From Desktop to Exascale: All in on GPUs

Michael A. Heroux
Senior Scientist, Sandia National Laboratories
Former Director of Software Technology, ECP
PI, PESO, Post-ECP Software-Ecosystem Stewardship Project
SIAM Fellow

SIAM Supercomputing Spotlights Series
September 11, 2024

Exascale Computing Project

Maintain **international leadership in HPC**

Promote the **health of the US HPC industry**

Deliver a **sustainable software ecosystem** used and maintained for years to come

Ensure that exascale systems can be used to deliver **mission-critical applications**

ECP EXASCALE COMPUTING PROJECT

7-year, $1.8B

US Department of Energy project funded 1000+ people at national labs, universities, US industries

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Application Development

- Develop and enhance the predictive capability of applications, **25 applications, 6 Co-Design Centers**

Software Technology

- Deliver expanded and vertically integrated software stack, **70 unique products**

Hardware and Integration

- Application integration and software deployment to facilities, exascale node and system design, **6 US HPC vendors**

NNSA National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

ECP Application Development

National security	Energy security	Economic security	Scientific discovery	Earth system	Health care
Next-generation, **stockpile stewardship** codes	Turbine **wind plant** efficiency	**Additive manufacturing** of qualifiable metal parts	**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 commercialization of **SMR**s	Reliable and efficient planning	Validate fundamental laws of nature	Stress-resistant crop analysis and catalytic	

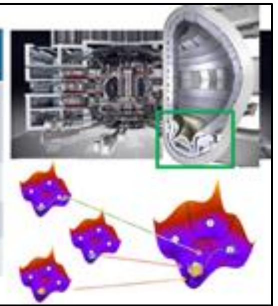
The 24 AD application projects

- Include **62 separate codes**
- Represent over **10 million lines of code**
- In some cases support large user communities
- Mostly started with MPI or MPI+OpenMP on CPUs

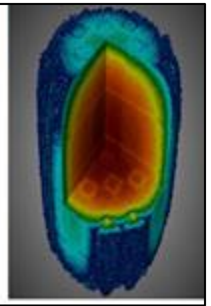
Manycore CPU → CPU+ GPU → CPU+ Multi-GPU → GPU Resident → Diverse Multi-GPU → Architecture-specific optimization

ECP libraries and tools supported diverse applications across multiple architectures

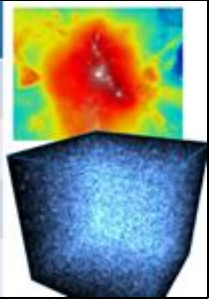
Project/PI	EXAALT: Molecular Dynamics Danny Perez
Challenge Problem	Damaged surface of Tungsten in conditions relevant to plasma facing materials in fusion reactors • 100,000 atoms • T=1200K
FOM Speedup	398.5
Nodes Used	7000
ST/CD Tools	Used in KPP Demo: **Kokkos, CoPa**



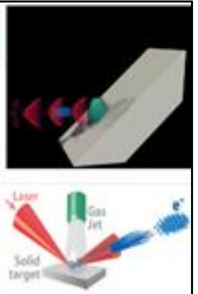
Project/PI	ExaSMR: Small Modular Reactors Steve Hamilton
Challenge Problem	NuScale-style Small Module Reactor (SMR) with depleted fuel and natural circulation • 213,860 Monte Carlo tally cells/6 reactions • 5.12×10^{12} particle histories/cycle, 40 cycles • 1098×10^{6} CFD spatial elements • 376×10^{9} CFD degrees of freedom • 1500 CFD timesteps
FOM Speedup	70
Nodes Used	6400
ST/CD Tools	Used in KPP Demo: **CEED** Additional: Trilinos



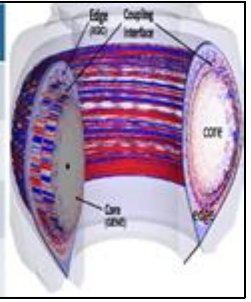
Project/PI	ExaSky: Cosmology Salman Habib
Challenge Problem	Two large cosmology simulations • gravity-only • hydrodynamics
FOM Speedup	271.65
Nodes Used	8192
ST/CD Tools	Used in KPP demo: **none** Additional: CoPa, VTK-m, CINEMA, HDF5.0



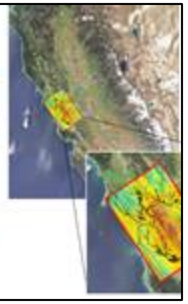
Project/PI	WarpX: Plasma Wakefield Accelerators Jean-Luc Vay
Challenge Problem	Wakefield plasma accelerator with a 1PW laser drive • 6.9×10^{12} grid cells • 14×10^{12} macroparticles • 1000 timesteps/1 stage
FOM Speedup	500
Nodes Used	8576
ST/CD Tools	Used in KPP Demo: **AMReX, libEnsemble** Additional: ADIOS, HDF5, VTK-m, ALPINE



Project/PI	WDMApp: Fusion Tokamaks Amitava Bhatacharjee
Challenge Problem	Gyrokinetic simulation of the full ITER plasma to predict the height and width of the edge pedestal
FOM Speedup	150
Nodes Used	6156
ST/CD Tools	Used in KPP Demo: **CODAR, CoPA, PETSc, ADIOS** Additional: VTK-m



Project/PI	EQSIM: Earthquake Modeling and Risk Dave McCallen
Challenge Problem	Impacts of Mag 7 rupture on the Hayward Fault on the bay area.
FOM Speedup	3467
Nodes Used	5088
ST/CD Tools	Used in KPP Demo: **RAJA, HDF5**



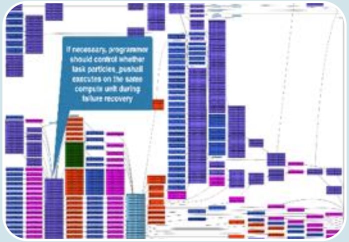
ECP Software Technologies

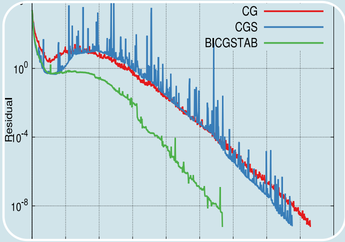
- Prepare SW stack for scalability with massive on-node parallelism
- Extend existing capabilities when possible, develop new when not
- Guide, and complement, and integrate with vendor efforts
- Develop and deliver high-quality and robust software products

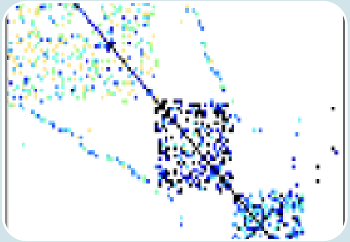
70 software products across 6 technical areas

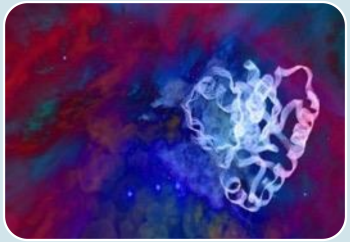
ECP Impact – Portable Libraries and Tools for Accelerators

ECP ST 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

ECP-sponsored Software Products: A Sample

Programming Models & Runtimes

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.

Four Parallel Node Programming Approaches

C++ abstractions

- 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

- Design application with a specific motif to use common software components
- Depend on co-design code (e.g. CEED, AMReX) to implement key functions on GPU.

Efficiently utilizing GPUs goes far beyond typical code porting

Port Code

- Rewrite, profile, and optimize
- Memory coalescing
- Loop ordering
- Kernel flattening

Adapt Numerics

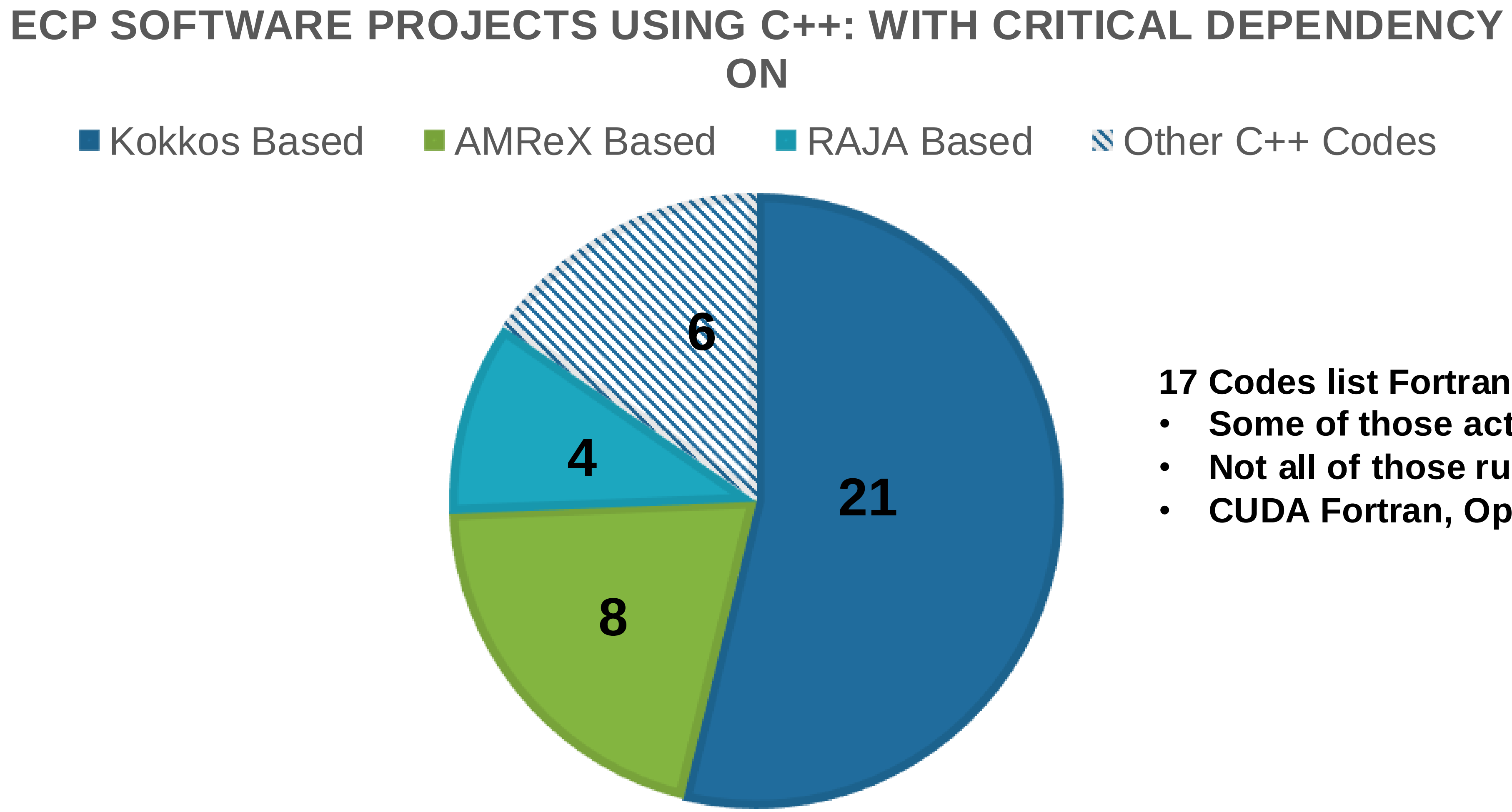
- Reduced synchronization
- Reduced precision
- Communication avoiding

