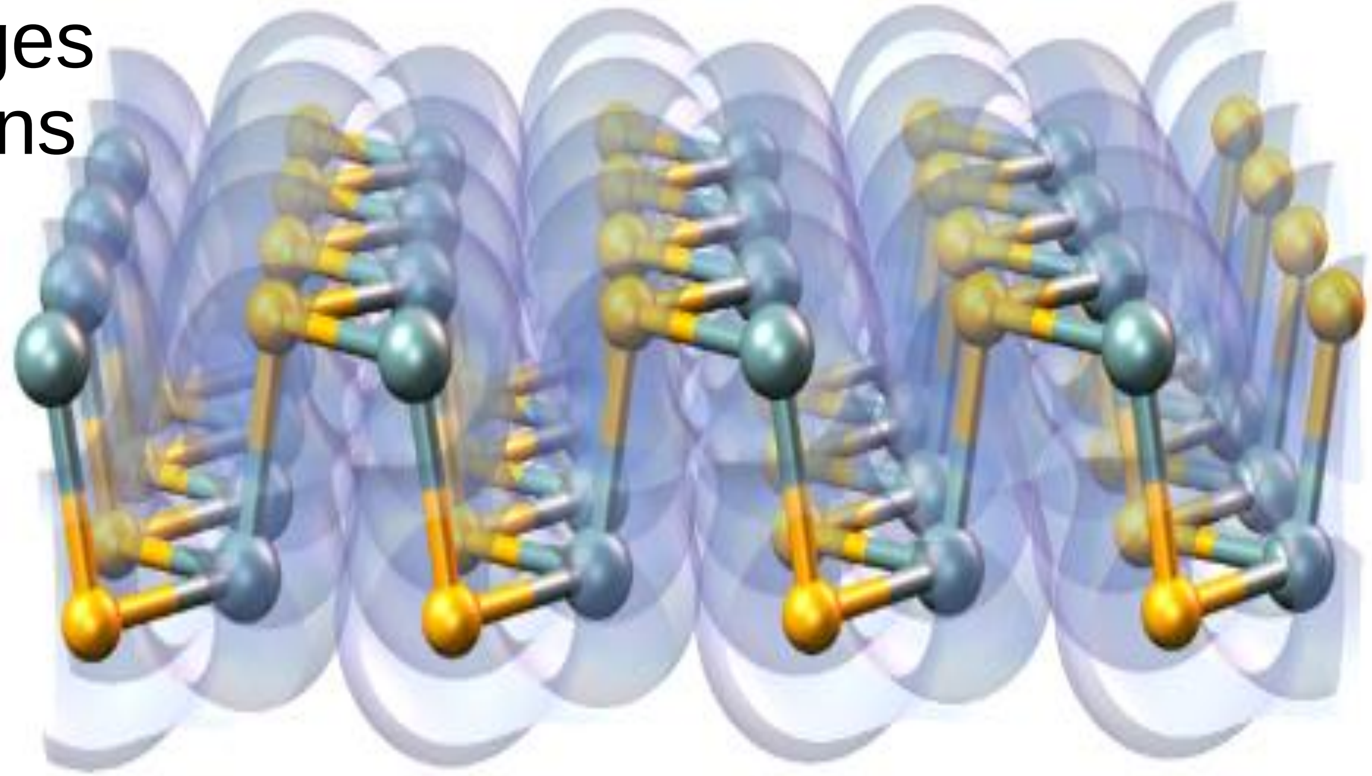
ECP: Perspectives on Challenges in Adapting Complex Applications to Exascale Systems

ASCAC
Sept 29, 2021
Virtual Meeting

Andrew Siegel, AD Director



EXASCALE
COMPUTING
PROJECT

Outline

- Overview of Exascale Computing Project (ECP)
- Current Status of Application Development in ECP
- What is so hard? Unique difficulties and challenges of ECP
- AD Case Studies illustrating unique complexities
  - Metagenomics
  - Data processing at X-ray facilities
  - Small modular nuclear reactors
  - The power grid
- Compilers, Programming models, general libraries, and application specific libraries
- Annual report

Exascale Computing Project (ECP) background

7
Years

$1.8B

- A seven-year, $1.8B R&D effort that launched in 2016

6
Core DOE
Labs

- Argonne
- Lawrence Berkeley
- Lawrence Livermore
- Oak Ridge
- Sandia
- Los Alamos

Application Development: 14 labs, 48 universities, 12 companies

3
Technical
Focus
Areas

- Hardware and Integration
- Software Technology
- **Application Development**

81
R&D Teams
1000
Researchers

- 81 research teams, roughly 10 researchers per team
- Apps projects span 9 DOE program offices + NIH

Four key ingredients of an ECP Application Development Project

SCIENTIFIC OR ENGINEERING GOAL

Challenge Problem

ALGORITHMIC INNOVATION

Not focused
on benchmarks

PORTING TO NEW HARDWARE

Achieving GPU
speedups
often not trivial

INTEGRATION

Encourage collaboration,
use and co-design of
external libraries

ECP Encourages innovation in all aspects of scientific computing

> *“The downside of ... benchmarks is that innovation is chiefly limited to the architecture and compiler. Better data structures, algorithms, programming languages, …cannot be used, since that would give a misleading result. The system could win because of, say, the algorithm, and not because of the hardware or the compiler. While these guidelines are understandable when the foundations of computing are relatively stable, as they were in the 1990s and the first half of this decade, they are undesirable during a programming revolution. For this revolution to succeed, we need to encourage innovation at all levels*.”
>
> -Hennessy and Patterson, Computer Architecture, A Quantitative Approach

Challenge Problem

Not focused on benchmarks

Achieving GPU speedups often not trivial

Encourage collaboration, use and co-design of external libraries

ECP Application Development (AD) Focus Area

National security

Next-generation, **stockpile stewardship** codes

Reentry-vehicle-environment simulation

Multi-ph
simulat
energ
physic

Energy security

Turbine **wind plant** efficiency

Design and commercialization of **SMRs**

combustion

Biofuel catalyst design

Economic security

Additive manufacturing of qualifiable metal parts

Reliable and efficient planning

Scientific discovery

Cosmological probe of the standard model of particle physics

Validate fundamental laws of nature

Demystify **origin of chemical elements**

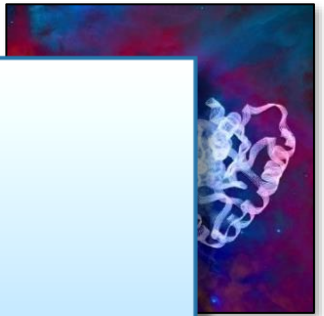
Earth system

Accurate regional impact assessments in **Earth system models**

Stress-resistant crop analysis and catalytic

Health care

Accelerate and translate **cancer research** (partnership with NIH)



AD includes **24 applications** and **6 co-design** projects

- Including **78 separate codes**
- Representing over **10 million lines of code**
- Many supporting large user communities
- Covering broad range of mission critical science and engineering domains
- Mostly started with MPI or MPI+OpenMP on CPUs

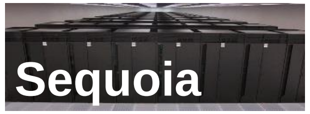
DOE HPC Roadmap to Exascale Systems

FY 2012

ORNL
Cray/AMD/NVIDIA

ANL
IBM BG/Q

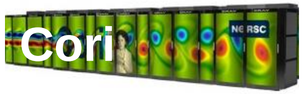
decommissioned



LLNL
IBM BG/Q

FY 2016

ANL
Cray/Intel KNL



LBNL
Cray/Intel Xeon/KNL

LANL/SNL
Cray/Intel Xeon/KNL

FY 2018

ORNL
IBM/NVIDIA

LLNL
IBM/NVIDIA

FY 2021

FRONTIER

ORNL
HPE/AMD

ANL
HPE/AMD/NVIDIA

LBNL
HPE/AMD/NVIDIA

FY 2022

ANL
Intel/HPE

CROSSROADS

LANL/SNL
HPE/Intel

FY 2023

LLNL
HPE/AMD

Exascale Systems

EXASCALE COMPUTING PROJECT

Version 2.0

AD: Where we are now from a porting perspective


Summit


Sierra

CPU ⇨ CPU/GPU ⇨ CPU/Multi-GPU ⇨ Diverse CPU/Multi-GPU

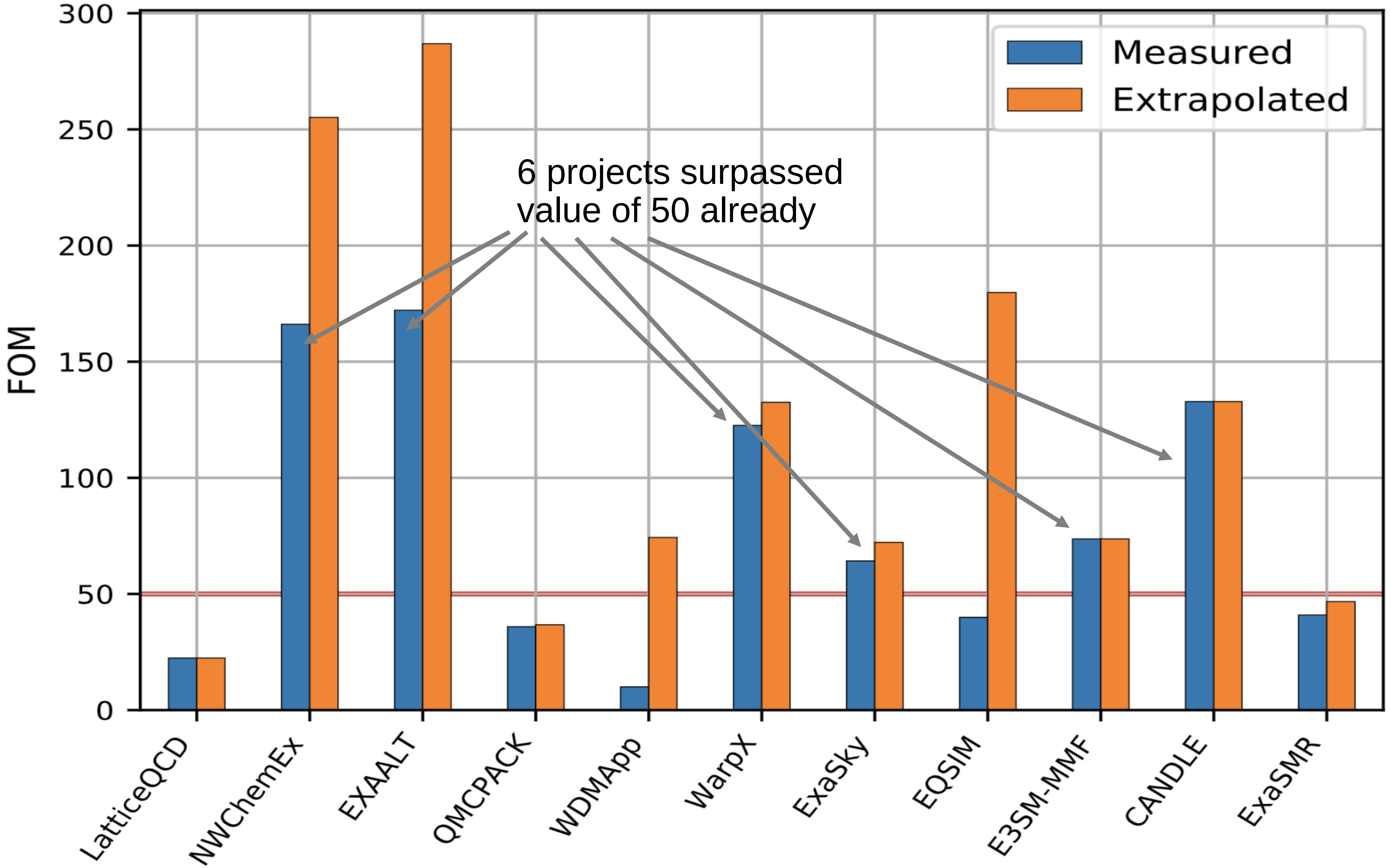
Summit/Sierra:
new AI-focused features

Early hardware is available now
for Intel, AMD, and NVIDIA

⇨ GPU-Resident

Current focus

Current Performance of Key ECP Applications



6 projects surpassed value of 50 already

Why is this so hard? What are the major challenges?

