Oak Ridge National Laboratory

Preparing for Post-Exascale Computing

Congratulations on 30th Anniversary!

Jeffrey S. Vetter

With many contributions from ACSR Section and Colleagues

The 30th Anniversary Symposium of the Center for Computational Sciences
University of Tsukuba
13 Oct 2022

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

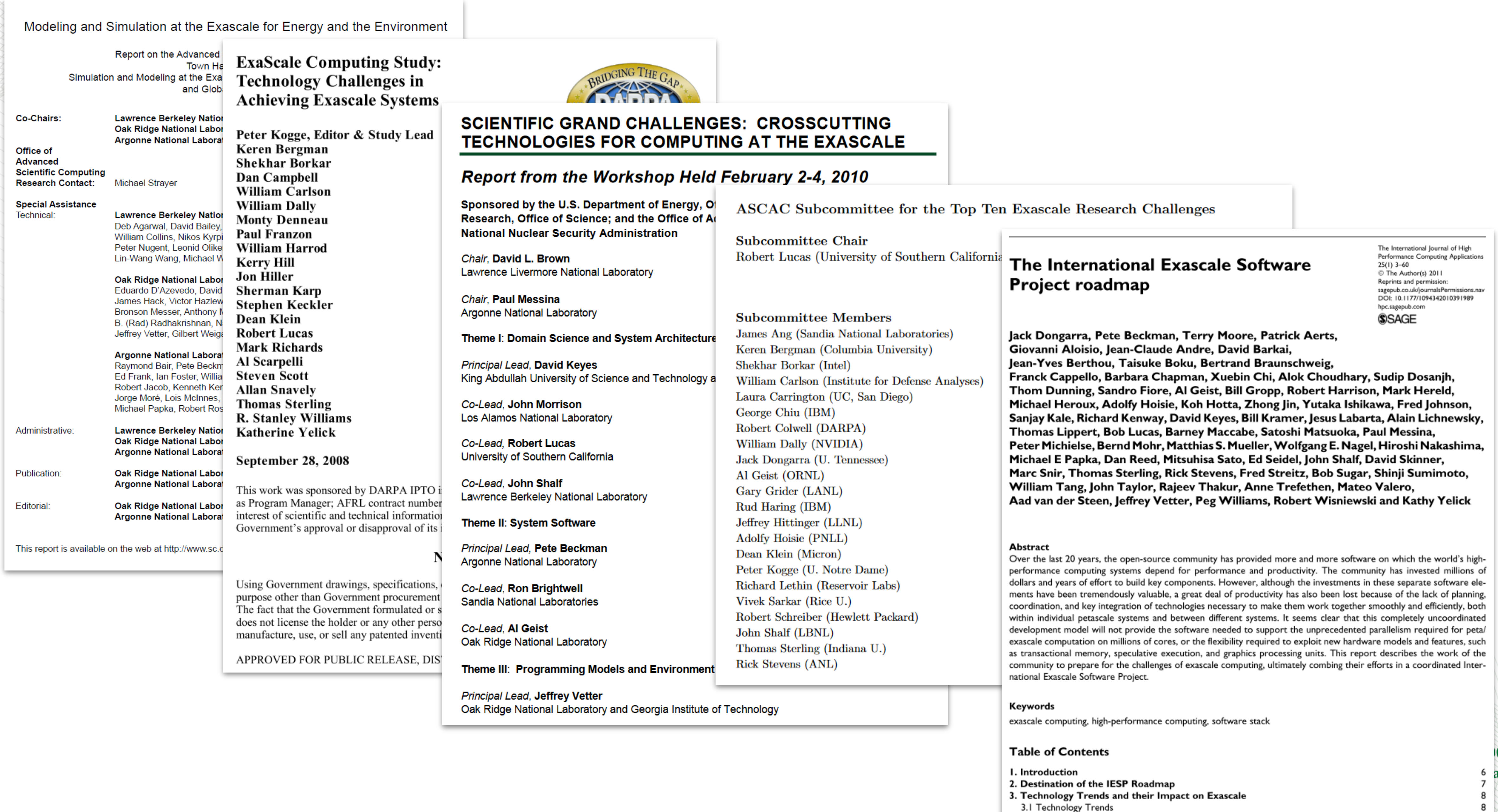
Future Technologies Group

https://www.ornl.gov/section/advanced-computing-systems-research (https://j.mp/acsrs)
vetter@computer.org

Highlights

- 15 years to go from Exascale ideation to deployment
  - Reports and predictions
- Current status
  - Systems (Frontier)
  - Exascale Computing Project
- Exascale: What did we get right, get wrong, overlook?
- Post Exascale?
  - Heterogeneous systems enabled by Heterogeneous integration and Chiplets
  - Codesign becomes even more important
- Abikso: Microelectronics codesign project

Exascale Reports (and predictions) from 2007 to 2014



Modeling and Simulation at the Exascale for Energy and the Environment

Report on the Advanced
Town Ha
Simulation and Modeling at the Exa
and Glob

Co-Chairs: Lawrence Berkeley Natior
Oak Ridge National Labor
Argonne National Laborat

Office of
Advanced
Scientific Computing
Research Contact: Michael Strayer

Special Assistance
Technical: Lawrence Berkeley Natior
Deb Agarwal, David Bailey,
William Collins, Nikos Kyrpi
Peter Nugent, Leonid Oliker
Lin-Wang Wang, Michael W

Oak Ridge National Labor
Eduardo D'Azevedo, David
James Hack, Victor Hazlew
Bronson Messer, Anthony M
B. (Rad) Radhakrishnan, N
Jeffrey Vetter, Gilbert Weig

Argonne National Laborat
Raymond Bair, Pete Beckm
Ed Frank, Ian Foster, Willia
Robert Jacob, Kenneth Ker
Jorge Moré, Lois McInnes,
Michael Papka, Robert Ros

Administrative: Lawrence Berkeley Natior
Oak Ridge National Labor
Argonne National Laborat

Publication: Oak Ridge National Labor
Argonne National Laborat

Editorial: Oak Ridge National Labor
Argonne National Laborat

This report is available on the web at http://www.sc.d

ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems

Peter Kogge, Editor & Study Lead
Keren Bergman
Shekhar Borkar
Dan Campbell
William Carlson
William Dally
Monty Denneau
Paul Franzon
William Harrod
Kerry Hill
Jon Hiller
Sherman Karp
Stephen Keckler
Dean Klein
Robert Lucas
Mark Richards
Al Scarpelli
Steven Scott
Allan Snavely
Thomas Sterling
R. Stanley Williams
Katherine Yelick

September 28, 2008

This work was sponsored by DARPA IPTO i
as Program Manager; AFRL contract number
interest of scientific and technical informatio
Government's approval or disapproval of its i

N

Using Government drawings, specifications,
purpose other than Government procurement
The fact that the Government formulated or s
does not license the holder or any other perso
manufacture, use, or sell any patented inventi

APPROVED FOR PUBLIC RELEASE, DIS

SCIENTIFIC GRAND CHALLENGES: CROSSCUTTING TECHNOLOGIES FOR COMPUTING AT THE EXASCALE

Report from the Workshop Held February 2-4, 2010

Sponsored by the U.S. Department of Energy, O
Research, Office of Science; and the Office of A
National Nuclear Security Administration

Chair, **David L. Brown**
Lawrence Livermore National Laboratory

Chair, **Paul Messina**
Argonne National Laboratory

Theme I: Domain Science and System Architecture

Principal Lead, **David Keyes**
King Abdullah University of Science and Technology a

Co-Lead, **John Morrison**
Los Alamos National Laboratory

Co-Lead, **Robert Lucas**
University of Southern California

Co-Lead, **John Shalf**
Lawrence Berkeley National Laboratory

Theme II: System Software

Principal Lead, **Pete Beckman**
Argonne National Laboratory

Co-Lead, **Ron Brightwell**
Sandia National Laboratories

Co-Lead, **Al Geist**
Oak Ridge National Laboratory

Theme III: Programming Models and Environment

Principal Lead, **Jeffrey Vetter**
Oak Ridge National Laboratory and Georgia Institute of Technology

ASCAC Subcommittee for the Top Ten Exascale Research Challenges

Subcommittee Chair
Robert Lucas (University of Southern Californi

Subcommittee Members
James Ang (Sandia National Laboratories)
Keren Bergman (Columbia University)
Shekhar Borkar (Intel)
William Carlson (Institute for Defense Analyses)
Laura Carrington (UC, San Diego)
George Chiu (IBM)
Robert Colwell (DARPA)
William Dally (NVIDIA)
Jack Dongarra (U. Tennessee)
Al Geist (ORNL)
Gary Grider (LANL)
Rud Haring (IBM)
Jeffrey Hittinger (LLNL)
Adolfy Hoisie (PNLL)
Dean Klein (Micron)
Peter Kogge (U. Notre Dame)
Richard Lethin (Reservoir Labs)
Vivek Sarkar (Rice U.)
Robert Schreiber (Hewlett Packard)
John Shalf (LBNL)
Thomas Sterling (Indiana U.)
Rick Stevens (ANL)

The International Exascale Software Project roadmap

The International Journal of High Performance Computing Applications
25(1) 3–60
© The Author(s) 2011
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1094342010391989
hpc.sagepub.com
SAGE

Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Zhong Jin, Yutaka Ishikawa, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr, Matthias S. Mueller, Wolfgang E. Nagel, Hiroshi Nakashima, Michael E Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski and Kathy Yelick

Abstract

Over the last 20 years, the open-source community has provided more and more software on which the world's high-performance computing systems depend for performance and productivity. The community has invested millions of dollars and years of effort to build key components. However, although the investments in these separate software elements have been tremendously valuable, a great deal of productivity has also been lost because of the lack of planning, coordination, and key integration of technologies necessary to make them work together smoothly and efficiently, both within individual petascale systems and between different systems. It seems clear that this completely uncoordinated development model will not provide the software needed to support the unprecedented parallelism required for peta/exascale computation on millions of cores, or the flexibility required to exploit new hardware models and features, such as transactional memory, speculative execution, and graphics processing units. This report describes the work of the community to prepare for the challenges of exascale computing, ultimately combing their efforts in a coordinated International Exascale Software Project.

Keywords

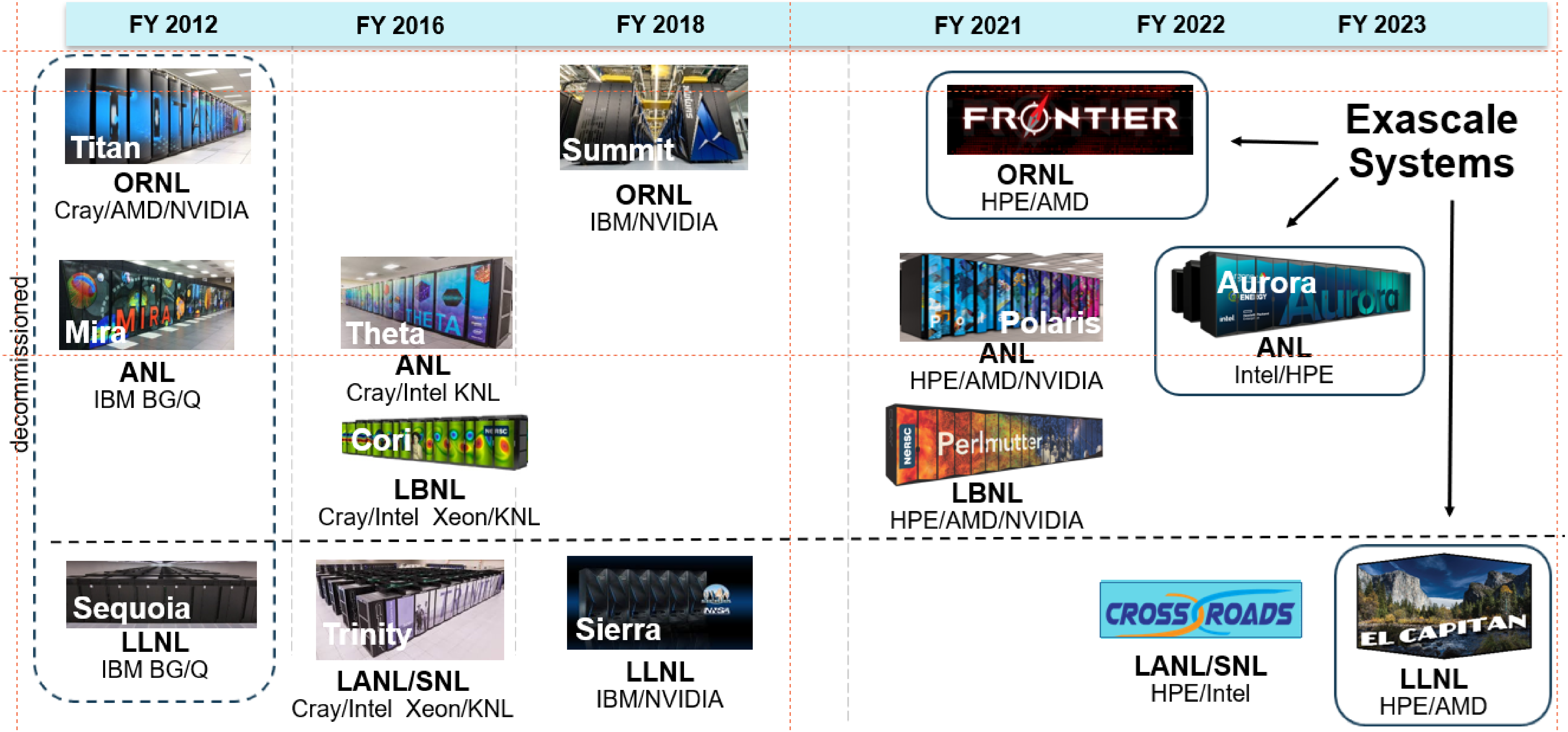
exascale computing, high-performance computing, software stack

Table of Contents

My first trip to Tsukuba!

