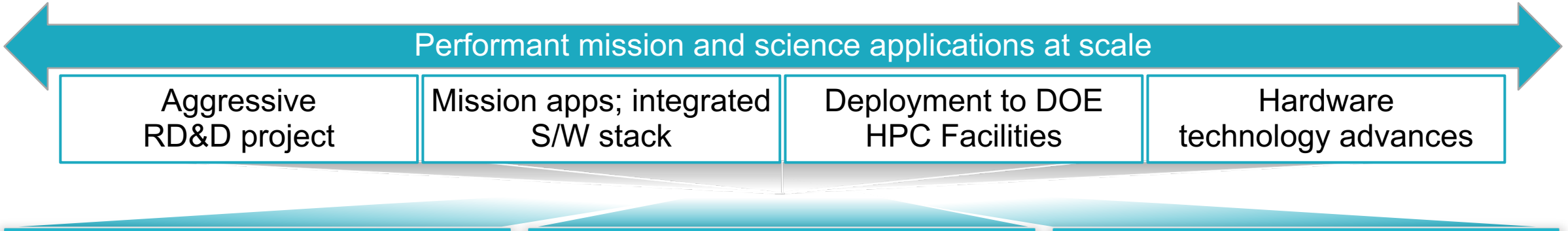
ECP Update

Douglas B. Kothe (ORNL), ECP Director
Lori Diachin (LLNL), ECP Deputy Director

DOE ASCR Advisory Committee (ASCAC) Meeting
March 29, 2022
Video Conference

ECP's Technical Focus Areas

Providing the necessary components to meet national goals



Application Development (AD)

Develop and enhance the predictive capability of applications critical to DOE

24 applications
National security, energy, Earth systems, economic security, materials, data

6 Co-Design Centers
Machine learning, graph analytics, mesh refinement, PDE discretization, particles, online data analytics

Software Technology (ST)

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

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

Hardware and Integration (HI)

Integrated *continuous testing & delivery* of ECP products on targeted systems at leading DOE HPC facilities

6 US HPC vendors
focused on exascale node and system design; application integration and software deployment to Facilities

ECP: From Baseline to Project Completion

FY20	FY21	FY22	FY23	FY24
• **CD-2/3 review and approval** • Did PathForward deliver? • Are AD and ST performance and integration projections on track? • Access to Aurora and Frontier early hardware	• **Status Independent Project Review (IPR)** ✓ AD application projections & ST capability assessment ✓ Assess Path-Forward impact ✓ Access to Frontier test and development system (TDS)	✓ **Status IPR** ✓ AD and ST readiness assessment • Access to Frontier full system • Access to El Capitan early hardware	• Access to Aurora TDS • Access to Aurora full system • Demonstrate and deliver KPP completion evidence with priority on Frontier • **Project completion (plan date)**	• Access to El Capitan full system (secure) • **CD-4 completion (DOE date)**

ECP Just Completed its Annual Independent Project Review (IPR)

Mar 15-17, 2022

1. Is the project making adequate progress to address the recommendations and comments from the February 2021 Independent Project Review? **YES**
2. Is the project on track for delivering on key milestones and meeting the ECP KPPs? **YES**
3. Has the completion of KPP-4 been adequately documented? **YES**
4. Are risks, including any associated with COVID-19, adequately identified and managed with appropriate responses for this phase of the project? Is there adequate contingency to successfully complete the project? **YES**
5. Are the critical internal and external dependencies being adequately managed and tracked? **YES**
6. Is the overall project being managed properly? **YES**

ECP's 2021 Review Recommendations: Proactive and Helpful

Motivated us to respond with success-oriented plans

Recommendation	Response
Identify long term options for supporting and evolving the software ecosystem developed and used throughout the ECP project. A starting point for this could be the ASCR/ECP Transition Report issued by ASCR's Advanced Scientific Computing Advisory Committee in April 2020.	✓ Decadal strategic vision and plan (to mid 2030s) for E4S formulated and documented. ✓ Regular interactions with ASCR Task Force on Software Stewardship in response to ASCAC Oct 2020 report on *Transitioning ASCR after ECP.* ✓ Establish Leadership Scientific Software (LSSw) portal to build community & understanding around development and sustainable delivery of leadership scientific software (lssw.io). ✓ Documented response to ASCR RFI on the *Stewardship of Software for Scientific and High-Performance Computing.* ✓ Monthly "Town Hall" meetings (5 held to date) to engage broader scientific software community on ECP software efforts and how to improve software sustainability in the future.
Work with DOE to develop a Contingency buy-down plan	✓ New *Contingency Management Plan* documented that encompasses contingency strategy, trigger events and dates, and overall scope of contingency: cost (risks, uncertainties), scope, schedule, standing army.
Demonstrate progress on, and communicate, a management plan for the end of the project including people, software ecosystem, and management practices.	✓ Formulated overall plan in new ECP *End of Project Plan* document. ✓ New companion ECP documents - *IT Tool Handbook* and *Contingency Management Plan* – directly support end of the project. ✓ Existing *Transition to Operations and Research* document. ✓ Plan is consistent with overall Sep 2020 recommendations given by the ASCR ASCAC subcommittee on ECP Transition.

