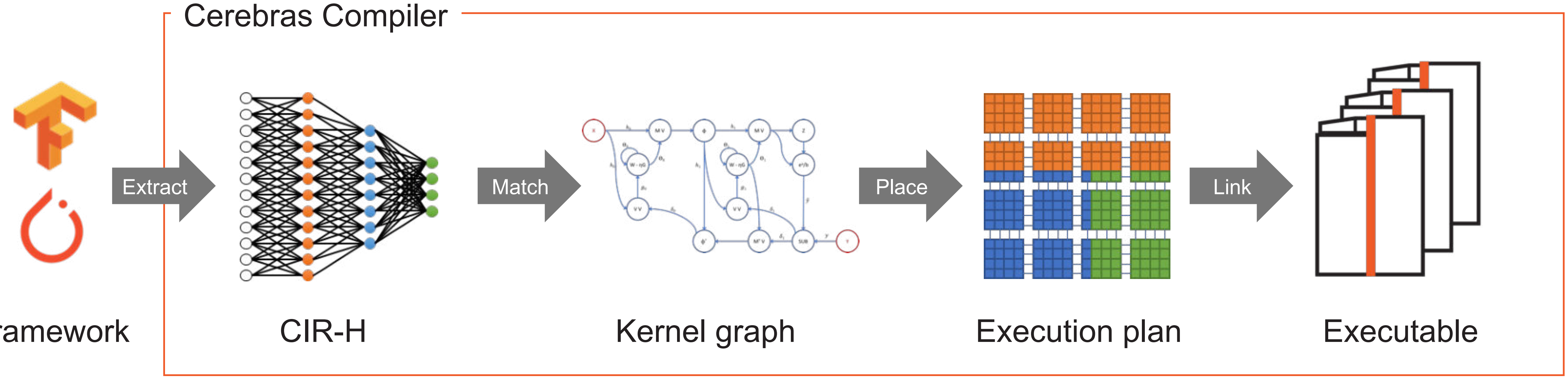
The Cerebras Software Platform



<u>Hardware Placement</u>

Assign kernels to **regions of fabric** and create **executable** to be run by CS-2.

The Cerebras Software Platform



Program a cluster-scale resource with the ease of a single node

ML Software Key Features

Network Architectures

- Transformers (TF and PyT)
  - E.g., BERT, RoBERTa, AIAYN, T5, GPT
- Multi-layer Perceptrons (MLP) (TF and PyT)
- Experimental (TF only)
  - UNet - limited functionality

General features

- Supports Train, Eval
- Trained weights in standard TF and PyT formats
- Monitor your runs with TensorBoard
- Multi-replica support for smaller models

* Please refer to the *Original Cerebras Installation* Model Zoo repository for architectures we support.
* We recommend starting from the Model Zoo implementations to build your models.

Topics of interest for ML applications

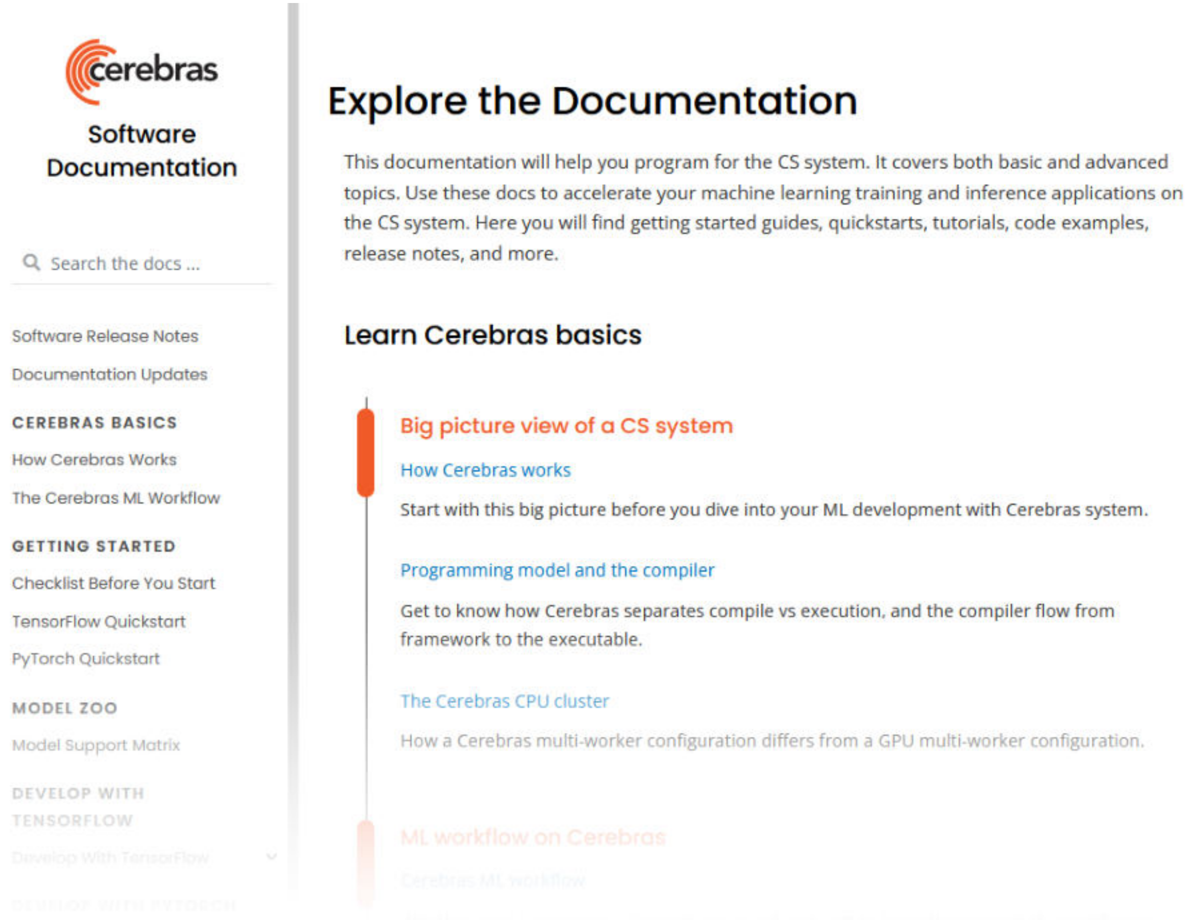
- Neocortex is best suited for running Transformer style models such as BERT, GPT, Transformer, T5, and ViT.
- Transformer style models cover a wide range of data modeling tasks such as:
  - Sequence classification - sentiment analysis, molecule properties
  - Sequence annotation - extractive summarization, protein binding site identification
  - Sequence generation - abstractive summarization, candidate drug generation
  - Sequence to sequence mapping - Natural language translation, code translation
  - Representation learning for biological sequences (genome, epigenome, protein)
- Many data modeling task that historically used other architectures can and have been reframed to leverage the transformer architecture. Some examples are:
  - Image classification: CNN → BERT(ViT)
  - Autoregressive sequence/time-series modeling: RNN → GPT
  - Graph modeling: GNN → BERT with adjacency attention mask