- International Exascale Software Project
- Meeting 3:
  Tsukuba, Japan
  Oct. 18-20, 2009

DOE HPC Roadmap to Exascale Systems



FY 2012

FY 2016

FY 2018

FY 2021

FY 2022

FY 2023

decommissioned

Titan
ORNL
Cray/AMD/NVIDIA

Mira
ANL
IBM BG/Q

Sequoia
LLNL
IBM BG/Q

Theta
ANL
Cray/Intel KNL

Cori
LBNL
Cray/Intel Xeon/KNL

Trinity
LANL/SNL
Cray/Intel Xeon/KNL

Summit
ORNL
IBM/NVIDIA

Sierra
LLNL
IBM/NVIDIA

Frontier
ORNL
HPE/AMD

Polaris
ANL
HPE/AMD/NVIDIA

Perlmutter
LBNL
HPE/AMD/NVIDIA

Aurora
ANL
Intel/HPE

Crossroads
LANL/SNL
HPE/Intel

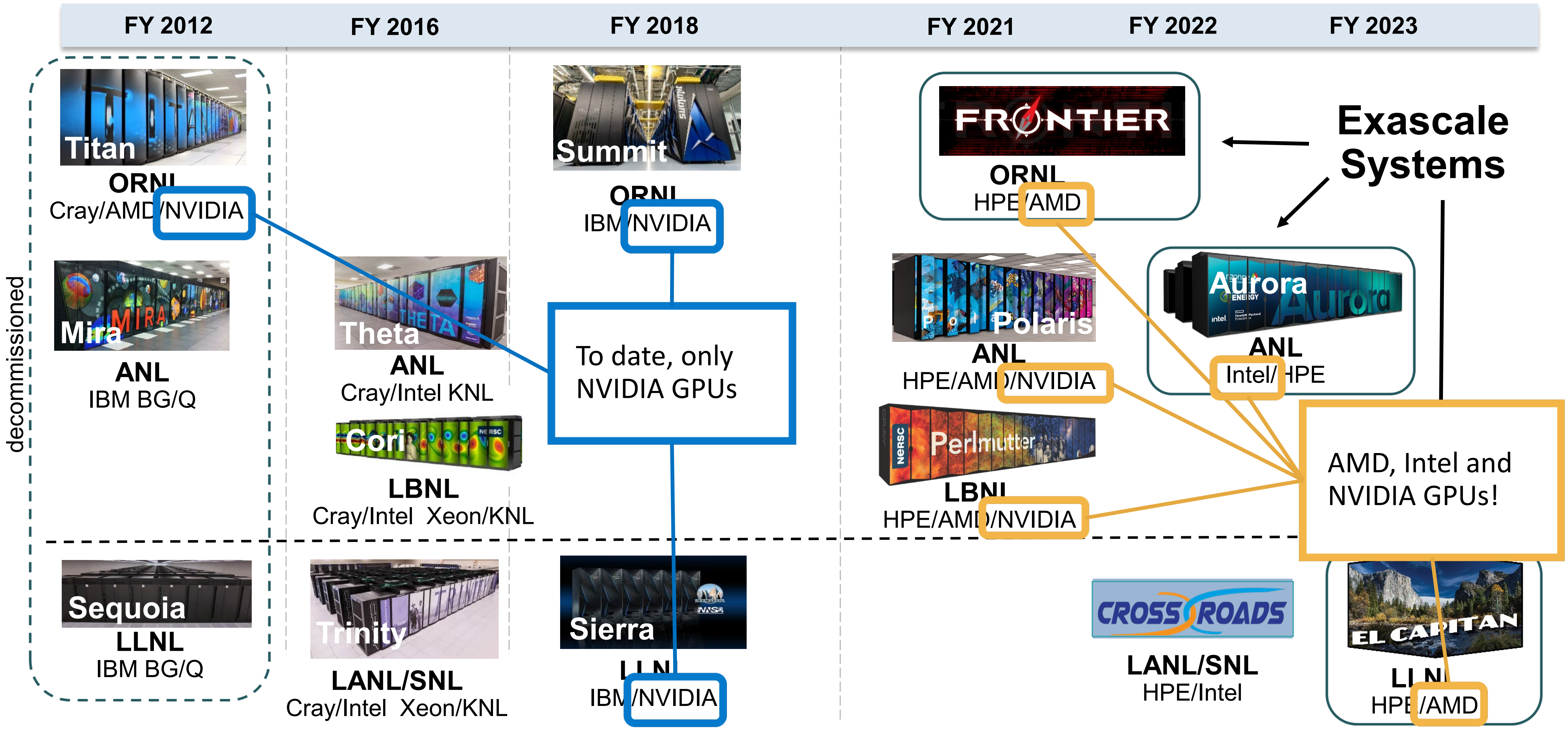
El Capitan
LLNL
HPE/AMD

Exascale Systems

ECP EXASCALE COMPUTING PROJECT

Version 2.0

DOE HPC Roadmap to Exascale Systems



FY 2012 | FY 2016 | FY 2018 | FY 2021 | FY 2022 | FY 2023

decommissioned

Titan
ORNL
Cray/AMD/NVIDIA

Mira
ANL
IBM BG/Q

Sequoia
LLNL
IBM BG/Q

Theta
ANL
Cray/Intel KNL

Cori
LBNL
Cray/Intel Xeon/KNL

Trinity
LANL/SNL
Cray/Intel Xeon/KNL

Summit
ORNL
IBM/NVIDIA

To date, only NVIDIA GPUs

Sierra
LLNL
IBM/NVIDIA

FRONTIER
ORNL
HPE/AMD

Exascale Systems

Polaris
ANL
HPE/AMD/NVIDIA

Aurora
ANL
Intel/HPE

Perlmutter
LBNL
HPE/AMD/NVIDIA

AMD, Intel and NVIDIA GPUs!

CROSSROADS
LANL/SNL
HPE/Intel

EL CAPITAN
LLNL
HPE/AMD

Frontier System

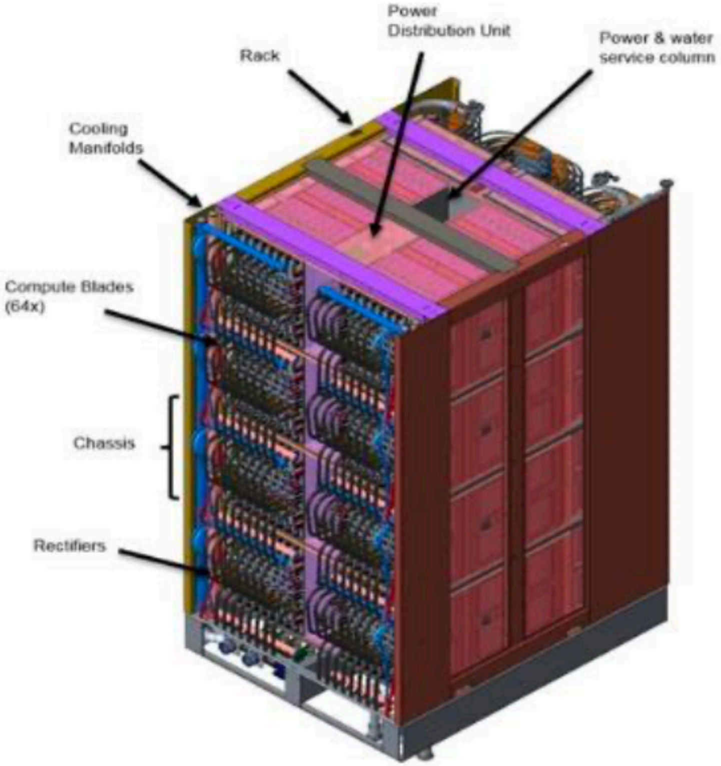
System

- 74 compute racks
- 29 MW Power Consumption
- 9,408 nodes
- 9.2 PB memory (4.6 PB HBM, 4.6 PB DDR4)
- Cray Slingshot network with dragonfly topology
- 37 PB Node Local Storage
- 716 PB Center-wide storage
- 4000 ft^2 foot print

Frontier Cabinet

Olympus rack

- 128 AMD nodes
- 8,000 lbs
- Supports 400 KW



Frontier Node

AMD extraordinary engineering

- 1 AMD "Trento" CPU (optimized Milan)
- 4 AMD MI250X GPUs
- 512 GiB DDR4 memory on CPU
- 512 GiB HBM2e total per node
- 4 Cassini NICs connected to the 4 GPUs

Compute blade

- 2 AMD nodes

All water cooled, even DIMMS and NICs

OAK RIDGE NATIONAL LABORATORY'S FRONTIER SUPERCOMPUTER

OAK RIDGE National Laboratory
U.S. DEPARTMENT OF ENERGY
Hewlett Packard Enterprise
AMD
FRONTIER

- 74 HPE Cray EX cabinets
- 9,408 AMD EPYC CPUs, 37,632 AMD GPUs
- 700 petabytes of storage capacity, peak write speeds of 5 terabytes per second using Cray Clusterstor Storage System
- 90 miles of HPE Slingshot networking cables

TOP500

1

1.1 exaflops of performance on the May 2022 Top500.

GREEN500

1, # 2

62.04 gigaflops/watt power efficiency on a single cabinet.

52.23 gigaflops/watt power efficiency on the full system.

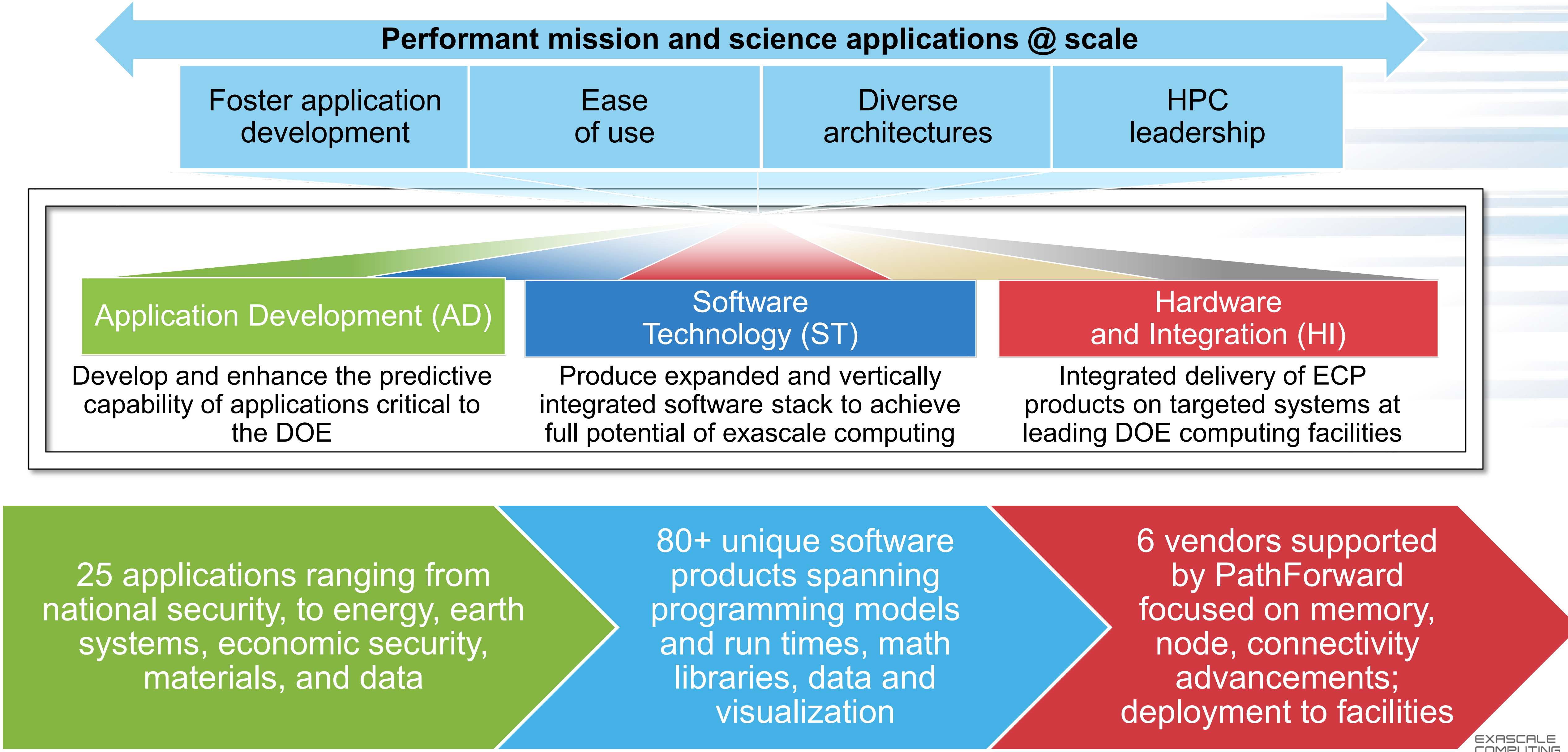
HPL-AI

1

6.88 exaflops on the HPL-AI benchmark.

