

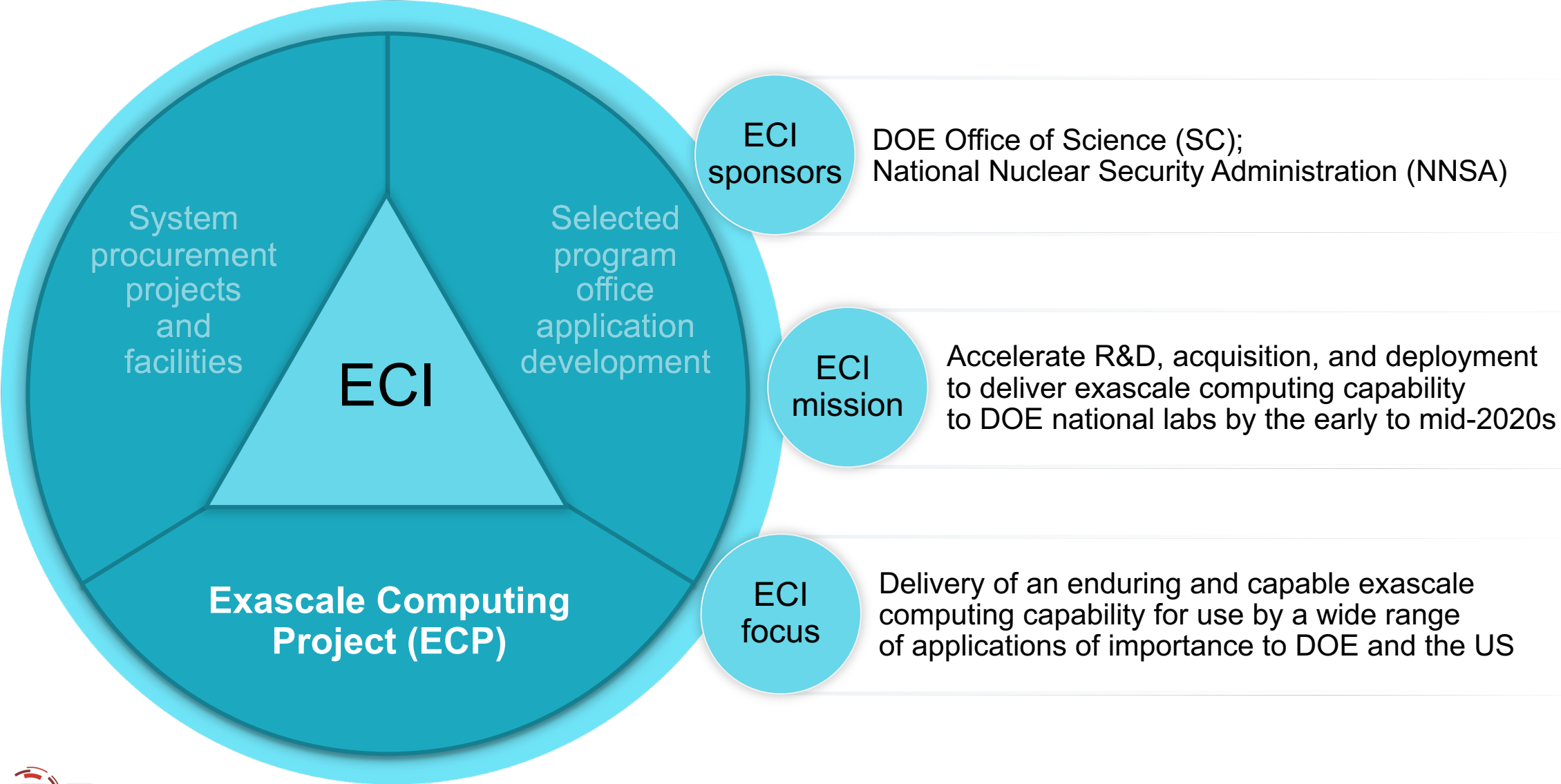
Exascale Computing Project Update



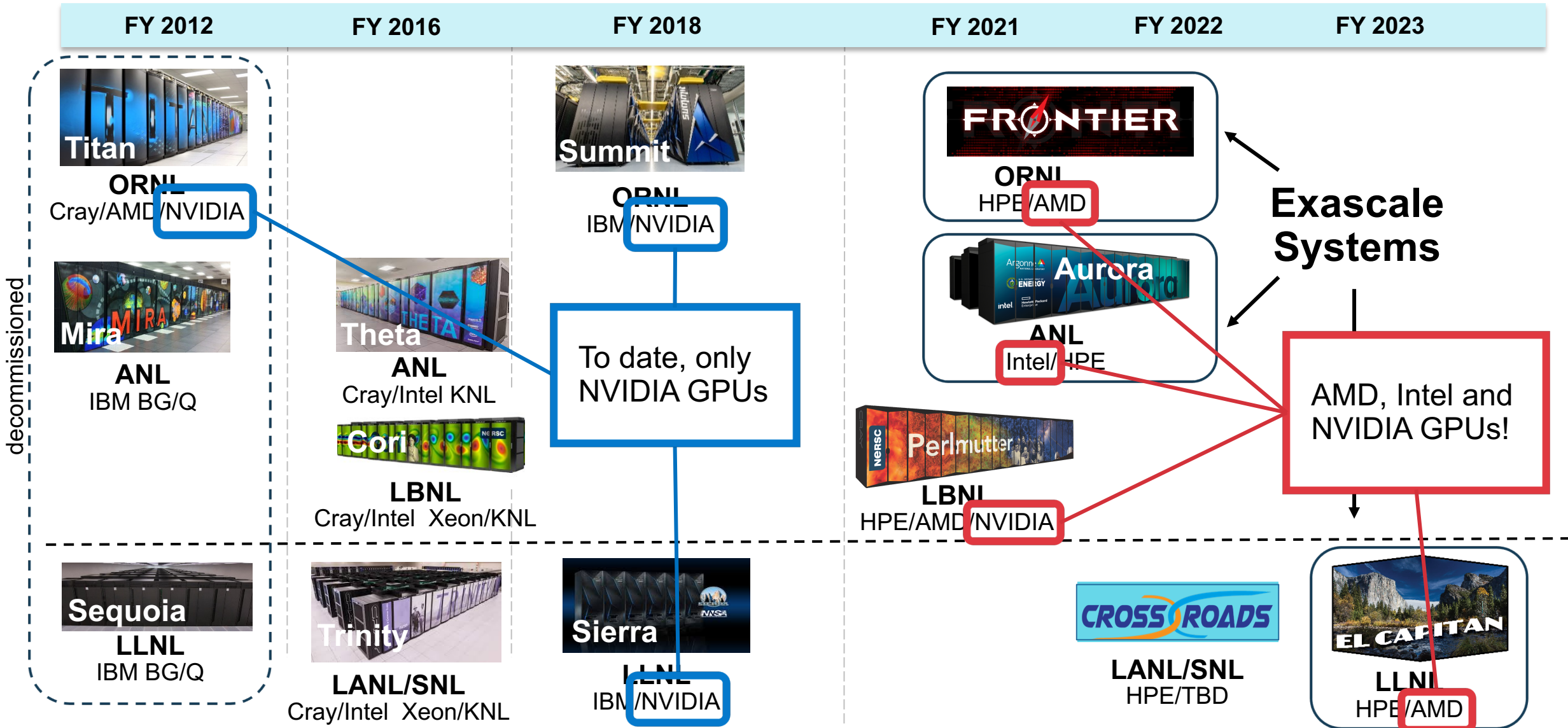
Douglas B. Kothe
Oak Ridge National Laboratory (ORNL)
Director, ECP

HPC User Forum
May 11, 2021
Video Conference

DOE Exascale Computing Initiative (ECI)



DOE HPC Roadmap to Exascale Systems



Specs of the Aurora & Frontier Exascale Systems are Here

Details of these impressive systems are becoming firm and real

System attributes	ALCF Now	NERSC Now	OLCF Now	NERSC Pre-Exascale	ALCF Pre-Exascale	OLCF Exascale	ALCF Exascale
Name (Planned) Installation	Theta 2016	Cori 2016	Summit 2017-2018	Perlmutter (2020-2021)	Polaris (2021)	Frontier (2021-2022)	Aurora (2022-2023)
System peak	> 15.6 PF	> 30 PF	200 PF	> 120PF	35 – 45PF	>1.5 EF	≥ 1 EF DP sustained
Peak Power (MW)	< 2.1	< 3.7	10	6	< 2	29	≤ 60
Total system memory	847 TB DDR4 + 70 TB HBM + 7.5 TB GPU memory	~1 PB DDR4 + High Bandwidth Memory (HBM) + 1.5PB persistent memory	2.4 PB DDR4 + 0.4 PB HBM + 7.4 PB persistent memory	1.92 PB DDR4 + 240TB HBM	> 250 TB	4.6 PB DDR4 +4.6 PB HBM2e + 36 PB persistent memory	> 10 PB
Node performance (TF)	2.7 TF (KNL node) and 166.4 TF (GPU node)	> 3	43	> 70 (GPU) > 4 (CPU)	> 70 TF	TBD	> 130
Node processors	Intel Xeon Phi 7320 64-core CPUs (KNL) and GPU nodes with 8 NVIDIA A100 GPUs coupled with 2 AMD EPYC 64-core CPUs	Intel Knights Landing many core CPUs Intel Haswell CPU in data partition	2 IBM Power9 CPUs + 6 Nvidia Volta GPUs	CPU only nodes: AMD EPYC Milan CPUs; CPU-GPU nodes: AMD EPYC Milan with NVIDIA A100 GPUs	1 CPU; 4 GPUs	1 HPC and AI optimized AMD EPYC CPU and 4 AMD Radeon Instinct GPUs	2 Intel Xeon Sapphire Rapids and 6 Xe Ponte Vecchio GPUs
System size (nodes)	4,392 KNL nodes and 24 DGX-A100 nodes	9,300 nodes 1,900 nodes in data partition	4608 nodes	> 1,500(GPU) > 3,000 (CPU)	> 500	> 9,000 nodes	> 9,000 nodes
CPU-GPU Interconnect	NVLINK on GPU nodes	N/A	NVLINK Coherent memory across node	PCIe		AMD Infinity Fabric Coherent memory across the node	Unified memory architecture, RAMBO
Node-to-node interconnect	Aries (KNL nodes) and HDR200 (GPU nodes)	Aries	Dual Rail EDR-IB	HPE Slingshot NIC	HPE Slingshot NIC	HPE Slingshot	HPE Slingshot
File System	200 PB, 1.3 TB/s Lustre 10 PB, 210 GB/s Lustre	28 PB, 744 GB/s Lustre	250 PB, 2.5 TB/s GPFS	35 PB All Flash, Lustre	N/A	695 PB + 10 PB Flash performance tier, Lustre	≥ 230 PB, ≥ 25 TB/s DAOS