Many interacting moving parts makes ECP a huge challenge

Over 7-year period a lot changes -- new fundamental methodologies are developed, new physical models added, etc.

GPU hardware is general purpose but has preferred computational motifs

---

Programming models/analysis tools, application building blocks take time to mature – broad community buy-in, co-design, expertise not unlimited.

Application-level libraries are critical for most apps and have to evolve fast to be useful

GPUs do best for codes that …

✓ … expose massive fine-grained parallelism required for efficient hardware multithreading

- Summit : 32X80X6X4600 → 73M-way parallelism

✓ … can be made GPU resident – concentrated performance bottlenecks, etc.

✓ … operate in the weak scaling regime – high value of $N_{1/2}$ relative to CPUs

✓ … have high arithmetic intensity

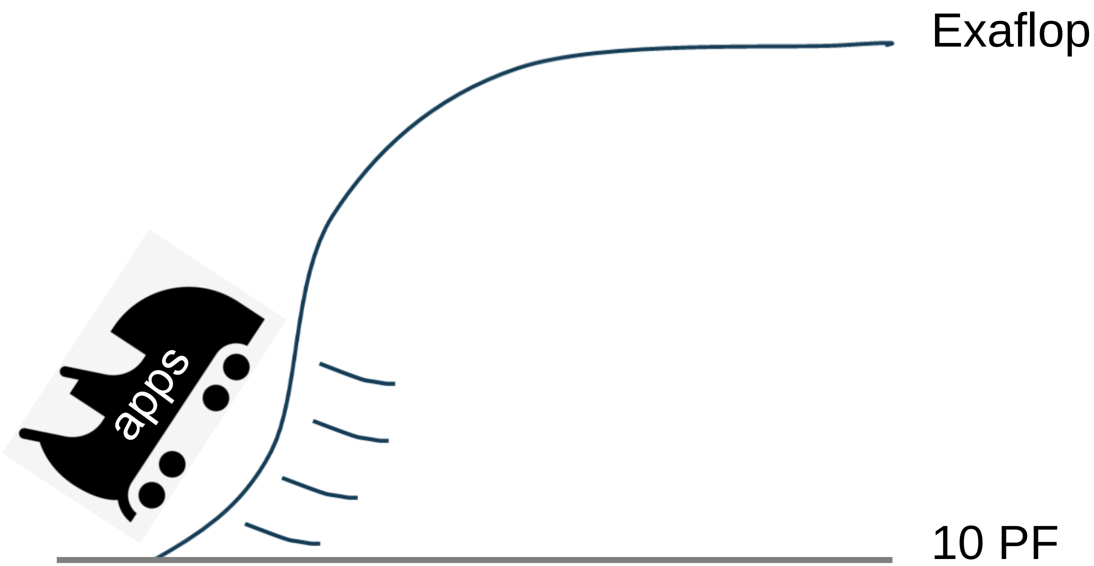
✓ … can be formulated as wide SIMD instructions with minimal branching logic

✓ … require extreme performance with relatively high FLOP to byte (of storage) ratio

✓ … can make use of specialized (tensor core) instructions

What happens when many of these conditions aren’t met?

Case Studies



ExaBiome: Exascale Computational Tools for Metagenomics

Kathy Yelick, UCB/LBL

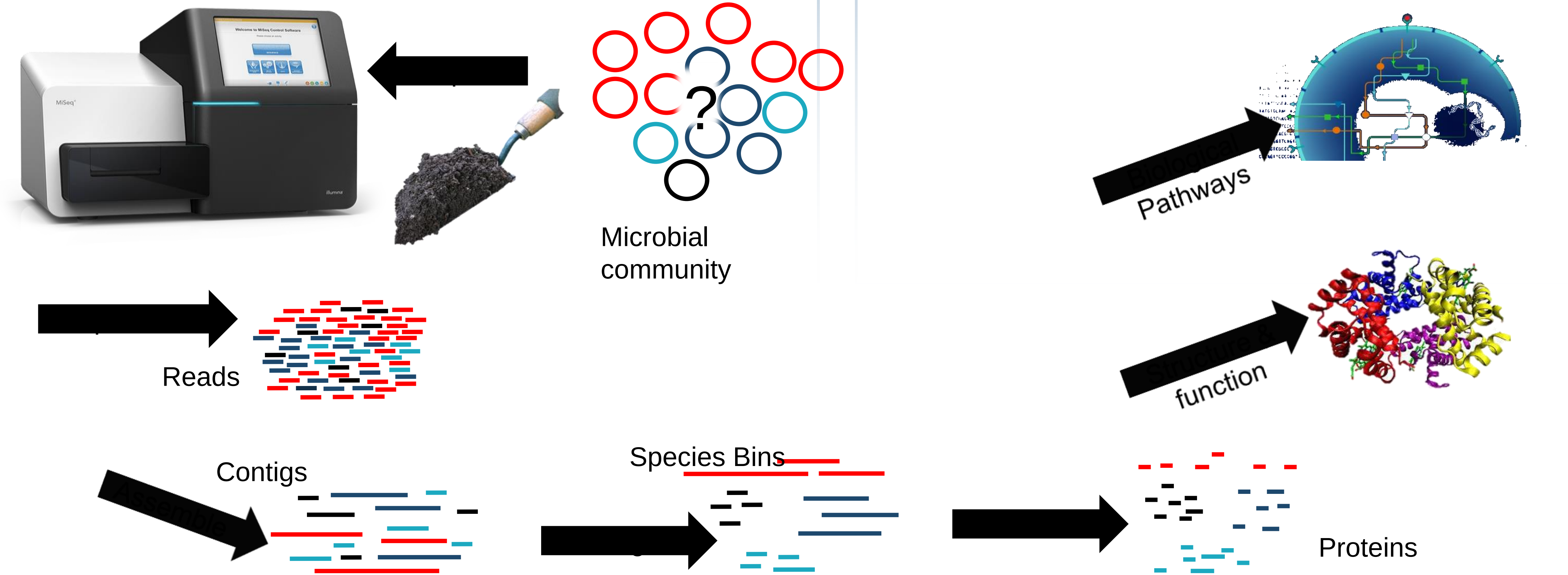
Main agency stakeholder: BER

Science Goal

- Demonstrate a high-quality assembly on at least 50 TB of environmental data (i.e., reads) that effectively use an exascale machine.
- Likely to become production assembler for JGI

Computational challenge

- Methods not robust,
  - intractable, heuristics required, verification difficult.
- Methods evolving at same time as codes/hardware
- Some computational motifs not ideal fit for GPUs



?

