Introduction to ECP Software Technologies

1st Workshop on NSF and DOE High Performance Computing Tools

Jonathan Carter (LBNL), Software Technologies Deputy Director

Michael Heroux (SNL), ECP ST Director
Rajeev Thakur (ANL), Programming Models & Runtimes
Jeffrey S. Vetter (ORNL), Development Tools
Lois Curfman McInnes (ANL), Mathematical Libraries
James Ahrens (LANL), Data & Visualization
Todd Munson (ANL), Software Ecosystem & Delivery
J. Robert Neely (LLNL), NNSA Software

NNSA National Nuclear Security Administration

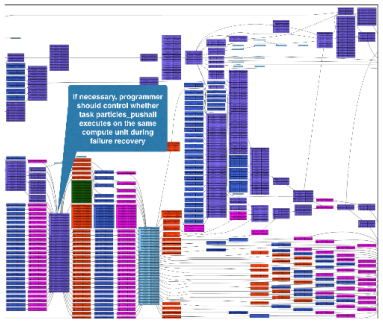
U.S. DEPARTMENT OF ENERGY | Office of Science

ECP **Software Technology (ST)**

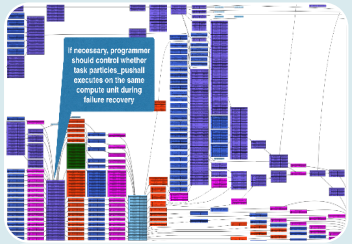
Goal

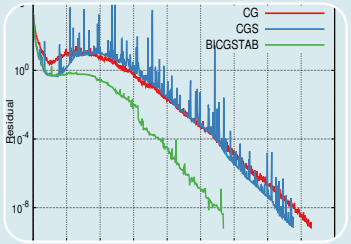
Build a comprehensive, coherent software stack that enables application developers to productively develop highly parallel applications that effectively target diverse exascale architectures

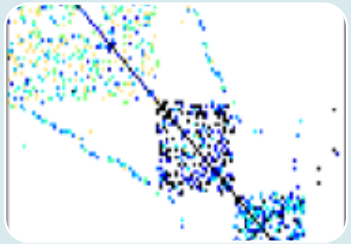
- Prepare SW stack for scalability with massive on-node parallelism
- Extend existing capabilities when possible, develop new when not
- Guide, and complement, and integrate with vendor efforts
- Develop and deliver high-quality and robust software products

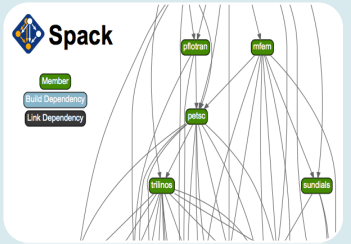


ECP software technologies are a fundamental underpinning in delivering on DOE's exascale mission









Programming Models & Runtimes

- Enhance & prepare OpenMP and MPI programming models (hybrid programming models, deep memory copies) for exascale
- 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 & 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

- 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
- Open source NNSA Software projects
- Subject to the same planning, reporting and review processes

ST leadership team

Rajeev Thakur, Programming Models and Runtimes (2.3.1)
Rajeev is a senior computer scientist at ANL and most recently led the ECP Software Technology focus area. His research interests are in parallel programming models, runtime systems, communication libraries, and scalable parallel I/O. He has been involved in the development of open source software for large-scale HPC systems for over 20 years.

Jeff Vetter, Development Tools (2.3.2)
Jeff is a computer scientist at ORNL, where he leads the Future Technologies Group. He has been involved in research and development of architectures and software for emerging technologies, such as heterogeneous computing and nonvolatile memory, for HPC for over 15 years.

Lois Curfman McInnes, Math Libraries (2.3.3)
Lois is a senior computational scientist in the Math and Computer Science Division of ANL. She has over 20 years of experience in high-performance numerical software, including development of PETSc and leadership of multi-institutional work toward sustainable scientific software ecosystems.

Jim Ahrens, Data and Visualization (2.3.4)
Jim is a senior research scientist at the Los Alamos National Laboratory (LANL) and an expert in data science at scale. He started and actively contributes to many open-source data science packages including ParaView and Cinema.

Todd Munson, Software Ecosystem and Delivery (2.3.5)
Todd is a computational scientist in the Math and Computer Science Division of ANL. He has nearly 20 years of experience in high-performance numerical software, including development of PETSc/TAO and project management leadership in the ECP CODAR project.