Example Projects/ Models

- GSK: new sequence modeling for genetic medicine
  - “Epigenomic language models powered by Cerebras” Trotter, 2021
- PubMedBERT
  - “Domain-specific language model pretraining for biomedical NLP” Gu, 2021
- AntiBERTa
  - “Deciphering the language of antibodies using self-supervised learning” Leem, 2021
- TAPE
  - “Evaluating protein transfer learning with TAPE” Rao, 2019
- SMILES-BERT
  - SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction" Wang, 2019

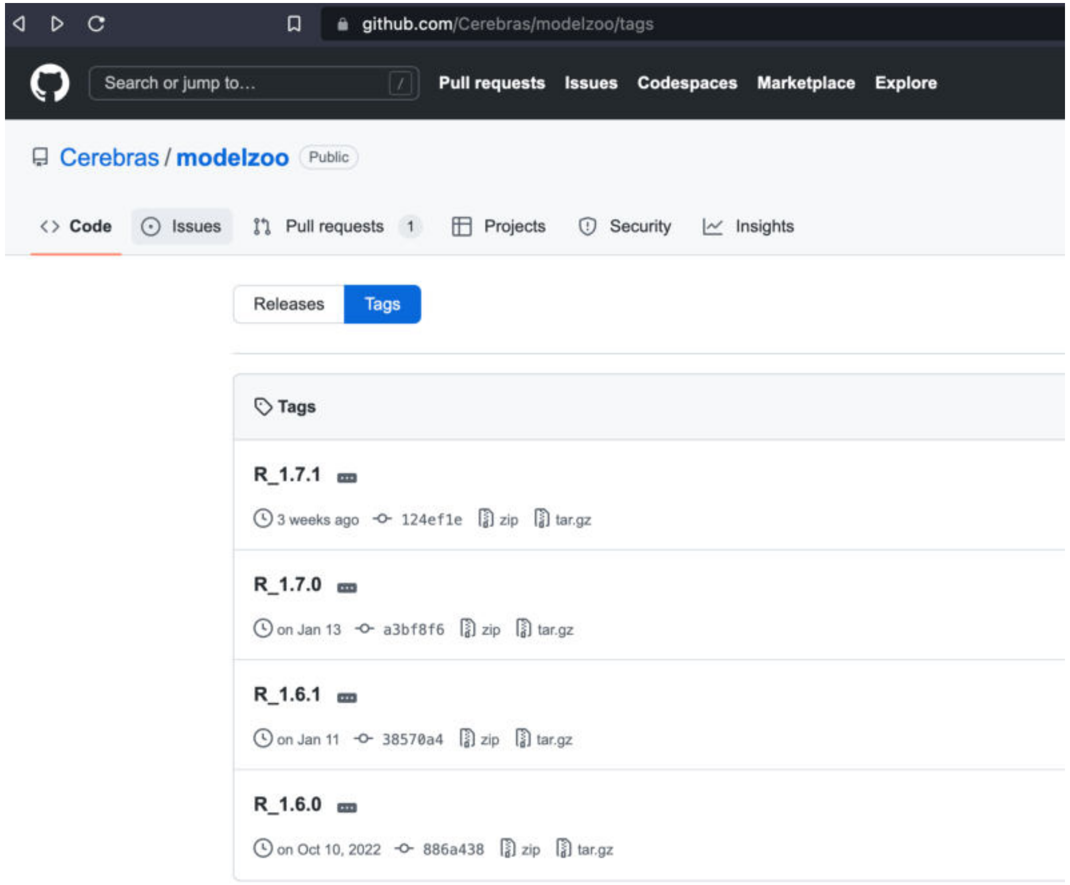
Resources

Documentation: docs.cerebras.net



Cerebras Model Zoo:

github.com/Cerebras/modelzoo/tree/original_cerebras_installation



Please check out the **Original Cerebras Installation** version/sections.