Microbial community

Reads

Contigs

Assemble

Species Bins

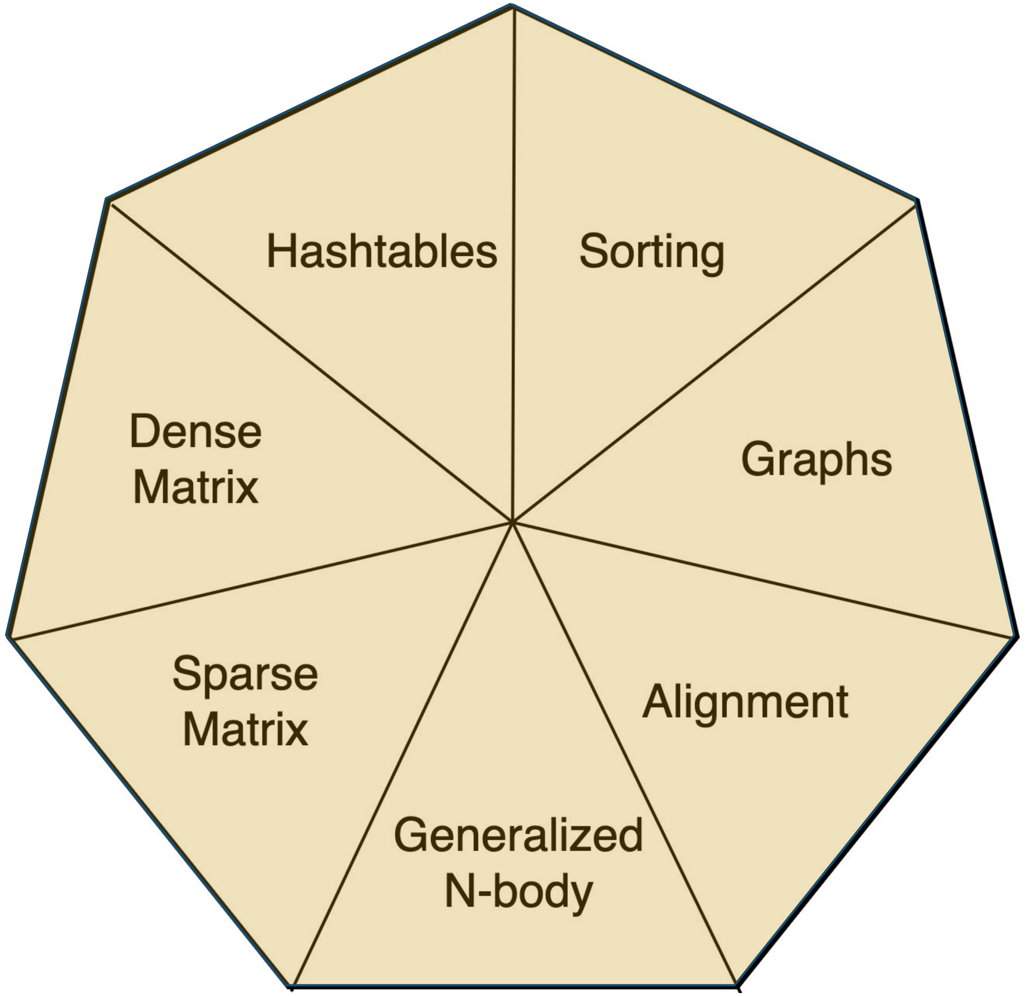
Proteins

Biological Pathways

Structure & function

Motifs of Genomic Data Analysis

These computational patterns dominate ExaBiome Project experience



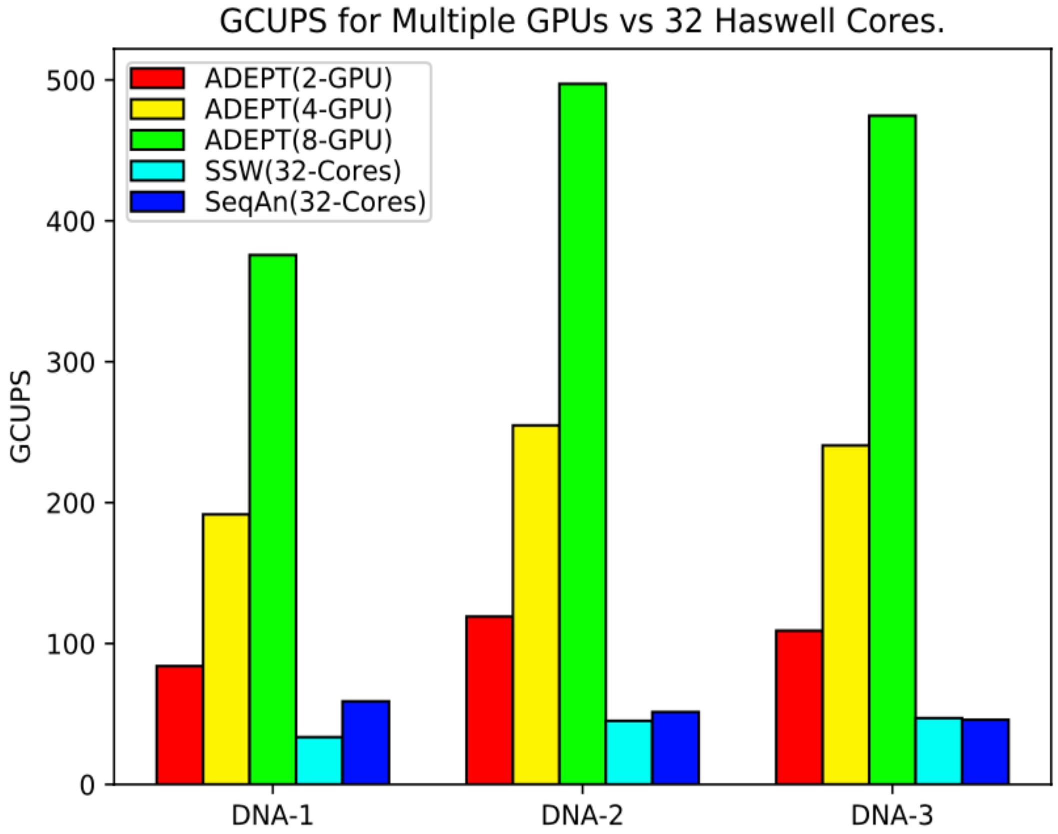
Application problems

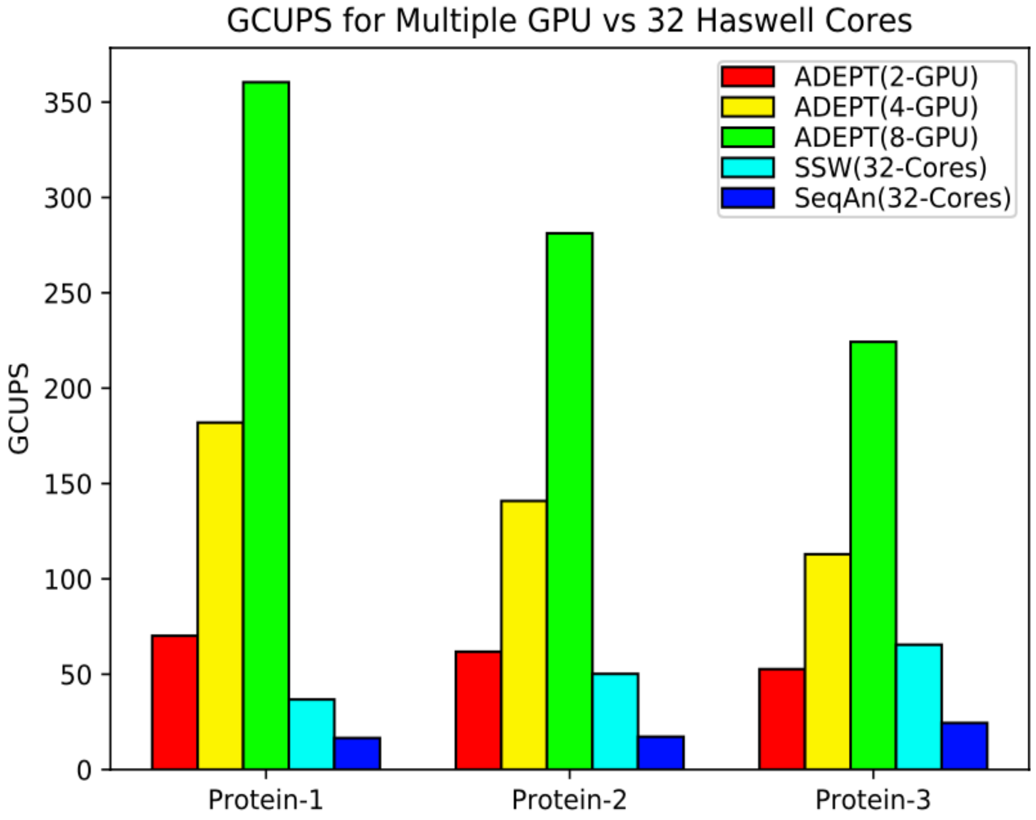
- Assemble genomes
- Compute distances
- Cluster (contigs, proteins,…)
- Annotate

GPUs and distributed memory platforms open up new approaches and science questions

ADEPT: Sequence alignment (Smith-Waterton) on GPUs

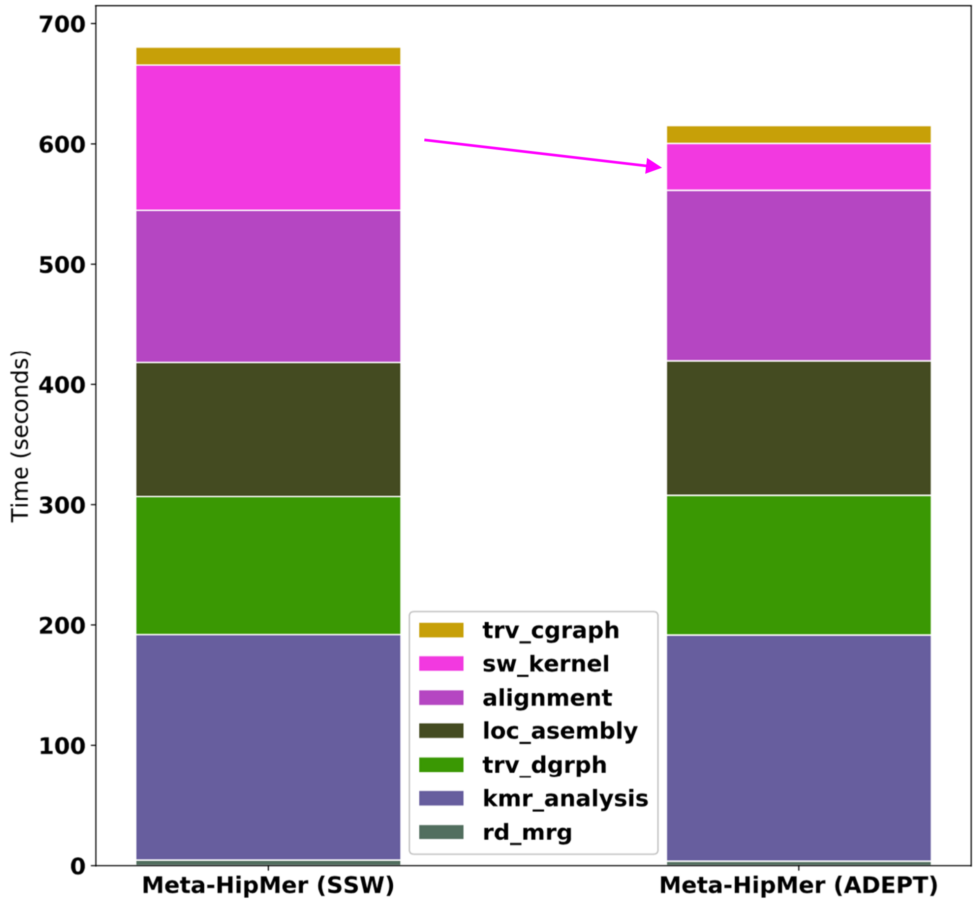
GCUPS = Giga-Cell Updates per second

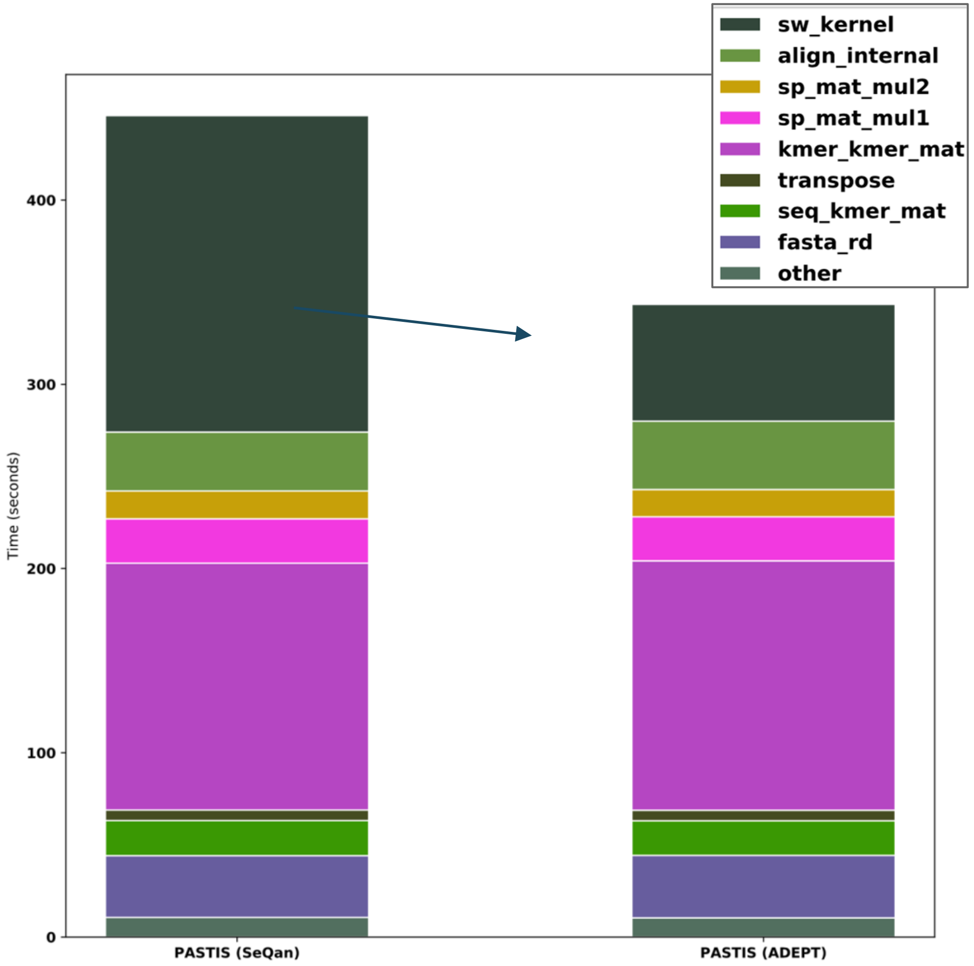




ADEPT: A Domain Independent Sequence Alignment Strategy for GPU Architectures." *MC Bioinformatics* (2020) 21: 406. https://doi.org/10.1186/s12859-020-03720-1