Sources: May 30, 2022 Top500 release

Exascale Computing Project has three technical areas to meet national goals



Performant mission and science applications @ scale

Foster application development

Ease of use

Diverse architectures

HPC leadership

Application Development (AD)

Develop and enhance the predictive capability of applications critical to the DOE

Software Technology (ST)

Produce expanded and vertically integrated software stack to achieve full potential of exascale computing

Hardware and Integration (HI)

Integrated delivery of ECP products on targeted systems at leading DOE computing facilities

25 applications ranging from national security, to energy, earth systems, economic security, materials, and data

80+ unique software products spanning programming models and run times, math libraries, data and visualization

6 vendors supported by PathForward focused on memory, node, connectivity advancements; deployment to facilities

https://www.exascaleproject.org/

EXASCALE COMPUTING PROJECT

Application Development KPP-1 and KPP-2 Readiness Overview

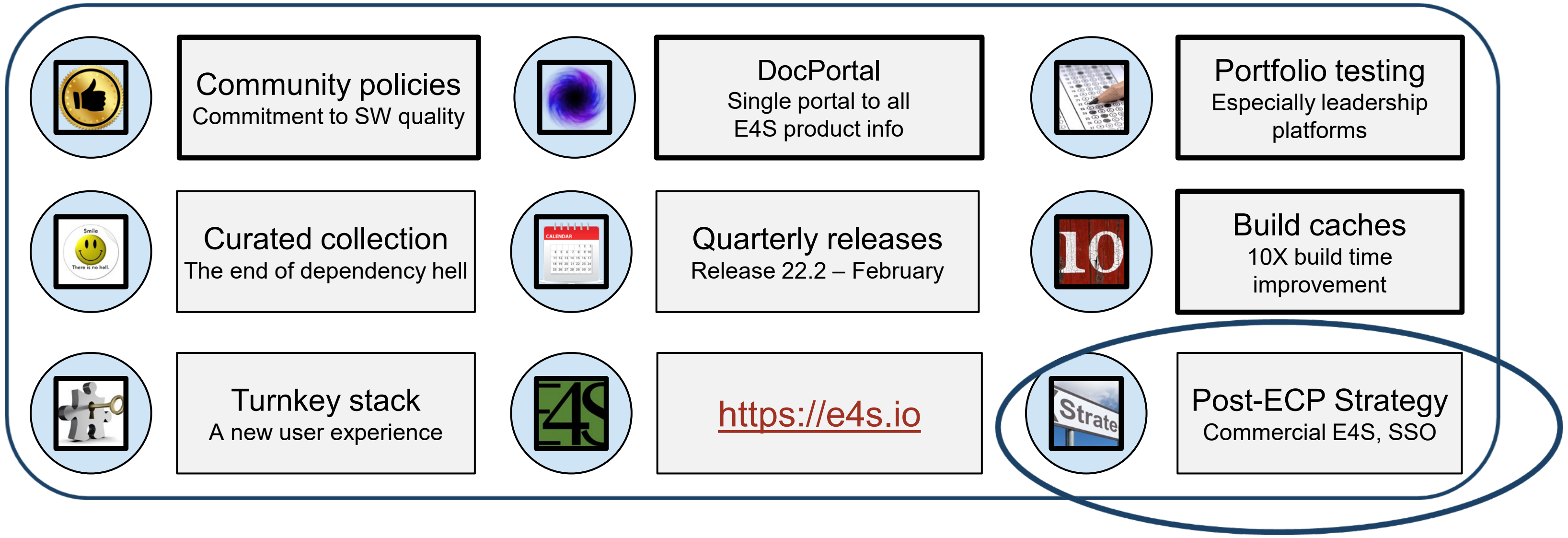
KPP-1 App	Aurora EAS (Intel Proprietary)	Frontier TDS
LatticeQCD	Verified	Improving Perf.
NWChemEx	Full Build/Test	Initial Build/Test
EXAALT	Verified	Improving Perf.
QMCPACK	Initial Build/Test	Improving Perf.
ExaSMR	Improving Perf.	Improving Perf.
WDMApp	Improving Perf.	Improving Perf.
WarpX	Verified	Improving Perf.
ExaSky	Improving Perf.	Improving Perf.
EQSIM	Initial Build/Test	Improving Perf.
E3SM-MMF	Improving Perf.	Improving Perf.
CANDLE	Ready	Improving Perf.

KPP-2 App	Aurora EAS (Intel Proprietary)	Frontier TDS
GAMESS	Improving Perf.	Improving Perf.
ExaAM	Initial Build/Test	Improving Perf.
ExaWind	Verified	Improving Perf.
Combustion-PELE	Initial Build/Test	Improving Perf.
MFIX-Exa	Verified	Improving Perf.
ExaStar	Full Build/Test	Improving Perf.
Subsurface	Stretch	Improving Perf.
ExaSGD	Stretch	Improving Perf.
ExaBiome	Stretch	Improving Perf.
ExaFEL	Full Build/Test	Blocked (ROCm)

ExaFel (Blocked ROCm) – The project team noticed during a recent work around to compiler bugs preventing compilation of Spinifel relies on an upstream compiler built locally by the OLCF team. Integration into a formal ROCm release is still pending, so we have chosen to mark this as formally blocked even though the team can currently make progress with the unofficial compiler build.

Extreme-scale Scientific Software Stack (E4S)

- E4S: HPC software ecosystem – a curated software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source**, **containers**, **cloud, binary caches**
- Leverages and enhances SDK interoperability thrust
- Not a commercial product – an open resource for all
- Growing functionality: May 2022: E4S 22.05 – 100+ full release products



Community policies
Commitment to SW quality

DocPortal
Single portal to all E4S product info

Portfolio testing
Especially leadership platforms

Curated collection
The end of dependency hell

Quarterly releases
Release 22.2 – February

Build caches
10X build time improvement

Turnkey stack
A new user experience

https://e4s.io

Post-ECP Strategy
Commercial E4S, SSO

https://spack.io

Spack lead: Todd Gamblin (LLNL)

https://e4s.io

E4S lead: Sameer Shende (U Oregon)

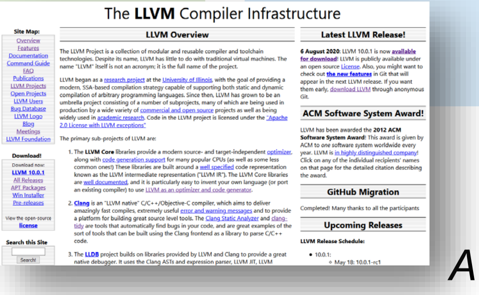
Also includes other products, e.g.,
AI: PyTorch, TensorFlow, Horovod
Co-Design: AMReX, Cabana, MFEM

ECP is Improving the LLVM Compiler Ecosystem

LLVM COMPILER INFRASTRUCTURE

LLVM

- Very popular open-source **compiler infrastructure**
- **Permissive** license
- **Modular**, well-defined IR allows use by a lot of different languages, ML frameworks, etc.
- **Backend infrastructure** allowing the efficient creation of backends for new (heterogeneous) hardware.
- A **state-of-the-art C++ frontend**, CUDA support, scalable LTO, sanitizers and other debugging capabilities, and more.



The **LLVM** Compiler Infrastructure

Site Map:

Overview
Features
Documentation
Command Guide
FAQ
Publications
LLVM Projects
Open Projects
LLVM Users
Bug Database
LLVM Logo
Blog
Meetings
LLVM Foundation

Download!

Download now:
LLVM 10.0.1
All Releases
APT Packages
Win Installer
Pre-releases

View the open-source
license

Search this Site

Search!

LLVM Overview

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.

LLVM began as a research project at the University of Illinois, with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of commercial and open source projects as well as being widely used in academic research. Code in the LLVM project is licensed under the "Apache 2.0 License with LLVM exceptions"

The primary sub-projects of LLVM are:

1. The **LLVM Core** libraries provide a modern source- and target-independent optimizer, along with code generation support for many popular CPUs (as well as some less common ones!) These libraries are built around a well specified code representation known as the LLVM intermediate representation ("LLVM IR"). The LLVM Core libraries are well documented, and it is particularly easy to invent your own language (or port an existing compiler) to use LLVM as an optimizer and code generator.
2. **Clang** is an "LLVM native" C/C++/Objective-C compiler, which aims to deliver amazingly fast compiles, extremely useful error and warning messages and to provide a platform for building great source level tools. The Clang Static Analyzer and clang-tidy are tools that automatically find bugs in your code, and are great examples of the sort of tools that can be built using the Clang frontend as a library to parse C/C++ code.
3. The **LLDB** project builds on libraries provided by LLVM and Clang to provide a great native debugger. It uses the Clang ASTs and expression parser, LLVM JIT, LLVM

Latest LLVM Release!

6 August 2020: LLVM 10.0.1 is now **available for download**! LLVM is publicly available under an open source License. Also, you might want to check out **the new features** in Git that will appear in the next LLVM release. If you want them early, download LLVM through anonymous Git.

ACM Software System Award!

LLVM has been awarded the **2012 ACM Software System Award**! This award is given by ACM to *one* software system worldwide every year. LLVM is in highly distinguished company! Click on any of the individual recipients' names on that page for the detailed citation describing the award.

GitHub Migration

Completed! Many thanks to all the participants

Upcoming Releases

LLVM Release Schedule:

- 10.0.1:
  - May 18: 10.0.1-rc1

+ SOLLVE

- Enhancing the implementation of OpenMP in LLVM
- Unified memory
- Prototype OMP features for LLVM
- OMP Optimizations
- OMP test suite
- Tracking OMP implementation quality
- Training

+ PROTEAS-TUNE

- Core optimization improvements to LLVM
  - OpenMP offload
- OpenACC capability for LLVM
  - Clacc
  - Flacc
- Autotuning for OpenACC and OpenMP in LLVM
- Integration with Tau performance tools
- SYCL characterizing and benchmarking
- Leading LLVM-DOE fork
- Training

+ FLANG

- Developing an open-source, production Fortran frontend
- Upstream to LLVM public release
- Support for OpenMP and OpenACC
- Recently approved by LLVM
- Initial implementation of serial F77 compiler for CPUs under review

+ HPCToolkit

- Improvements to OpenMP profiling interface OMPT
- OMPT specification improvements
- Refine HPCT for OMPT improvements

+ NNSA

- Enhancing LLVM to optimize template expansion for FlexCSI, Kokkos, RAJA, etc.
- Flang testing and evaluation
- Kitsune and Tapir

Vendors

- Increasing dependence on LLVM
- Many vendors import and redistribute LLVM
- Contributions and collaborations with many vendors through LLVM
  - AMD
  - ARM
  - Cray
  - HPE
  - IBM
  - Intel
  - NVIDIA

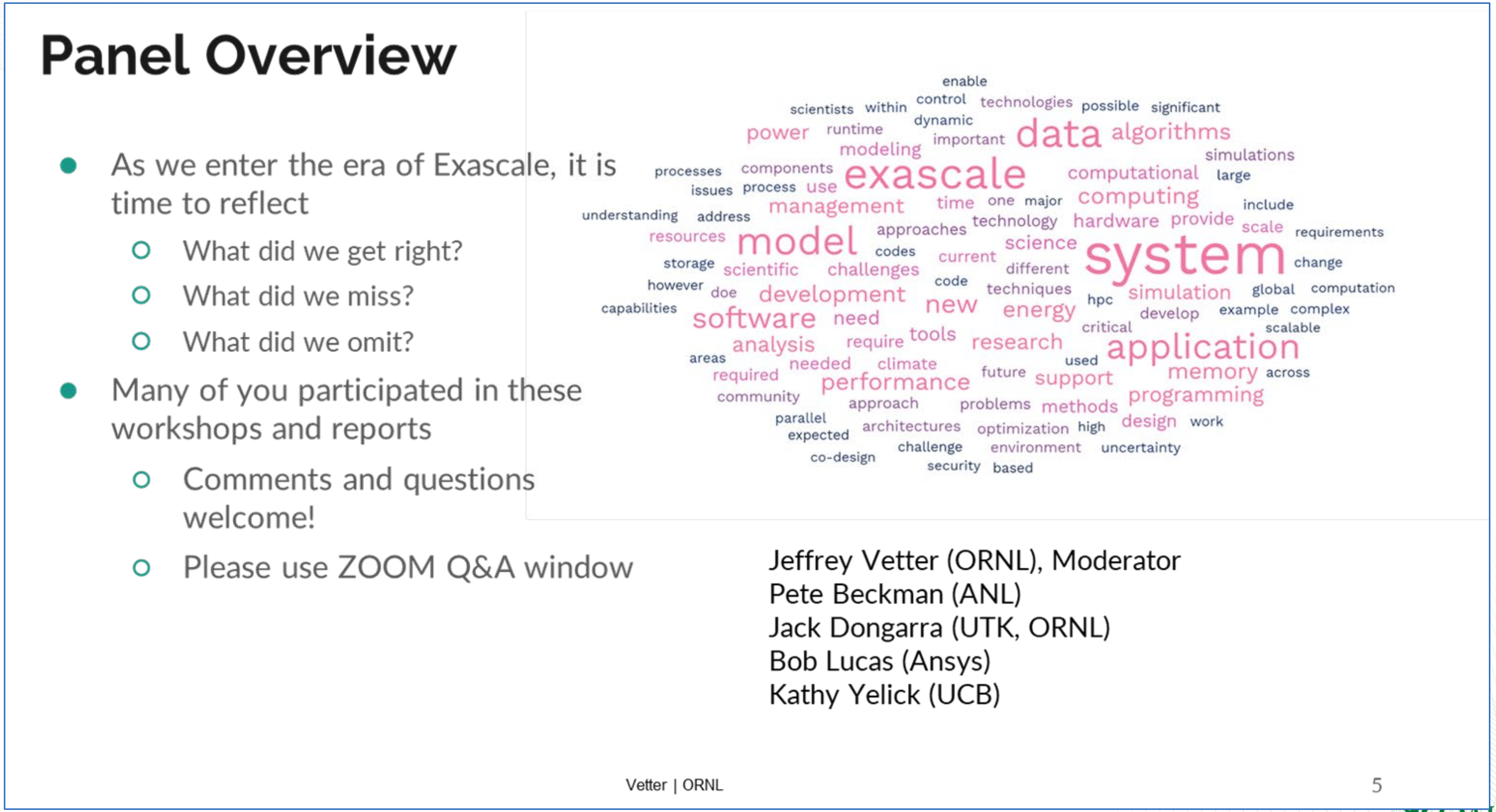
Other ECP activities with LLVM emerging organically.