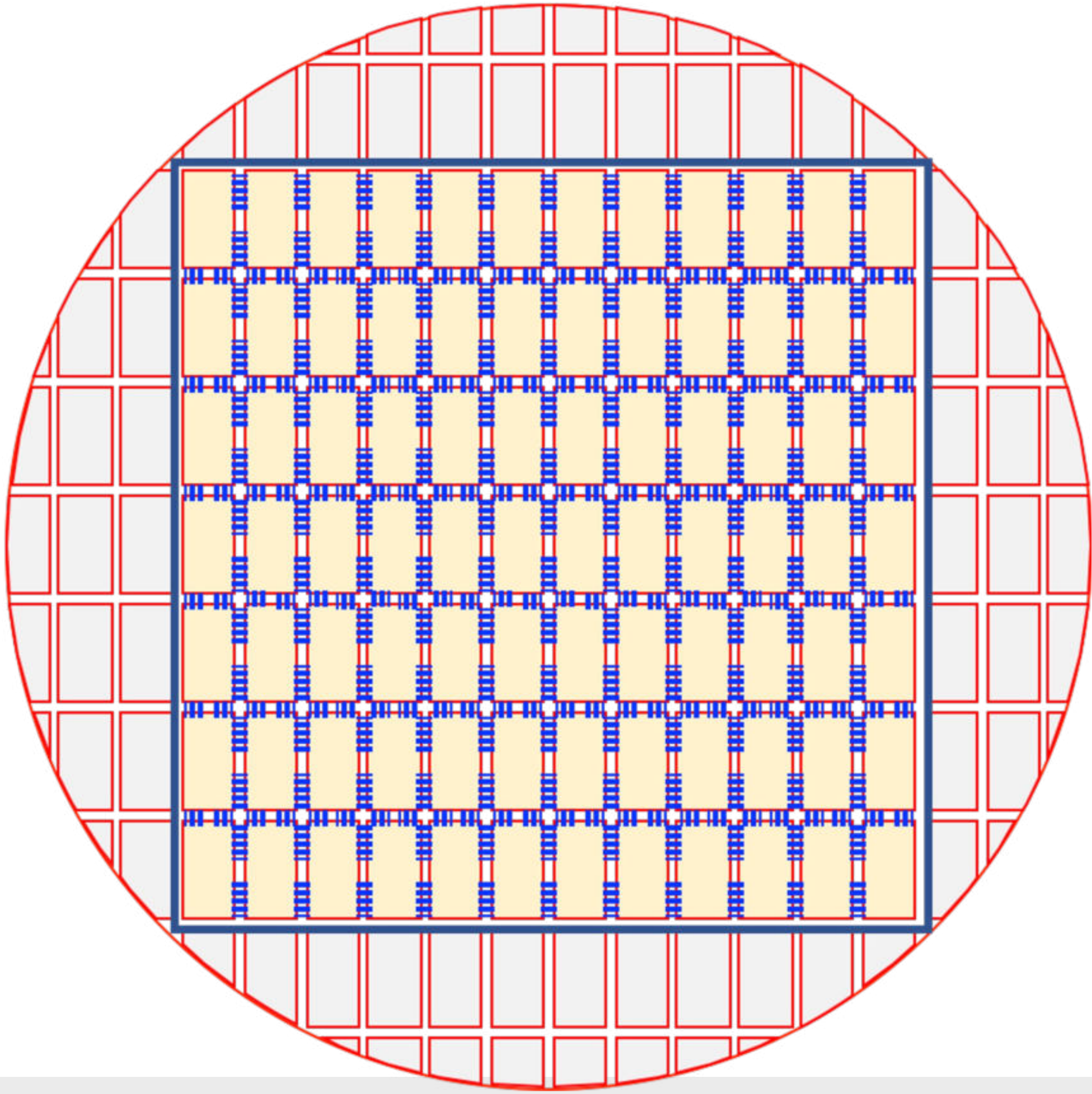
CS-2 for HPC via SDK

Does your application scale poorly across nodes?

Examples: *FFT-based solvers, particle simulators, non-linear problems with iterative solvers*

The Cerebras solution:

- The WSE-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



Is your application constrained by data access?

Examples: *Stencil based PDE solvers, linear algebra solvers, signal processing, sparse tensor math, big data analysis*

The Cerebras solution:

- The CS-2 system has **40 GB of SRAM** uniformly distributed across the wafer that is **1 cycle** away from the processing element
  - Speeds up memory access by orders of magnitude
- The CS-2 system is capable of **1.2 Tb/s bandwidth** onto the chip
  - Stream data onto the chip as required