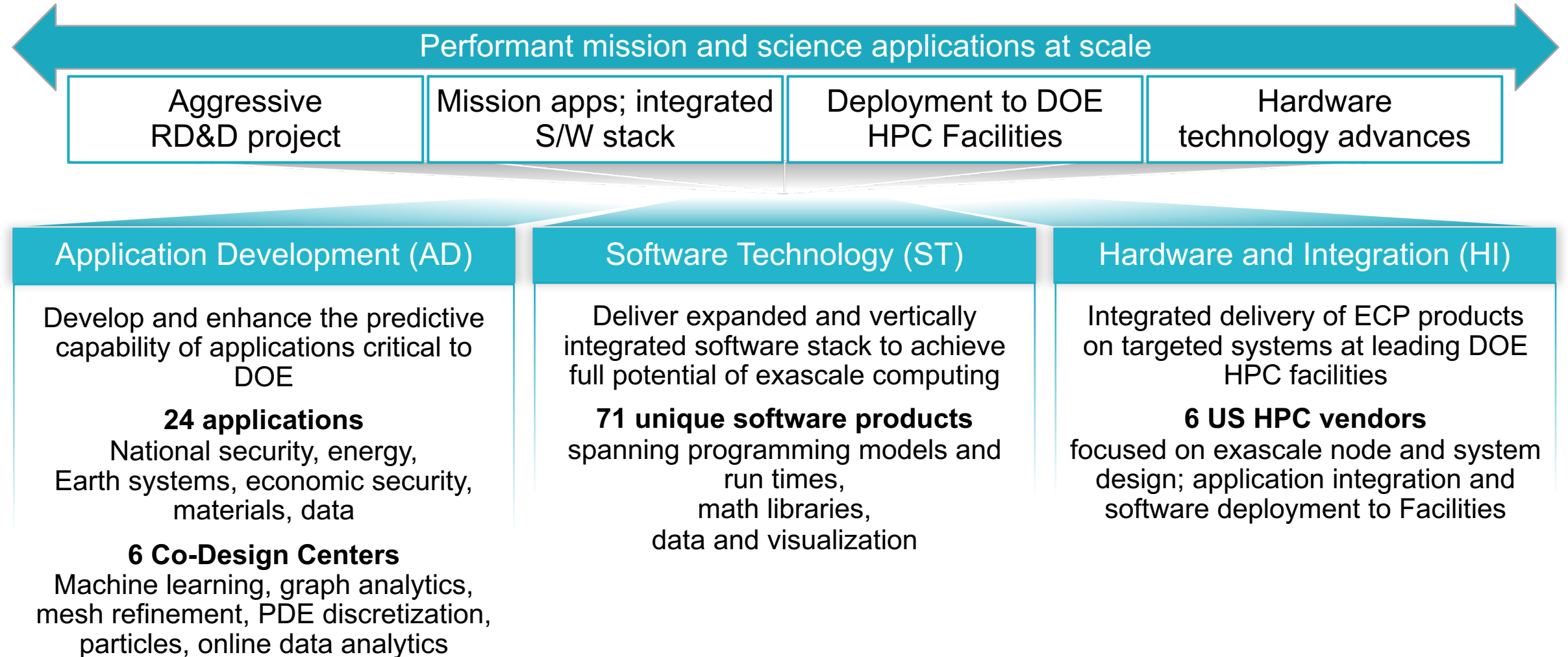
U.S. DEPARTMENT OF
ENERGY

Office of
Science

ASCR Computing Upgrades At-a-Glance
November 24, 2020

ECP's Technical Focus Areas

Providing the necessary components to meet national goals



ECP Application Portfolio: 24 First-Movers of Strategic Importance to DOE

Starting Point

- **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 S&E domains
- Mostly all MPI or MPI+OpenMP on CPUs
- Each envisioned innovative S&E enabled by 100X increase in computing power
- Path to harnessing 100-fold improvement initially unknown likely to have disruptive impact on software unlike anything in last 30 years

Current status

- All applications have, with their own unique development plans, made tremendous progress in model and algorithm development and software architecture redesign / refactor. Most applications have integrated and adopted software abstraction layers or co-designed motif-based components and frameworks to ensure efficient and portable GPU implementations.
- Many application have already realized >50X increase in science work rate performance on the Summit system at ORNL since starting ECP development activities in 2016

→ **Massive software investments**

National security

Energy security

Economic security

Scientific discovery

Earth system

Health care

Next-generation, stockpile stewardship

Turbine wind plant

Additive manufacturing

Cosmological probe of the standard model of particle physics

Accurate regional impact assessments in Earth system models

Accelerate and translate cancer research (partnership with NIH)

Reentry-vehicle environment simulation

Design and optimization of SIMS

Reliable and reproducible metal parts

Validate fundamental laws of nature

Stress-resistant crop analysis and catalytic conversion of biomass-derived

Multi-physics simulations of high energy density physics conditions

and fusion reactor materials design

Scientific-based risk assessment

Plasma wakefield accelerator design

Metagenomics for analysis of biogeochemical cycles, climate change, environmental



ECP applications are using a variety of programming models and implementation strategies

GPU-specific kernels

- Isolate the computationally-intensive parts of the code into CUDA/HIP/SYCL kernels.
- Refactoring the code to work well with the GPU is the majority of effort.

Loop pragma models

- Offload loops to GPU with OpenMP or OpenACC.
- Most common portability strategy for Fortran codes.

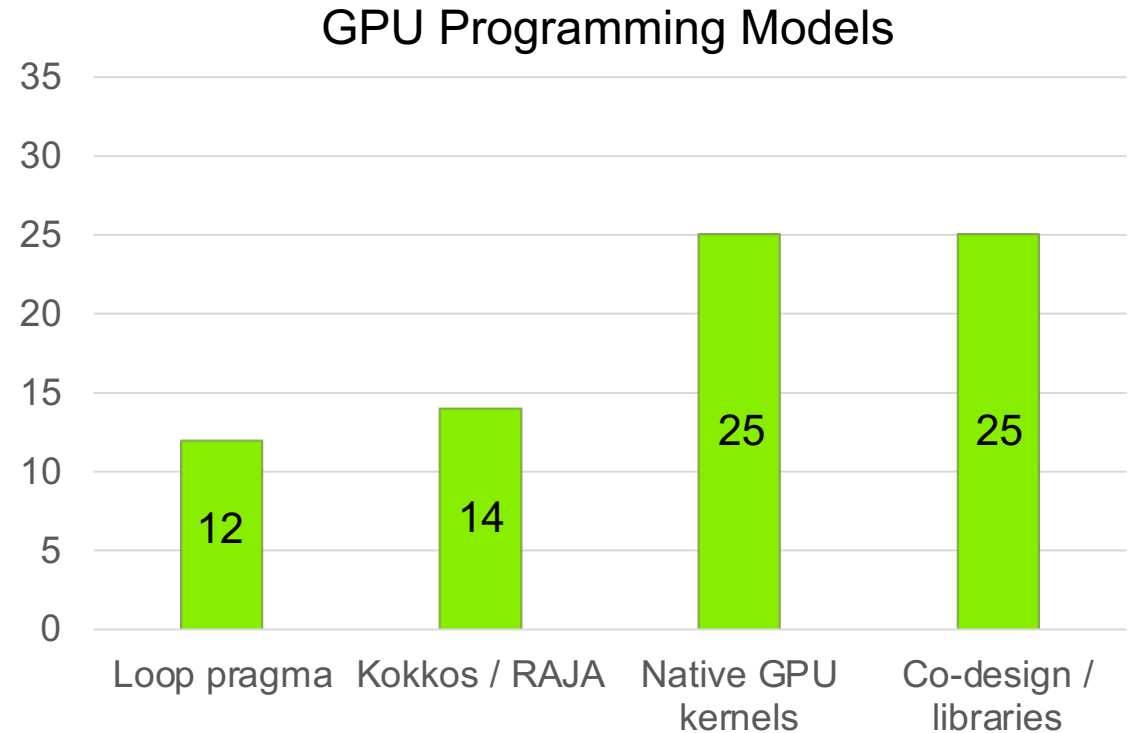
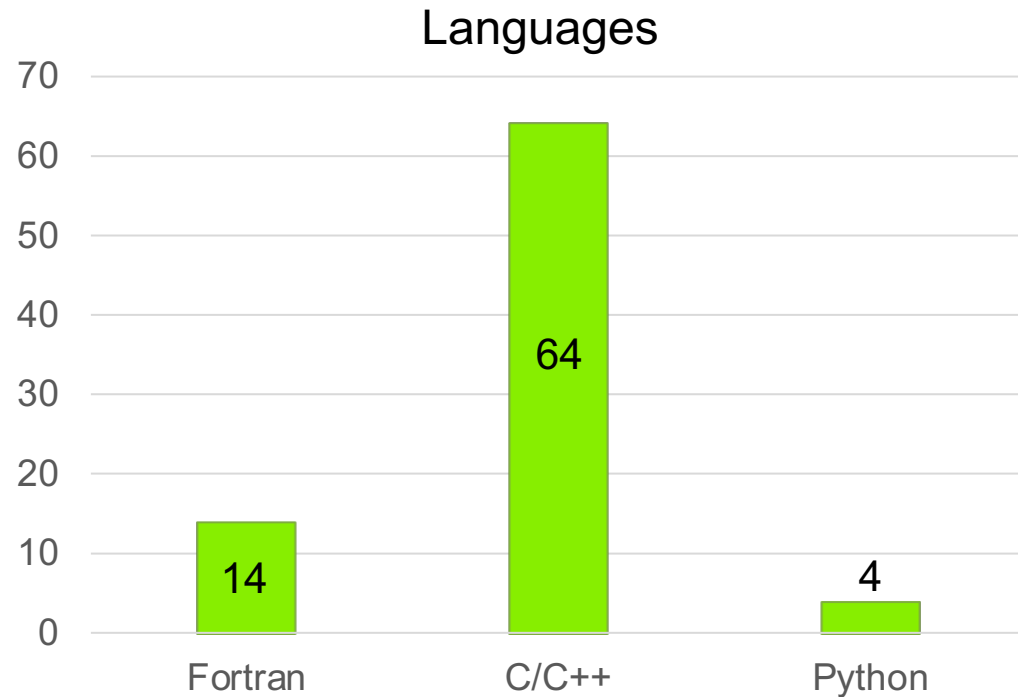
C++ abstraction layers