Active involvement with broad LLVM community: LLVM Dev, EuroLLVM
ECP personnel had 10+ presentations at the 2020 Dev Meeting

OAK RIDGE National Laboratory

So how did we do?

Reflections from ECP Panel at ECP AHM in May 2022



Panel Overview

- As we enter the era of Exascale, it is time to reflect
  - What did we get right?
  - What did we miss?
  - What did we omit?
- Many of you participated in these workshops and reports
  - Comments and questions welcome!
  - Please use ZOOM Q&A window

enable
scientists within control technologies possible significant
dynamic
power runtime important data algorithms
modeling simulations
processes components exascale computational large
issues process use time one major computing include
understanding address management technology hardware provide
resources model approaches science scale requirements
codes
storage scientific challenges current system change
different
however doe code techniques
development global computation
capabilities new hpc simulation
software need energy develop example complex
critical scalable
analysis require tools research application
areas needed climate used
required future support memory across
performance programming
community approach problems methods
parallel architectures optimization high design work
expected challenge environment uncertainty
co-design security based

Jeffrey Vetter (ORNL), Moderator
Pete Beckman (ANL)
Jack Dongarra (UTK, ORNL)
Bob Lucas (Ansys)
Kathy Yelick (UCB)

Vetter | ORNL 5

15 years is an eternity in computing - How did our predictions do?

Hits

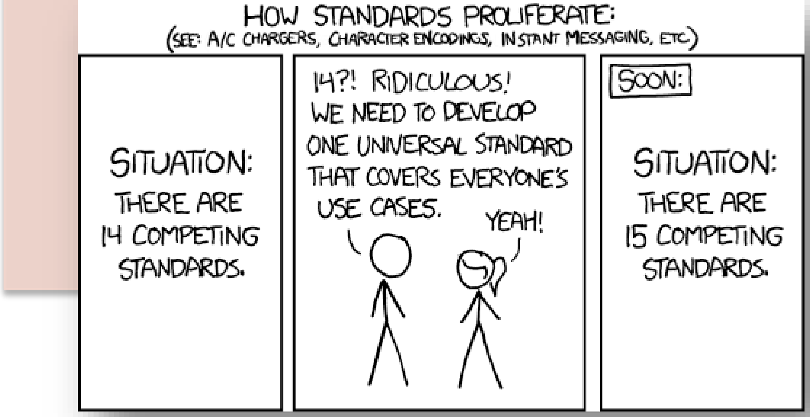
- System power came in at O(20MW) not O(1GW)
- Few major software rewrites / evolution
  - So far, FORTRAN -> C++ is the main conversion
- ECP included applications, software, and hardware
  - ~70 teams, ~1000 researchers
  - IESP
- Concurrency (1B-way parallelism)
- Open-source software

Misses

- Systems deployed 4 years later than expected (of 2018)
- Programming systems are multiplying and immature/incomplete
- Hardware diversity
- Demise of vendor interest in HPC
- Fault tolerance

Overlooked

- Productive programming models (ala AI/ML): Python, Jupyter, Julia
- Cost of ECP + NRE + Procurements approaches $3.6B USD
- AI/ML is not predicted (or even mentioned)
- Cloud deployment models
- Green/sustainable computing


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Pondering Post-Exascale Computing

- Thinking about the next 10 years

Important Architectural Trends

- Heterogeneous integration
- Chiplets
- Ecosystems and Standards
  - CXL, UCIe, BoW, …
- Open-source Tools and IP
  - RISC-V, OpenLane, Silicon Compiler, etc
- Open foundries

- ***Codesign will be more important than ever***

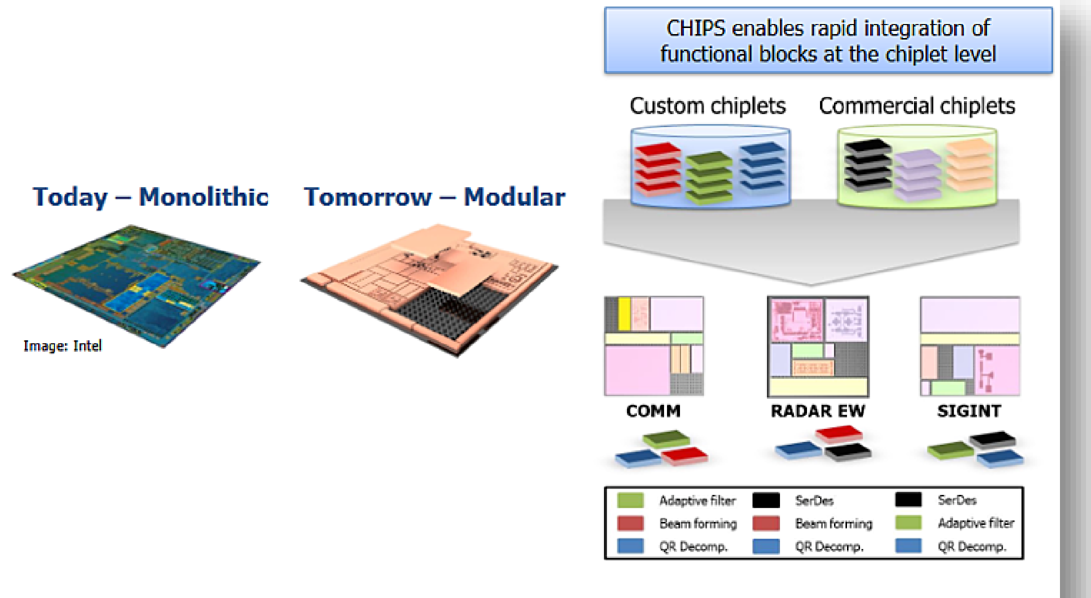
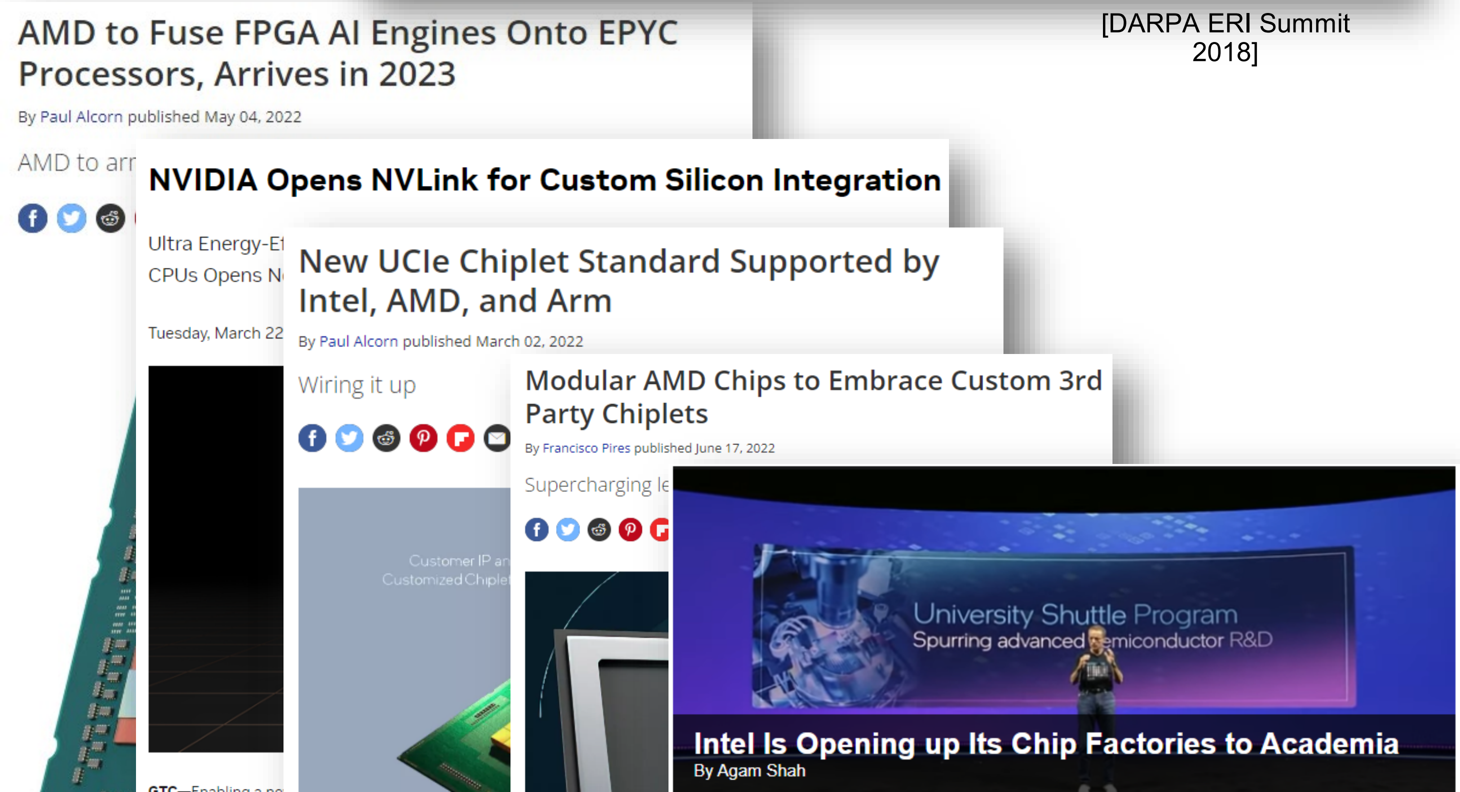


Figure 1. CHIPS Vision (DARPA)

[DARPA ERI Summit 2018]



AMD to Fuse FPGA AI Engines Onto EPYC Processors, Arrives in 2023

By Paul Alcorn published May 04, 2022

AMD to arr

NVIDIA Opens NVLink for Custom Silicon Integration

Ultra Energy-Ef

CPUs Opens N

Tuesday, March 22

GTC—Enabling a ne

New UCIe Chiplet Standard Supported by Intel, AMD, and Arm

By Paul Alcorn published March 02, 2022

Wiring it up

Customer IP an

Customized Chiplet

Modular AMD Chips to Embrace Custom 3rd Party Chiplets

By Francisco Pires published June 17, 2022

Supercharging le

University Shuttle Program

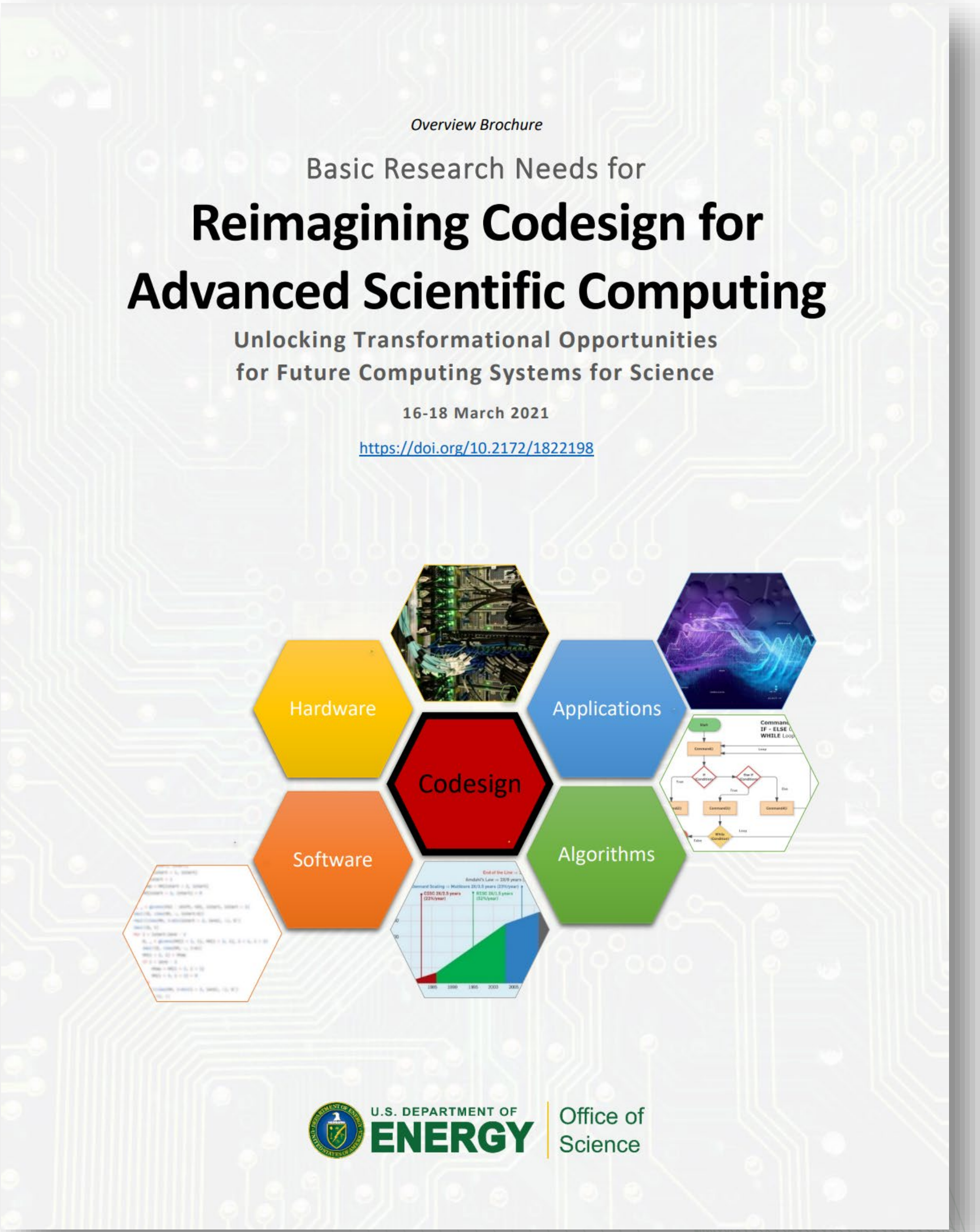
Spurring advanced semiconductor R&D

Intel Is Opening up Its Chip Factories to Academia

By Agam Shah

Reimagining Codesign

- 2021 Workshop
- Four priority research directions
  - Drive Breakthrough Computing Capabilities with Targeted Heterogeneity and Rapid Design
  - Software and Applications that Embrace Radical Architecture Diversity
  - Engineered Security and Integrity from Transistors to Applications
  - Design with Data-Rich Processes
- We must make codesign agile, more accurate, and use real workloads