Cerebras SDK

Language

CSL: Cerebras Software Language

Host APIs with Python

Libraries

Optimized primitives

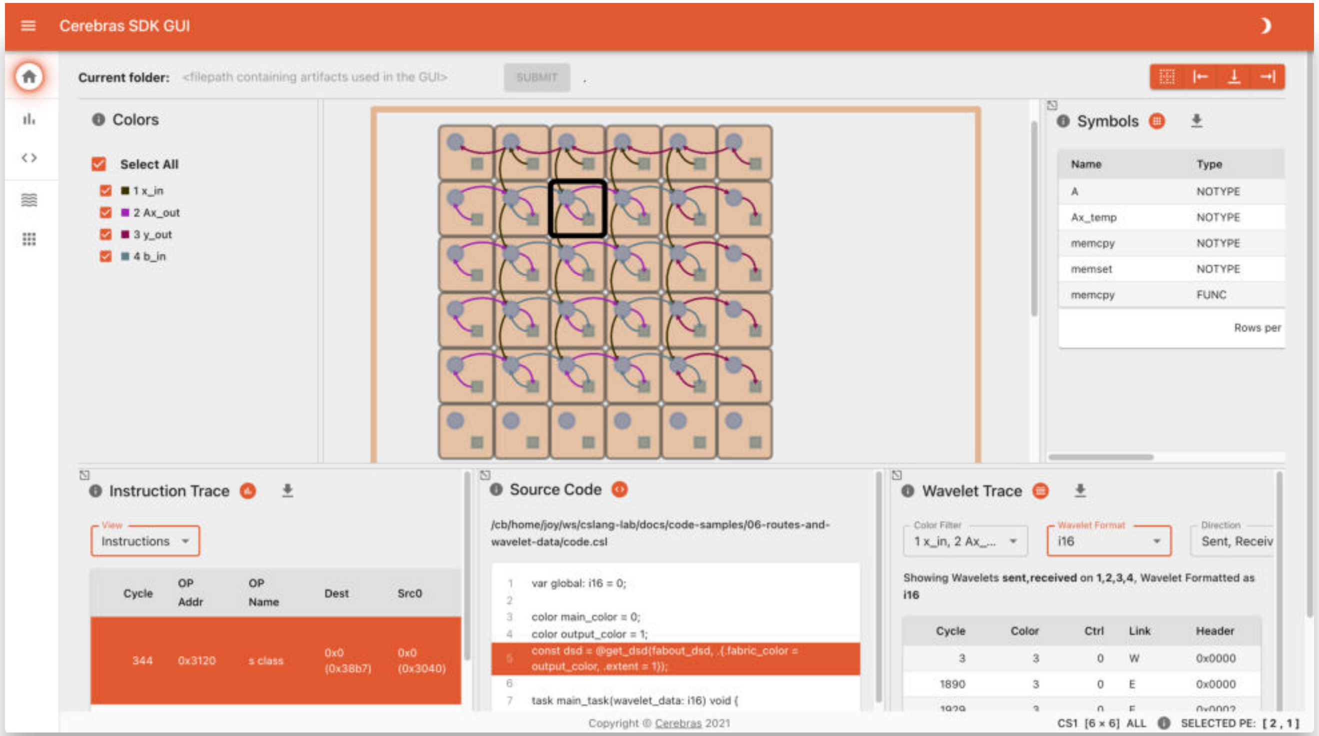
Tools

Simulator

Debugger

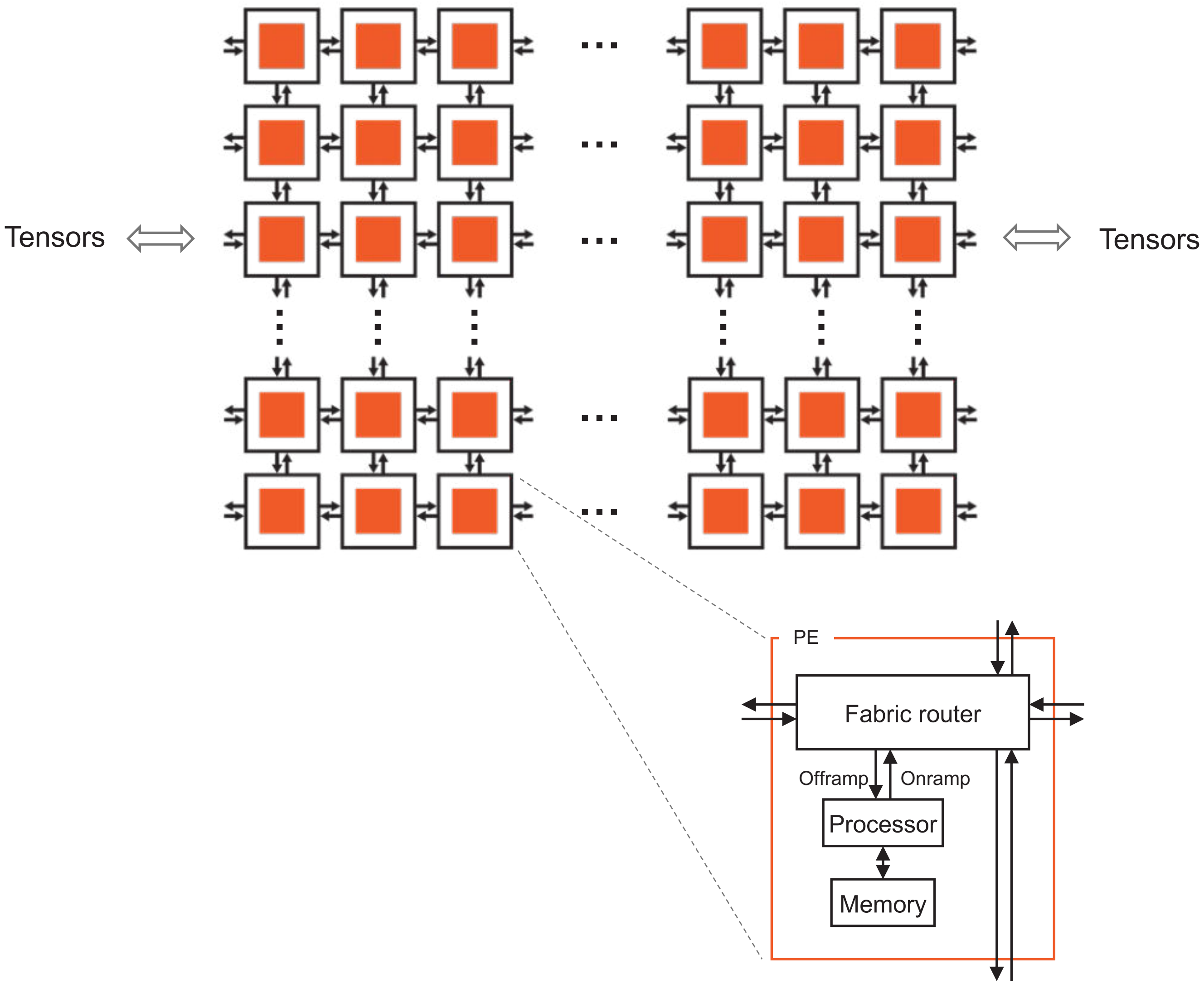
Performance profiler

Visualization



A general-purpose parallel-computing platform and API allowing software developers to write custom programs ("kernels") for Cerebras systems.