Adapt Models

- Mathematical representation
- “On the fly” recomputing vs. lookup tables
- Prioritization of new physical models

ECP efforts have de-risked (but not removed) these activities for others

DOE Exascale Computing Project



17 Codes list Fortran as critical

- **Some of those actually moved on**
- **Not all of those run on GPUs**
- **CUDA Fortran, OpenMP, OpenACC**

Large Majority of C++ Codes Chose Abstraction over Vendor Models!

Kokkos Programming Model

What is Kokkos?

A C++ Programming Model for Performance Portability

- Implemented as a template library on top of CUDA, OpenMP, HIP, SYCL, …
- Aims to be descriptive not prescriptive
- Aligns with developments in the C++ standard
- Replaces usage of CUDA, OpenMP, HIP, etc.

Expanding solution for common needs of modern science/engineering codes

- Math libraries based on Kokkos
- Tools which enable insight into Kokkos

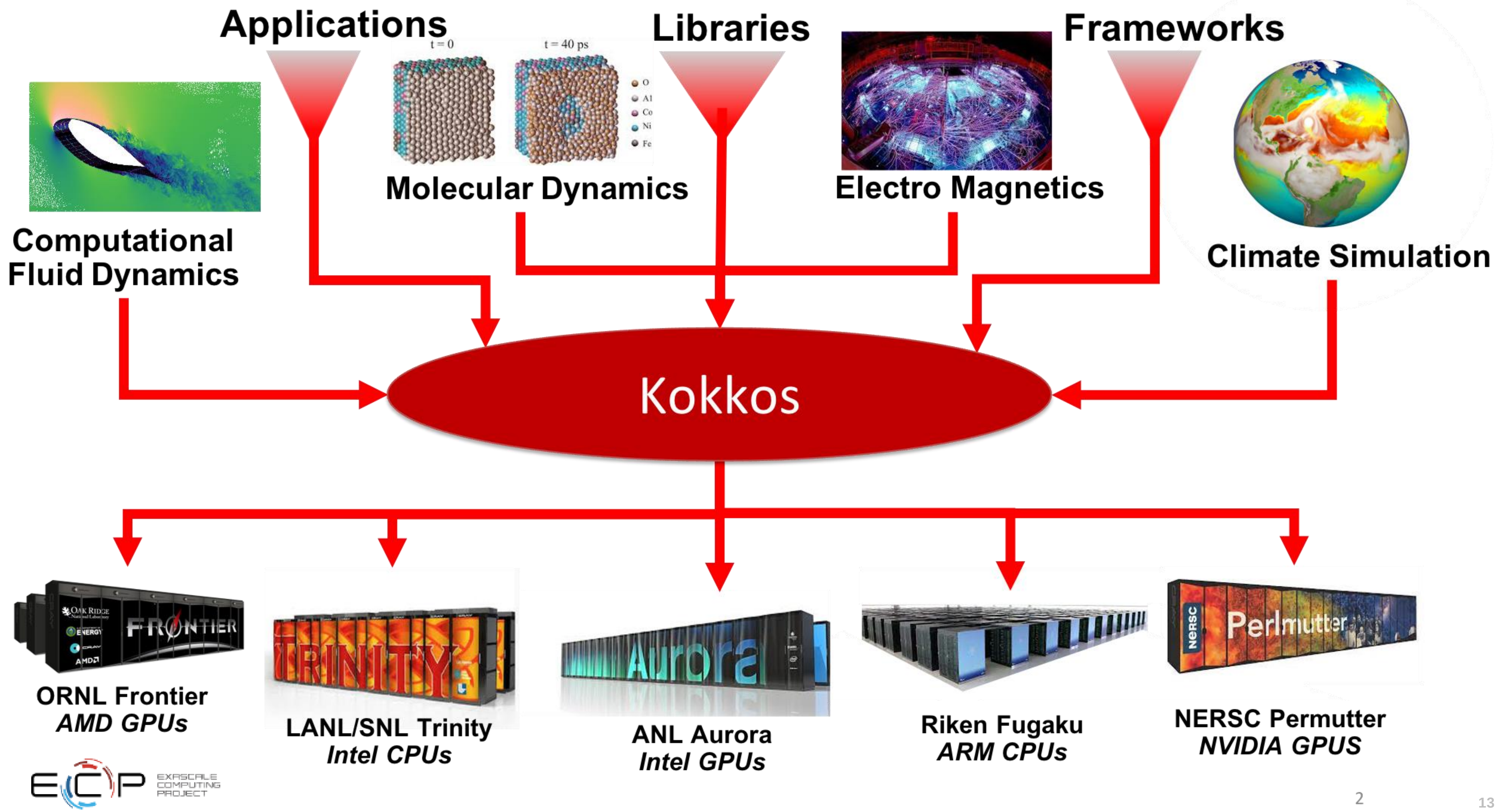
It is Open Source

- Maintained and developed at https://github.com/kokkos

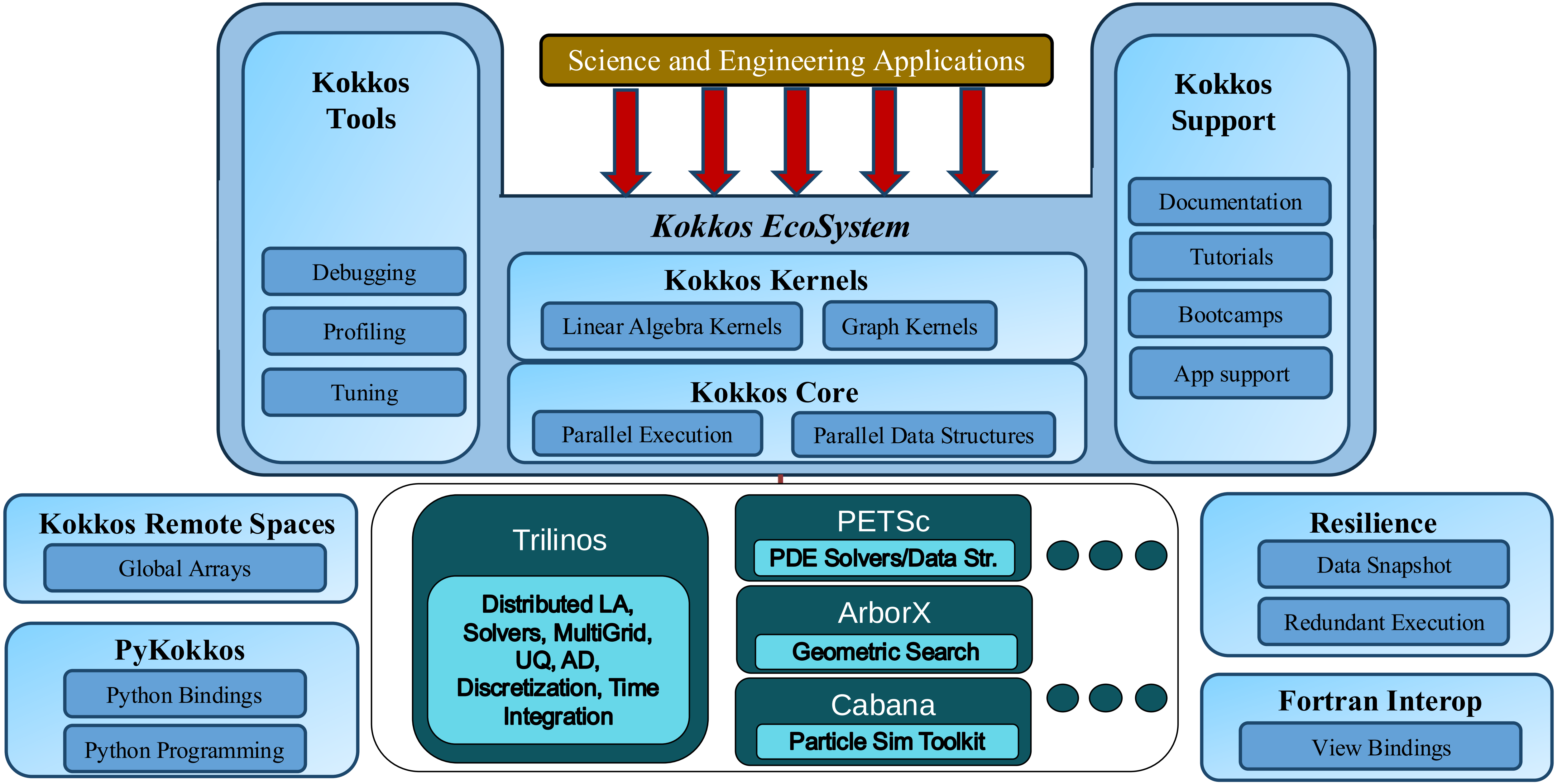
It has many users at wide range of institutions.