- Fully abstract loop execution and data management using advanced C++ features.
- Kokkos and RAJA developed by NNSA in response to increasing hardware diversity.

Co-design frameworks and libraries

- Design application with a specific motif to use common software components
- Depend on well-integrated library to manage data, execute key functions on GPU.

ECP applications are using a variety of programming models and implementation strategies



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!

Trends in Internode Programming

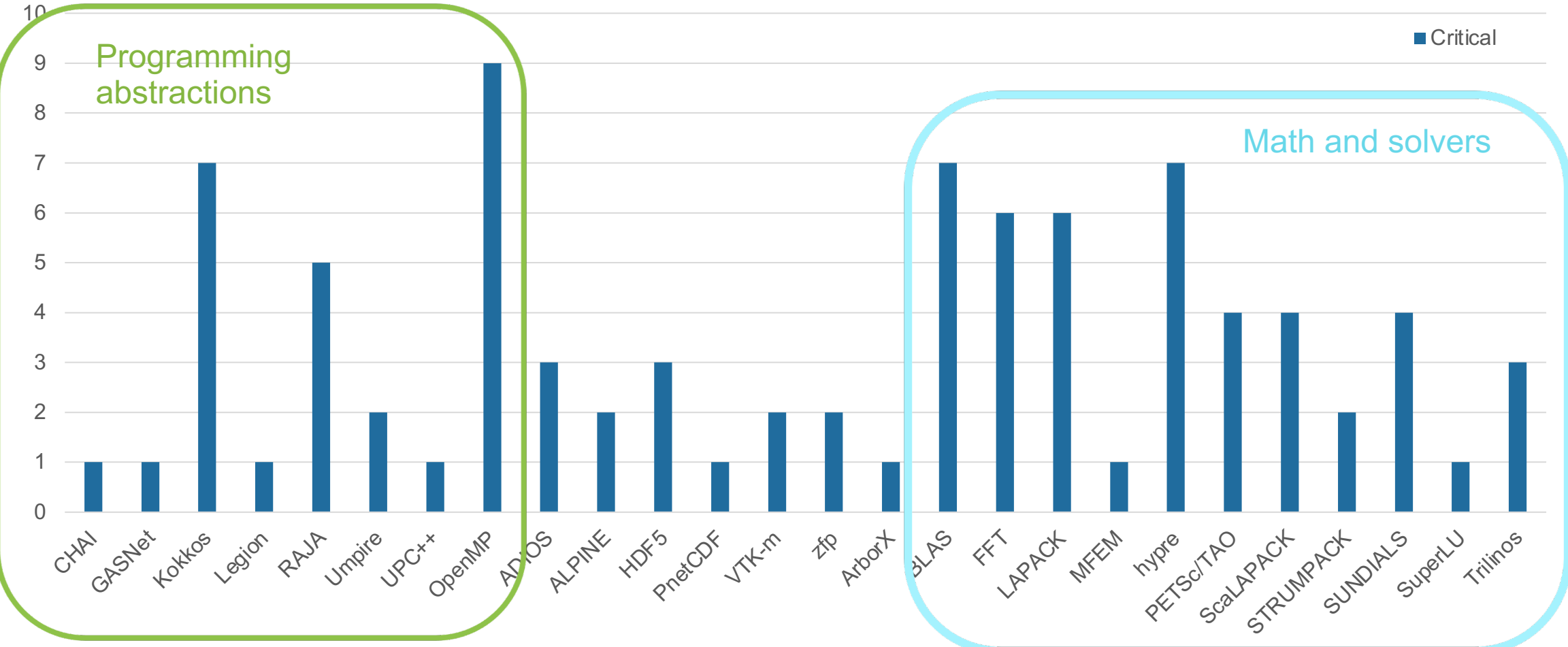
- Individual compute nodes are becoming very powerful because of accelerators
- As a result, fewer total number of nodes are needed
- MPI will continue to serve as the main internode programming model
- Nonetheless, improvements are needed in the MPI Standard and in MPI implementations
 - Hybrid programming (integration with GPUs & GPU memory and with the node programming model)
 - Scalability, low-latency communication, optimized collective algorithms
 - Overall resilience and robustness
 - Optimized support for exascale interconnects and lower-level communication paradigms (OFI, UCX)
 - Scalable process startup and management
- PGAS models (e.g., UPC++) are also available to be used by applications that rely on them, and they face similar challenges as MPI on exascale systems

Trends in Intranode Programming

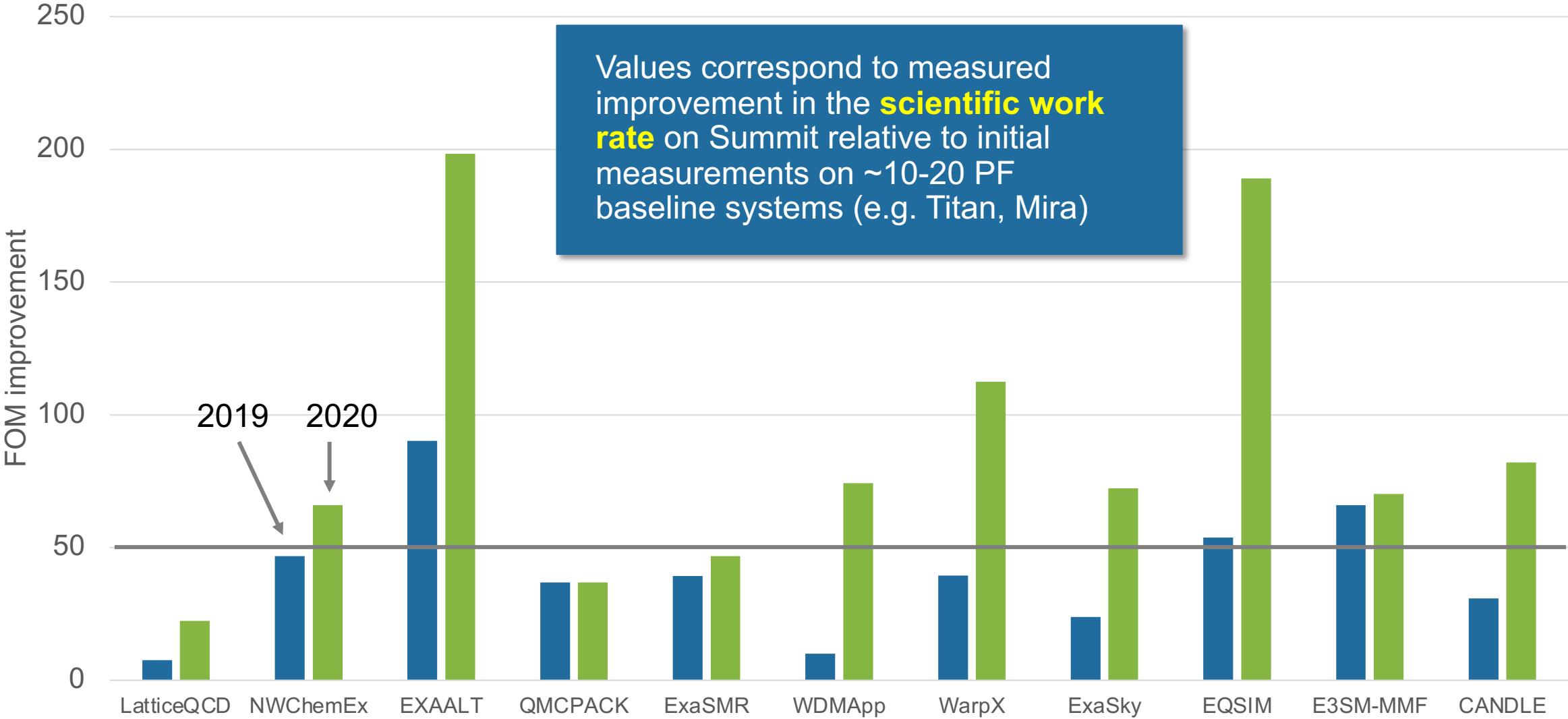
- Main challenge for exascale is in achieving performance and portability for intranode programming
- Vendor-supported options for GPUs
 - NVIDIA: CUDA, OpenACC
 - Intel: SYCL/DPC++ (C++ abstractions on top of OpenCL)
 - AMD: HIP (similar to CUDA)
- OpenMP (portable standard)
 - Supports accelerators via the `target` directive (since OpenMP version 4.0, July 2013)
 - Subsequent releases of OpenMP (4.5 and 5.0) have further improved support for accelerators
 - Supported by vendors on all platforms
- Kokkos and RAJA (developed at SNL and LLNL)
 - Portable, heterogenous-node programming via C++ abstractions
 - Support complex node architectures with multiple types of execution resources and multilevel memories
 - Many ECP applications use Kokkos and RAJA to write portable code for a variety of CPUs and GPUs