Rob Neely, NNSA ST (2.3.6)
Rob is an Associate Division Leader in the Center for Applied Scientific Computing (CASC) at LLNL, chair of the Weapons Simulation & Computing Research Council, and lead for the Sierra Center of Excellence. His efforts span applications, CS research, platforms, and vendor interactions.

ECP ST Subprojects

- WBS
- Name
- PIs
- Project Managers (PMs)

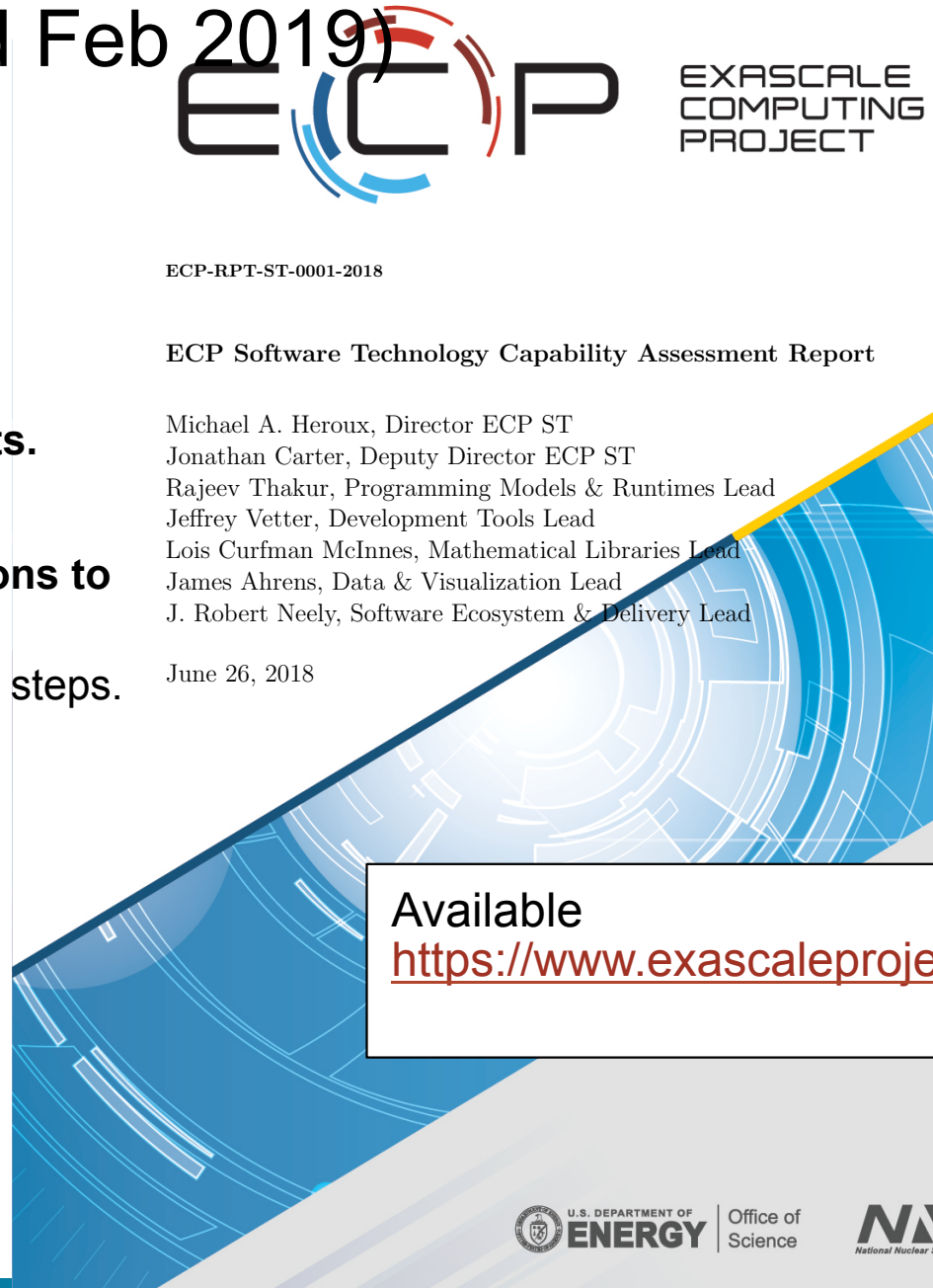
ECP ST Stats

- 33 L4 subprojects

WBS	WBS Name	CAM/PI	PM
2.3	**Software Technology**	**Heroux, Mike, Carter, J.**	
2.3.1	**Programming Models & Runtimes**	**Thakur, Rajeev**	
2.3.1.01	PMR SDK	Shende, Sameer	Shende, Sameer
2.3.1.07	Exascale MPI (MPICH)	Balaji, Pavan	Bayyapu, Neelima
2.3.1.08	Legion	McCormick, Pat	McCormick, Pat
2.3.1.09	PaRSEC	Bosilca, George	Carr, Earl
2.3.1.14	Pagoda: UPC++/GASNet for Lightweight Communication and Global Address Space Support	Baden, Scott	Hargrove, Paul (and PI)
2.3.1.16	SICM	Lang, Michael	Vigil, Brittney
2.3.1.17	OMPI-X	Bernholdt, David	TBD PMA Through ORNL PMO
2.3.1.18	RAJA/Kokkos	Trott, Christian Robert	Trott, Christian
2.3.1.19	Argo: Low-level resource management for the OS and runtime	Beckman, Pete	Gupta, Rinku
2.3.2	**Development Tools**	**Vetter, Jeff**	
2.3.2.01	Development Tools Software Development Kit	Miller, Barton	Tim Haines
2.3.2.06	Exa-PAPI++: The Exascale Performance Application Programming Interface with Modern C++	Dongarra, Jack	Jagode, Heike
2.3.2.08	Extending HPCToolkit to Measure and Analyze Code Performance on Exascale Platforms	Mellor-Crummey, John	Mellor-Crummey, John
2.3.2.10	PROTEAS-TUNE	Vetter, Jeff	Glassbrook, Dick
2.3.2.11	SOLLVE: Scaling OpenMP with LLVm for Exascale	Chapman, Barbara	Kong, Martin
2.3.2.12	FLANG	McCormick, Pat	Perry-Holby, Alexis
2.3.3	**Mathematical Libraries**	**McInnes, Lois**	
2.3.3.01	Extreme-scale Scientific xSDK for ECP	Yang, Ulrike	Yang, Ulrike
2.3.3.06	Preparing PETSc/TAO for Exascale	Smith, Barry	Munson, Todd
2.3.3.07	STRUMPACK/SuperLU/FFTX: sparse direct solvers, preconditioners, and FFT libraries	Li, Xiaoye	Li, Xiaoye