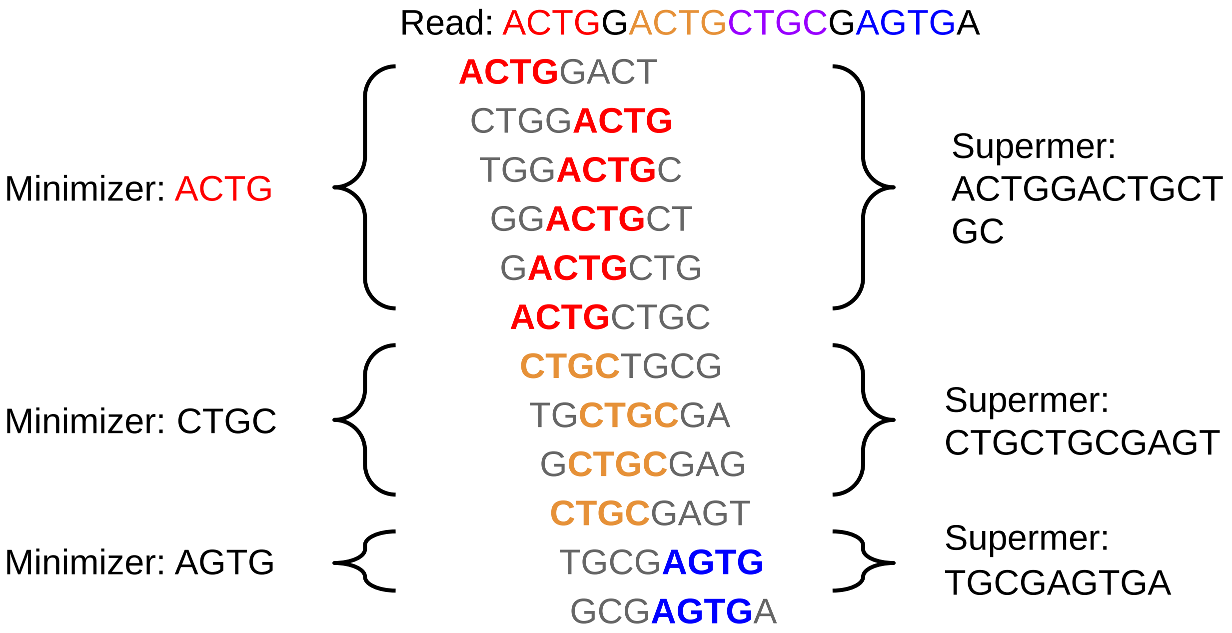
ADEPT: Impact on ExaBiome Applications





Meta-HipMer: De novo assmeber
PASTIS: Protein similarity graph construction pipeline

K-mer Counting: Reducing Communication



Read: ACTGGACTGCTGCGAGTGA

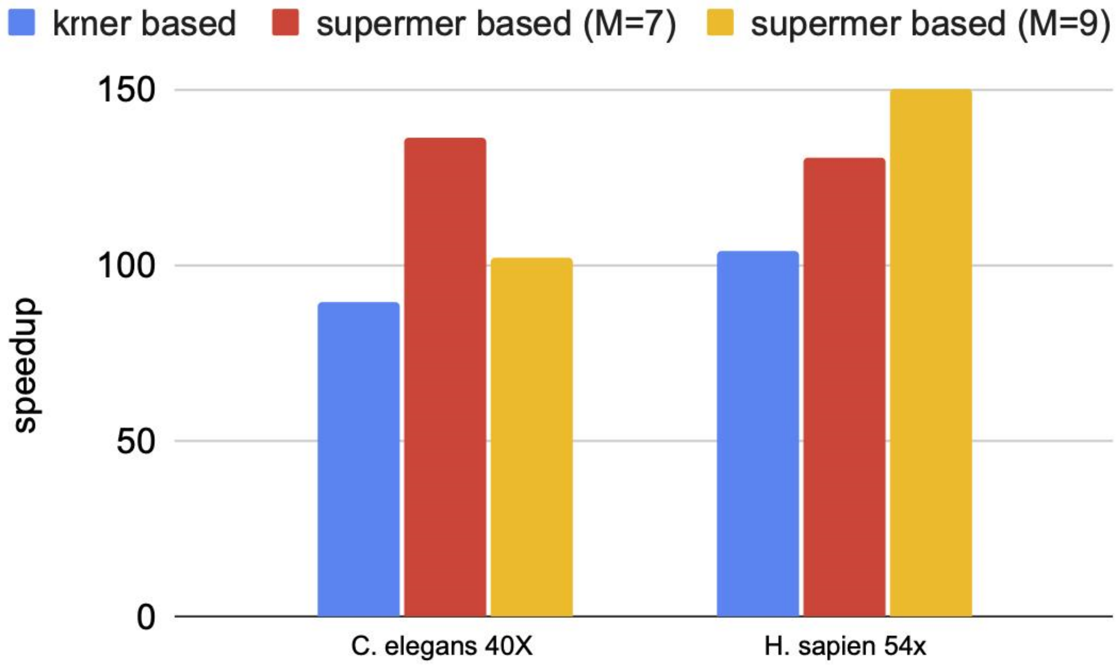
Minimizer: ACTG
ACTGGACT
CTGGACTG
TGGACTGC
GGACTGCT
GACTGCTG
ACTGCTGC
Supermer: ACTGGACTGCTGC

Minimizer: CTGC
CTGCTGCG
TGCTGCGA
GCTGCGAG
CTGCGAGT
Supermer: CTGCTGCGAGT

Minimizer: AGTG
TGCGAGTG
GCGAGTGA
Supermer: TGCGAGTGA

Reduce communication with “Supermers”

- Multiple contiguous k-mer
- map to the same process ID with minimizer-based hashing
- Saves volume (bandwidth) and number of messages (latency)



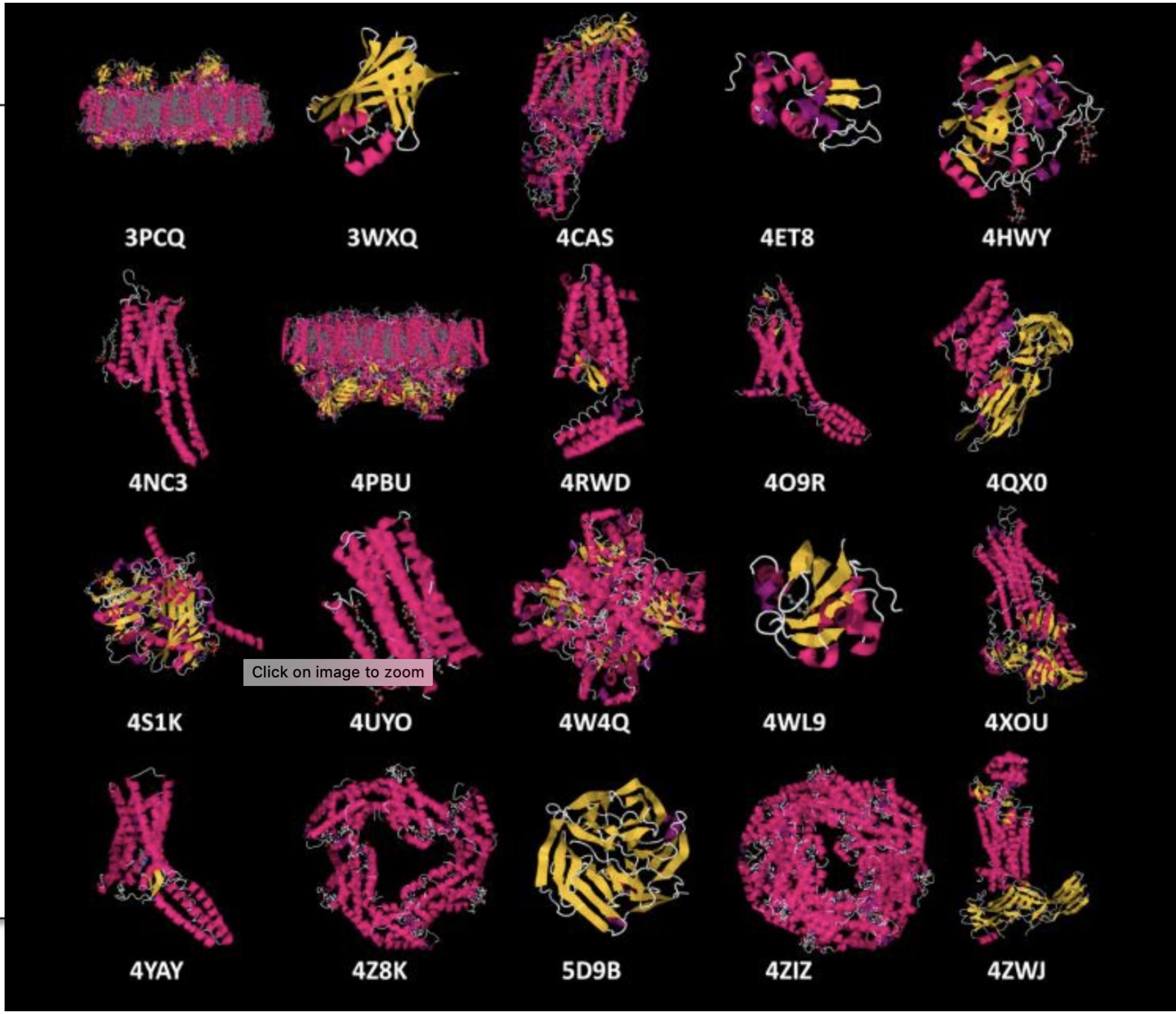
Speedup on 64 Summit nodes

- 6 GPUs / node
- baseline: 42 cores / node

ExaFEL: Real time particle imaging from light sources

Amedeo Perazzo, SLAC

Main agency stakeholder: BES



ExaFEL:
Data Analytics for High Repetition Rate Free Electron Lasers

FEL data challenge:

- **Ultrafast X-ray pulses** from LCLS are used like flashes from a high-speed strobe light, producing stop-action movies of atoms and molecules
- Both **data processing** and **scientific interpretation** demand intensive computational analysis

LCLS-II will increase **data throughput by three orders of magnitude** by 2025, creating an exceptional scientific computing challenge

Project Goals:

- **Serial Femtosecond Crystallography (SFX):** using x-ray tracing in nanocrystallography reconstruction (*challenge problem*)
- **Single Particle Imaging (SPI):** simultaneously determine conformational states, orientations, intensity, and phase from single particle diffraction images
- **Real time end-to-end workflows**: automate the coordination of resources to execute end-to-end workflows from SLAC to NERSC

Science Goal

- Detector data rates at light sources are advancing exponentially
- LCLS will increase its data throughput by three orders of magnitude by 2025.
- Data analysis must be carried out quickly to allow users to iterate their experiments and extract the most value from scarce beam time.
- **The grand challenge: Enabling new photon science from the LCLS will require near real-time analysis (~10 min) of data bursts, requiring burst computational intensities exceeding an exaflop**

Computational challenges

- Complex multi-component workflow, integration of DOE HPC and experimental facilities
- Moving trom SFX to single particle imagining algorithms (M-TIP).
- Non-uniform FFTs on GPUs
- Improving algorithms for SFX: X-ray tracing for pixel-level resolution
- Maximum likelihood estimation non-linear, sparse optimization loop

Example data rate for LCLS-II (early science)

•1 x 4 Mpixel detector @ 5 kHz = 40 GB/s

Example LCLS-II and LCLS-II-HE (mature facility)