Contingency Management Plan: Strategy

Aggressively yet prudently apply cost contingency to mitigate risks, exploit opportunities, and support a possible early completion date extension

Goals

- Conclude before the required formal end of project (CD-4) date with project KPPs well past threshold
- Minimal remaining cost contingency @ project completion. Any cost contingency funds must apply to appropriate ECP scope and in a timely fashion for ECP staff to execute

Priorities

1. Maximize probability of achieving threshold KPPs with minimal reduction of baseline scope
2. Maximize probability of achieving objective KPPs with full baseline scope
3. Aggressively address prioritized opportunities

Assumptions

- KPP threshold success; adding staff hard; urgency for contingency actions (e.g., cannot buy time), team makeup constant regardless of scope

Opportunities considered (if/when trigger events occur)

Usability, Portability, Sustainability

Identify and implement strategies to promote usability, portability, sustainability, flexibility, and agility of the suite of tools, codes, products

Expand Outreach and Transition of Technologies to US Agencies, Industry, and Academia

Workshops, targeted tutorials, BoFs, or other formalized engagements to inform potential stakeholders & users on ECP apps & software products

Readying for Emerging Technologies in the Next 5–10 Years

Preparing for computational and data science technology milestones likely to be realized over the next decade (right-sizing E4S for edge, expanded workflows, co-designed hardware targeting motifs, etc.)

Increase Stakeholder Engagement

Demonstrate capabilities of codes and products to targeted agencies and program offices

Planning for a Successful Conclusion: Sustainability and Adoption

ECP's End of Project Management Plan formulated and documented

Successful project completion requires that ECP act now on key tasks

People

Maintain teams of researchers focused on demonstrating challenge problems on exascale platform

- Early finish date extension
- Opportunity-based funding
- Collaboration with senior leadership at participating institutions

Software Ecosystem Development & Sustainability

Develop and implement plans for the transition of ECP apps & software products to the broader community

- Software artifacts developed by ECP: how they are currently managed and deployed
- Ecosystem management plan through ECP completion
- Activities to maximize the potential for long-term sustainment of the ECP software ecosystem post-ECP
- ECP's evolving vision for software sustainability documented in response to a Feb 2021 IPR recommendation

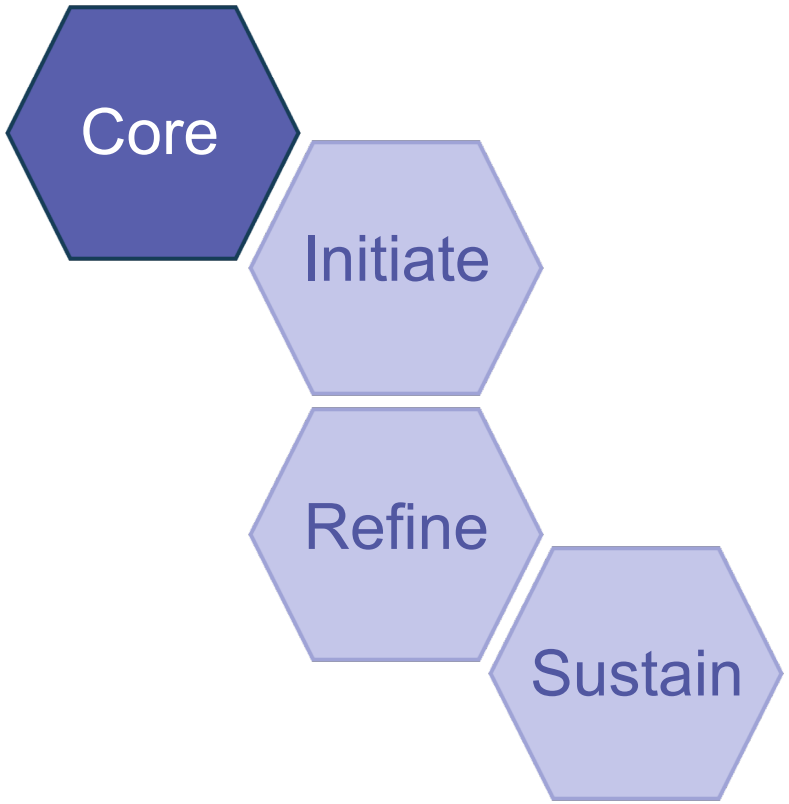
Management Tools and Practices

Document the legacy of tools and project management practices implemented to manage RD&D projects