CS-2 Dataflow Programming



To the programmer, the CS-2 appears as a logical 2D array of 850k individually programmable Processing Elements (PEs)

Flexible compute
- General purpose CPU
- 16- and 32-bit native FP and integer data types
- Tasks triggered by the arrival of data packets

Flexible communication
- Programmable router
- Static or dynamic routes (aka colors)
- Data packets (aka wavelets) passed between PEs
- 1 cycle for PE-to-PE communication

Fast memory
- 40GB on-chip SRAM
- Data and instructions
- 1 cycle read/write

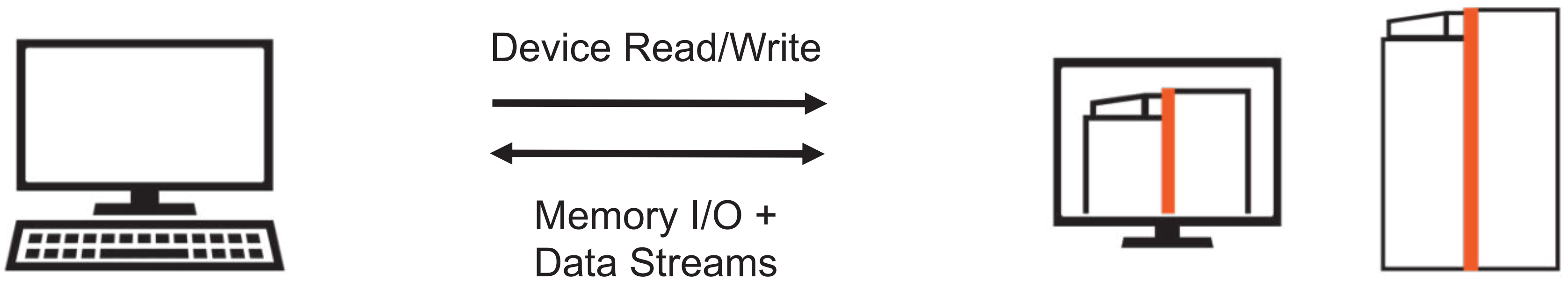
From a Programmer’s Perspective

Host CPU(s): Python

- Loads program onto simulator or CS-2 system
- Streams in/out data from one or more workers
- Reads/writes device memory

Device: CSL

- Target software simulator or CS-2
- CSL programs run on groups of cores on the WSE, specified by programmer
- Executes dataflow programs



CSL: Language Basics

- Types
- Functions
- Control structures
- Structs/Unions/Enums
- Comptime

Straight from C
(via Zig)

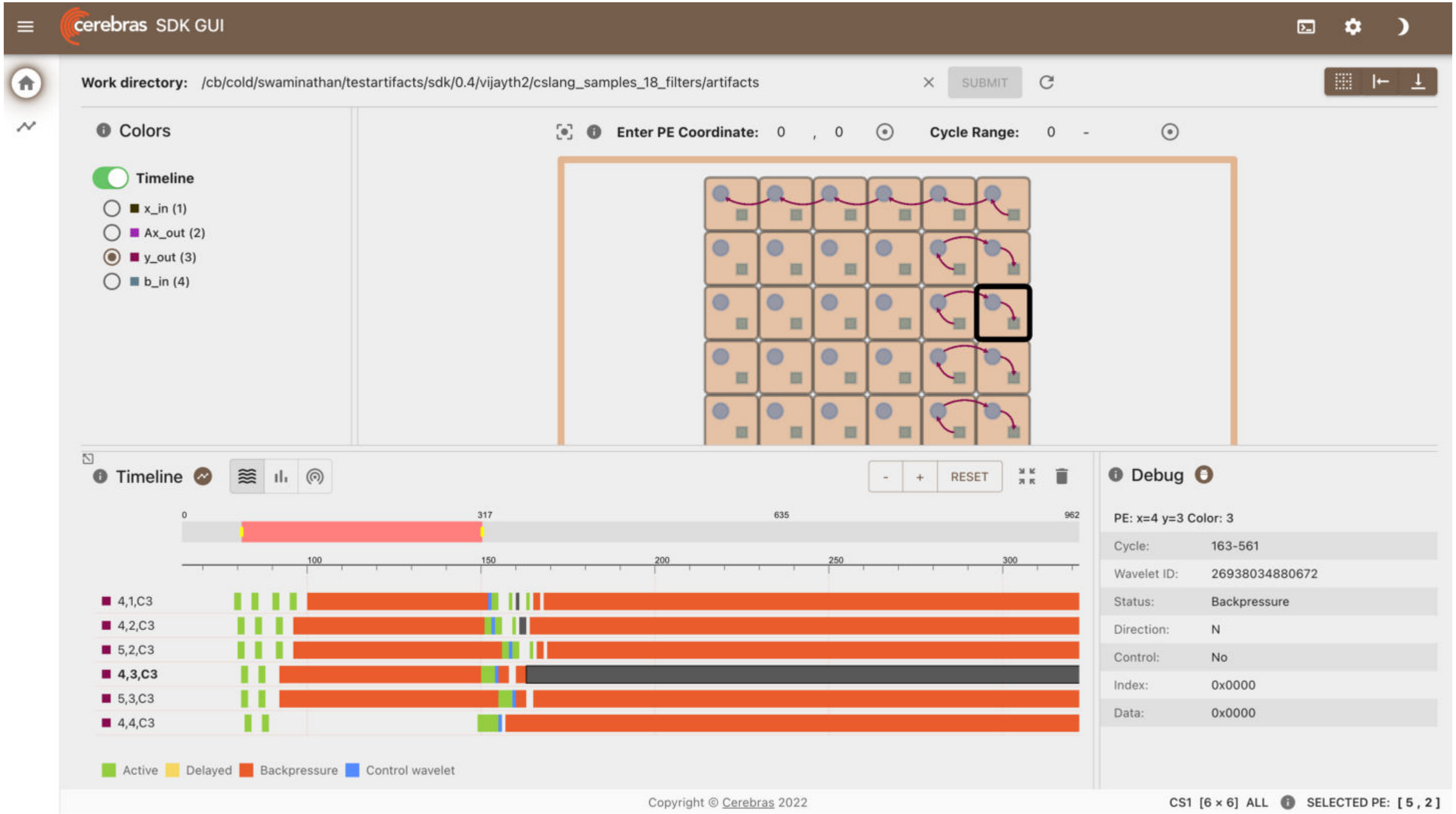
- Builtins
- Module system
- Params
- Tasks
- Data Structure Descriptors
- Layout specification

CSL specific

Used for writing device kernel code

Familiar to C/C++/HPC programmers

Simulation Debug Tools



Private Documentation: sdk.cerebras.net

cerebras

SDK Documentation

Search the docs ...