•2 planes x 8 Mpixel ePixUHR @ 50 kHz = 1.6 TB/s

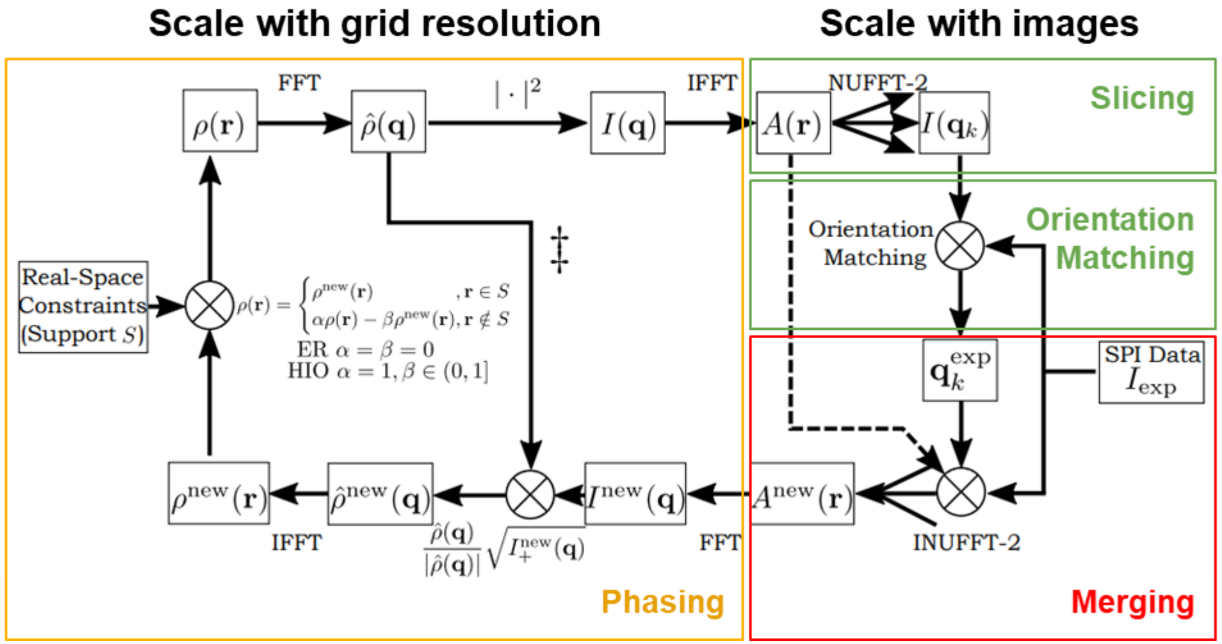
- Collaboration with ESNET to incorporate SENSE software into ExaFEL
- ExaFEL intended as exemplar for all light source facilities !

SPI Acceleration on Summit

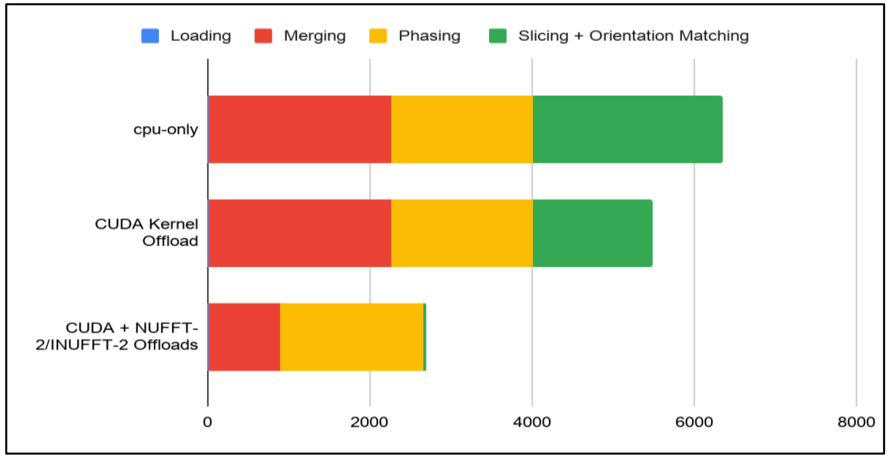
Single-node analysis: 1,500 images

- 1 CPU vs 1 GPU
- spinifel proxy-app

Optimization level	Wall Time (s)	Speed Up
CPU only	6345	-
CUDA kernels offload	5495	13%
CUDA kernel + NUFFT-2/INUFFT-2 offloads	2697	57%



Time (s) spent in different modules



ExaSGD: Exascale Computational Tools for the Power Grid

Slaven Peles, PNNL

Main agency stakeholder: OE

Engineering Goal

- Enable the timely analysis of national-scale grid models with large numbers of contingency constraints that reflect realistic failure scenarios.
- Enable regional and national stakeholders to assess the reliability of electric energy production in the context of uncertain power generation, severe weather disruptions and cyber attacks.
- Enable power grid operators to to small-scale analyses that effectively leverage CPU+GPU computing hardware to accelerate their calculations. This will enable power grid operators to quickly adapt and respond with much more realistic grid models.

Negotiated underfrequency load-shedding with OE as major application driver → real time control.

Computational challenges

- Massive non-linear optimization
- Large, sparse indefinite linear systems
- Compressed dense systems

Modified optimization algorithm uses compression to generate dense linear systems : ideal for GPUs

$$R_1(x) = \min_{y_1} \; f_1(x, y_1) \quad \text{s.t.} \; g_1(x, y_1) = b_1, \; y_1 \geq 0.$$

$$\begin{bmatrix} i_\alpha \\ 0 \end{bmatrix} = \begin{bmatrix} Y^{\text{bus}}_{\alpha,\alpha} & Y^{\text{bus}}_{\alpha,\beta} \\ Y^{\text{bus}}_{\beta,\alpha} & Y^{\text{bus}}_{\beta,\beta} \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}$$

1. **CPU** preprocessing

HiOp Kron Reduction

$$i_\alpha = \underbrace{\left(Y^{\text{bus}}_{\alpha,\alpha} - Y^{\text{bus}}_{\alpha,\beta}\left(Y^{\text{bus}}_{\beta,\beta} \setminus Y^{\text{bus}}_{\beta,\alpha}\right)\right)}_{Y^{\text{red}}} \; v_\alpha$$

CPU implementation

2. Optimization loop on **GPU** via HiOp-MDS

HiOp new mixed dense-sparse (MDS) linear algebra

$$\begin{bmatrix} H^s & 0 & (J^s)^T \\ 0 & H^d & (J^d)^T \\ J^s & J^d & 0 \end{bmatrix} \begin{bmatrix} \Delta x^s \\ \Delta x^d \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

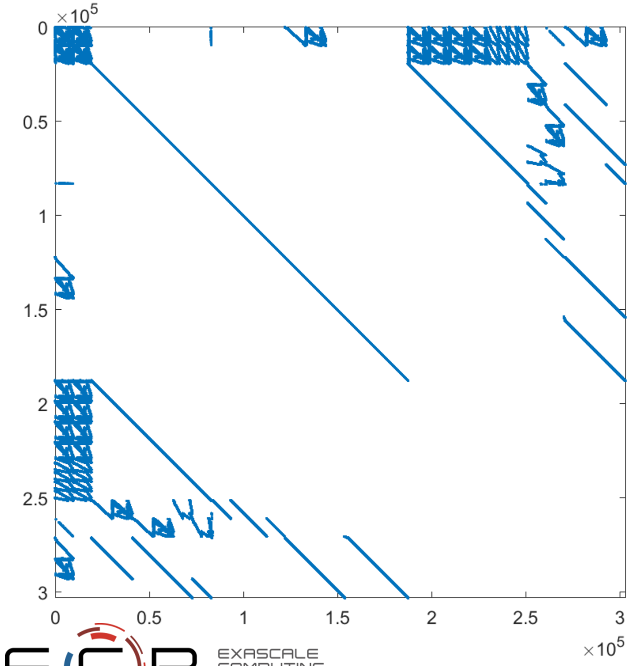
HiOp Schur complement reduction

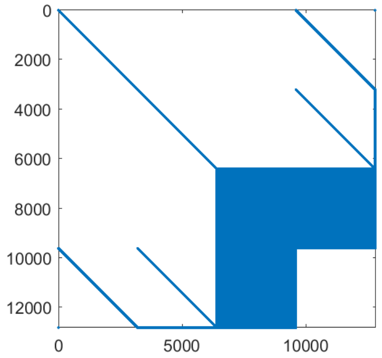
$$\begin{bmatrix} H^d & (J^d)^T \\ J^d & -J^s(H^s)^{-1}(J^s)^T \end{bmatrix} \begin{bmatrix} \Delta x^d \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} \tilde{r}_x \\ \tilde{r}_\lambda \end{bmatrix}$$

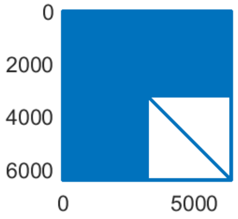
MAGMA/SLATE GPU solver

CPU sequential implementation
RAJA/Umpire portable implementation

18-72% of peak GPU perf. depending on how much stability is needed

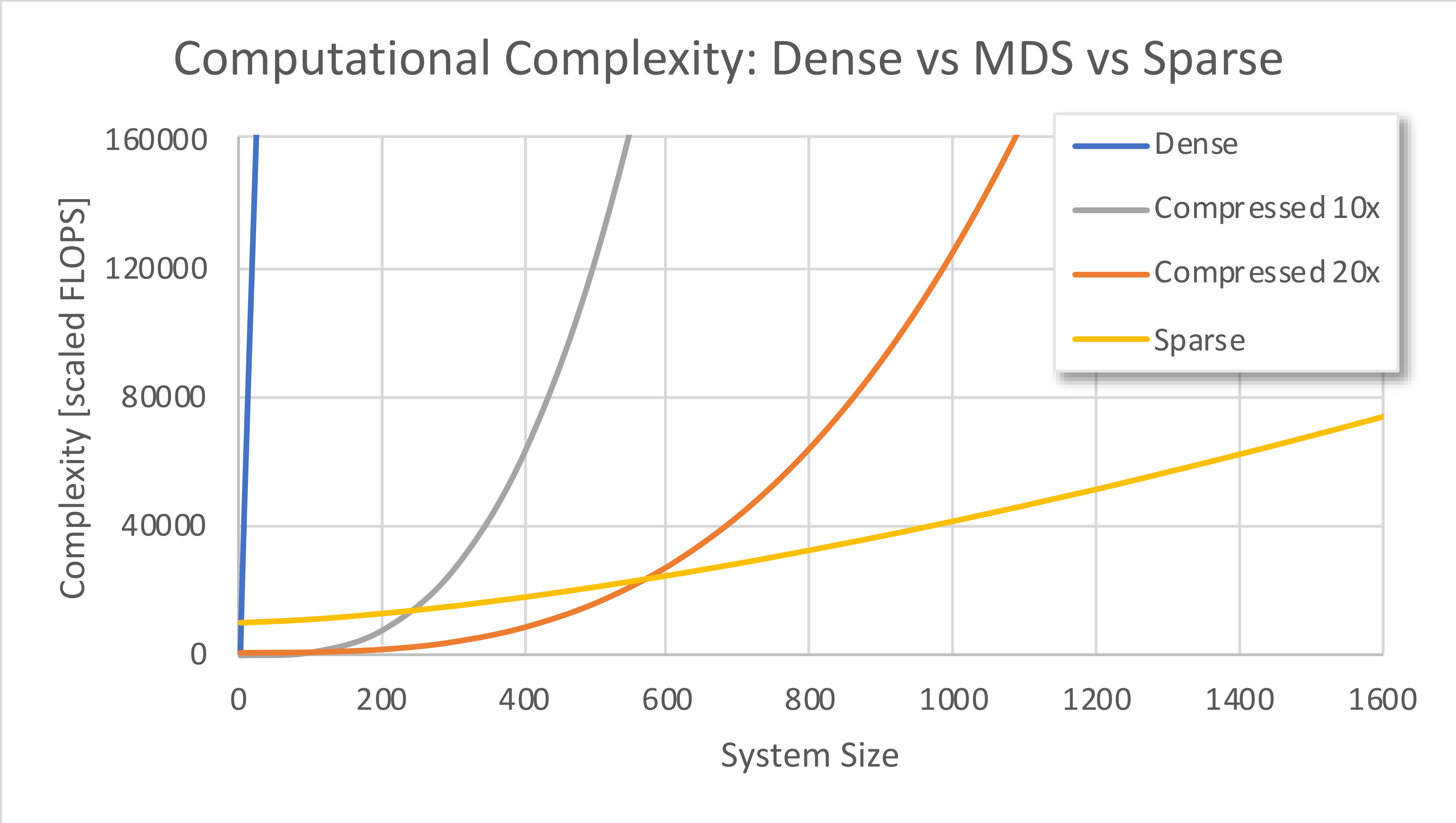






HiOp's modified optimization algorithm leads to dense system more suitable for solving on GPU. Available in release 0.3: https://github.com/LLNL/hiop