Critical dependencies highlighted at recent reviews

All dependencies listed in AD reviews (with standard languages, compilers, tools omitted)



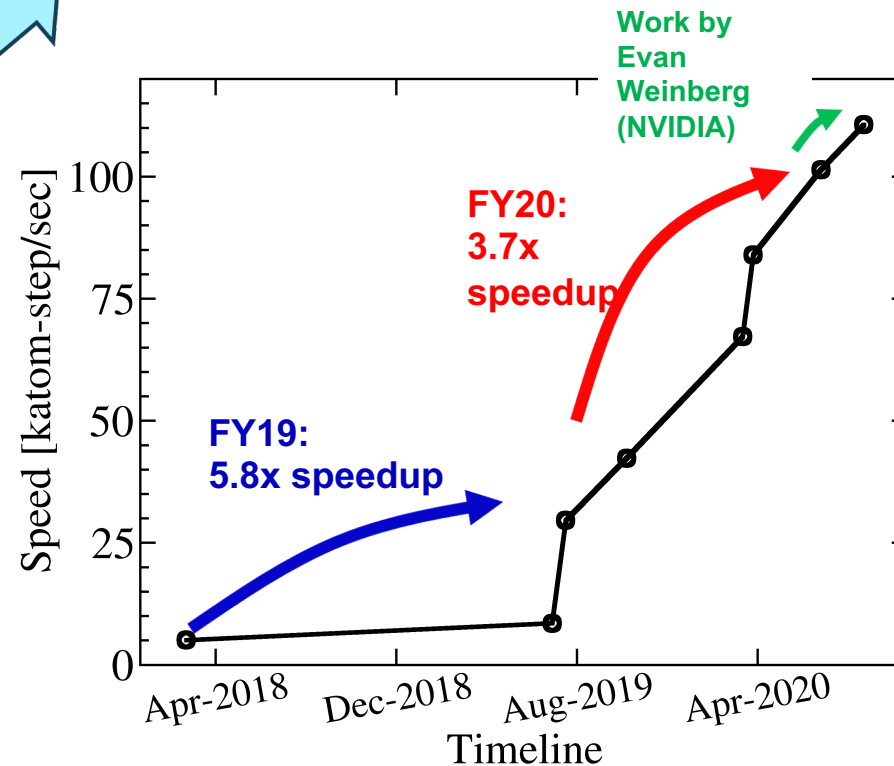
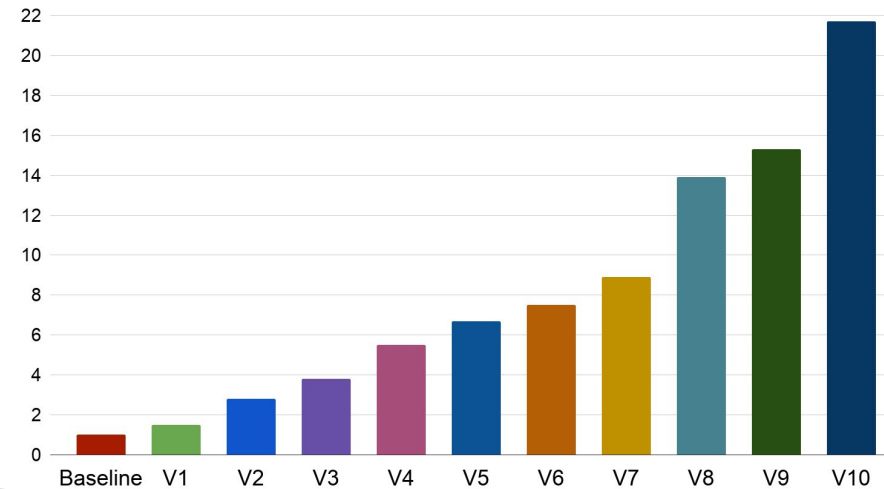
FY20 vs. FY19 Summit Performance for Selected ECP Applications



Application Exemplar: Molecular Dynamics (EXAALT) Optimization on V100

- Production LAMMPS/Kokkos version is now highly optimized for GPUs
- Significant performance improvement over the last year. Tentative FOM increased **from 88x to 288x**.
- Performance improvements come from a range of different sources, enabled by algorithmic restructuring
- **22x** improvement in performance compared to pre-ECP version of code on V100.
- Joint effort of EXAALT, CoPA, NERSC/NESAP, NVIDIA, and HPE

SNAP progress relative to baseline for 2J14



ECP Energy Applications: Code Base Summary

Domain (ECP Subproject)	Computational Motifs	Programming Models	Key Integration Points
Wind Energy (ExaWind)	Low-mach CFD, unstructured grids, block AMR, turbulence	C++/Fortran, Kokkos, AMReX	Kokkos, AMReX, xSDK
Combustion Engines (Combustion-PELE)	Low-mach/compressible CFD, reactive flow, block AMR + EB, particles, turbulence	C++ [☆] , AMReX	AMReX, SUNDIALS
Nuclear Energy (ExaSMR)	Low-mach CFD, turbulence, unstructured grids, Monte Carlo particle transport, neutron depletion	C++ [☆] , HIP/CUDA + Thrust OpenMP5 libparanumal/OCCA	CEED (libparanumal/OCCA)
(Fossil Fuel Combustion) MFIX-Exa	Low-mach multiphase CFD, DEM (particles), block AMR + EB	C++, AMReX	AMReX
Fusion (WDMApp)	PIC (particles), gyrokinetics, structured/unstructured grids	C++/Fortran, Cabana (Kokkos), CUDA, OpenACC	CoPA (Cabana), PETSc, ADIOS
Accelerator Physics (WarpX)	PIC (particles), block AMR, Maxwell Equations, FFT	C++ [☆] , AMReX	AMReX, heFFTe

ECP Application: Wind Energy

PI: Mike Sprague (NREL)

Challenge Problem and Codes

Predictive simulation of a wind farm with tens of megawatt-scale wind turbines dispersed over an area of 50 square kilometers

- Base: 2x2 wind farm; 3x3 km²
- Stretch: 3x3 wind farm; 4x4 km²

Code Base

- *Nalu-Wind*: unstructured, finite-volume
- *AMR-Wind*: block-AMR background solver
- *OpenFAST*: wind-turbine simulation code

Physical Models, Motifs, Programming Models

Models

Multi-scale, multi-physics computational fluid dynamics, coupled fluid-structure models, moving/deforming meshes

Motifs

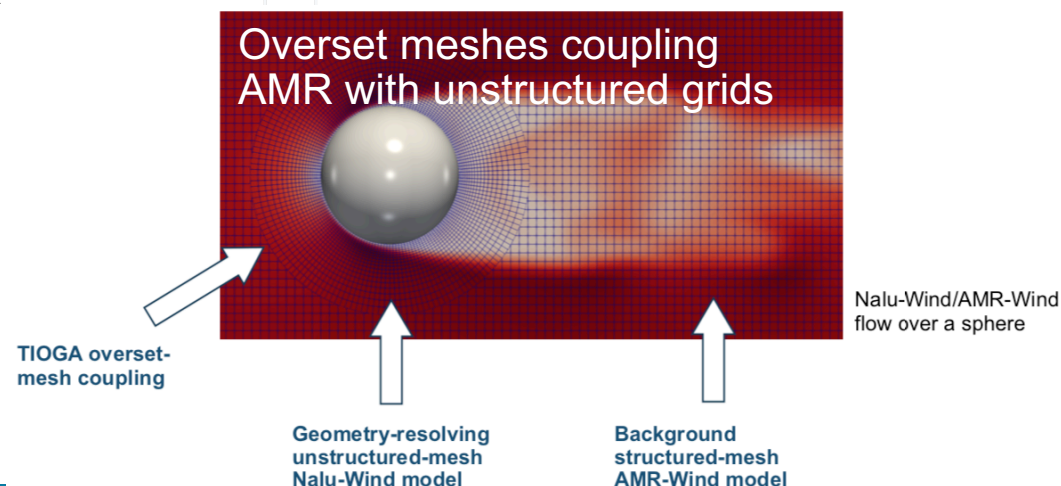
- PDEs, sparse linear algebra
- unstructured grids, AMR

Programming Models

- Kokkos
- AMReX

Challenges

- Linear solver performance per timestep in strong-scaling limit
 - Multiple solvers implemented through xSDK
- Use of UVM on AMD
- Kokkos enabled clean porting to Summit
 - Aurora/Frontier an open question



ECP Application: Internal Combustion Engines

PI: Jackie Chen (SNL)

Challenge Problem and Codes

Simulation of the in-cylinder processes in an internal combustion engine utilizing low-temperature combustion (LTC) through reactivity controlled compression ignition (RCCI)

- Base: 2 fuels under multipulse injection
- Stretch: multi-phase fuel injection with soot emission

Code Base

- *Pele-C*: compressible
- *Pele-LM*: low-mach
- *PelePhysics*: physical parameter databases

Physical Models, Motifs, Programming Models

Models

turbulence, chemical kinetics, polydisperse spray, soot, thermal radiation, non-ideal fluids

Motifs

- PDEs, AMR, sparse iterative linear solvers
- implicit-explicit schemes, ODEs, linear algebra
- Particles

Programming Models

- AMReX

Challenges

- Methods for advancing chemistry terms in time on heterogenous architectures
 - SUNDIALS for explicit/implicit time marching
 - CPUs for spatially stiff chemistry, GPUs for remainder of domain
- Solver performance in Pele-LM
- Conversion from C++ to Fortran still ongoing

ECP Application: Nuclear Energy (Small Modular Reactors)

PI: Steve Hamilton (ORNL)

Challenge Problem and Codes

Simulation of SMR core physics with in-vessel coolant loop

- Base: CFD-neutronics coupling
- Stretch: CFD-neutronics + depletion for single state point