- Tools used – now covered in detail in new ECP *IT Tool Handbook*
- Effective practices in *Performance Measurement Plan, Project Controls Plan, Risk Management Plan, Communications Outreach Plan*
- Project closeout planning

An Evolving Vision for Software Sustainability

A software center with core efforts + “sprint-like” campaigns



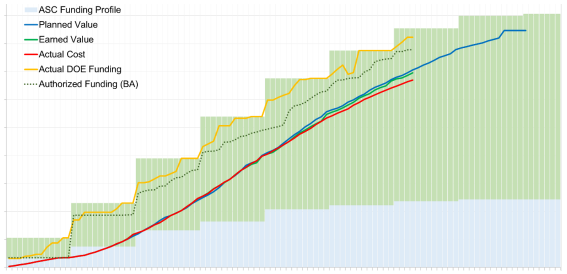
Software Campaigns

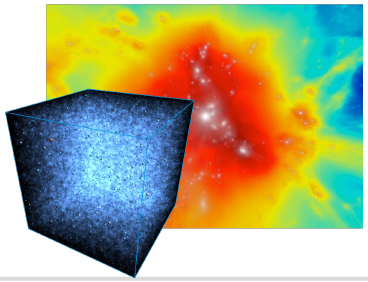
Starting point: ECP’s Extreme-Scale Scientific Software Stack, **E4S**, a Spack-based distribution of software tested for interoperability and portability to multiple architectures (e4s.io)

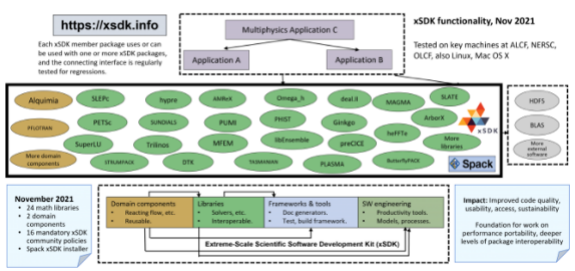
	Sprint 1 FY 2024-26	Sprint 2 FY 2027-29	Sprint 3 FY 2030-32
Next phase core SW	✓	✓	✓
Establish AI/ML SDK	✓		
Next phase AI/ML		✓	✓
Scope Edge SDK	✓		
Establish Edge SDK		✓	
Next phase Edge			✓
Scope Quantum SDK		✓	
Establish Quantum SDK			✓
Contingency	✓	✓	✓

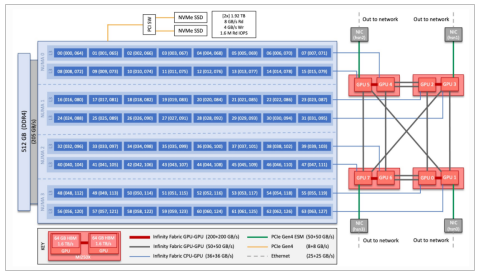
A component of ECP’s response to a Feb 2021 IPR Recommendation: *Identify long term options for supporting and evolving the software ecosystem developed and used throughout the ECP project.* Note, any software sustainability activities planned for post-ECP are not in ECP’s scope.

Summary of ECP's Key Efforts (by Focus Area) this Past Year









Project Management

- Navigation and sustainment of ECP funding through ORNL procurement system upgrade
- COVID-19 Impact Survey administered to ECP subproject teams
- Analysis & impact assessment of 1Q extension of ECP early finish date with opportunity plan
- Project performance assessment and reporting, aided by numerous live dashboards that query & analyze real-time data

Application Development

- Porting, perf analysis on exascale early access systems (EAS). Initial access to Frontier TDS, full system Perlmutter
- Progression from CPU to CPU / GPU to Multi-CPU / Multi-GPU to Diverse CPU / Multi-GPU to GPU Resident
- Continuous stakeholder engagement
- FY21 reviews & FY20/21 key milestone assessment reports
- KPP-1/2 verification process: vetted contracts for quantified completion criteria