Kokkos is NOT just for GPUs!

kokkos


Applications
Libraries
Frameworks
t = 0
t = 40 ps
O
Al
Co
Ni
Fe
Molecular Dynamics
Electro Magnetics
Computational Fluid Dynamics
Climate Simulation
Kokkos
ORNL Frontier
AMD GPUs
LANL/SNL Trinity
Intel CPUs
ANL Aurora
Intel GPUs
Riken Fugaku
ARM CPUs
NERSC Permutter
NVIDIA GPUS
ECP EXASCALE COMPUTING PROJECT
2

The Kokkos Ecosystem



Kokkos Core Abstractions

Kokkos

Data Structures

Memory Spaces ("Where")

- HBM, DDR, Non-Volatile, Scratch

Memory Layouts

- Row/Column-Major, Tiled, Strided

Memory Traits ("How")

- Streaming, Atomic, Restrict

Parallel Execution

Execution Spaces ("Where")

- CPU, GPU, Executor Mechanism

Execution Patterns

- parallel_for/reduce/scan, task-spawn

Execution Policies ("How")

- Range, Team, Task-Graph

CG Solve: The AXPBY

Simple data parallel loop: Kokkos::parallel_for

Easy to express in most programming models

Bandwidth bound

Serial Implementation:

```
void axpby(int n, double* z, double alpha, const double* x,
                        double beta,  const double* y) {
  for(int i=0; i<n; i++)
    z[i] = alpha*x[i] + beta*y[i];
}
```

Parallel Pattern: for loop

String Label: Profiling/Debugging

Execution Policy: do n iterations

Loop Body

Iteration handle: integer index

Kokkos Implementation:

```
void axpby(int n, View<double*> z, double alpha, View<const double*> x,
                        double beta,  View<const double*> y) {
  parallel_for("AXpBY", n, KOKKOS_LAMBDA ( const int i) {
    z(i) = alpha*x(i) + beta*y(i);
  });
}
```

Kokkos Support

The Kokkos Lectures

- 8 lectures covering most aspects of Kokkos
- 15 hours of recordings
- > 500 slides
- >20 exercises

Extensive Wiki

- API Reference
- Programming Guide

Slack as primary direct support

https://kokkos.link/the-lectures

- Module 1: Introduction
  - Introduction, Basic Parallelism, Build System
- Module 2: Views and Spaces
  - Execution and Memory Spaces, Data Layout
- Module 3: Data Structures and MDRangePolicy
  - Tightly Nested Loops, Subviews, ScatterView,…
- Module 4: Hierarchical Parallelism
  - Nested Parallelism, Scratch Pads, Unique Token
- Module 5: Advanced Optimizations
  - Streams, Tasking and SIMD
- Module 6: Language Interoperability
  - Fortran, Python, MPI and PGAS
- Module 7: Tools
  - Profiling, Tuning , Debugging, Static Analysis
- Module 8: Kokkos Kernels
  - Dense LA, Sparse LA, Solvers, Graph Kernels

Kokkos Kernels

BLAS, Sparse and Graph Kernels on top of Kokkos and its View abstraction

- Scalar type agnostic, e.g. works for any types with math operators
- Layout and Memory Space aware

Can call vendor libraries when available

Views contain size and stride information => Interface is simpler

```
// BLAS
int M, N, K, LDA, LDB; double alpha, beta; double *A, *B, *C;
dgemm('N','N', M, N, K, alpha, A, LDA, B, LDB, beta, C, LDC);
```

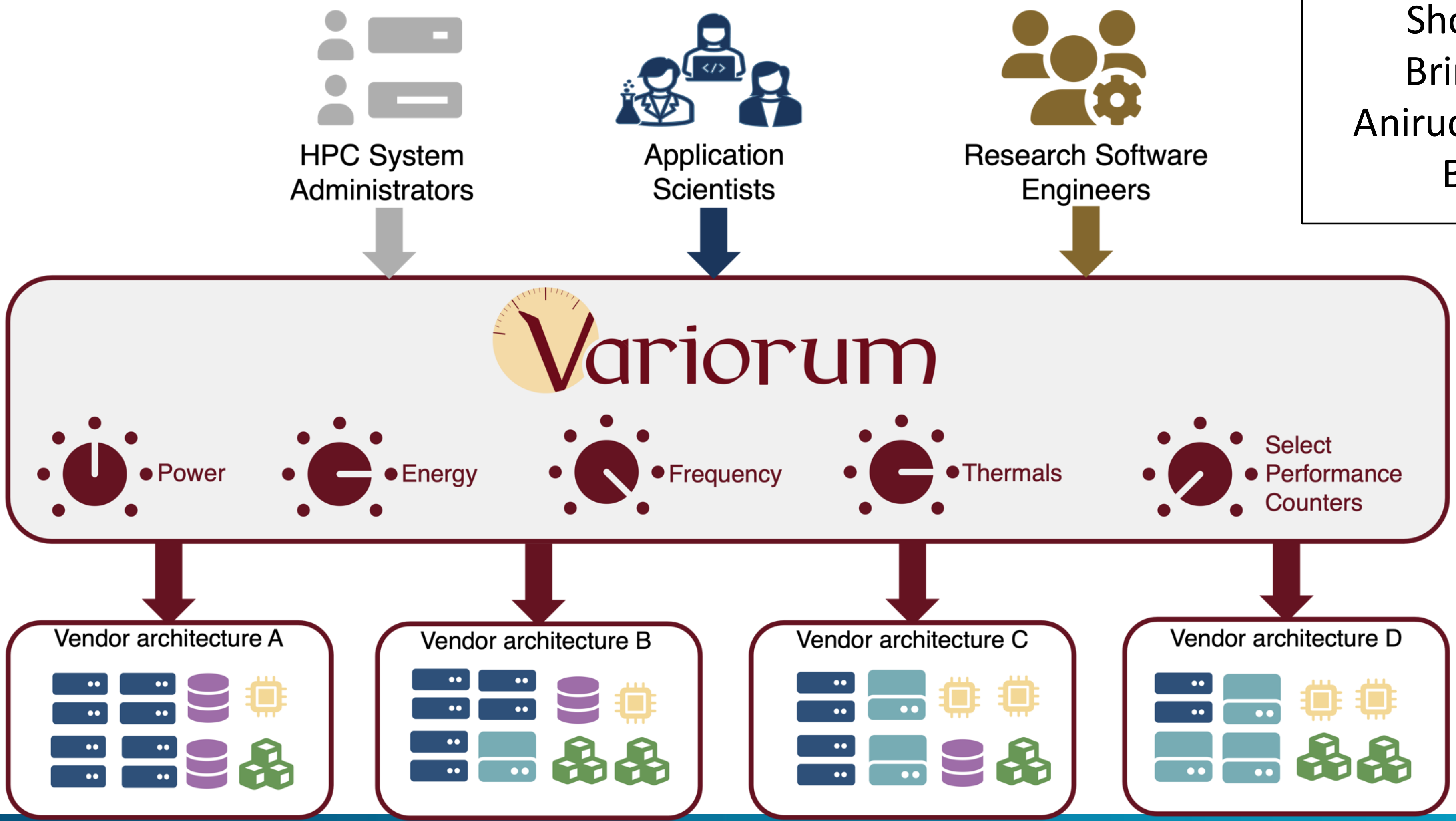
vs.

```
// Kokkos Kernels
double alpha, beta; View<double**> A,B,C;
gemm('N','N', alpha, A, B, beta, C);
```

Development Tools

Variorum provides safe, user-space, vendor neutral access for all users: administrators, application scientists and RSEs

PI: Tapasya Patki
Team: Kathleen Shoga, Stephanie Brink, Eric Green, Aniruddha Marathe, Barry Rountree