Compressed formulation still too expensive for largest problems



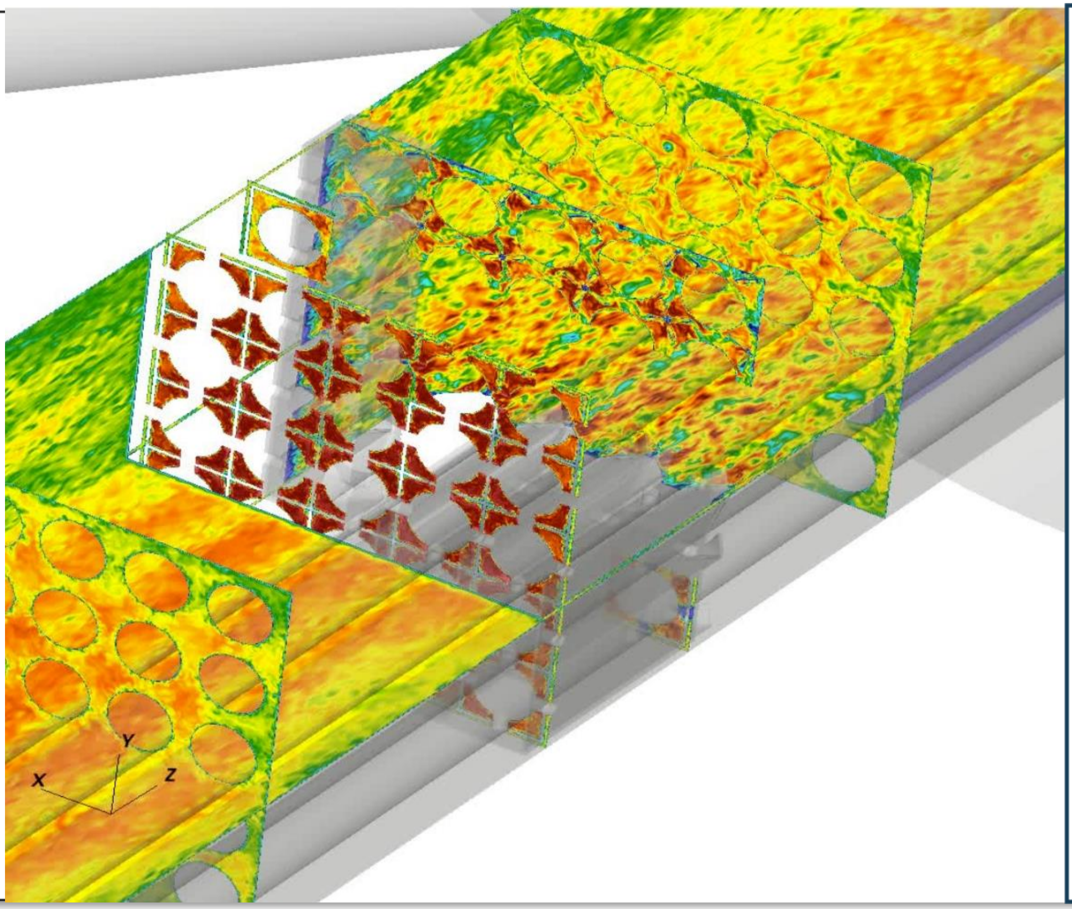
Research into sparse, indefinite GPU solvers needed

Test case	**Size**	**NNZ**	**MA57** reference CPU only	**SuperLU (ECP – LBNL)**		**STRUMPACK (ECP – LBNL)**		**KLU + cuSolve (NVIDIA)**		**SSIDS (STFC, UK Gov.)**	
				CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
73-bus	4,766	23,762	0.01s	0.08s	0.65s	0.06s	0.82s	0.01s	0.01s*	0.14s	2.03s
10k-bus	238,072	1,111,991	0.54s	4.06s	4.95s	2.82s	3.71s	0.81s	0.25s*	2.40s	4.76s
70k-bus	1,640,411	7,671,693	5.30s	30.46s	35.58s	24.4s	26.8s	13.26s	3.26s*	32.25s	197.66s

ExaSMR: Exascale Computational Tools for Nuclear Reactor Design

Steve Hamilton, ORNL

Main agency stakeholders:
NE, FES, NNL

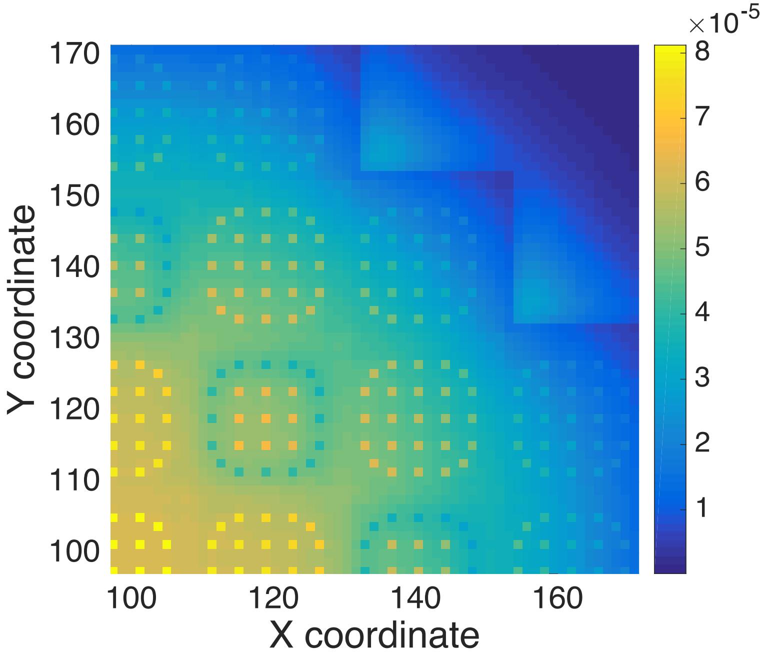


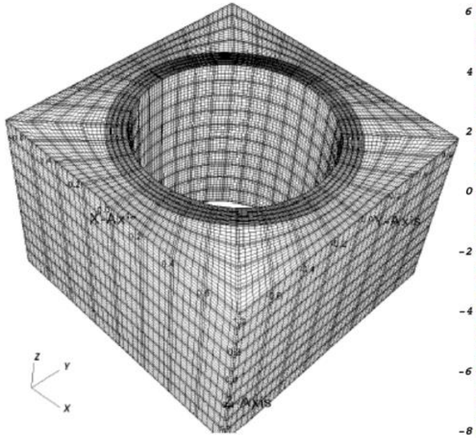
Engineering Goal

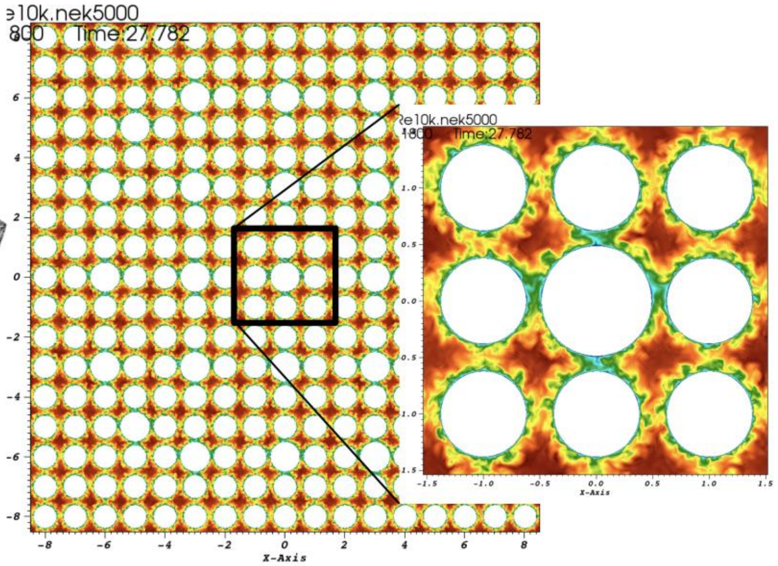
- Simulation of full NuScale SMR model core by coupling continuous-energy Monte Carlo neutronics with CFD
  - Complete in-vessel coolant loop (natural circulation flow)
  - Hybrid LES/RANS turbulence model
  - Sub-pin resolution fission power
  - Isotopic depletion (stretch goal)

Computational challenges

- Monte Carlo methods on GPUs
- Coupled Monte Carlo (transport), deterministic (CFD)
- Strong-scaling CFD