SDK Release Notes

Documentation Updates

START HERE

A Conceptual View

Kernel Development Flow

QUICKSTART

Installation and Setup

Quickstart

DEVELOPMENT GUIDES

Working With Code Samples

CSL Code Examples

CSL Language Guide

DEBUGGING

Debugging Guide

Route Visualizer

API REFERENCE

SDK API Reference

Documentation for Developing with CSL

This is the documentation for developing kernels for Cerebras system. Here you will find getting started guides, quickstarts, tutorials, code examples, release notes, and more.

Start Here

Computing with Cerebras

A conceptual, "mental model" view.

Quickstart

Compile and run

Quickstart with a single PE or multiple PEs.

Kernel Development Flow

Steps to develop your kernel

Define layout, assign code to PE and configure routes and colors.

Working with Code Samples

Learn how to run the code samples

A glimpse into the run script.

Program the WSE

CSL examples

Manipulate sparse tensors, configure fabric switches and more.

Debug

Learn how to use the debugger

Trace the instructions, monitor the tasks at a specific PE and trace wavelets.

Visualize the Fabric

SDK API Reference

Using CSL

Examples Included in the SDK

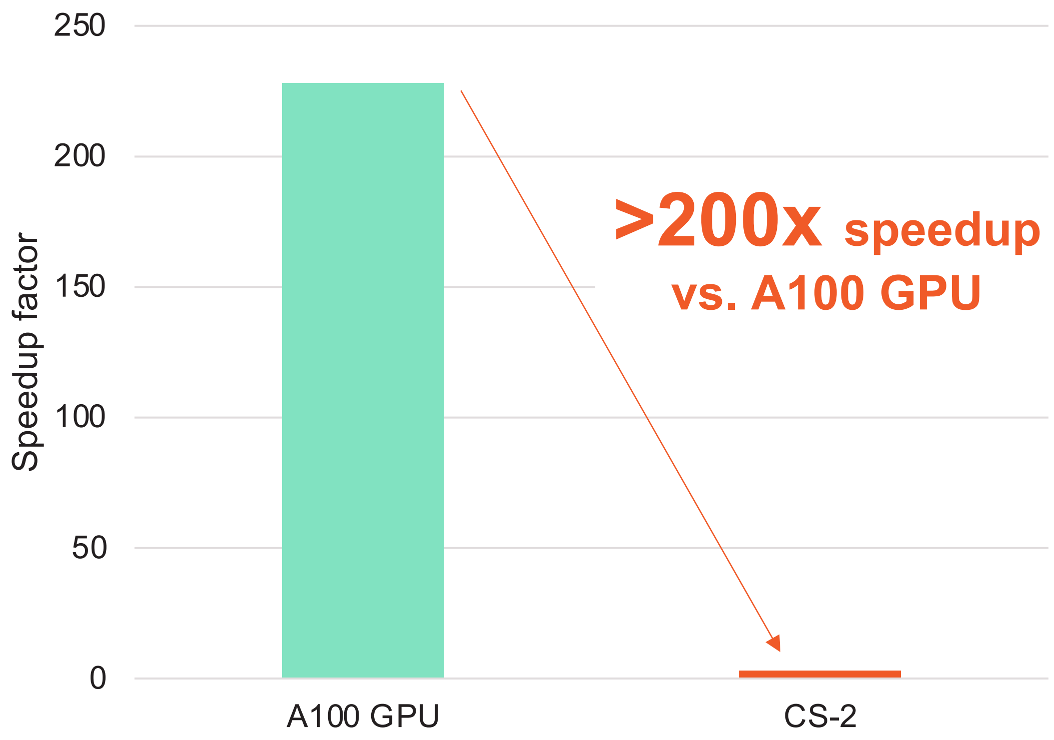
- Basic tasks and colors
- Multiple source files
- Multi-PE kernel
- Basic parameters
- Wavelet-triggered Tasks
- Arrays and Pointers
- Sparse Tensors
- Memory DSDs
- Fabric DSDs
- Reduction
- Basic Branches
- Initializers
- Modules
- Loops
- Kernel Parameterization
- Fabric Switches
- GEMV
- FFT
- Stencil (Finite Differences)
- Shift-Add Multiply

Accelerated energy research at TotalEnergies

Objective: Enable order-of-magnitude speedups on a wide range of simulations: batteries, biofuels, wind flows, drillings, and CO2 storage

Challenge: Participate in Total study to evaluate hardware architectures, using finite difference seismic modelling code as a benchmark

Outcome: Cerebras CS-2 system outperformed a A100 AI GPU by >200X using code written in the Cerebras Software Language (CSL). System now installed and running at customer facility in Houston, TX



"We count on the CS-2 system to boost our multi-energy research and give our research 'athletes' that extra competitive advantage."

Dr. Vincent Saubestre, CEO and President, TotalEnergies Research & Technology USA

TotalEnergies

Mathias Jacquelin, Mauricio Araya-Polo, Jie Meng, "Massively Scalable Stencil Algorithm"
Presented at SC22, https://arxiv.org/abs/2204.03775

Topics of interest for HPC applications

- Structured grid based PDE and ODE solvers
- Dense linear algebra
- Sparse linear algebra
- Particle methods with regular communication
- Monte Carlo type problems that can fill the wafer
- Towards development of HPL, HPCG type benchmarks
- Custom ML kernels

Recap

- CS-2 is a dense and powerful single system, powered by 1 enormous chip
  - Cluster-scale compute on a single device => good fit for large DL models
  - 40GB SRAM with massive memory bandwidth => good fit for sparse problems
- CS-2 for Deep Learning
  - Leveraging TensorFlow and PyTorch frontends
  - No low-level programming required
  - Cerebras Software takes care of distributing computations across 850,000 cores
- CS-2 for HPC via SDK
  - Use CSL language for low-level programming on the wafer
  - Users decide how to distribute the computations
  - Cannot be integrated with TensorFlow/ PyTorch workloads
- Next: CS-2 for field equations via NETL's WFA