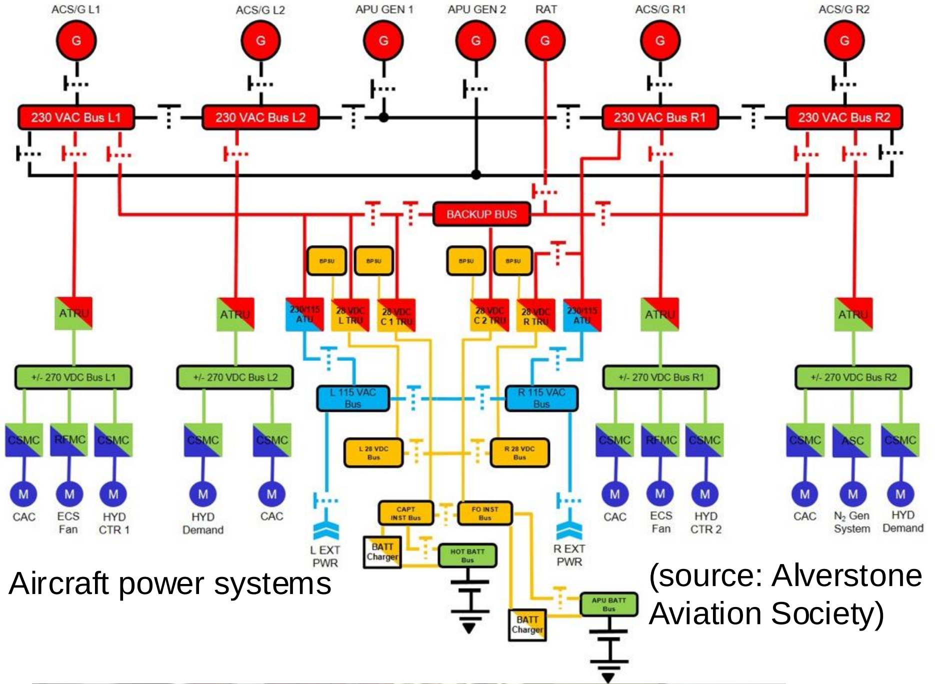
2023 R&D 100 WINNER

EXASCALE COMPUTING PROJECT

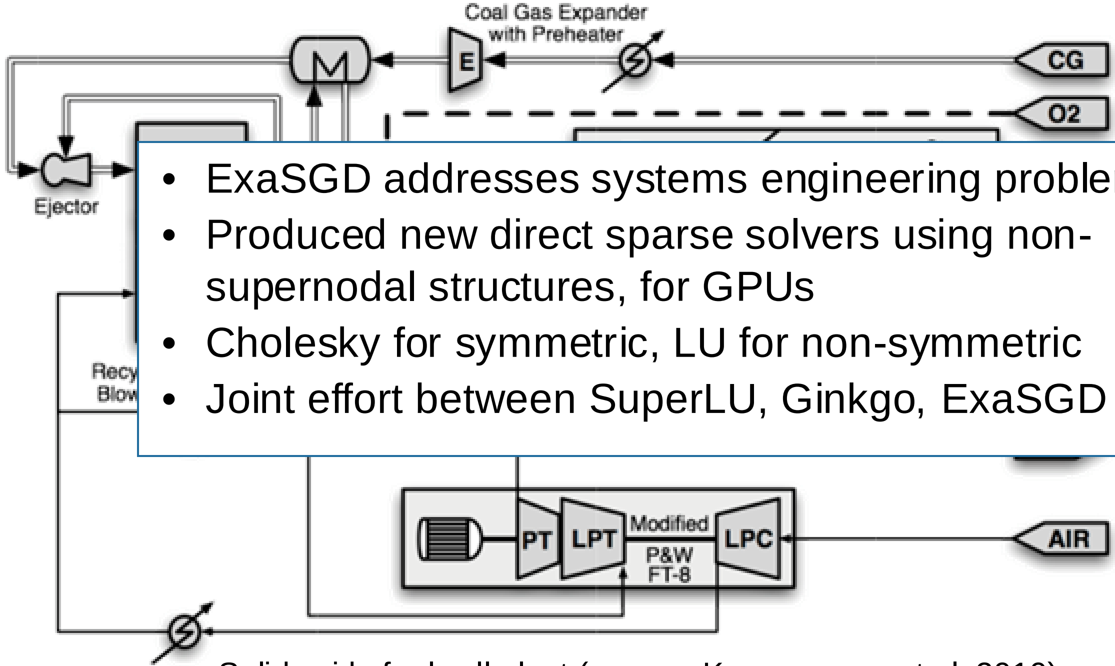
`Variorum`: Vendor-neutral user space library for power management

- Power management capabilities (and their interfaces, domains, latency, capabilities) widely differ from one vendor to the next, needing common interfaces
- `Variorum`: Platform-agnostic vendor-neutral, simple front-facing APIs
  - Evolved from *libmsr*, and designed to target several platforms and architectures
  - Abstract away tedious and chaotic details of low-level knobs
  - Implemented in C, with function pointers to specific target architecture
  - Integration with higher-level power management software through JSON
- Integrated with Flux, GEOPM, LDMS, Kokkos, Caliper and PowerAPI to enable a PowerStack
- Supported on all upcoming exascale systems (Aurora, Frontier, El Capitan) and several other supercomputers: architecture support includes CPU support for ARM, AMD, Intel, IBM; and GPU support for NVIDIA, AMD and Intel.

Systems Engineering Domain



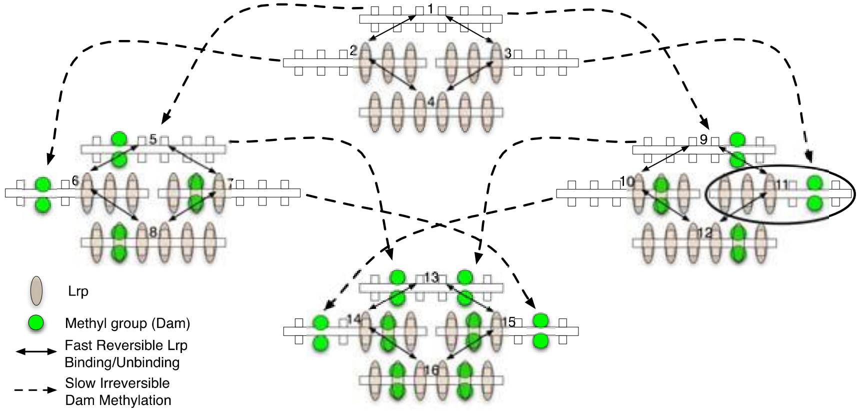
Aircraft power systems (source: Alverstone Aviation Society)



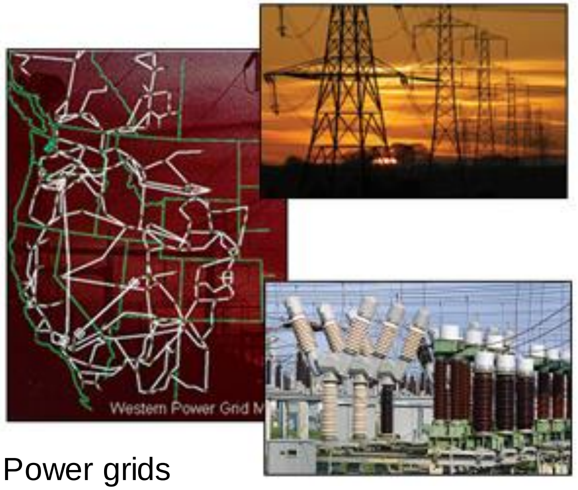
- ExaSGD addresses systems engineering problems
- Produced new direct sparse solvers using non-supernodal structures, for GPUs
- Cholesky for symmetric, LU for non-symmetric
- Joint effort between SuperLU, Ginkgo, ExaSGD teams

Solid oxide fuel cell plant (source: Kameswaran at al. 2010)

Buildings (source: EEB Hub, B661 2014)



Gene regulatory networks (source: Peles et al. 2006)

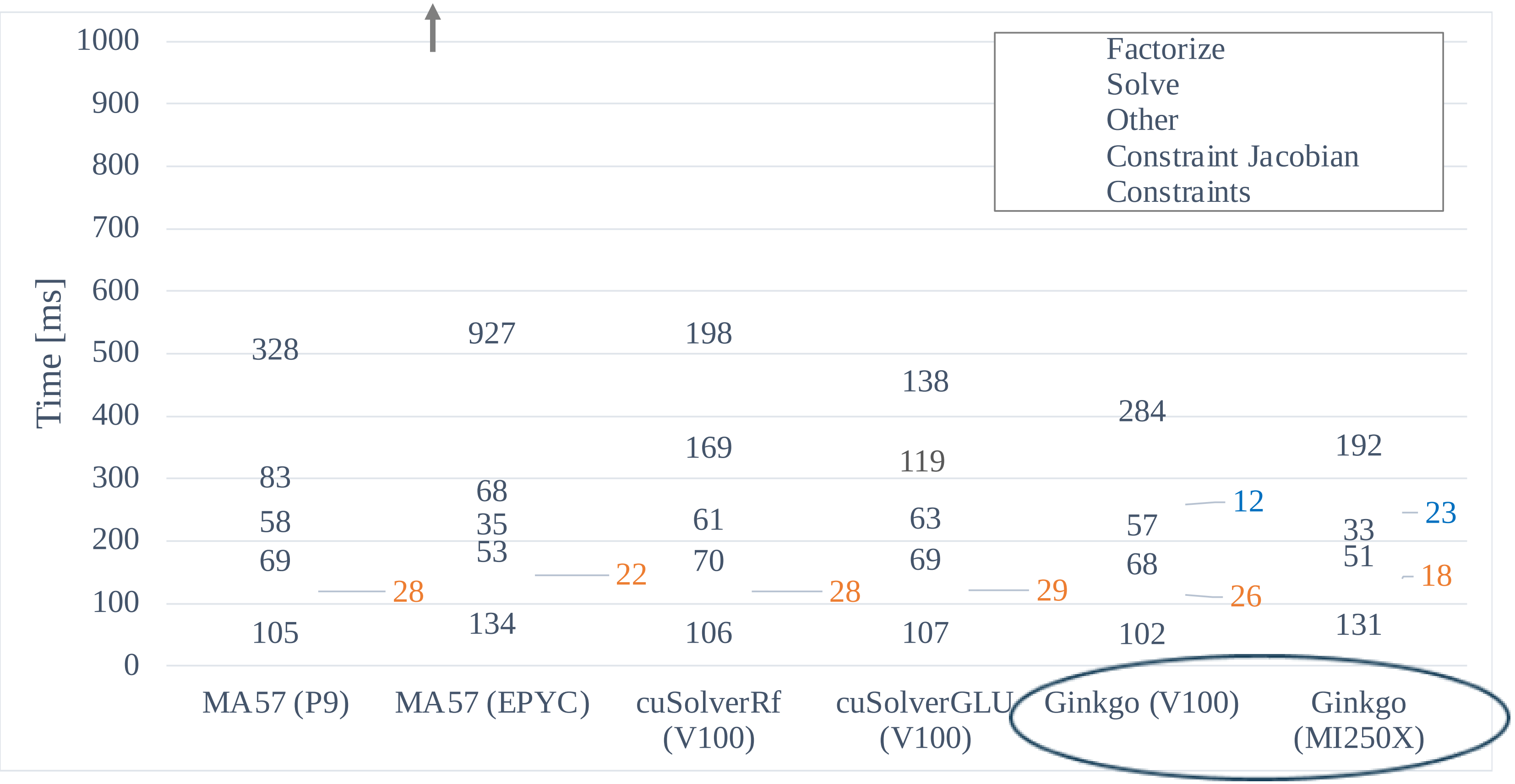


Power grids (source: PNNL)