Code Base

- *NekRS*: spectral finite element low-mach CFD
- *Shift*: Monte Carlo neutronics + depletion
- *OpenMC*: Monte Carlo neutronics

Physical Models, Motifs, Programming Models

Models

neutronics, incompressible fluid flow, heat transfer, turbulent flow

Motifs

- Monte Carlo, Particles,
- PDEs on unstructured grids, High-order FE, Spectral element methods, Sparse linear algebra,
- Structured grids, Dense linear algebra

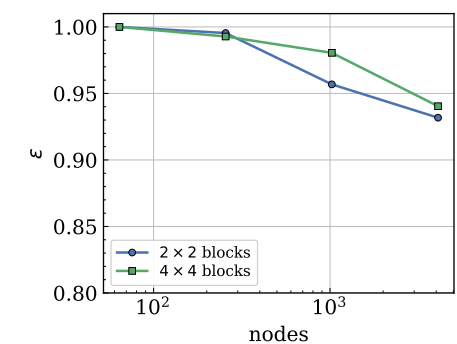
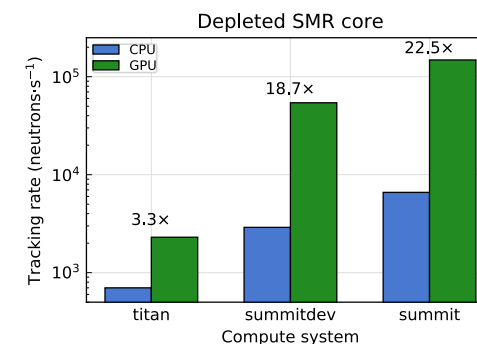
Programming Models

- CUDA/HIP/Thrust
- OpenMP5
- OCCA

Challenges

- Mapping Monte Carlo particle transport to SIMT architectures
 - Random access data patterns
 - Branch divergence
- Unstructured mesh generation
- Iteration to steady-state

Continuous-energy Monte Carlo transport on GPUs



ECP Application: Chemical Looping Reactors

PI: Madhava Syamlal (NETL)

Challenge Problem and Codes

Simulation of NETL's 50 kW chemical loop reactor (CLR) containing 5B particles for 5 minutes of operational time

- Base: exit gas compositions reach stationary state
- Stretch: exit gas concentrations reach stationary state

Code Base

- *MFIX-Exa*: fluid flow + DEM solid particle transport

Physical Models, Motifs, Programming Models

Models

Low-Mach number fluid flow with multi-species, reactive transport coupled to discrete solid particle transport

Motifs

- Particles,
- Sparse Linear Algebra,
- AMR, Structured Grids

Programming Models

- AMReX

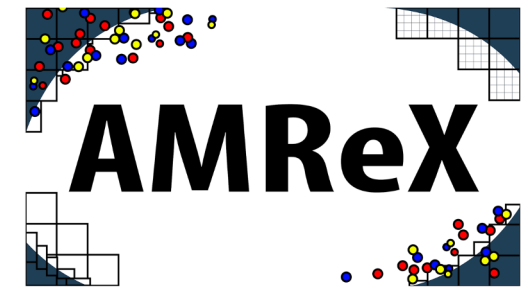
Challenges

- Mapping DEM to adaptive grids with multiple levels of refinement
- Embedded boundary treatments between grid solve and particles
- DEM algorithmic efficiency on GPU

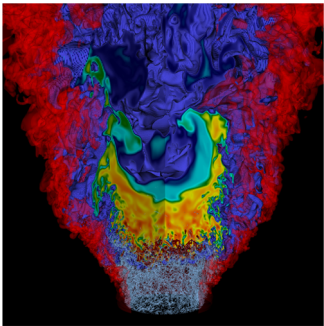
ECP Co-Design Portfolio

Project	PI Name, Inst	Short Description/Objective
Proxy Apps	David Richards, LLNL	Improve the quality of proxy applications and maximize the benefit received from their use; maintain and distribute the ECP Proxy App Suite
CODAR	Ian Foster, ANL	Advance understanding of the constraints, mappings, and configuration choices that determine interactions of applications, data analysis and reduction, and exascale platforms
AMReX	John Bell, LBNL	Build framework to support development of block-structured adaptive mesh refinement algorithms for solving systems of partial differential equations on exascale architectures
CEED	Tzanio Kolev, LLNL	Develop next-generation discretization software and algorithms that will enable finite element applications to run efficiently on future hardware
CoPA	Susan Mniszewski, LANL	Create co-designed numerical recipes and performance-portable libraries for particle-based methods
ExaGraph	Mahantesh Halappanavar, PNNL	Develop methods and techniques for efficient implementation of key combinatorial (graph) algorithms
ExaLearn	Frank Alexander, BNL	Deliver state-of-the-art machine learning and deep learning software at the intersection of applications, learning methods, and exascale platforms

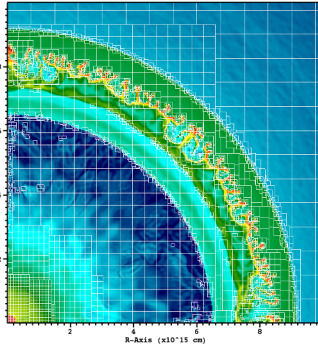
AMReX provides portability to ECP applications through multiple low-level implementations



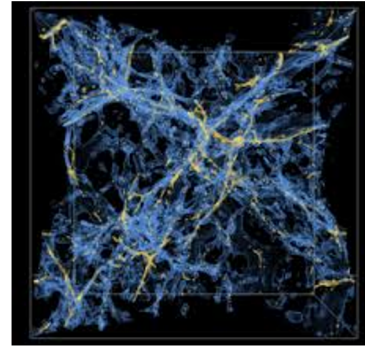
Combustion-PELE
(PeleC and PeleLM)



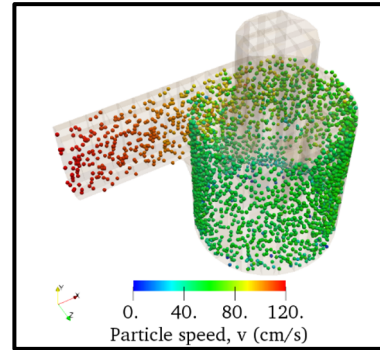
ExaStar
(Castro)



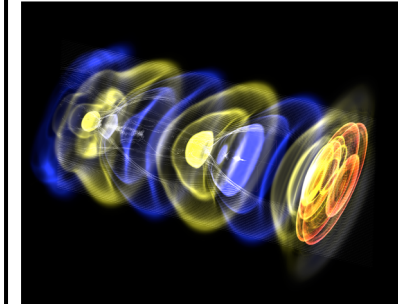
ExaSky
(Nyx)



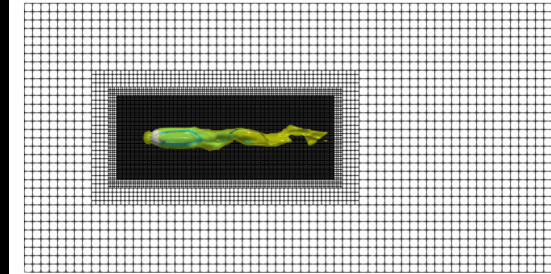
MFIX-Exa



WarpX



ExaWind
(AMR-Wind)



AMReX

MPI

OpenMP

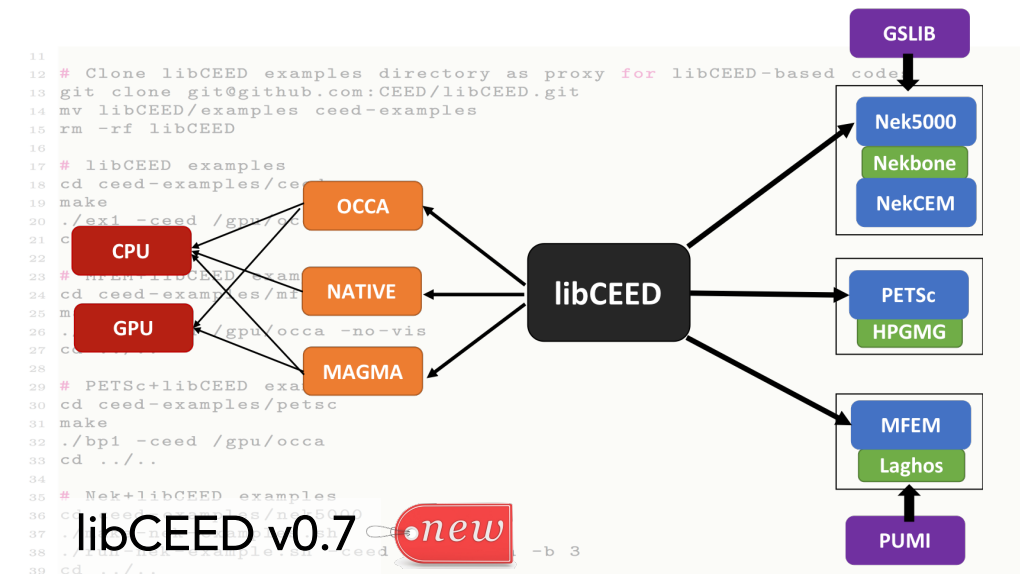
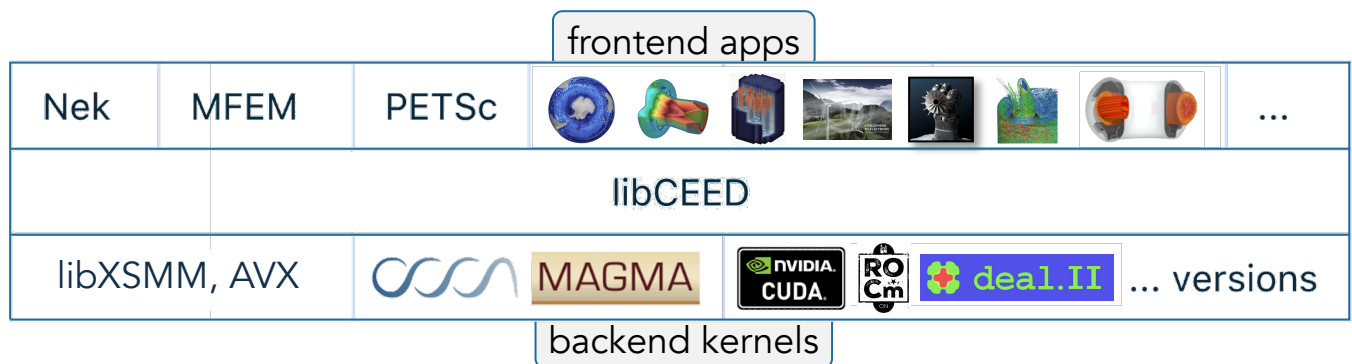
OpenACC

CUDA

HIP

DPC++

CEED provides multiple back-ends, including through its OCCA portability layer



- ✓ API between *frontend apps* and *backend kernels*
- ✓ *Efficient operator description* (not global matrix)
- ✓ Clients use any backend as a run-time option
- ✓ Backend can be added as plugins without recompiling
- ✓ Backends compete for best performance, latency vs throughput, optimize for order/device, use JIT, ...