2.3.3.12	Enabling Time Integrators for Exascale Through SUNDIALS/ Hypre	Woodward, Carol	Woodward, Carol
2.3.3.13	CLOVER: Computational Libraries Optimized Via Exascale Research	Dongarra, Jack	Carr, Earl
2.3.3.14	ALExa: Accelerated Libraries for Exascale/ForTrilinos	Turner, John	TBD PMA Through ORNL PMO
2.3.4	**Data and Visualization**	**Ahrens, James**	
2.3.4.01	Data and Visualization Software Development Kit	Atkins, Chuck	Atkins, Chuck
2.3.4.09	ADIOS Framework for Scientific Data on Exascale Systems	Klasky, Scott	TBD PMA Through ORNL PMO
2.3.4.10	DataLib: Data Libraries and Services Enabling Exascale Science	Ross, Rob	Ross, Rob
2.3.4.13	ECP/VTK-m	Moreland, Kenneth	Moreland, Kenneth
2.3.4.14	VeloC: Very Low Overhead Transparent Multilevel Checkpoint/Restart/Sz	Cappello, Franck	Ehling, Scott
2.3.4.15	ExaIO - Delivering Efficient Parallel I/O on Exascale Computing Systems with HDF5 and Unify	Byna, Suren	Bagha, Neelam
2.3.4.16	ALPINE: Algorithms and Infrastructure for In Situ Visualization and Analysis/ZFP	Ahrens, James	Turton, Terry
2.3.5	**Software Ecosystem and Delivery**	**Munson, Todd**	
2.3.5.01	Software Ecosystem and Delivery Software Development Kit	Willenbring, James M	Willenbring, James M
2.3.5.09	SW Packaging Technologies	Gamblin, Todd	Gamblin, Todd
2.3.6	**NNSA ST**	**Neely, Rob**	
2.3.6.01	LANL ATDM	Mike Lang	Vandenbusch, Tanya Marie
2.3.6.02	LLNL ATDM	Becky Springmeyer	Gamblin, Todd
2.3.6.03	SNL ATDM	Jim Stewart	Trujillo, Gabrielle

ECP Software Technology Capability Assessment Report (2nd Version Released Feb 2019)

- Document elements:
  1. Executive summary
  2. Project Description
     - **SDKs, Delivery strategy, project restructuring, new projects.**
     - Technical areas overview.
     - **Deliverables: Products, Standards committees, contributions to external products.**
     - Project two-pages: with description, activities, challenges, next steps.
- 212 pages (191 public), updated twice a year.