Thank you!

https://cerebras.net/

Using the WFA for Scientific Computing on the WSE

Dirk Van Essendelft*

*dirk.vanessendelft@netl.doe.gov

Neocortex Spring 2023 CFP

February 28, 2023

NATIONAL ENERGY TECHNOLOGY LABORATORY

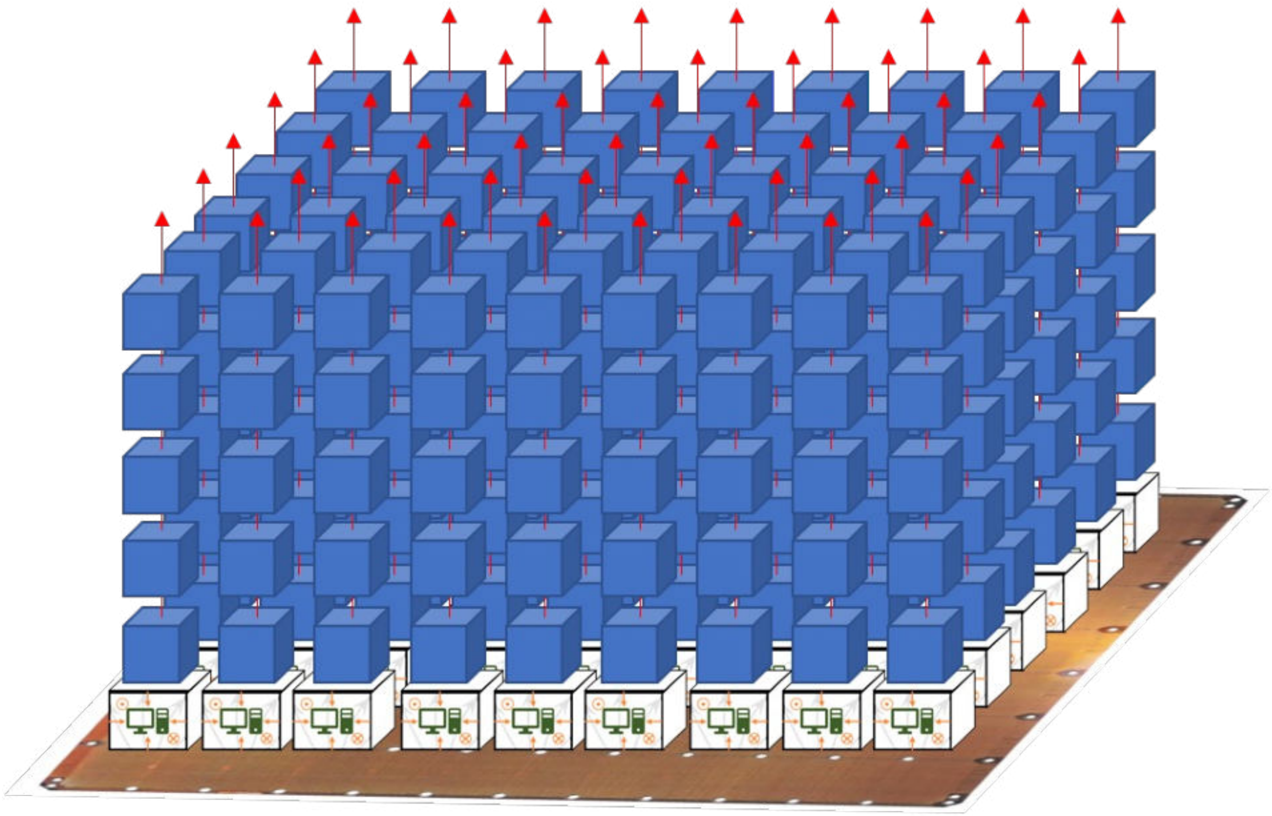
What is the WFA

DSL for Solving Spatial-Temporal Problems on Structured Grids

Simple Python Front End (like Numpy)

```
from WSE_FE.WSE_Interface import WSE_Interface
from WSE_FE.WSE_Array import WSE_Array
from WSE_FE.WSE_Loops import WSE_For_Loop
import numpy as np

# Instantiate the WSE Interface
wse = WSE_Interface()

# define constants
c = 0.1
center = 1.0 - 6.0 * c

# Create the initial temperature field and BC's
T_init = np.ones((102, 102, 102))*500.0
T_init[1:-1, 1:-1, 0] = 300.0
T_init[1:-1, 1:-1, -1] = 400.0

# Instantiate the WSE Array objects needed
T_n = WSE_Array(name='T_n', initData=T_init)

# Loop over time
with WSE_For_Loop('time_loop', 400000):
    T_n[1:-1, 0, 0] = center * T_n[1:-1, 0, 0]\
        + c * (T_n[2:, 0, 0] + T_n[:-2, 0, 0]
               + T_n[1:-1, 1, 0] + T_n[1:-1, 0, -1]
               + T_n[1:-1, -1, 0] + T_n[1:-1, 0, 1])

wse.make_WSE(answer=T_n)
```