- ✓ Extensible backends
 - **CPU**: reference, vectorized, libXSMM
 - **CUDA** using NVRTC cuda-gen
 - **OCCA** (JIT): CPU, OpenMP, OpenCL, CUDA
 - **MAGMA**

CoPA: Cabana particle library is built on a Kokkos portability layer

CabanaMD
Molecular dynamics proxy app

CabanaPIC
Particle-in-cell proxy app

ExaMPM
Material point method proxy app

XGC
Plasma PIC

Picasso
Continuum Mechanics PIC

Cabana

- Flexible particle data layout
- Performance portable, multi-node **particle** and **particle-grid** motifs

hypre
Preconditioners and solvers

heFFTe
Performance portable, multi-node FFTs

MPI
Multi-node computation

ArborX
Geometric search

FFTW

cuFFT

rocFFT

Kokkos
On-node performance portability

CUDA

OpenMP

HIP

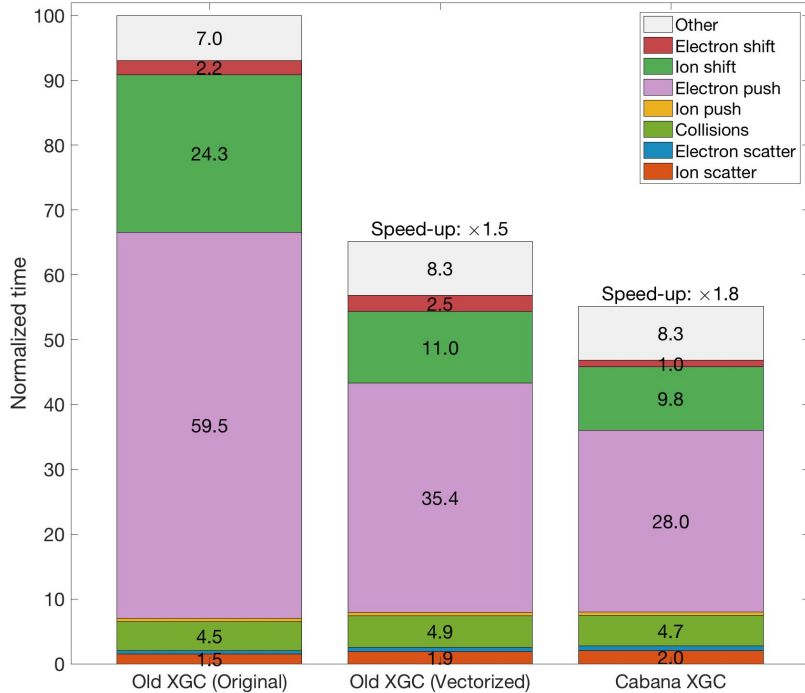
OpenMP Target

SYCL

Cabana provides performance portability to the XGC legacy PIC code

Outperforms hand-tuned OpenMP implementation on Cori (KNL)

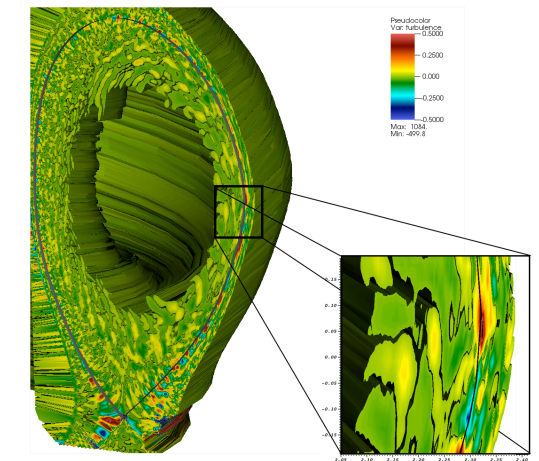
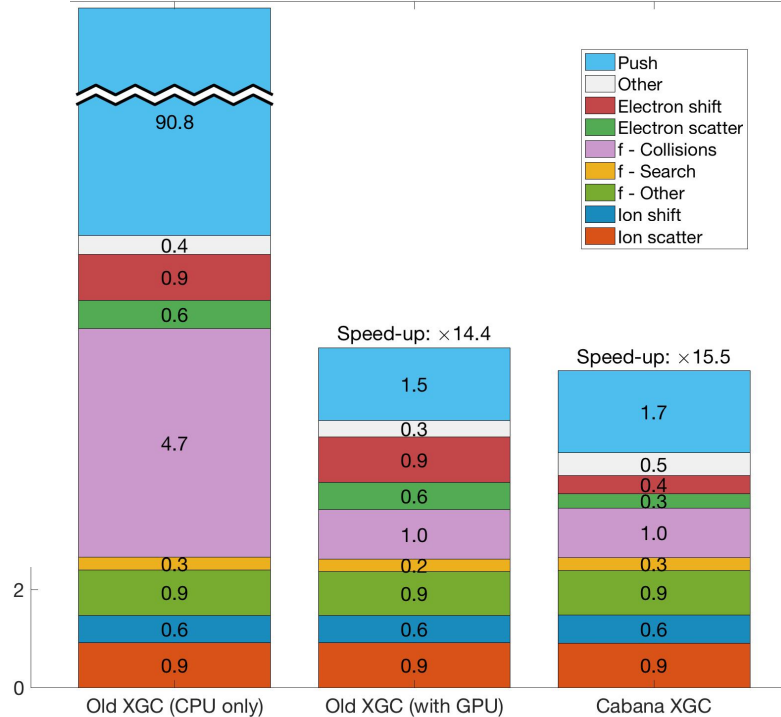
XGC versions on Cori KNL, 370k mesh, 12M particles/node, 2 planes, 512 KNL nodes



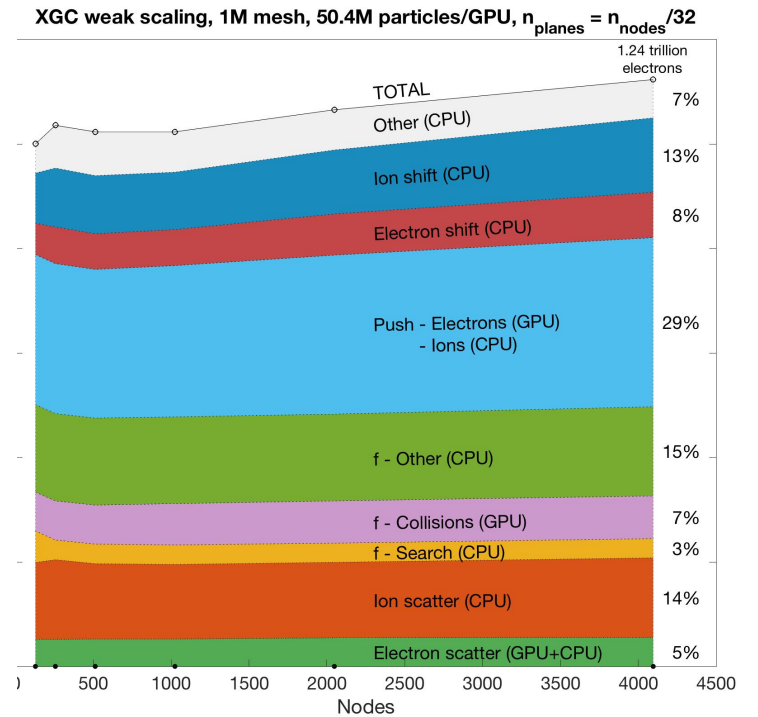
2019 results

Outperforms hand-tuned CUDA implementation on Summit

XGC on 256 Summit nodes, 1M mesh, 25.2M particles/GPU, 8 planes



...while retaining scalability (1.2 T electrons on full Summit)



Experiences on early access hardware and next steps

- Teams have been working on small test machines with AMD and Intel GPUs
- Portability abstractions have sped up porting process and reduced duplication of effort.
- Integration with ST, co-design centers means common components can be optimized for multiple codes.
- Goal is to discover missing features, learn where performance needs improvement, and identify bugs.
- Work on Summit showed that robust performance analysis tools are essential to achieve high performance, this is one of the challenges of new architectures.

We cannot yet claim complete success with performance portable abstractions, but early evidence is looking good!