Software Technology

- Porting, perf analysis on exascale EAS. Initial access to Frontier TDS, full system Perlmutter
- 5 E4S releases: Spack-based distribution of GPU based images for Intel, AMD, NVIDIA. Deployed to NERSC, OLCF
- FY21 reviews & FY20/21 key milestone CAR reports
- KPP-3 verification process: agreed upon targets for quantified capability integrations
- Sustainability efforts: LSSw web portal, town hall meetings, response to ASCR RFI, ASCR Task Force engagement

Hardware and Integration

- Every PathForward milestone delivered (KPP-4 objective)
- Key milestone report focused on role and efforts of app performance engineers at ALCF, OLCF, NERSC
- Implemented targeted ST vendor support at LCFs and increased facility user support for ECP users
- ECP-Facility engagement plan updated for allocation and management of exascale resources

Crusher: Frontier Test and Development System (TDS) for ECP

Available to ECP users from November 30, 2021 through the life of ECP

“Crusher” TDS system

- 2 cabinets of Frontier HW
- 192 Frontier nodes
- 40 PF
- Slingshot 11 w/ Cassini
- Same Programing Environment as Frontier:
  - Compilers
  - Debuggers
  - Performance tools
  - Libraries
  - Etc.

See also https://www.hpcwire.com/2022/03/28/ahead-of-frontiers-deployment-this-year-1-5-cabinet-crusher-serves-science/

Progress on Crusher by ECP KPP-1 Applications

Apps selected to demonstrate performance improvement for mission-critical problems

Science Area	ECP Project	Crusher (TDS)
Quantum Chromodynamics	**LatticeQCD**	Improving Performance
Chemistry (Biomass Conversion)	**NWChemEx**	Initial Build/Test
Extreme Materials (MD)	**EXAALT**	Improving Performance
Quantum Materials (QMC)	**QMCPACK**	Initial Build/Test
Nuclear Reactors (SMRs)	**ExaSMR**	Improving Performance
Magnetic Fusion	**WDMApp**	Improving Performance
Accelerator Design	**WarpX**	Improving Performance
Cosmology	**ExaSky**	Improving Performance
Earthquakes	**EQSIM**	Improving Performance
Earth System	**E3SM-MMF**	Improving Performance
Cancer Research	**CANDLE**	Improving Performance

Progress on Crusher by ECP KPP-2 Applications

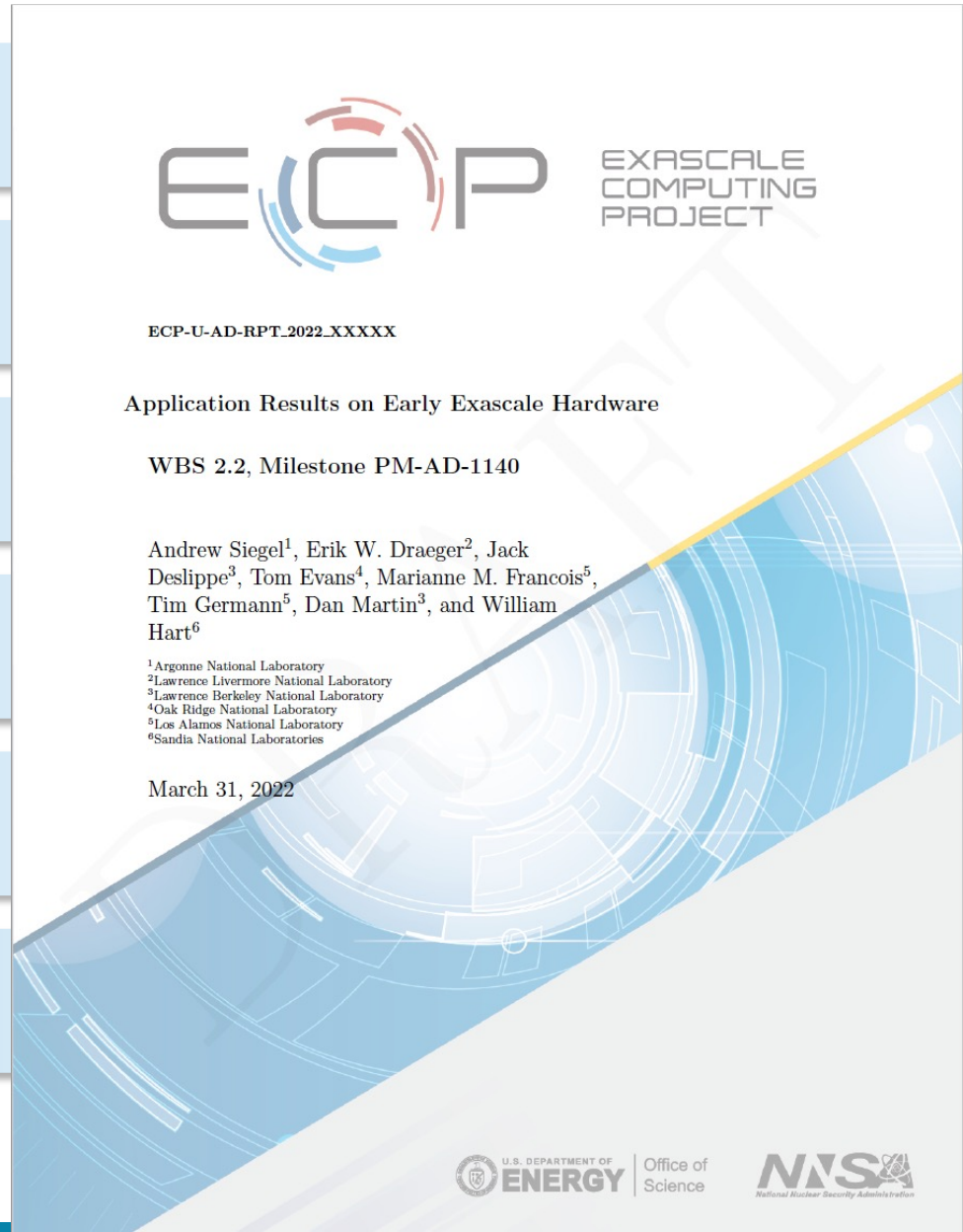
Apps selected to broaden the reach of exascale science and mission capability