Available
https://www.exascaleproject.org

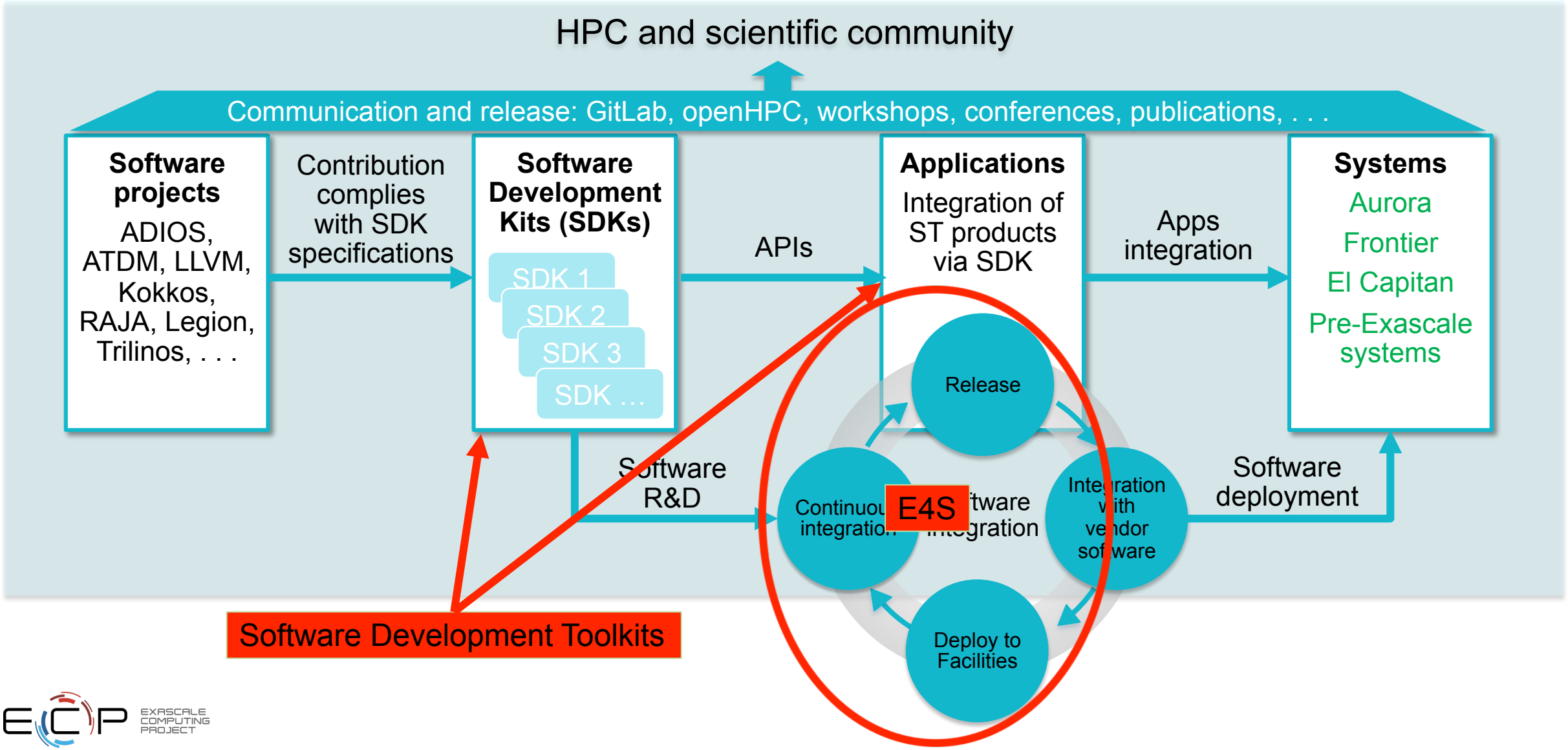
ECP ST Product Dictionary List

Widely-recognized product names.