Interesting Themes Emerging from 2020 AD 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

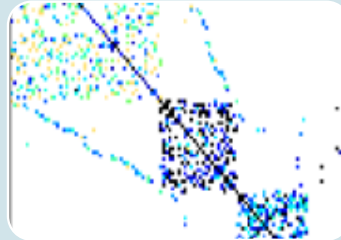
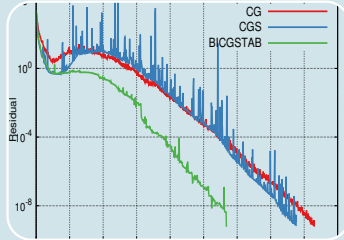
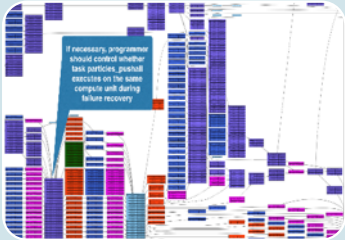
WBS 2.2, Milestone PM-AD-1110

Andrew Siegel¹, Erik Draeger², Jack Deslippe³,
Tom Evans⁴, Tim Germann⁵, Dan Martin²,
and William Hart⁶

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

December 2, 2020

ECP ST has six technical areas



Programming Models & Runtimes

- Enhance and get ready for exascale the widely used MPI and OpenMP programming models (hybrid programming models, deep memory copies)
- Development of performance portability tools (e.g. Kokkos and Raja)
- Support alternate models for potential benefits and risk mitigation: PGAS (UPC++/GASNet), task-based models (Legion, PaRSEC)
- Libraries for deep memory hierarchy and power management

Development Tools

- Continued, multifaceted capabilities in portable, open-source LLVM compiler ecosystem to support expected ECP architectures, including support for F18
- Performance analysis tools that accommodate new architectures, programming models, e.g., PAPI, Tau

Math Libraries

- Linear algebra, iterative linear solvers, direct linear solvers, integrators and nonlinear solvers, optimization, FFTs, etc
- Performance on new node architectures; extreme strong scalability
- Advanced algorithms for multi-physics, multiscale simulation and outer-loop analysis
- Increasing quality, interoperability, complementarity of math libraries

Data and Visualization

- I/O via the HDF5 API
- Insightful, memory-efficient in-situ visualization and analysis – Data reduction via scientific data compression
- Checkpoint restart

Software Ecosystem

- Develop features in Spack necessary to support all ST products in E4S, and the AD projects that adopt it
- Development of Spack stacks for reproducible turnkey deployment of large collections of software
- Optimization and interoperability of containers on HPC systems
- Regular E4S releases of the ST software stack and SDKs with regular integration of new ST products

NNSA ST

- Open source NNSA Software projects
- Projects that have both mission role and open science role
- Major technical areas: New programming abstractions, math libraries, data and viz libraries
- Cover most ST technology areas
- Subject to the same planning, reporting and review processes

Kokkos and RAJA

- C++ performance portability abstractions developed at Sandia and Livermore labs
- Primarily funded by NNSA
- ECP ST provides additional funding to develop optimized backends for Aurora and Frontier (OpenMP, SYCL/DPC++, HIP) and for outreach to ECP applications
- Organized as one project in ECP ST. The two teams collaborate on common features, C++ and backend support, and outreach activities
- The combined project involves six labs: SNL, LLNL, ANL, LANL, LBNL, and ORNL
 - PI: Christian Trott (SNL); Co-PIs: Rich Hornung (LLNL), Nevin Liber (ANL), Galen Shipman (LANL), Jack Deslippe (LBNL), Damien Lebrun-Grandie (ORNL)

Kokkos: Preparation for Exascale Platforms

- Kokkos provides a production-quality solution for C++ performance portability
 - Kokkos Core: C++ template-based library for the programming model
 - Kokkos Tools: Profiling, debugging and tuning tools that connect into Kokkos Core
 - Kokkos Kernels: Math libraries based on Kokkos Core
- Distributed development team: SNL, ORNL, ANL, LBNL, and LANL
- Long-term goal is alignment and extension of the ISO C++ standard
 - Fundamental capabilities from Kokkos are proposed for the C++ standard
- Kokkos is currently used on most DOE and many European production HPC systems
 - Numerous applications use Kokkos to run on Sierra, Summit, Astra, Trinity, Theta, PizDaint, and others
- Extensively used across ECP AD and ST projects
 - Exawind, EXAALT, WDMApp, ExaAM, SNL and LANL ATDM apps, ExaGraph, CoPA, ALExa (ArborX and DTK), Kokkos Kernels, Trilinos, FleCSI, ...
- Numerous workshops and training events
 - Now available “The Kokkos Lectures”: 15 hours of recorded lectures <https://kokkos.link/the-lectures>

RAJA: Preparation for Exascale Platforms

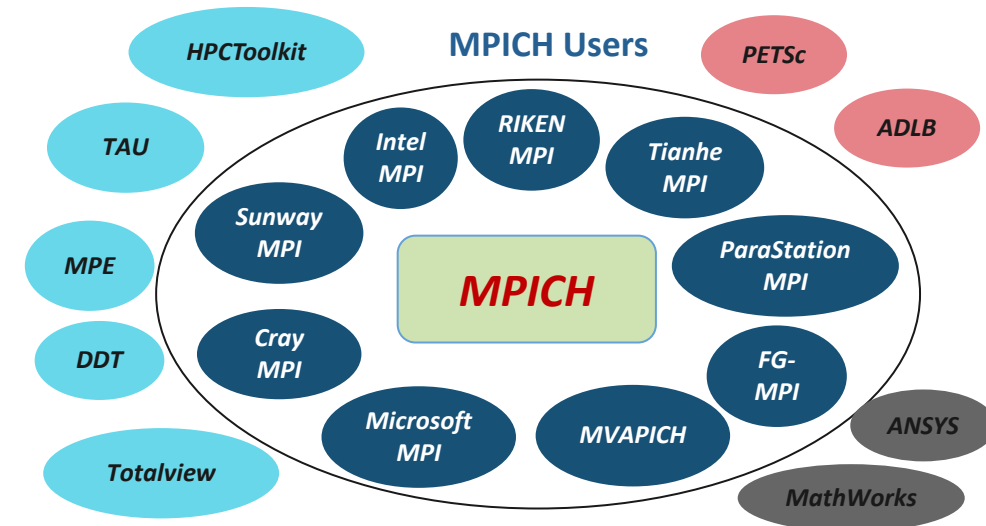
- A suite of performance-portability tools developed at LLNL
 - RAJA (C++ kernel execution abstractions); CHAI (C++ array abstractions); Umpire (memory management)
- Used in a diverse set of ECP and LLNL mission applications and support libraries
 - Supports majority of production LLNL weapons program apps and apps in other LLNL programs
 - ECP: SW4 (EQSIM), GEOSX (Subsurface), MFEM (CEED), ExaSGD, SUNDIALS, DevilRay (Alpine)
- Exascale platform support
 - Full support for HIP back-end in public releases since January 2020. Key part of El Capitan CoE activities.
 - SYCL back-end in progress, including SW4 & RAJA Performance Suite running on pre-Aurora systems now
- ECP applications have demonstrated substantial GPU node (Sierra) vs. CTS-1 node speedups
 - SW4: 28X speedup on Sierra node (4 NVIDIA Volta GPUs) vs. CTS-1 CPU-only node (36 core Intel Xeon).
 - Hayward fault earthquake simulation of unprecedented resolution (26B grid points) runs in 10.3 hours on 256 nodes of Sierra (6% of machine). Same problem (grid size, run time) requires 8196 nodes of Cori-II (85% of machine)
 - GEOSX: 14X speedup on Sierra node vs. CTS-1 CPU-only node for explicit time integration solid mechanics model needed for ECP challenge problem

Exascale MPI (MPICH)

- PI: Yanfei Guo (ANL)
- MPICH has been a key influencer in the adoption of MPI
 - First or most comprehensive implementation of each new version of the MPI standard
 - Allows supercomputing centers to not compromise on what features they demand from vendors
- R&D 100 Award in 2005
- MPICH and its derivatives are the world's most widely used MPI implementations on large-scale supercomputers
- MPICH will be the primary MPI implementation on Aurora, Frontier, and El Capitan (via Intel MPI and Cray MPI)

Ongoing activities

- MPI Forum and standardization efforts
- Addressing ECP application-specific issues
- Efficient and transparent support for GPUs from multiple vendors (Intel, AMD, NVIDIA)
- New library for efficient noncontiguous data communication (MPI derived datatypes)