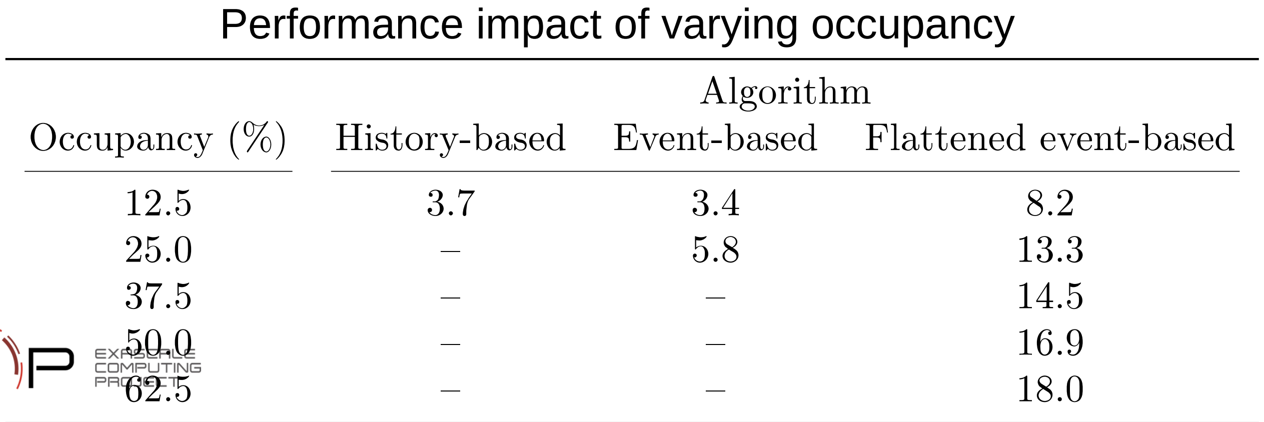






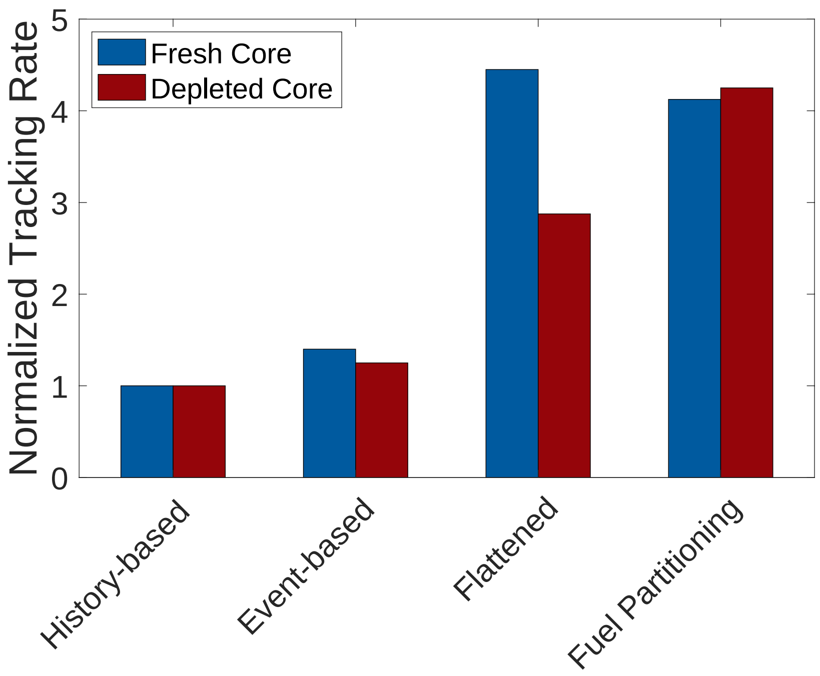
Shift on GPUs

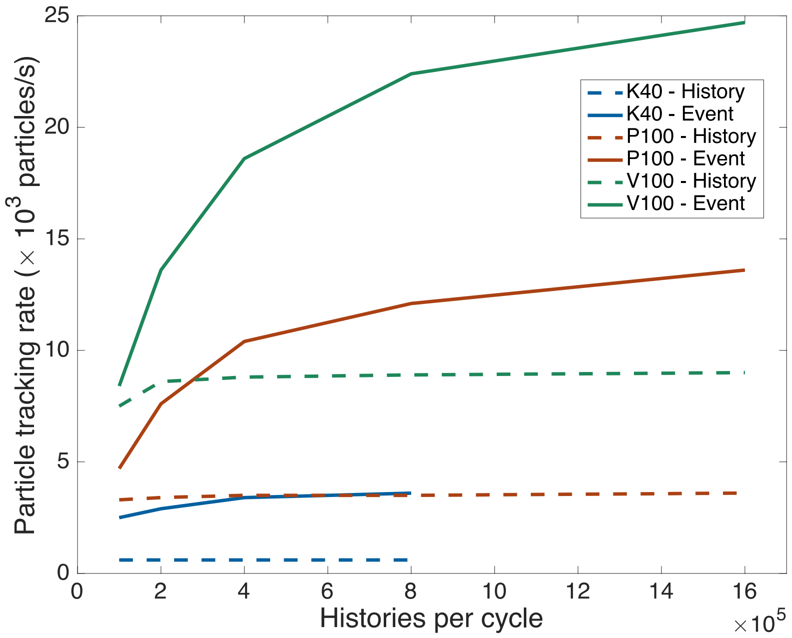
- Shift ported to Nvidia GPUs using CUDA
- Initial Shift GPU implementation used *history-based* algorithm
  - “Fat kernel” approach (>10k LOC in single kernel)
  - Massive thread divergence
  - Low occupancy
- Optimized implementation uses an *event-based* approach
  - Particles requiring similar processing collected together
  - Smaller, targeted kernels
  - Occupancy increase from 12.5% to 62.5%
  - Requires many particles in flight for ideal performance



Performance impact of varying occupancy

	Algorithm		
Occupancy (%)	History-based	Event-based	Flattened event-based
12.5	3.7	3.4	8.2
25.0	–	5.8	13.3
37.5	–	–	14.5
50.0	–	–	16.9
62.5	–	–	18.0



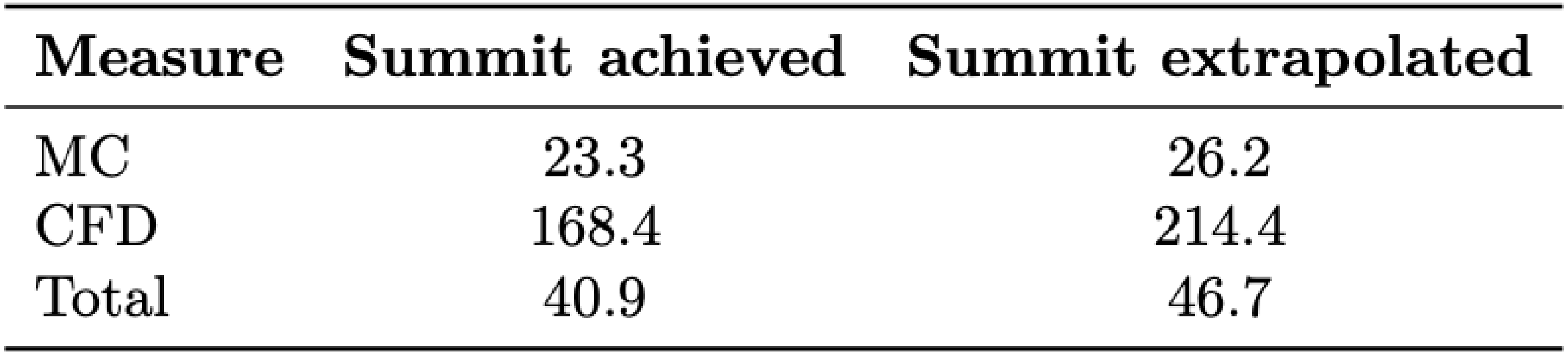


Performance Figure of Merit

- Overall FOM is harmonic average of individual physics components:

$$\mathrm{W}_{\mathrm{MC}} = \frac{\text{particles}}{\text{wall clock seconds}} \qquad \mathrm{W}_{\mathrm{CFD}} = \frac{\text{degrees of freedom}}{\text{wall clock seconds per time step}}$$

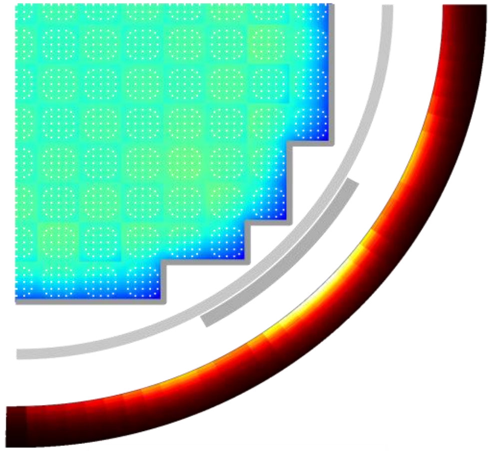
ExaSMR Figure of Merit progress to date



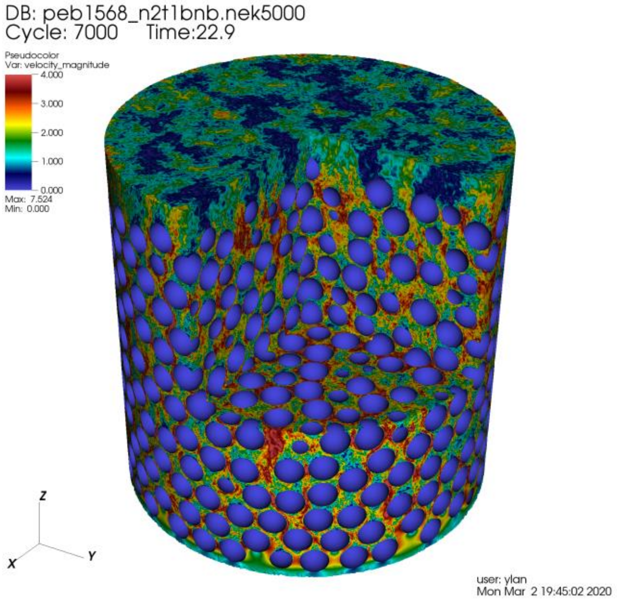
Measure	**Summit achieved**	**Summit extrapolated**
MC	23.3	26.2
CFD	168.4	214.4
Total	40.9	46.7

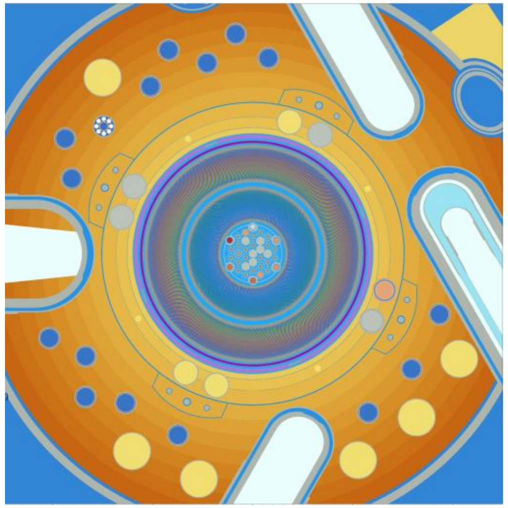
Goal is to achieve 50x performance improvement on Frontier or Aurora

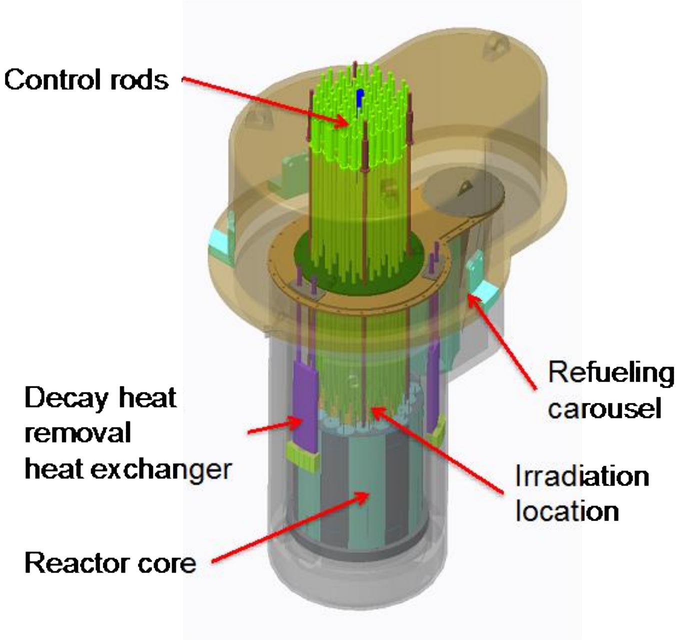
Applications beyond SMRs

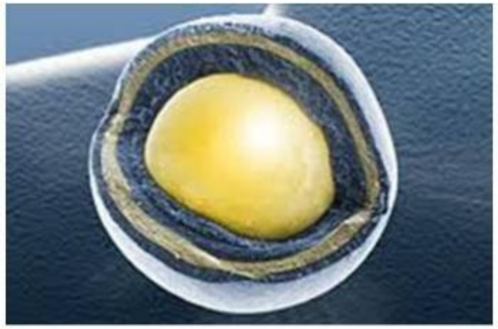


- Advanced reactors – pebble beds, molten salt
- Micro-reactors
- Ex-core vessel fluence and dosimetry
- Radiation shielding