Simple Pencil Memory Layout

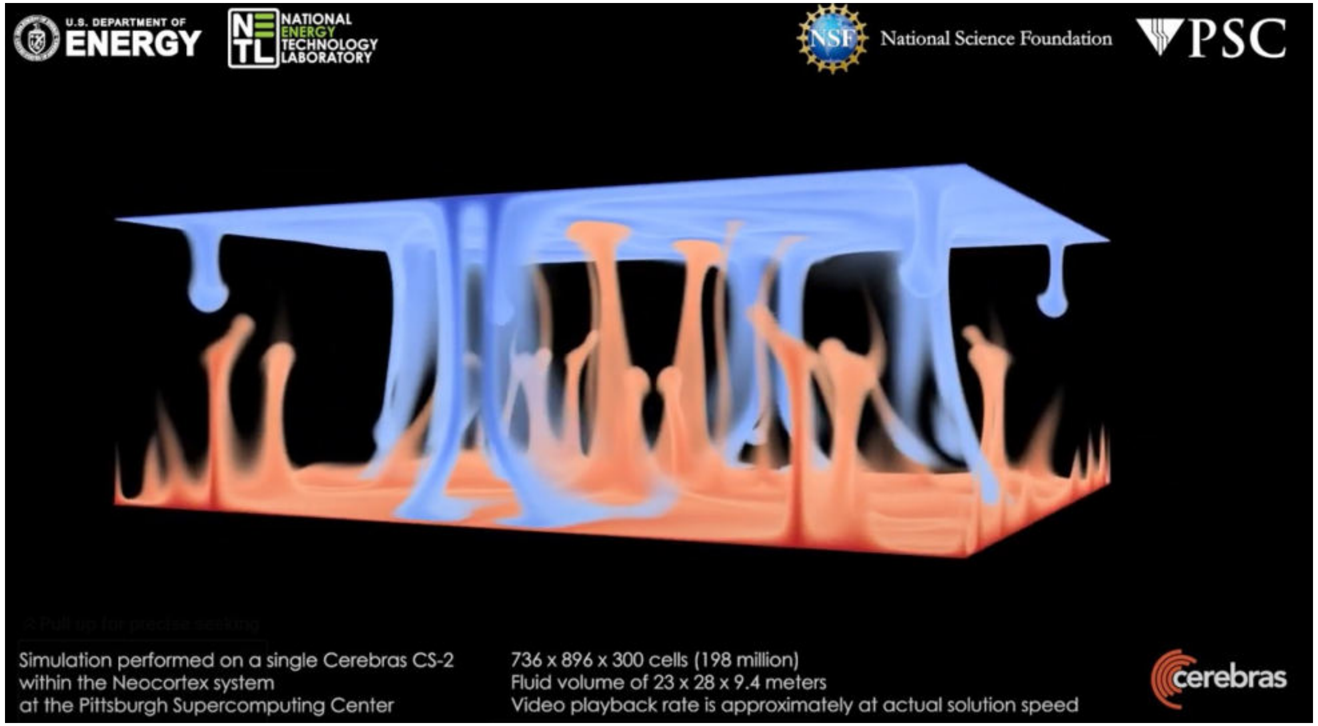


https://dirk-netl.github.io/WSE_FE/

Near Real Time Scientific Modeling

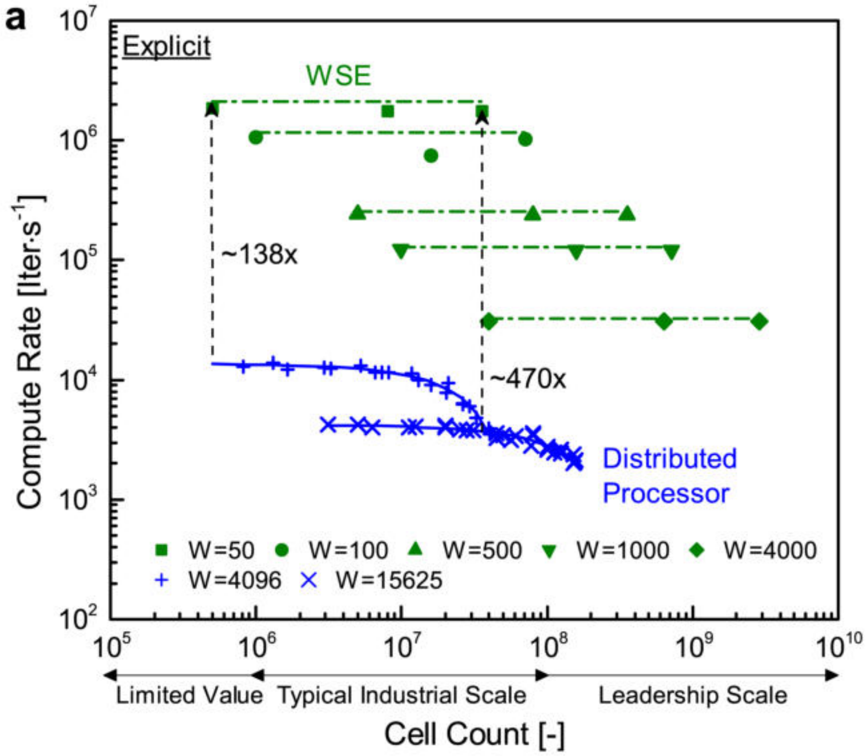
Exceptionally Fast PDE Solutions On Wafer Scale Engine

CFD Demonstration Completely on WSE



https://www.youtube.com/watch?v=5ad9f70ORvQ

Several Hundred Times Faster Than Distributed Computing



https://arxiv.org/abs/2209.13768

Seeking Beta Testers for Scientific Computing

Project Guidelines

- Problem Requirements
  - Must lay out on a Hex grid (3d or many 2d parallel)
  - Should involve Spatial Locality
  - Should be Data Intense
  - Single Precision, <40GB
- Problem Examples
  - Computational Fluid Dynamics (FVM, FDM, FEM, LBM)
  - Structural Mechanics
  - Geomechanics
  - Weather/Climate
  - Materials – Ising Model, Density Functional Theory
  - CNN/RNN inference
- Project Requirements
  - Build a Python class that imports the WFA and contains a "Library" to solve your scientific problem
  - Post on a public github
- What to expect
  - Development at the high-level Python interface that is similar to Numpy
  - A container (provided by Cerebras) to compile and generate binaries and run on the WSE
  - An unpolished product (Beta level) that will require some hand holding from our team on slack
    - Be prepared for bugs and unpolished documentation
  - Exceptional speed if successful, strong scaling that can't be matched elsewhere