***MPICH is not just software
It's an ecosystem***

ECP's Software Development Kits (SDKs) Span All Technology Areas

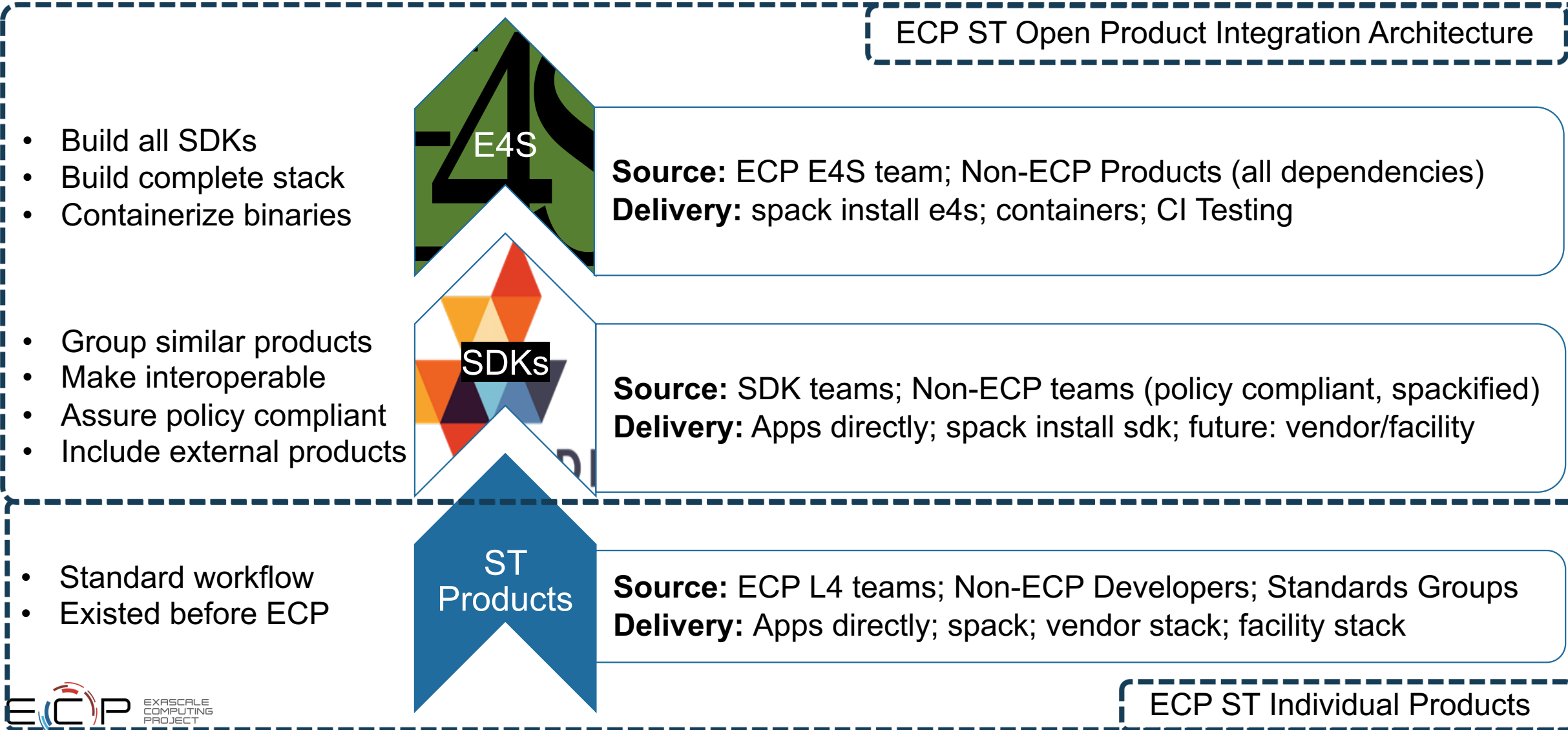
ECP's Extreme Scale Scientific Software Stack (E4S) embodies the latest Software Technology products developed in ECP and packaged in SDKs. The latest Nov 2020 release (<https://e4s.io>) includes 67 distinct products using the Spack package manager in a full-feature containerized release. E4S also supports AI/ML packages such as TensorFlow, PyTorch, and Horovod. E4S is available for download from Dockerhub, with bare metal and custom containers also supported using the E4S Spack build cache.

PMR Core (17)	Compilers and Support (7)	Tools and Technology (11)	xSDK (16)	Visualization Analysis and Reduction (9)	Data mgmt, I/O Services, Checkpoint restart (12)	Ecosystem/E4S at-large (12)
QUO	openarc	TAU	hypr	ParaView	SCR	mpiFileUtils
Papyrus	Kitsune	HPCToolkit	FleSCI	Catalyst	FAODEL	TriBITS
SICM	LLVM	Dyninst Binary Tools	MFEM	VTK-m	ROMIO	MarFS
Legion	CHILL autotuning comp	Gotcha	Kokkoskernels	SZ	Mercury (Mochi suite)	GUFI
Kokkos (support)	LLVM openMP comp	Caliper	Trilinos	zfp	HDF5	Intel GEOPM
RAJA	OpenMP V & V	PAPI	SUNDIALS	VisIt	Parallel netCDF	BEE
CHAI	Flang/LLVM Fortran comp	Program Database Toolkit	PETSc/TAO	ASCENT	ADIOS	FSEFI
PaRSEC*		Search (random forests)	libEnsemble	Cinema	Darshan	Kitten Lightweight Kernel
DARMA		Siboka	STRUMPACK	ROVER	UnifyCR	COOLR
GASNet-EX		C2C	SuperLU		VeloC	NRM
Qthreads		Sonar	ForTrilinos		IOSS	ArgoContainers
BOLT			SLATE		HXHIM	Spack
UPC++			MAGMA			
MPICH			DTK			
Open MPI			Tasmanian			
Umpire			TuckerMPI			
AML						

Legend

- PMR
- Tools
- Math Libraries
- Data and Vis
- Ecosystems and delivery

Delivering an Open, Hierarchical software Ecosystem



Vision for E4S Now and in the Future

- E4S has emerged as a new top-level component in the DOE HPC community, enabling fundamentally new relationships
- E4S has similar potential for new interactions with other US agencies, US industry and international collaborators. NSF and UK are examples
- The E4S portfolio can expand to include new domains (ML/AI), lower—level components (OS), and more.
- E4S can provide better (increased quality), faster (timely delivery of leading edge capabilities) and cheaper (assisting product teams)
- E4S v1.2 Released for SC20 – Details in breakout session – aka, E4S 20.10 (YY.MM), targeting quarterly releases
- E4S support at NERSC: <https://docs.nersc.gov/applications/e4s/> - E4S 20.10 installed at NERSC

ST Capability Assessment Report (CAR)

- Products discussed here are presented with more detail and further citations.
- We classify ECP ST product deployment as broad, moderate, or experimental.
 - Broad and moderate deployment is typically suitable for collaboration.
 - Web links are available for almost all products.
 - 67 of 71 of ECP ST products are available as part of the Extreme-scale Scientific Software Stack (E4S) <http://e4s.io>.



ECP-RPT-ST-0002-2020-Public

ECP Software Technology Capability Assessment Report-Public

Michael A. Heroux, Director ECP ST
Lois Curfman McInnes, Deputy Director ECP ST
Rajeev Thakur, Programming Models & Runtimes Lead
Jeffrey S. Vetter, Development Tools Lead
Sherry Li, Mathematical Libraries Lead
James Ahrens, Data & Visualization Lead
Todd Munson, Software Ecosystem & Delivery Lead
Kathryn Mohror, NNSA ST Lead

November 19, 2020

<https://www.exascaleproject.org/wp-content/uploads/2021/01/ECP-ST-CAR-v2.5.pdf>

ECP is on track for successfully delivering on its objectives

- Application Development (AD)
 - Results on Summit have consistently exceeded expectations.
 - Early access hardware has allowed teams to begin tangible preparation for non-Nvidia GPUs.
 - Teams show keen awareness of what is needed to succeed and are eager to surpass threshold requirements
 - Applications are on track to meet project key parameters for performance and science outcomes
- Software Technology (ST)
 - Established a working testing framework and process in collaboration with HI and DOE HPC Facilities
 - Making excellent progress and are on track with delivery of key milestones and performance parameters
 - A software portfolio (E4S) that provides regular integration, testing, documentation and QA for ECP products
- Hardware & Integration (HI)
 - AD and ST teams are making productive use of Early Access Systems (EAS)
 - Integration of performance engineers on application teams showing to be best practice
 - Training programs are targeting EAS systems, well attended and received by ST and AD teams
 - ST products are being tested with Continuous Integration (CI) on the EAS platforms and NERSC Perlmutter

Questions?



Supplemental Material



ECP Applications

Risks to achieving their base challenge problems are becoming more focused and measurable

Domain*	Base Challenge Problem	Risks and Challenges
Wind Energy	2x2 5 MW turbine array in 3x3x1 km ³ domain	Linear solvers; structured / unstructured overset meshes
Nuclear Energy	Small Modular Reactor with complete in-vessel coolant loop	Coupled CFD + Monte Carlo neutronics; MC on GPUs
Fossil Energy	Burn fossil fuels cleanly with CLR	AMR + EB + DEM + multiphase incompressible CFD
Combustion	Reactivity controlled compression ignition	AMR + EB + CFD + LES/DNS + reactive chemistry
Accelerator Design	TeV-class 10 ²⁻³ times cheaper & smaller	AMR on Maxwell's equations + FFT linear solvers + PIC
Magnetic Fusion	Coupled gyrokinetics for ITER in H-mode	Coupled continuum delta-F + stochastic full-F gyrokinetics
Nuclear Physics: QCD	Use correct light quark masses for first principles light nuclei properties	Critical slowing down; strong scaling performance of MG-preconditioned Krylov solvers
Chemistry: GAMESS	Heterogeneous catalysis: MSN reactions	HF + MP2 + coupled cluster (CC) + fragmentation methods
Chemistry: NWChemEx	Catalytic conversion of biomass	CCSD(T) + energy gradients
Extreme Materials	Microstructure evolution in nuclear matls	AMD via replica dynamics; OTF quantum-based potentials
Additive Manufacturing	Born-qualified 3D printed metal alloys	Coupled micro + meso + continuum; linear solvers

ECP Applications

Risks to achieving their base challenge problems are becoming more focused and measurable

Domain*	Challenge Problem	Computational Hurdles
Quantum Materials	Predict & control matls @ quantum level	Parallel on-node perf of Markov-chain Monte Carlo; OpenMP
Astrophysics	Supernovae explosions, neutron star mergers	AMR + nucleosynthesis + GR + neutrino transport
Cosmology	Extract “dark sector” physics from upcoming cosmological surveys	AMR or particles (PIC & SPH); subgrid model accuracy; in-situ data analytics
Earthquakes	Regional hazard and risk assessment	Seismic wave propagation coupled to structural mechanics
Geoscience	Well-scale fracture propagation in wellbore cement due to attack of CO ₂ -saturated fluid	Coupled AMR flow + transport + reactions to Lagrangian mechanics and fracture
Earth System	Assess regional impacts of climate change on the water cycle @ 5 SYPD	Viability of Multiscale Modeling Framework (MMF) approach for cloud-resolving model; GPU port of radiation and ocean
Power Grid	Large-scale planning under uncertainty; underfrequency response	Parallel nonlinear optimization based on discrete algebraic equations; multi-period optimization
Cancer Research	Scalable machine learning for predictive preclinical models and targeted therapy	Increasing accelerator utilization for model search; exploiting reduced/mixed precision; resolving data management or communication bottlenecks
Metagenomics	Discover and characterize microbial communities through genomic and proteomic analysis	Graph algorithms, distributed hashing, matrix operations and other discrete algorithms
FEL Light Source	Protein and molecular structure determination using streaming light source data	Parallel structure determination for ray tracing and single-particle imaging