- MPI – MPICH, OpenMP
- C++/C/Fortran – LLVM
- Fortran – Flang
- hypre – hypre

ADIOS
AML
Ascent
BLAS
C
C++
Caliper
Catalyst
CHAI
Cinema
CUDA
Darshan
DTK
Dyninst
E4S
FFT
FleCSI
Flux
Fortran
GASNet
Ginkgo
HDF5
HPCToolkit
hypre
Kokkos
KokkosKernels
LAPACK
Legion
libEnsemble
MarFS
MFEM
MPI
OpenACC
OpenCL
OpenMP
PAPI
Papyrus
Paraview
PaRSEC
PETSc/TAO
PnetCDF
PowerStack
RAJA
MPI-IO
ScaLAPACK
SCR
SICM
Spack
SPOT
STRUMPACK
SUNDIALS
SuperLU
SYCL
SZ
TASMANIAN
TAU
Trilinos
UMap
Umpire
Unify
UPC++
VeloC
Visit
VTK-m
xSDK
ZFP

ECP's Software and Application Delivery and Deployment has two principal components: SDKs and E4S



Software Development Kits (SDKs): Key delivery vehicle for ECP

A collection of related software products (packages) where coordination across package teams improves usability and practices, and foster community growth among teams that develop similar and complementary capabilities

- **Domain scope**
  Collection makes functional sense
- **Interaction model**
  How packages interact; compatible, complementary, interoperable
- **Community policies**
  Value statements; serve as criteria for membership
- **Meta-infrastructure**
  Invokes build of all packages (Spack), shared test suites
- **Coordinated plans**
  Inter-package planning. Augments autonomous package planning
- **Community outreach**
  Coordinated, combined tutorials, documentation, best practices

ECP ST SDKs: Grouping similar products for collaboration & usability

Programming Models & Runtimes Core

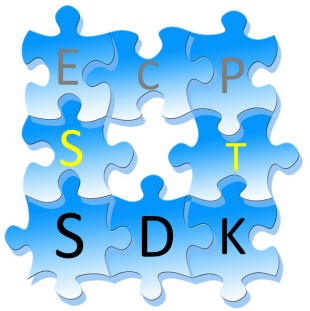
Tools & Technologies

Compilers & Support

Math Libraries (xSDK)

Viz Analysis and Reduction

Data mgmt., I/O Services & Checkpoint/ Restart



"Unity in essentials, otherwise diversity"

Extreme-scale Scientific Software Stack (E4S)

A Spack-based distribution of ECP ST products and related and dependent software tested for interoperability and portability to multiple architectures
Lead: Sameer Shende, University of Oregon

- Provides distinction between SDK usability / general quality / community and deployment / testing goals
- Will leverage and enhance SDK interoperability thrust
- Releases:
  - Oct: E4S 0.1: 24 full, 24 partial release products
  - Jan: E4S 0.2: 37 full, 10 partial release products
- Current primary focus: Facilities deployment
- Ideal mechanism for collaborations with other institutions, agencies, countries

http://e4s.io

Current E4S Release and Installed Packages

- Adios
- Bolt
- Caliper
- Darshan
- Gasnet
- GEOPM
- GlobalArrays
- Gotcha
- HDF5
- HPCToolkit
- Hypre
- Jupyter
- Kokkos
- Legion
- Libquo
- Magma
- MFEM
- MPICH
- OpenMPI
- PAPI
- Papyrus
- Parallel netCDF
- ParaView
- PETSc/TAO
- Program Database Toolkit (PDT)
- Qthreads
- Raja
- SCR
- Spack
- Strumpack
- Sundials
- SuperLU
- Swift/T
- SZ
- Tasmanian
- TAU
- Trilinos
- VTKm
- Umpire

Packages installed using Spack

- UnifyCR
- Veloc
- xSDK
- Zfp

Plan is to release all ST products through E4S

ECP ST staff contribute to ISO and *de facto* standards groups:
Assuring sustainability through standards

ST contributes requirements, analysis, design, prototype and reference implementations to vendor & community products that address ECP mission needs.

- **ECP ST Product contributions:** ST efforts provide fundamental software contributions to all of these projects.
  - Examples: MPICH, OpenMPI, SOLLVE (OpenMP, LLVM)
  - Others: Kokkos/RAJA (C++)
- **MPI/OpenMP:** Several key leadership positions.
- Heavy involvement in all aspects.
- **C++:** Getting HPC requirements considered, contributing working code.
- **Fortran:** Flang front end for LLVM.
- ***De facto:*** Specific HPC efforts.

Standards Effort	ECP ST Committee Participants
MPI Forum	15
OpenMP	15
BLAS	6
C++	4
Fortran	4
OpenACC	3
LLVM	2
PowerAPI	1

How ST products are delivered

- From source: Direct installation from open source repos.
- In collaboration with HI:
  - Strong collaboration focused on deployment.
  - To facilities, in apps, to vendors
- Through standards: MPI, OpenMP, LLVM.
- Through vendors: Integrated into vendor stacks.
- Via E4S/SDKs: Turnkey multi-product builds.
- In Containers: Complete pre-built environments.

Questions?