Overview Brochure

Basic Research Needs for

Reimagining Codesign for Advanced Scientific Computing

Unlocking Transformational Opportunities for Future Computing Systems for Science

16-18 March 2021

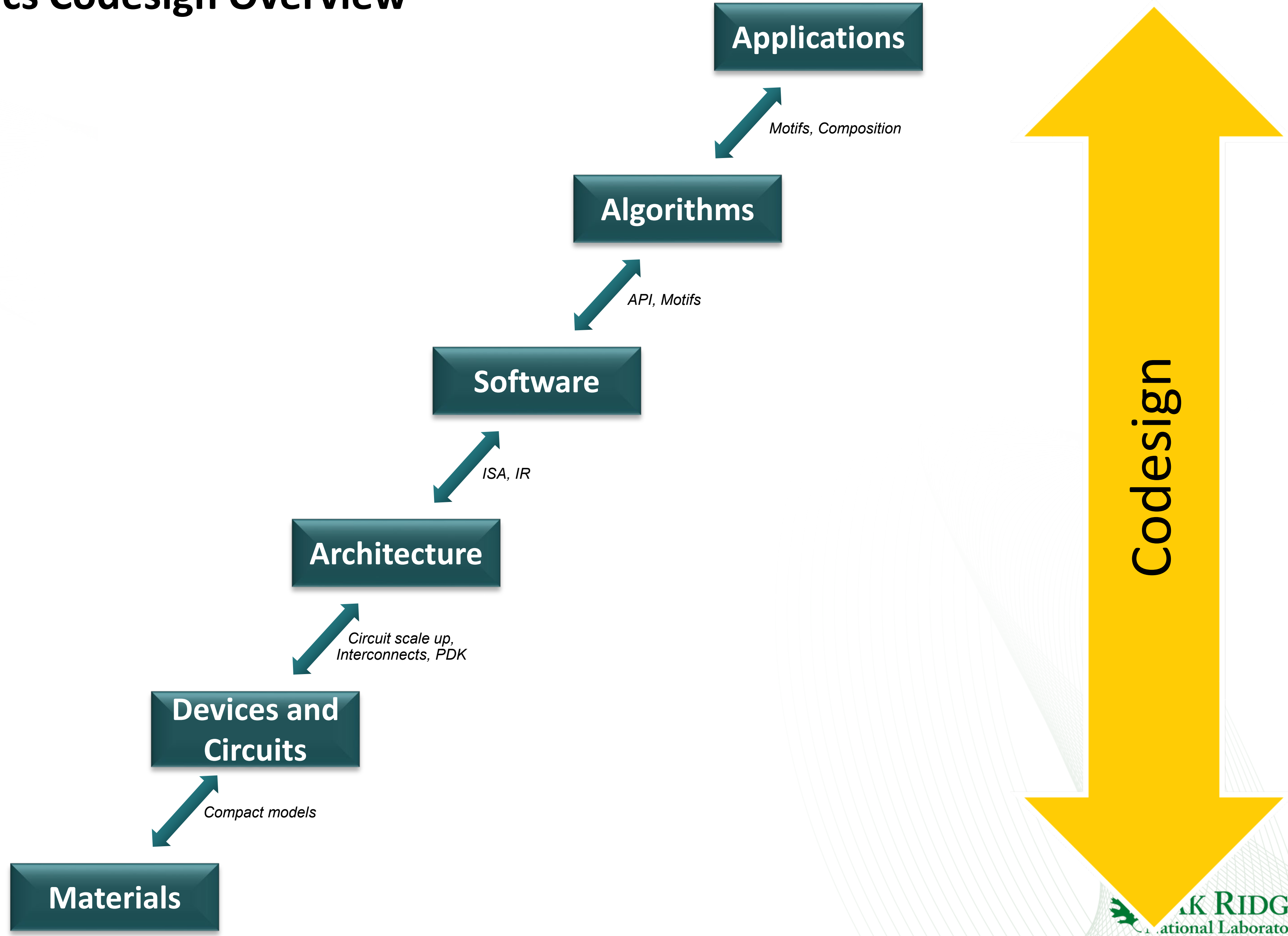
https://doi.org/10.2172/1822198

U.S. DEPARTMENT OF ENERGY | Office of Science

https://www.osti.gov/biblio/1822198-reimagining-codesign-advanced-scientific-computing-unlocking-transformational-opportunities-future-computing-systems-science

Abisko: Microelectronics Codesign

Abisko Microelectronics Codesign Overview



Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory

Sandia National Laboratories

Arizona State University

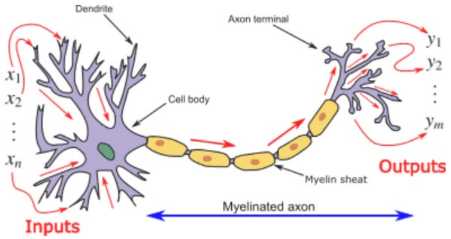
Georgia Tech

Harvard University

Collaborator

Fermilab

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet



Source: Wikipedia

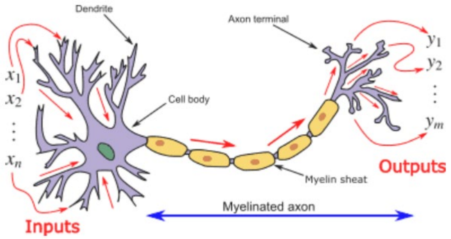
Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory · Sandia National Laboratories · Arizona State University · Georgia Tech · Harvard University

Collaborator

Fermilab

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet



Source: Wikipedia

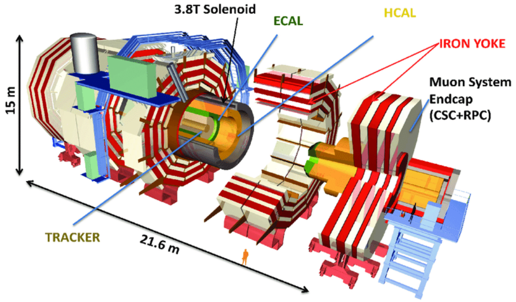
Motivation

- Transportation
- CMS Sensors

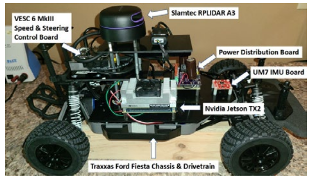
Applications

Motifs, Composition

CMS



F1Tenth



VESC 6 MkIII Speed & Steering Control Board
Slamtec RPLIDAR A3
Power Distribution Board
UM7 IMU Board
Nvidia Jetson TX2
Traxxas Ford Fiesta Chassis & Drivetrain

Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory

Sandia National Laboratories

Arizona State University

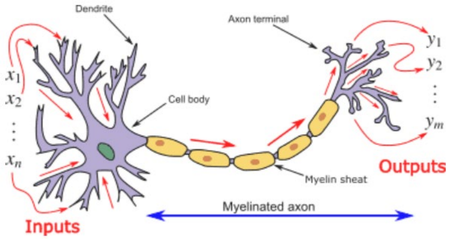
Georgia Tech

Harvard University

Collaborator

Fermilab

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet



Source: Wikipedia

Algorithms

- ML: SLAYER, Whetstone, EONS, ePropp, STDP
- Non-ML: Graph algorithms, CSP
- Simulators: NEST, Brian2

Motivation

- Transportation
- CMS Sensors

Applications

Motifs, Composition

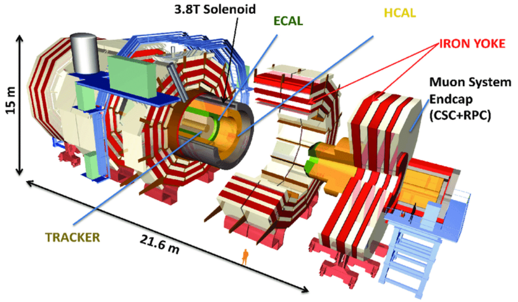
Algorithms

API, Motifs

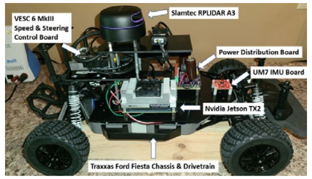
nest::

BRIAN

CMS



F1Tenth


VESC 6 MkIII Speed & Steering Control Board
Slamtec RPLIDAR A3
Power Distribution Board
UM7 IMU Board
Nvidia Jetson TX2
Traxxas Ford Fiesta Chassis & Drivetrain

Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory

Sandia National Laboratories

Arizona State University

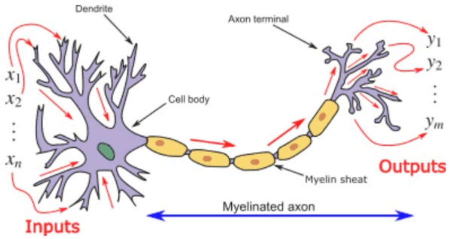
Georgia Tech

Harvard University

Collaborator

Fermilab

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet



Source: Wikipedia

Motivation

- Transportation
- CMS Sensors

Algorithms

- ML: SLAYER, Whetstone, EONS, eProp, STDP
- Non-ML: Graph algorithms, CSP
- Simulators: NEST, Brian2

Software

- DSL and API for neuromorphic co-processing
- Built on LLVM and MLIR
- Portable across Abisko chiplet, GPUs, etc.

Applications

Motifs, Composition

Algorithms

nest::

BRIAN

API, Motifs

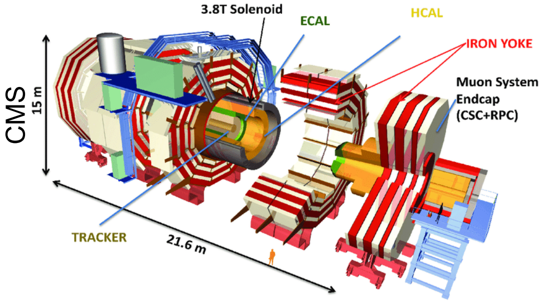
Software

LLVM COMPILER INFRASTRUCTURE

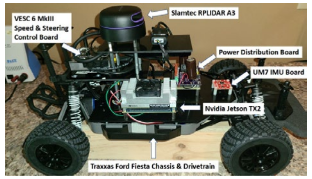
MLIR

XACC

ISA, IR



F1Tenth



Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory

Sandia National Laboratories

Arizona State University

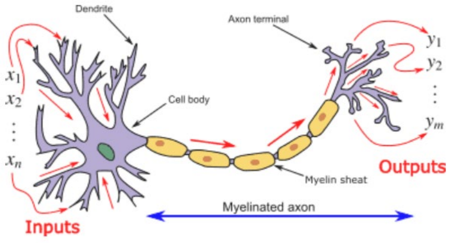
Georgia Tech

Harvard University

Collaborator

Fermilab

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet

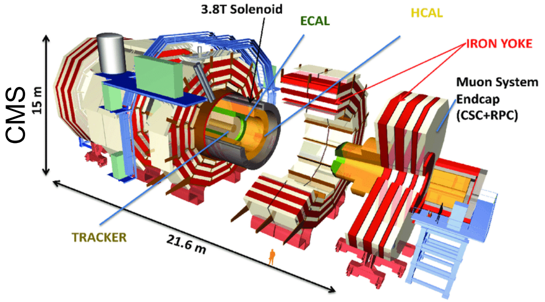


Source: Wikipedia

Motivation

- Transportation
- CMS Sensors

Applications



Motifs, Composition

Algorithms

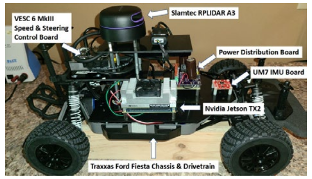
- ML: SLAYER, Whetstone, EONS, eProp, STDP
- Non-ML: Graph algorithms, CSP
- Simulators: NEST, Brian2

Algorithms

nest::

BRIAN

F1Tenth



VESC 6 MkIII Speed & Steering Control Board
Slamtec RPLIDAR A3
Power Distribution Board
UM7 IMU Board
Nvidia Jetson TX2
Traxxas Ford Fiesta Chassis & Drivetrain

API, Motifs

Software

- DSL and API for neuromorphic co-processing
- Built on LLVM and MLIR
- Portable across Abisko chiplet, GPUs, etc.