Linear Solver Performance within Optimization Algorithm

Average per iteration times (including first iteration on CPU)

- Each GPU solution outperforms all CPU baselines
- Ginkgo performance improves on a better GPU
- Iterative refinement configuration affects linear solver performance and optimization solver convergence

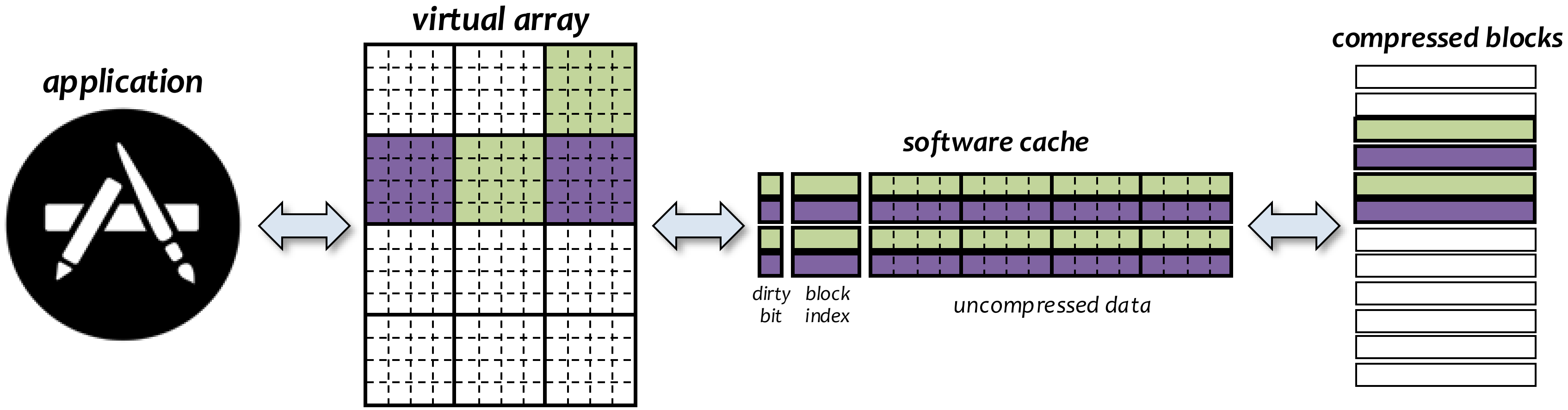


Ginkgo provides the first portable GPU-resident sparse direct linear solver for non-supernodal systems

EXASCALE COMPUTING PROJECT

Data and Visualization

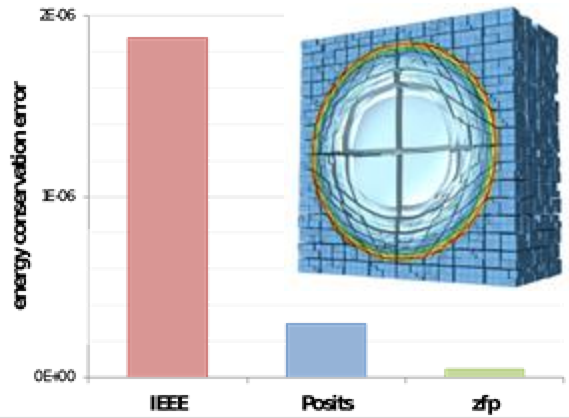
ZFP compressed multidimensional array primitive
Addressing growing gap of ops vs bw vs memory



2023 R&D 100 WINNER

- Fixed-length compressed blocks enable fine-grained read & write **random access**
  - C++ compressed-array classes hide complexity of compression & caching from user
  - User specifies per-array storage footprint in bits/value
- Absolute and relative **error tolerances** supported for offline storage, sequential access
- Fast, hardware friendly, and parallelizable: **150 GB/s throughput** on NVIDIA Volta
- **HPC tool support**: ADIOS, CEED, CONDUIT, HDF, intel, VTK-m

Inline compression boosts solution accuracy by 40x over IEEE in CEED code



Prepared by LLNL under Contract DE-AC52-07NA27344.

And Many More…

- ECP generated a
  - Collection of portable GPU-capable libraries and tools for AMD, Intel, and NVIDIA devices
  - Designed for future adaptation to next-generation highly-concurrent node architectures
  - Foundation for others who will make the transition from CPU to GPU and beyond

ECP Software Technology works on products that apps need now and in the future

Key themes:
- Focus: GPU node architectures and advanced memory & storage technologies
- Create: New high-concurrency, latency tolerant algorithms
- Develop: New portable (Nvidia, Intel, AMD GPUs) software product
- Enable: Access and use via standard APIs

Software categories:
- **Next generation established products:** Widely used HPC products (e.g., MPICH, OpenMPI, PETSc)
- **Robust ~~emerging~~ products:** Address key new requirements (e.g., **Kokkos, RAJA, Ginkgo, Spack**)
- **New products:** Enable exploration of emerging HPC requirements (e.g., **Variorum, zfp**)

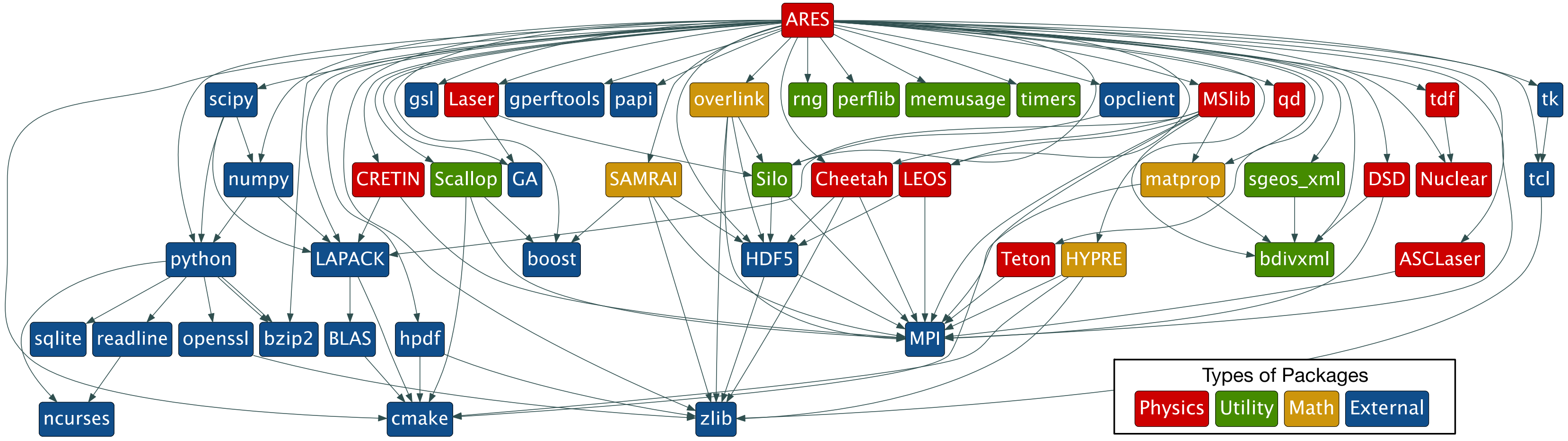
Legacy: A stack that enables performance portable application development on leadership platforms

2023 R&D 100 WINNER

Example Products	Engagement
MPI – Backbone of HPC apps	Explore/develop MPICH and OpenMPI new features & standards
OpenMP/OpenACC –On-node parallelism	Explore/develop new features and standards
C++ Performance Portability Abstractions	Lightweight APIs for compile-time polymorphisms
LLVM/Vendor compilers	Injecting HPC features, testing/feedback to vendors
Perf Tools - PAPI, TAU, HPCToolkit	Explore/develop new features
Math Libraries: BLAS, sparse solvers, etc.	Scalable algorithms and software, critical enabling technologies
IO: HDF5, MPI-IO, ADIOS	Standard and next-gen IO, leveraging non-volatile storage
Viz/Data Analysis	ParaView-related product development, node concurrency

Spack: How we build and test our software

Spack automates the build of LLNL multi-physics codes



	Linux			*BG/Q*	*Cray XE6*
	MVAPICH	*MVAPICH2*	*OpenMPI*	*BG/Q MPI*	*Cray MPI*
GCC	C P L D			C P L D	
Intel 14	C P L D				
Intel 15	C P L D	D			
PGI		D	C P L D		C L D
Clang	C P L D			C L D	
XL				C P L D	

- ARES is a 1, 2, and 3-D radiation hydrodynamics code
- The ARES configuration shown here has **47 dependencies**
- ARES team runs nightly builds for 36 different configurations

  4 code versions:
  - **(C)**urrent Production
  - **(P)**revious Production
  - **(L)**ite
  - **(D)**evelopment

Spack enabled testing on 3 platforms, 6 compilers, and 3 MPI implementations.