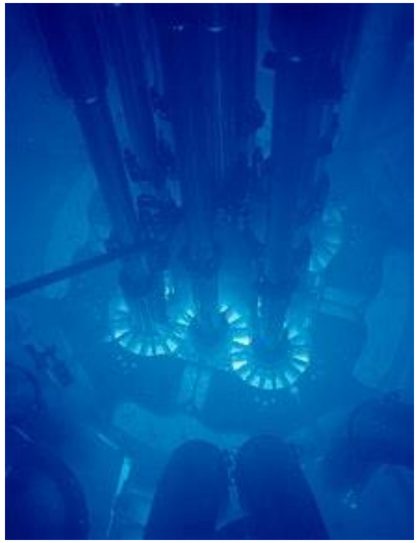








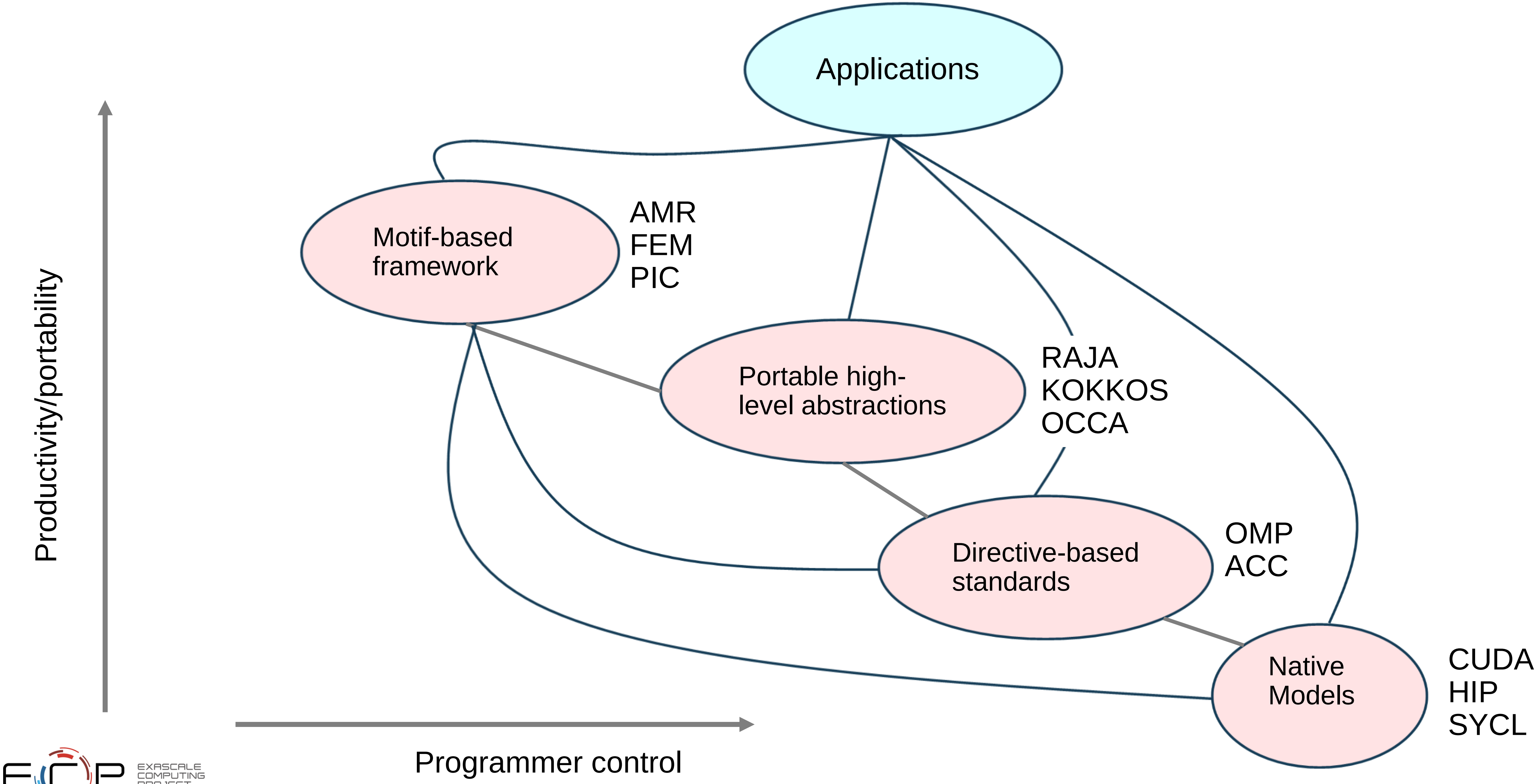
TRISO coated particle fuel



ExaSMR catalyzed a new joint ASCR/HEP project in charged particle tracking: CELERITAS

Compilers, Programming models, general libraries, and application specific libraries

Programming model choice balances risk/control with productivity



Applications

Motif-based framework

AMR
FEM
PIC

Portable high-level abstractions

RAJA
KOKKOS
OCCA

Directive-based standards

OMP
ACC

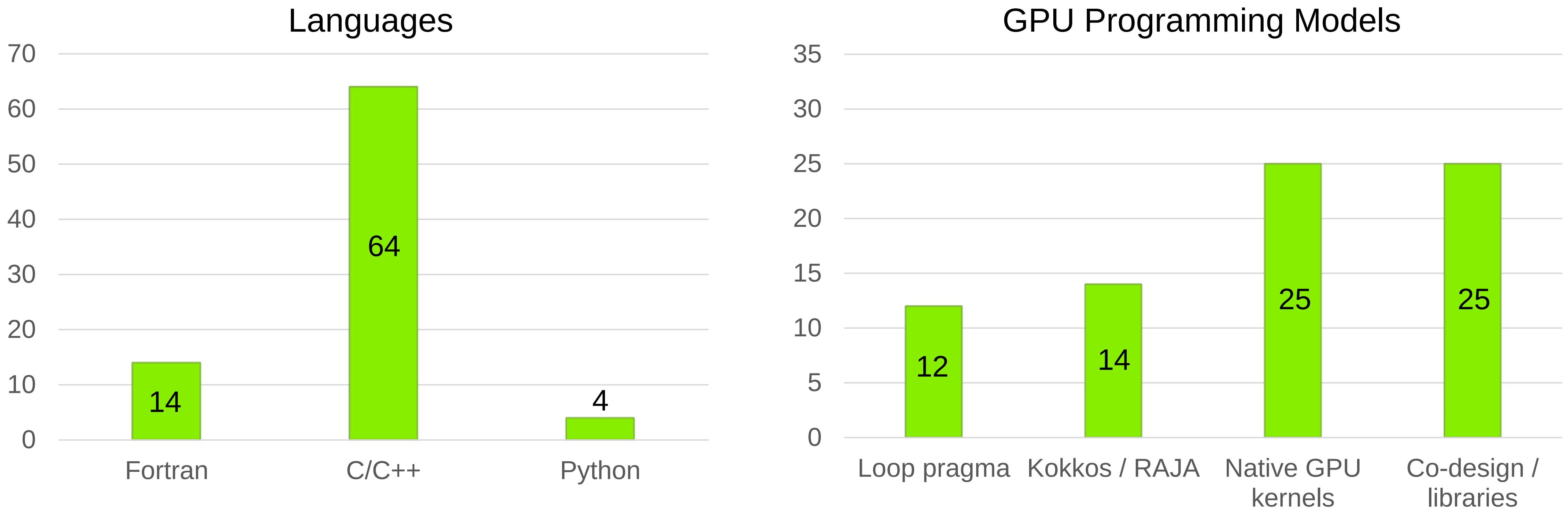
Native Models

CUDA
HIP
SYCL

Productivity/portability

Programmer control

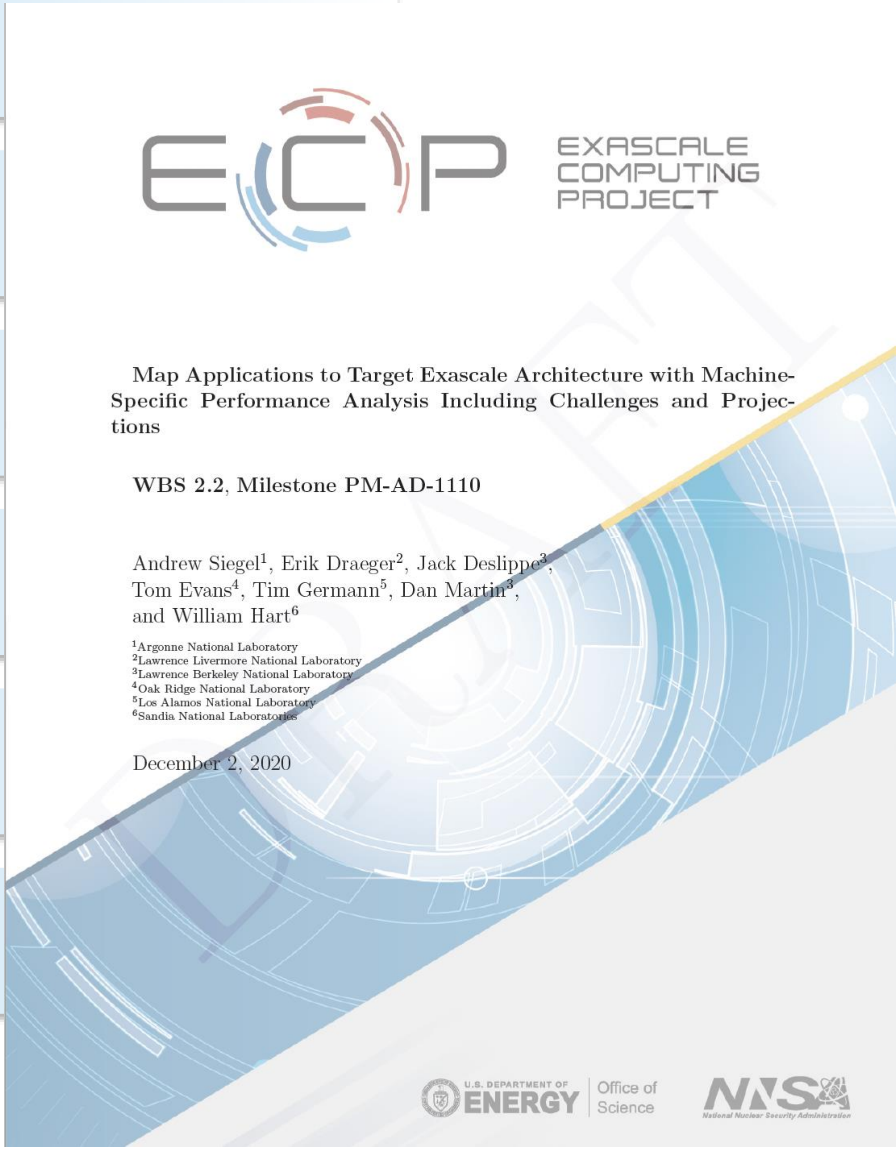
Distribution of ECP Programming Models



There has been significant movement in programming models and languages since the beginning of the project, mostly toward C++ and abstraction layers/libraries. However, we need all codes to run well!

Interesting Themes Emerging from 2020 Report

- ✓ **Use of mixed precision**
- ✓ **Strong Scaling**
- ✓ **Optimized libraries on early access machines**
- ✓ **Performance of OpenMP offload**
- ✓ **GPU Resident + Unified Virtual Memory**
- ✓ **Relative increased cost of inter-node comm.**



Map Applications to Target Exascale Architecture with Machine-Specific Performance Analysis Including Challenges and Projections

WBS 2.2, Milestone PM-AD-1110

Andrew Siegel[1], Erik Draeger[2], Jack Deslippe[3], Tom Evans[4], Tim Germann[5], Dan Martin[3], and William Hart[6]

[1]Argonne National Laboratory
[2]Lawrence Livermore National Laboratory
[3]Lawrence Berkeley National Laboratory
[4]Oak Ridge National Laboratory
[5]Los Alamos National Laboratory
[6]Sandia National Laboratories

December 2, 2020

Final Thoughts

- Very hard to push the frontiers alone
  - Broad collaborations of diverse teams
  - Adoption of libraries, enabling tools
  - Co-design of application-level libraries

- Hardware and programming models drive methods, models, algorithms as much as the reverse.

- ECP not just about porting and benchmarks – innovation at all levels with subtle interplay among them is key to progress.

- Don’t underestimate how long it takes software to mature on new systems.