Software

LLVM COMPILER INFRASTRUCTURE

MLIR

XACC

ISA, IR

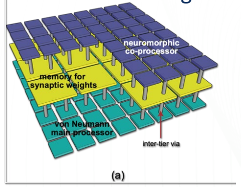
Architecture

- Design neuromorphic chiplet
- RISC-V neuromorphic extensions
- Heterogeneous integration with contemporary technologies

Architecture

RISC-V

2.5D and 3D integration



neuromorphic co-processor
memory for synaptic weights
von Neumann main processor
inter-tier via
(a)

Simulation/Emulation

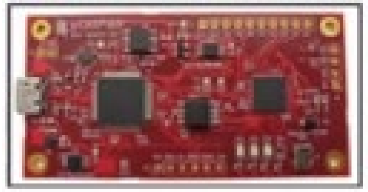
ALADDIN

gem5

Circuit scale up, Interconnects, PDK

Chiplet Architectures

- Design an (analog) SNN chiplet that can be easily integrated with contemporary technologies
  - Heterogeneous integration with mixed processes
  - Compatible with existing processes
- Extensive advances in chiplets, packaging, and heterogeneous integration recently
  - Open Domain-Specific Architecture
  - UCIe, BoW, TSMC SoIC-CoW, Intel Foveros
- Using open toolchain and architecture to explore chiplet designs: RISC-V, OpenLane



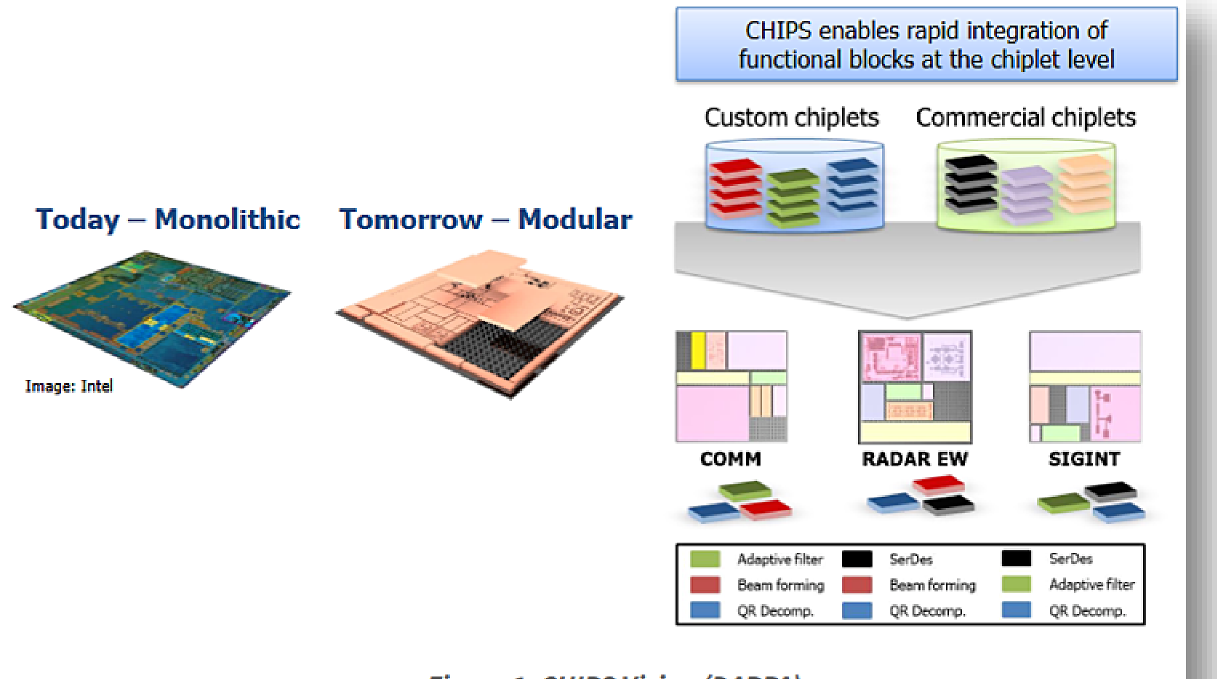


Figure 1. CHIPS Vision (DARPA)

[DARPA ERI Summit 2018]

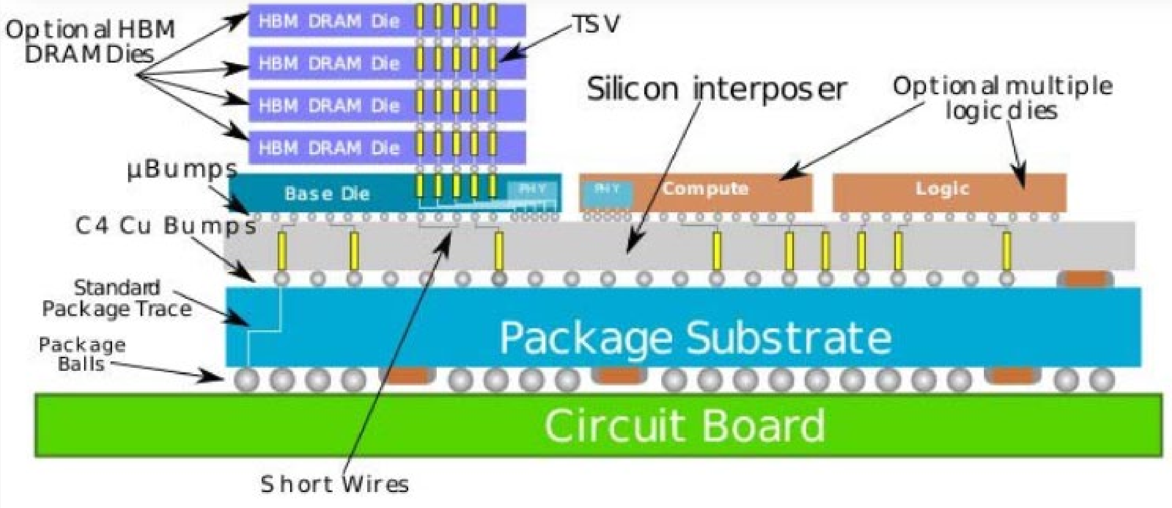
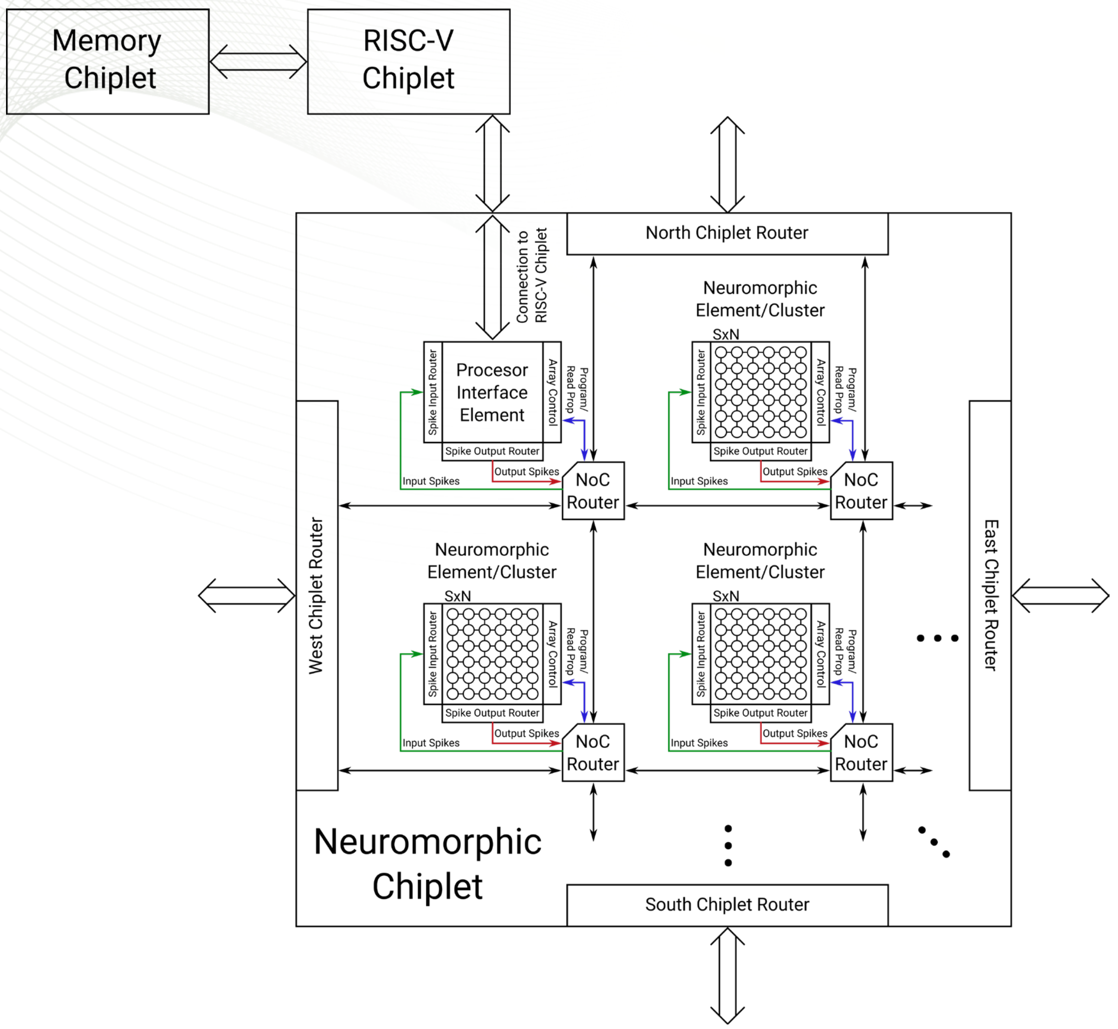


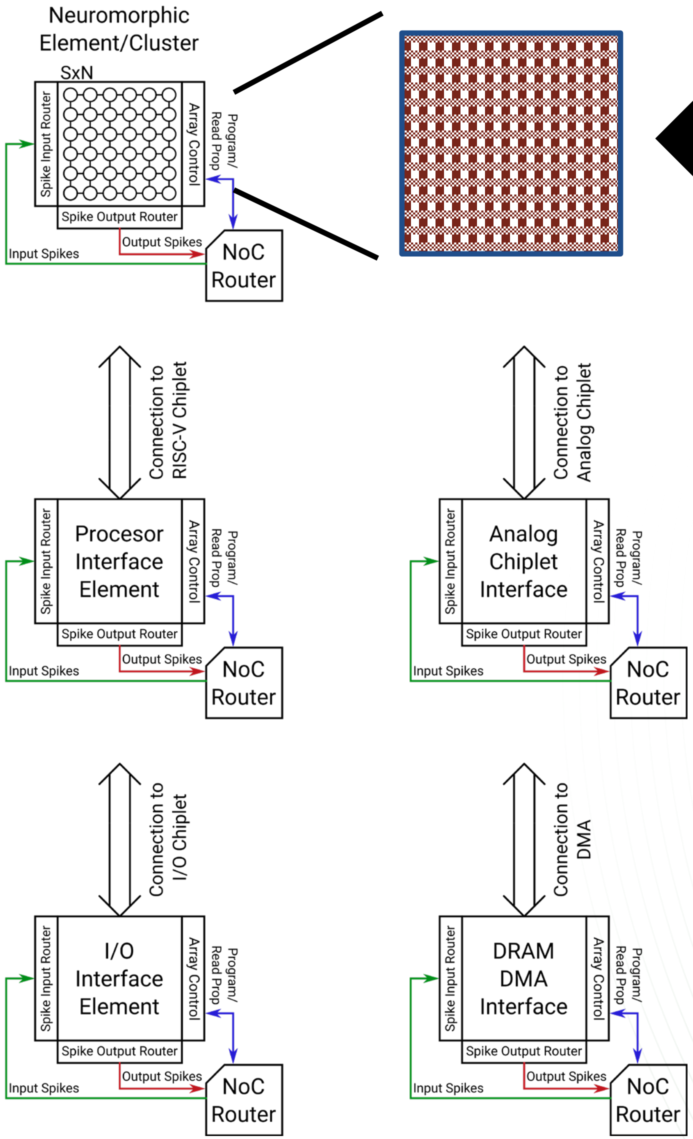
Figure 21. An example showing the use of 2D and 3D interconnections (courtesy TSMC)

[IEEE HIR 2021]

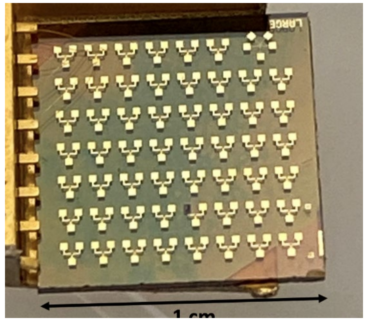
Evaluation of 2.5D Chiplet for Neuromorphic Computing



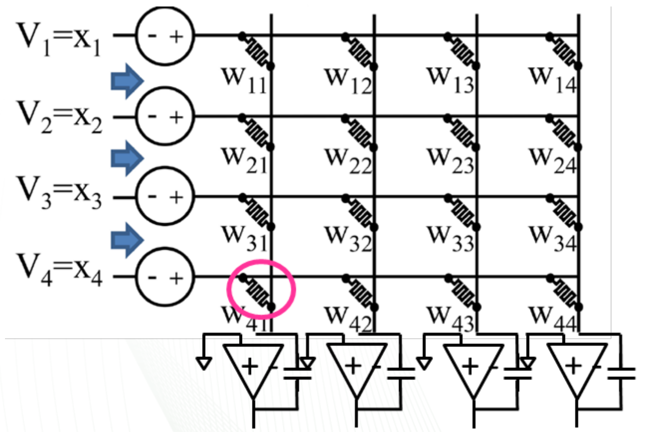
Element Types:



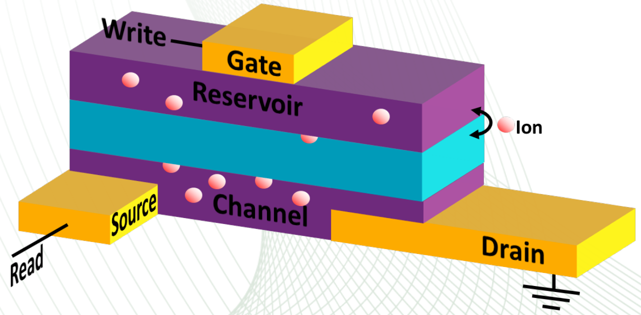
refine
improve



Sandia VO$_2$ ECRAM

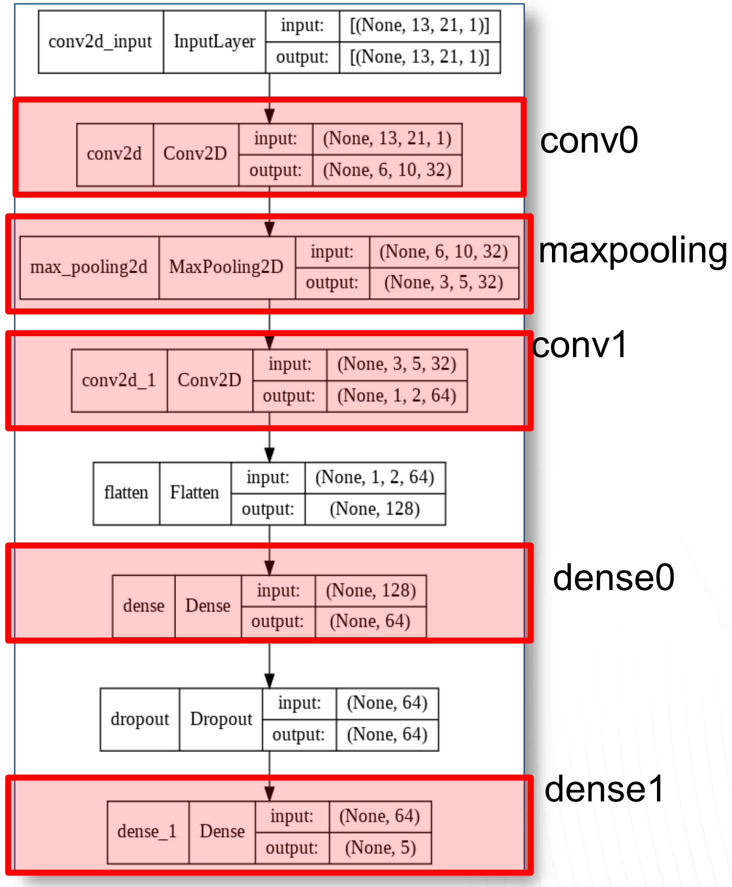


$V_1=x_1$
$V_2=x_2$
$V_3=x_3$
$V_4=x_4$
- +
W_{11} W_{12} W_{13} W_{14}
W_{21} W_{22} W_{23} W_{24}
W_{31} W_{32} W_{33} W_{34}
W_{41} W_{42} W_{43} W_{44}



ASIC Flow for Digital NN (baseline)

- Investigate the performance of fully customized ASIC design for ultra-fast NN inference
  - ORNL: HLS and RTL
  - Geogia Tech: ASIC Synthesis and PD
- Model details:
  - Fixed NN architecture with quantized weights
  - Experimented with 2bit or 3bit of inputs (limited by FermiLab implementation)
- Flow:
  - Vitis HLS to generate RTL
  - Catapult logic synthesis
  - Customized backend layout tool (incl. tech mapping, placement and routing)
- Achieved clock frequency of 1~ 2GHz in a 28nm technology



conv2d_input | InputLayer | input: [(None, 13, 21, 1)] | output: [(None, 13, 21, 1)]

conv2d | Conv2D | input: (None, 13, 21, 1) | output: (None, 6, 10, 32)

max_pooling2d | MaxPooling2D | input: (None, 6, 10, 32) | output: (None, 3, 5, 32)

conv2d_1 | Conv2D | input: (None, 3, 5, 32) | output: (None, 1, 2, 64)

flatten | Flatten | input: (None, 1, 2, 64) | output: (None, 128)

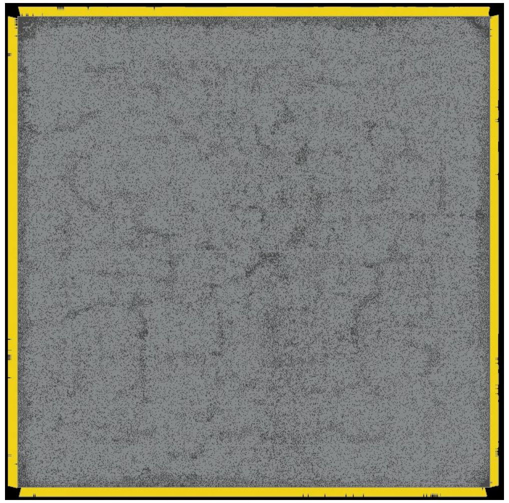
dense | Dense | input: (None, 128) | output: (None, 64)

dropout | Dropout | input: (None, 64) | output: (None, 64)

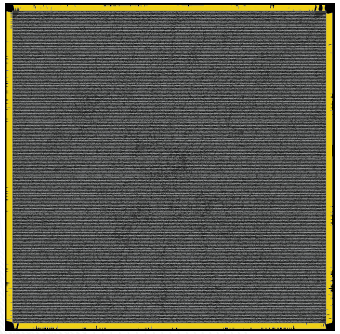
dense_1 | Dense | input: (None, 64) | output: (None, 5)

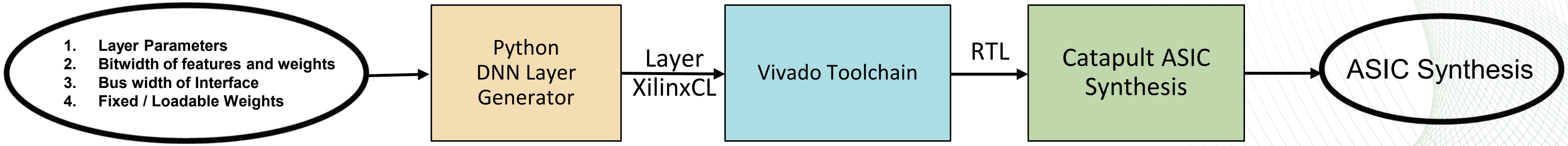
conv 0

3bits 794 um



2bits 526 um





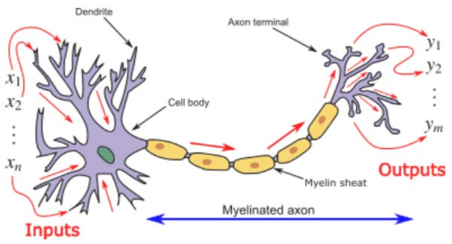
1. Layer Parameters
2. Bitwidth of features and weights
3. Bus width of Interface
4. Fixed / Loadable Weights

Python DNN Layer Generator → Layer XilinxCL → Vivado Toolchain → RTL → Catapult ASIC Synthesis → ASIC Synthesis

Abisko Microelectronics Codesign Overview

Collaborator

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet

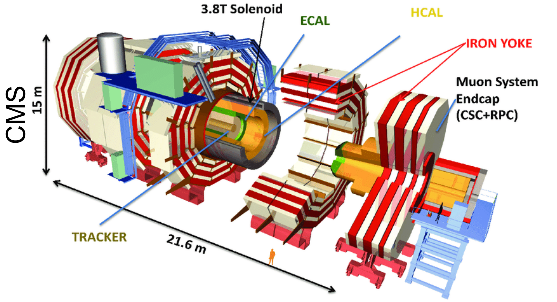


Source: Wikipedia

Motivation

- Transportation
- CMS Sensors

Applications



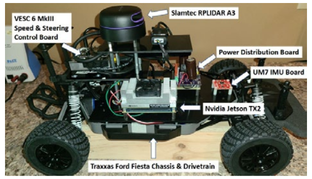
Motifs, Composition

Algorithms

- ML: SLAYER, Whetstone, EONS, eProp, STDP
- Non-ML: Graph algorithms, CSP
- Simulators: NEST, Brian2

Algorithms

F1Tenth



API, Motifs

Software

- DSL and API for neuromorphic co-processing
- Built on LLVM and MLIR
- Portable across Abisko chiplet, GPUs, etc.

Software

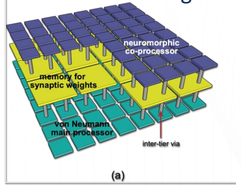
ISA, IR

Architecture

- Design neuromorphic chiplet
- RISC-V neuromorphic extensions
- Heterogeneous integration with contemporary technologies

Architecture

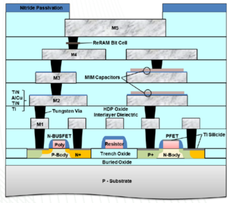
2.5D and 3D integration



neuromorphic co-processor
memory for synaptic weights
von Neumann main processor
inter-tier via
(a)

Simulation/Emulation

MesaFAB ReRAM



Circuit scale up, Interconnects, PDK

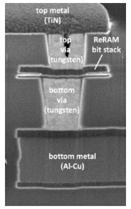
Devices and Circuits

- ion insertion (reversible doping) sets analog states
- mRaman captures transition linear, non-linear switching
- Will extend to 36x36 x-bar array
- Electronic and other optical spectroscopies

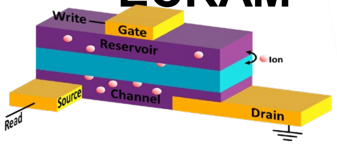
Devices and Circuits

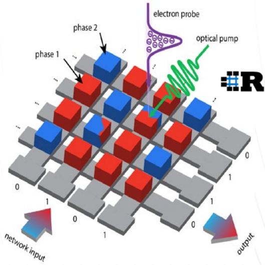
Compact models

TaOx ReRAM



ECRAM





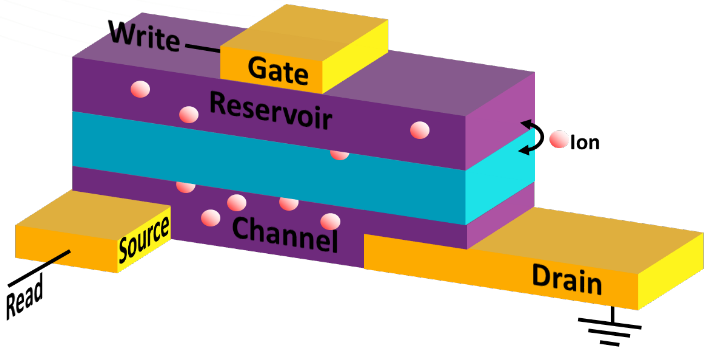
ROSS SIM

Devices and Circuits

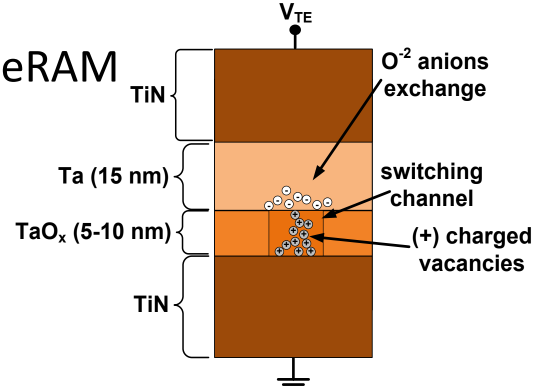
- Goals
  - Harness the interplay between mobile defects (ions and vacancies) and electronic properties to realize functional elements for spiking and non-spiking analog neuromorphic networks
  - Create and validate small network models; generate device and network data for co-design
  - Understand and mitigate radiation induced degradation mechanisms at the device and circuit level

Devices

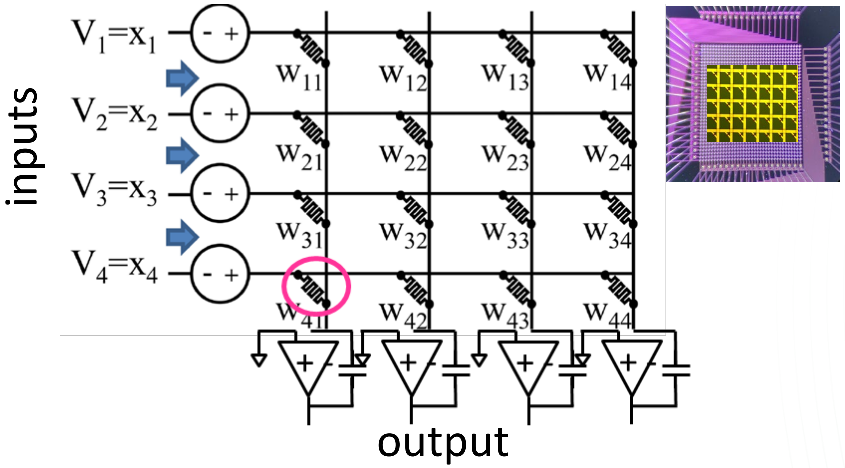
1) ECRAM



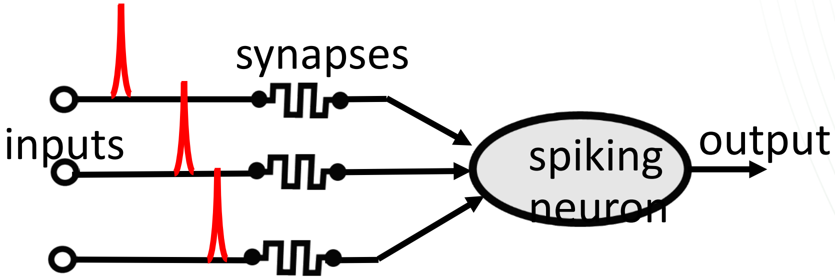
2) ReRAM



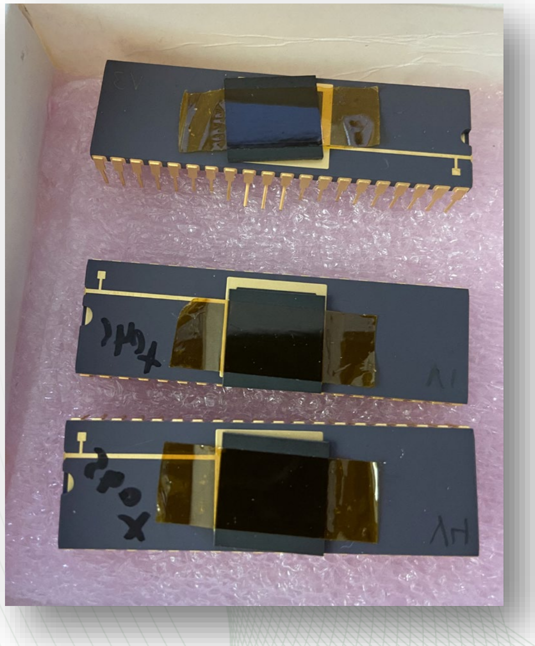
Circuits



$V_1=x_1$
$V_2=x_2$
$V_3=x_3$
$V_4=x_4$
inputs
w_{11} w_{12} w_{13} w_{14}
w_{21} w_{22} w_{23} w_{24}
w_{31} w_{32} w_{33} w_{34}
w_{41} w_{42} w_{43} w_{44}
output