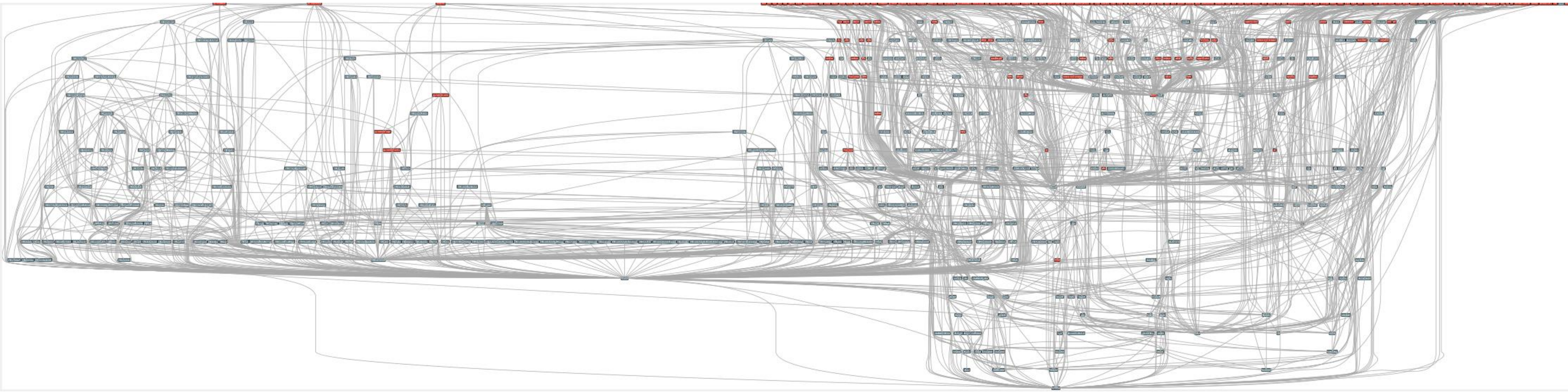
Spack has been ambitious about customizability from the start: "Just say what you want to install"

```
$ spack install mpileaks                                   unconstrained
$ spack install mpileaks@3.3                               @ custom version
$ spack install mpileaks@3.3 %gcc@4.7.3                    % custom compiler
$ spack install mpileaks@3.3 %gcc@4.7.3 +threads           +/- build option
$ spack install mpileaks@3.3 cppflags="-O3 –g3"            set compiler flags
$ spack install mpileaks@3.3 target=cascadelake            set target microarchitecture
$ spack install mpileaks@3.3 ^mpich@3.2 %gcc@4.9.3         ^ dependency constraints
```

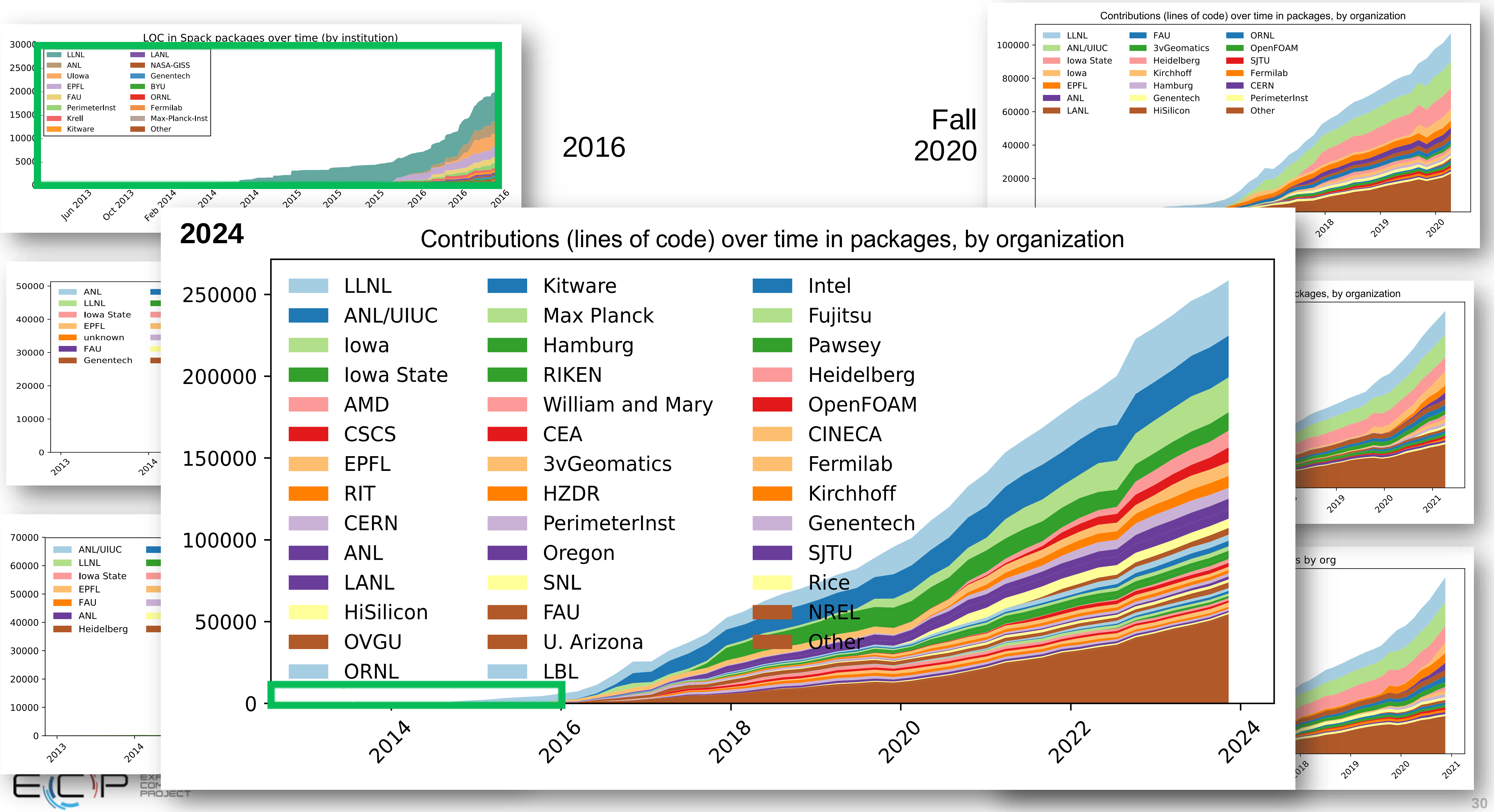
- DOE people never settled for the "off-the-shelf" configuration
  - Always want to customize library versions, build options, flags
- Composition is a fundamental part of Spack's model
  - Allows you to **swap components** like compilers, libraries, GPU runtimes, etc.
- Most distributions build a much more static stack

EXASCALE COMPUTING PROJECT

Now, the E4S stack is more than 600 packages!



– Red boxes are the packages in it (about 100)
– Blue boxes are what ***else*** you need to build it (about 600)
– E4S sits on top of *hundreds* of open source community projects


2016
Fall 2020
2024
LOC in Spack packages over time (by institution)
LLNL
ANL
UIowa
EPFL
FAU
PerimeterInst
Krell
Kitware
LANL
NASA-GISS
Genentech
BYU
ORNL
Fermilab
Max-Planck-Inst
Other
Jun 2013
Oct 2013
Feb 2014
Contributions (lines of code) over time in packages, by organization
LLNL
ANL/UIUC
Iowa State
Iowa
EPFL
ANL
LANL
FAU
3vGeomatics
Heidelberg
Kirchhoff
Hamburg
Genentech
HiSilicon
ORNL
OpenFOAM
SJTU
Fermilab
CERN
PerimeterInst
Other
Contributions (lines of code) over time in packages, by organization
LLNL
ANL/UIUC
Iowa
Iowa State
AMD
CSCS
EPFL
RIT
CERN
ANL
LANL
HiSilicon
OVGU
ORNL
Kitware
Max Planck
Hamburg
RIKEN
William and Mary
CEA
3vGeomatics
HZDR
PerimeterInst
Oregon
SNL
FAU
U. Arizona
LBL
Intel
Fujitsu
Pawsey
Heidelberg
OpenFOAM
CINECA
Fermilab
Kirchhoff
Genentech
SJTU
Rice
NREL
Other
250000
200000
150000
100000
50000
0
2014
2016
2018
2020
2022
2024
ANL
LLNL
Iowa State
EPFL
unknown
FAU
Genentech
ANL/UIUC
Heidelberg

Extreme-scale Scientific Software Stack (E4S)

https://spack.io

Spack lead: Todd Gamblin (LLNL)

- 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**
- Not a commercial product – an open resource for all
- Growing functionality: May 2024: E4S 24.05 – **140+ full release products**
- **One of the most important legacies of ECP**

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	Regular releases Release 24.11 – November	Build caches 10X build time improvement
Turnkey stack A new user experience	https://e4s.io	Post-ECP Strategy PESO & CASS

https://e4s.io

E4S lead: Sameer Shende (U Oregon)

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

Highlights from Sean Ziegeler talk at ECP IAC Meeting

Fast Follower Case Study: DoD Use of E4S

Winter 2024 ECP Industry and Agency Council Meeting

31 January 2024

- Port and use of Spack/E4S without DOE assistance
- Phase 1:
  - Five HPC systems
  - 3-4 compiling systems
  - 2 MPIs
  - 100 products + dependencies
- Phase 2:
  - Deep dive with 16 products
  - Used by DoD, in E4S

Fast Follower Case Study: DoD Use of E4S

Winter 2024 ECP Industry and Agency Council Meeting

31 January 2024

Selected Results: Flux

- **A scheduler (like PBS) but can be launched by the user and manage a set of nodes assigned to the user through the actual system scheduler**
  - Not currently installed on HPCMP systems
- **Install**

  `spack install flux-core %gcc`
- **Exciting option for users with lots of miscellaneous tasks to run**

```
ƒ                                                                    ∞
    nodes [||||||||||||||||||||||||1/1]                  0 pending
    cores [||||                  8/48]                   8 running
     gpus [                       0/0]      0 complete,  0 failed

 size: 1                     uptime: 2.1m                       0.49.0
       JOBID      USER ST NTASKS NNODES RUNTIME NAME
    ƒXSaqhHH  seanzig  R      1      1     59s testjob8
    ƒVcNs9NX  seanzig  R      1      1    1.1m testjob7
    ƒSP1ouiw  seanzig  R      1      1    1.2m testjob6
    ƒQMqpfvF  seanzig  R      1      1    1.3m testjob5
    ƒNK2xE5V  seanzig  R      1      1    1.3m testjob4
    ƒLfZeUXZ  seanzig  R      1      1    1.4m testjob3
    ƒK7h21h9  seanzig  R      1      1    1.4m testjob2
    ƒGbB4zNb  seanzig  R      1      1    1.5m testjob1
```

Highlights from Sean Ziegeler talk at ECP IAC Meeting

- Port and use of Spack/E4S without DOE assistance
- Phase 1:
  - Five HPC systems
  - 3-4 compiling systems
  - 2 MPIs
  - 100 products + dependencies
- Phase 2:
  - Deep dive with 16 products
  - Used by DoD, in E4S

Fast Follower Case Study: DoD Use of E4S

Winter 2024 ECP Industry and Agency Council Meeting

31 January 2024

Selected Results: Flux

Selected Results: TAU

- **Performance analysis tool**
  - Installed on HPCMP systems as part of Computational Science Environment (CSE)
- **Install options:** `+papi +fortran +io +pdt +pthreads +mpi +openmp +craycnl`
- **Potentially useful options:** `+comm +python +shmem`
- **For GPUs:** `+cuda +rocm +rocprofiler +roctracer`
- **Easier to build and configure versus CSE TAU installation:**
  - Configuring for CUDA works and is straightforward
  - Spack can be configured to use a newer PAPI library which has a wider variety of counters
  - Performance measurements from Spack & CSE versions were comparable
  - Tau_exec from Spack installation was more reliable
  - Except: had to modify the PAPI package script to include options not offered by default

Highlights from Sean Ziegeler talk at ECP IAC Meeting

- Port and use of Spack/E4S without DOE assistance
- Phase 1:
  - Five HPC systems
  - 3-4 compiling systems
  - 2 MPIs
  - 100 products + dependencies
- Phase 2:
  - Deep dive with 16 products
  - Used by DoD, in E4S

Highlights from Sean Ziegeler talk at ECP IAC Meeting

- Port and use of Spack/E4S without DOE assistance
- Phase 1:
  - Five HPC systems
  - 3-4 compiling systems
  - 2 MPIs
  - 100 products + dependencies
- Phase 2:
  - Deep dive with 16 products
  - Used by DoD, in E4S

Fast Follower Case Study: DoD Use of E4S

Winter 2024 ECP Industry and Agency Council Meeting

31 January 2024

Selected Results: Flux

Selected Results: TAU

Selected Results: Kokkos & RAJA

- **C++ performance-portability frameworks**
  - Not provided on HPCMP systems
  - Exception is Kokkos as part of LAMMPS
- **Install**

```
spack install --keep-stage kokkos %gcc@12.1.0 +cuda cuda_arch=70
                        +openmp +examples +wrapper +tests    # Cray EX
spack install --keep-stage raja %gcc@12.1.0 +cuda cuda_arch=70
                        +openmp +examples +exercises +tests
```

- **Both built and worked effectively on the test systems**
  - Kokkos-kernels used to test: GPU (V100) runs 2.6x-7.8x faster than CPU (Dual AMD EPYC 7H12, 128 cores)
  - Built-in tests: GPU runs 2.3-50x faster than CPU

Fast Follower Case Study: DoD Use of E4S

Winter 2024 ECP Industry and Agency Council Meeting

31 January 2024

Selected Results: Flux

Selected Results: TAU

Selected Results: Kokkos & RAJA

Selected Results: LAMMPS

- **Classical molecular dynamics code with a focus on materials modeling**
  - Provided on some systems as needed, most frequently installed manually by users due to customization
  - A frequent customization is represented in the spack install flags (green) here
  - Enterprise version has the following LAMMPS options also turned on: colloid, kspace, manybody, molecule, phonon, rigid, and shock, and is an MPI build
- **Install**

```
spack install lammps %oneapi@2022.2.0 -ffmpeg -jpeg -kim -png   # Cray EX
spack install lammps %aocc@3.0.0 -ffmpeg -jpeg -kim -png       # Cray EX
spack install lammps %intel@19.0.4.243 -ffmpeg -jpeg -kim -png  # HPE SGI
```

- **Spack build is 2X faster than HPCMP build for initial tests**
  - Cause of the performance increase is currently unknown, but some possibilities have been ruled out

Highlights from Sean Ziegeler talk at ECP IAC Meeting

- Port and use of Spack/E4S without DOE assistance
- Phase 1:
  - Five HPC systems
  - 3-4 compiling systems
  - 2 MPIs
  - 100 products + dependencies
- Phase 2:
  - Deep dive with 16 products
  - Used by DoD, in E4S

Fast Follower Case Study: DoD Use of E4S

Winter 2024 ECP Industry and Agency Council Meeting

31 January 2024

Selected Results: Flux

Selected Results: TAU

Selected Results: Kokkos & RAJA

Selected Results: LAMMPS

- **Classical molecular dynamics code with a focus on materials modeling**
  - Provided on some systems as needed, most frequently installed manually by users due to customization
  - A frequent customization is represented in the spack install flags (green) here
  - Enterprise version has the following LAMMPS options also turned on: colloid, kspace, manybody, molecule, phonon, rigid, and shock, and is an MPI build
- **Install**

```
spack install lammps %oneapi@2022.2.0 -ffmpeg -jpeg -kim -png   # Cray EX
spack install lammps %aocc@3.0.0 -ffmpeg -jpeg -kim -png       # Cray EX
spack install lammps %intel@19.0.4.243 -ffmpeg -jpeg -kim -png  # HPE SGI
```

- **Spack build is 2X faster than HPCMP build for initial tests**
  - Cause of the performance increase is currently unknown, but some possibilities have been ruled out

Highlights from Sean Ziegeler talk at ECP IAC Meeting

- Port and use of Spack/E4S without DOE assistance
- Phase 1:
  - Five HPC systems
  - 3-4 compiling systems
  - 2 MPIs
  - 100 products + dependencies
- Phase 2:
  - Deep dive with 16 products
  - Used by DoD, in E4S

E4S 24.05 Release

- **140+ HPC-AI packages** on ARM, x86_64, ppc64le platforms, 128K+ binaries in E4S build Cache
- **Growing suite of AI/ML packages**
  - DeepHyper, Google.generativeai (Gemini API), OpenAI (API), TorchBraid, Pandas, Scikit-Learn, JAX, PyTorch, TensorFlow, Horovod, OpenCV, and LBANN with support for GPUs
  - E4S DocPortal updated with AI/ML tools
- OS upgrade for containers: **Ubuntu 22.04 LTS**
- Upgraded
  - **CUDA from version 11 to 12**,
  - **ROCm upgraded from version 5.4 to 5.7.1.**
- **New tools**: Laghos, Glvis, netcdf-fortran, fpm, e4s-cl, and e4s-alc.
- **New applications**: Nek5000, Nekbone, Laghos and previously supported GROMACS, CP2K, Xyce, Quantum Espresso, ExaGo, LAMMPS, WARPX, Dealii, and OpenFOAM.
- **Adaptive Computing's HPC Cloud on-demand data center (ODDC)** web-based platform
  - Multi-user, multi-node ParaTools Pro for E4S images on AWS marketplace with support for aarch64 (Graviton) as well as x86_64 with NVIDIA GPUs with
  - VNC based remote desktop and torque (qsub) for multi-node execution
  - https://adaptivecomputing.com/

CI Testing:
- Frank: Extensive multi-device system
- Includes Grace-Grace, Grace-Hopper
- 7M+ CI builds

E4S 24.11 Release – Look for SC24 announcement

Looking Forward
Post-ECP Software Stewardship and Advancement

8 Software Stewardship Organizations (SSOs)

DOE Office of Advanced Scientific Computing Research (ASCR) Post-ECP Projects

COLABS

Training, workforce development, and building the RSE community

CORSA

Partnering with foundations to provide sustainable pathways for scientific software

FASTMATH

Stewardship, advancement, and integration for math and ML/AI packages

PESO

Stewarding, evolving and integrating a cohesive ecosystem for DOE software

RAPIDS

Stewardship, advancement, and integration for data, visualization and ML/AI packages

S4PST

Stewardship, advancement and engagement for programming systems

STEP

Stewardship, advancement of software tools for understanding performance and behavior

SWAS

Stewardship and project support for scientific workflow software and its community

Announcing CASS

The Consortium for the Advancement of Scientific Software

https://cass.community

CASS Basics

- A newly-formed organization
- Sponsored by DOE Office of Advanced Scientific Computing Research (ASCR)
- Established by DOE Software Stewardship Organizations (SSOs)

CASS Goals

- Forum for SSO collaboration and coordination
- Bigger than the sum of its parts
- Vehicle for advancing the scientific software ecosystem

CASS Status

- Defining governance structure
- Establishing community awareness
- Building a team of teams
- Collaborating on outreach

Software Stewardship Organization (SSO) Basics

- Each SSO represents a specific software ecosystem concern
- **Product SSOs:** Programming systems, performance tools, math packages, data/viz packages
- **Portfolio SSO:** Curating & delivering software stack to the community
- **Community SSOs:** Workforce, partnerships

Engage with CASS

- Review slides June 11-13 CASS Community BOF Days: https://cass.community/bofs
- Visit https://cass.community

Consortium for the Advancement of Scientific Software (CASS)

- **CASS Formation Team**
  - David Bernholdt
  - Phil Carns
  - Lois Curfman McInnes
- **Representatives of Software Stewardship Organizations (SSOs)**
  - **COLABS:** Anshu Dubey, David Bernholdt
  - **CORSA:** Greg Watson, Elaine Raybourn
  - **FASTMath:** Esmond Ng, Todd Munson
  - **PESO: Mike Heroux, Todd Gamblin**
  - **RAPIDS:** Rob Ross, Lenny Oliker
  - **S4PST:** Keita Teranishi, Damian Rouson
  - **STEP:** Terry Jones, Phil Carns
  - **SWAS:** Rafael Ferreira da Silva, Lavanya Ramakrishnan

PESO: Partnering for Scientific Software Ecosystem Stewardship Opportunities

About PESO

- Five-year post-ECP software-ecosystem stewardship and advancement project
- Partnership with CASS
- PESO leads portfolio activities of E4S+Spack curation, testing, delivery

Key PESO goals

- Enable applications to realize benefits of a software ecosystem
- Emphasize software product quality, the continued fostering of software product communities, and the delivery of products, working with CASS

Key PESO Activities

- **Partnerships**: We lead CASS efforts for diverse and inclusive workforce with sustainable career paths. We shepherd BSSw Fellows Program and BSSw.io portal.
- **Services**: We provide services including software product management, integration, and delivery, as well as software quality assurance and security.
- **Products**: We deliver & support products via Spack & E4S, provide porting & testing platforms leveraged across product teams to ensure code stability & portability.