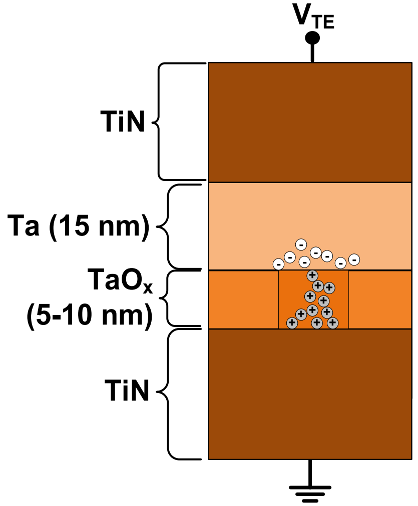


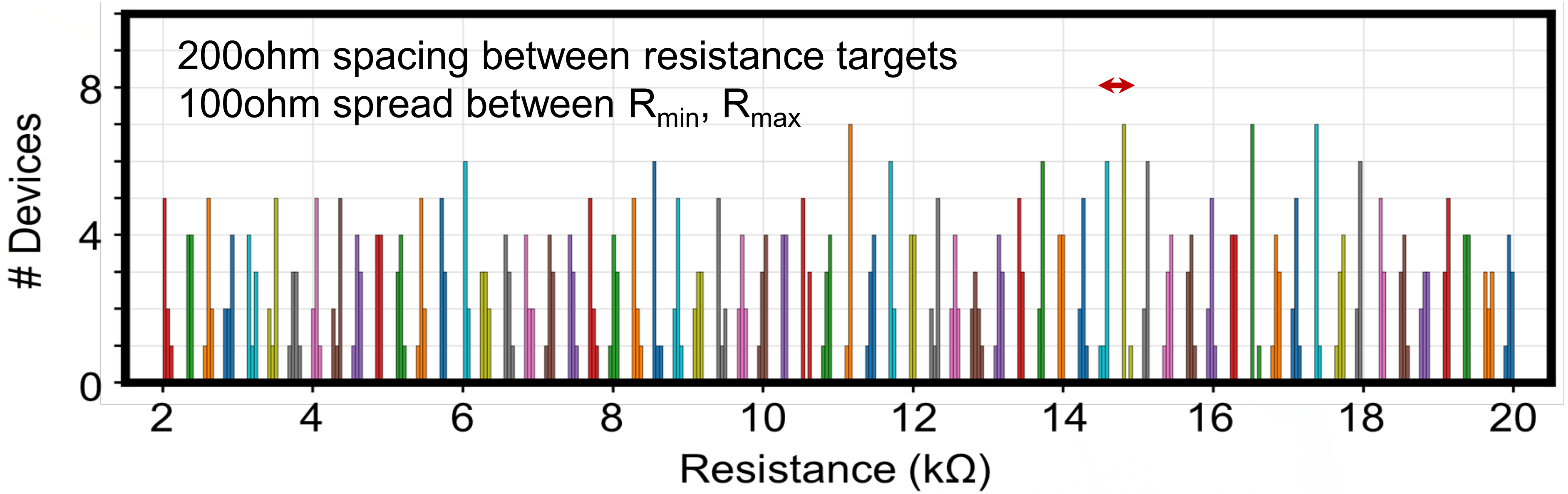
Prototypes

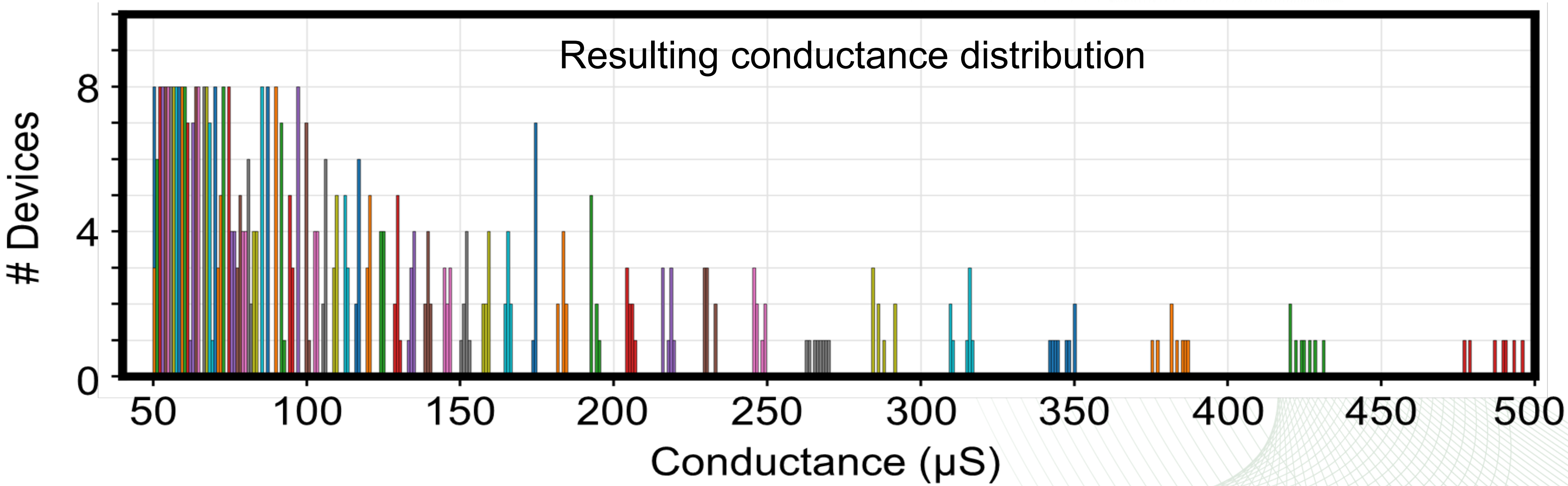


Experimental TaOx ReRAM Conductance Distributions

Developed TaOx weight mapping and programming routine for optimizing inference accuracy







Abisko Microelectronics Codesign Overview

Oak Ridge National Laboratory

Sandia National Laboratories

Arizona State University

Georgia Tech

Harvard University

Collaborator

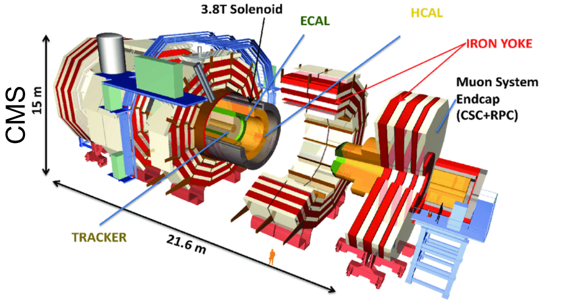
Fermilab

1. Develop better techniques for codesign from algorithms to devices and materials
2. Design Spiking Neural Network chiplet that can be integrated with contemporary computer architectures
3. Explore new devices and materials for the SNN chiplet (neuron, synapse, plasticity, etc.)
4. Design language abstractions and runtime support for SNN chiplet

Motivation

- Transportation
- CMS Sensors

Applications



Motifs, Composition

Algorithms

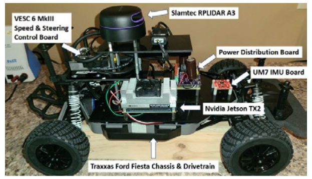
- ML: SLAYER, Whetstone, EONS, eProp, STDP
- Non-ML: Graph algorithms, CSP
- Simulators: NEST, Brian2

Algorithms

nest::

BRIAN

F1Tenth



VESC 6 MkIII Speed & Steering Control Board
Slamtec RPLIDAR A3
Power Distribution Board
UM7 IMU Board
Nvidia Jetson TX2
Traxxas Ford Fiesta Chassis & Drivetrain

API, Motifs

Software

- DSL and API for neuromorphic co-processing
- Built on LLVM and MLIR
- Portable across Abisko chiplet, GPUs, etc.

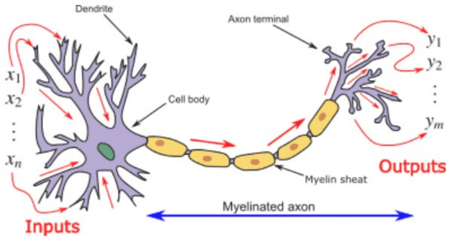
Software

LLVM COMPILER INFRASTRUCTURE

MLIR

XACC

ISA, IR



Source: *Wikipedia*

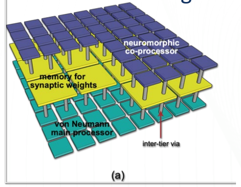
Architecture

- Design neuromorphic chiplet
- RISC-V neuromorphic extensions
- Heterogeneous integration with contemporary technologies

Architecture

RISC-V

2.5D and 3D integration



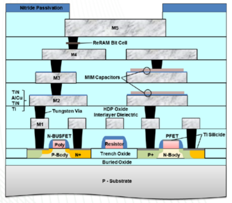
neuromorphic co-processor
memory for synaptic weights
von Neumann main processor
inter-tier via
(a)

Simulation/Emulation

ALADDIN

gem5

MesaFAB ReRAM



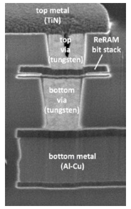
Circuit scale up, Interconnects, PDK

Devices and Circuits

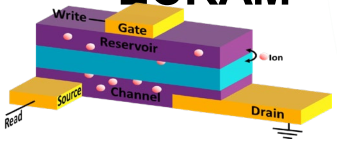
- ion insertion (reversible doping) sets analog states
- mRaman captures transition linear, non-linear switching
- Will extend to 36x36 x-bar array
- Electronic and other optical spectroscopies

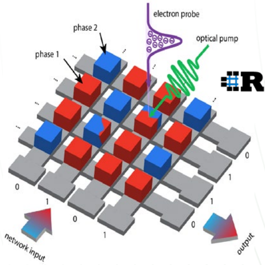
Devices and Circuits

TaOx ReRAM



ECRAM





ROSS SIM

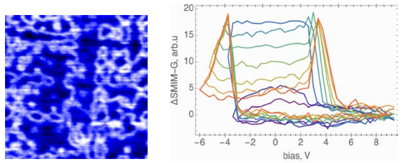
Compact models

Materials

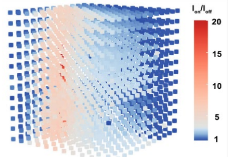
- Non-equilibrium probes to few nm
- Data-driven modeling
- On-demand neuromorphism

Materials

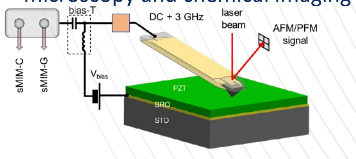
Domain wall memristor



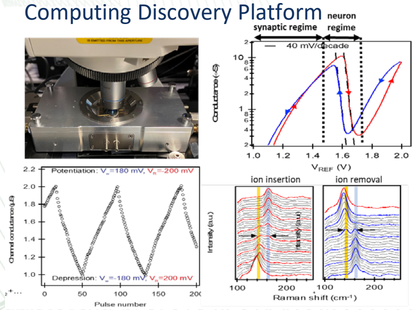
Computational data mining



CNMS scanning probe microscopy and chemical imaging



Computing Discovery Platform



Conclusions

Recap

- Exascale is here!
- Our predictions were reasonably accurate, but we completely missed some
  - AI/ML
  - Programming systems remain major challenge
- Post-exascale
  - Heterogeneous integration and Chiplet architectures are vastly diversifying the architectural landscape
  - Post exascale will be accelerated by recent major semiconductor investments
- Abisko microelectronics codesign project developing a chiplet for analog SNN
- Start building your own chiplets today!

- Visit us (post COVID ☺)
  - We host interns and other visitors year round
    - Faculty, grad, undergrad, high school, industry
- Jobs at ORNL
  - Visit https://jobs.ornl.gov
- Contact me vetter@ornl.gov
- Experimental Computing Lab
  - Lots of emerging archs
  - https://excl.ornl.gov