PI: Michael Heroux, Sandia National Laboratories
Collaborating Institutions: ANL, Berkeley, BNL, Kitware, LLNL, LANL, ORNL, PNNL, SNL, SHI, UO
ASCR Program: Software Stewardship and Advancement
ASCR PM: William Spotz
Resources: https://pesoproject.org, https://e4s.io, https://hpsf.io

The PESO Project exists to preserve, sustain, and advance the investments made by the Exascale Computing Project in a robust, versatile, and portable HPC software ecosystem and the people who make the ecosystem effective.

PESO is unique in its organization by composing itself of many members of other software stewardship organizations (SSOs) to best ensure tight integration across the SSOs and support the mission of the Consortium for the Advancement of Scientific Software (CASS).

PESO: Partnering for Scientific Software Ecosystem Stewardship Opportunities

PESO

https://pesoproject.org

Are these challenges familiar? Investing in the Spack/E4S ecosystem can help

- Installing 3rd party libs and tools
- Managing library version updates
- Adopting new libraries and tools
- Managing portability across CPUs/GPUs
- Long build times
- Out-of-date algorithms
- Awareness of latest SW practices
- Opportunities to improve your software

Spack and E4S "Powers of 10" possibilities

100,000+
Lines of code replaced with high-quality libs & tools
Using robust open-source libraries and tools helps you eliminate native source code that is less capable, more fragile, and harder to maintain

100+
Speedup using advanced devices like GPUs
ECP-sponsored applications realize a factor of 100 or more science-speedup by reformulating algorithms and software to exploit GPUs

10,000+
Community members via ecosystem collaborations
Shared experiences using the same ecosystem increases community knowledge base, improved software capabilities, cross-teaming

10+
Reduction in build times via Spack build caches
Spack build caches reduce re-build times by a factor of 10 or more, greatly reducing staff wait times and accelerating debugging and reconfiguration activities

1,000+
Code teams share ecosystem costs & benefits
Pooling investments and making the software ecosystem available everywhere optimizes cost and benefit sharing across many users

1
Source code base for all computing systems
Your single source code base that achieves performance portability, now and in the future, using libraries, tools, and compilers available via Spack and E4S

The High-Performance Software Foundation

We launched the High Performance Software Foundation (HPSF) at ISC24

HIGH PERFORMANCE SOFTWARE FOUNDATION

Scientific Achievements

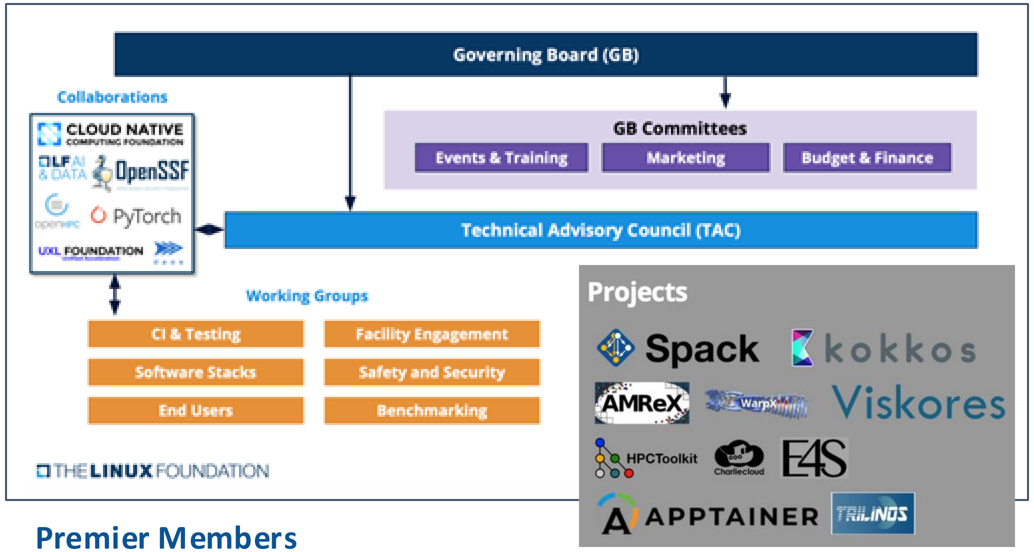
- NNSA led the formation of HPSF over the past 18 months, in close collaboration with DOE/ASCR, industry, and the Linux Foundation
- 15 founding members, 6 initial projects from industry, academia, labs around the world
- HPSF provides open source projects with:
  - A neutral home
  - Collaborations
  - Open governance
  - Funds for project infrastructure, working groups, events, other initiatives

Significance and Impact

- Initial membership raised more funds than expected: **~$1M annual budget**
- ISC kickoff BOF session was standing-room only; generating much excitement
- Expect to grow membership and projects over time
  - 2 NNSA, 2 ASCR, 2 European projects in the pipeline
  - Expecting at least 2 more general members within the year

Approach

- Separate financial (GB), technical (TAC), and project governance
- Interact directly with key projects in the HPC ecosystem
- Grow contributor base through increased adoption, training, events, outreach
- Build a portable, accelerated software stack for HPC and beyond



Premier Members

aws, Hewlett Packard Enterprise, Sandia National Laboratories, Lawrence Livermore National Laboratory

General Members

AMD, intel, NVIDIA, Argonne National Laboratory, Oak Ridge National Laboratory, Los Alamos National Laboratory, kitware, cea

Associate Members

University of Oregon, University of Maryland, सी-डैक CDAC

PI(s)/Facility Lead(s): Todd Gamblin, Christian Trott
Collaborating Institutions: NNSA/ASC, PESO, CORSA, Linux Foundation
DOE Programs: NNSA/ASC, ASCR/NGSST
DOE PMs: Si Hammond, William Spotz, David Rabson, Hal Finkel
Resources: https://hpsf.io

NNSA National Nuclear Security Administration

Some opportunities going forward

- The growing success of generative AI is spurring disruptive changes in scientific software
- GPUs are really AI devices (CPUs are still relevant and can benefit from concurrency too)
- A recent Hyperion survey estimates the high-end AI market is **30 times larger** than traditional HPC
  - Bad news: We are no longer a strong market driver
  - Good news: The HPC market is still large but the AI market is huge. We can benefit greatly, if we want …
- Here are three ways to benefit:

1. **New algorithms adapted to AI devices**
   New computing devices, such as GPUs designed for AI workflows, are suitable for traditional scientific codes but require extensive algorithm and software changes to realize their potential.
2. **AI models to complement or replace analytic models**
   AI inference engines are compelling components, complementing or replacing traditional modeling and simulation approaches.
3. **Leverage AI tools in our research work**
   Generative AI tools are transforming the entire research enterprise, especially software activities. AI tools assist in producing requirements, specifications, designs, source code, tests, and more. These changes are exciting but come with risks.

Summary

- The Exascale Computing Project (ECP):
  - Explored many approaches to performance-portable accelerator-based computing
  - Focused on very high end but most innovations were on the node: suitable for desktop on up
  - De-risked the path for many others: 62 applications from many domains showed a path forward
- In addition to transforming your own software base, you could consider using ECP-developed products:
  - Kokkos, zfp, Spack are a few
  - E4S provides many others
  - HPSF-sponsored products are also emerging
- PESO, CASS, and other software stewardship projects are evolving to meet future needs
- Happy to talk further: mheroux@acm.org

Post-ECP and Final Remarks

DOE/ECP has **learned a lot about producing software contributions** to the HPC community:

- Improved planning, executing, tracking, assessing, integrating, and delivering
- Improved interactions with the broader HPC software and hardware community
- Direct engagement with industry, US agencies, and international collaborators

In post-ECP efforts **we propose to continue and expand these efforts:**

- Further engage with commercial partners to provide a rich, robust software ecosystem
- Evolve a stable, sustainable business model for engaging with agencies and industry
- Engage with cloud providers, software foundations, and others to optimize cost & benefit sharing
- Further the ECP strategy for direct industry and agency engagement

We intend to **realize the potential of the ECP legacy across the HPC community:**

- **Realize the potential of software ecosystems** by leveraging "powers of 10" advantages
- **Increase the trustworthiness, sustainability, and cost effectiveness** of our software in the future
- **Support existing and emerging needs for AI/ML libraries and tools**

We want to work with the HPC community to realize the legacy of ECP, and beyond

- We have many new ways to interact
- Many new opportunities to pursue

Thank you!

ECP EXASCALE COMPUTING PROJECT