Science Area	ECP Project	Crusher (TDS)
Chemistry (Catalysis)	**GAMESS**	Initial Build / Test
Additive Manufacturing	**ExaAM**	Improving Performance
Wind Energy	**ExaWind**	Improving Performance
Combustion	**PELE**	Improving Performance
Fossil Energy	**MFIX-Exa**	Improving Performance
Astrophysics	**ExaStar**	Improving Performance
Subsurface	**Subsurface**	Improving Performance
Power Grid	**ExaSGD**	Improving Performance
Metagenomics	**ExaBiome**	Initial Build / Test
FEL Light Source Interactions with Matter (LCLS)	**ExaFEL**	Improving Performance

ECP's Annual Review of its Application Portfolio

Always Yields Interesting and Emerging Themes – public document forthcoming.

- ✓ **Sparse solver research challenges**
- ✓ **OpenMP offload performance**
- ✓ **Co-maturation of vendor compilers, software stack**
- ✓ **ST and CD integration success stories**
- ✓ **Maturity of performance analysis tools**
- ✓ **Network performance**



EXASCALE COMPUTING PROJECT

ECP-U-AD-RPT_2022_XXXXX

Application Results on Early Exascale Hardware

WBS 2.2, Milestone PM-AD-1140

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

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

March 31, 2022

ECP's Extreme Scale Software Stack (E4S) and associated Software Development Kits (SDKs) are providing tremendous value (e4s.io)

Activity	SDKs	E4S
Planning	Transparent and collaborative requirements, analysis and design, delivery – better plans, less effort, improved complementarity	Campaign-based portfolio planning coordinated with Facilities, vendors, community ecosystem, non-DOE partners
Implementation	Leverage shared knowledge, infrastructure, best practices	ID and assist product teams with cross-cutting issues
Cultivating Community	Within a specific technical domain: Portability layers, LLVM coordination, sparse solvers, etc.	Across delivery and deployment, with software teams, facilities' staff, with non-DOE users in industry, US agencies
Resolving issues, sharing solutions	Performance bottlenecks and tricks, coordinated packaging and use of substrate, e.g., Desul for RAJA and Kokkos	Build system bugs and enhancements, protocols for triage, tracking & resolution, leverage across & beyond DOE
Improving quality	Shared practice improvement, domain-specific quality policies, reduced incidental differences and redundancies, per-commit CI testing of portfolio	Portfolio-wide quality policies with assessment process and quality improvement efforts, documentation portal, portfolio testing on many platforms not available to developers. Address supply chain needs
Path-finding	Collaborative exploration and development of leading-edge tools and processes	Exploration and development of leading-edge packaging and distribution tools and workflows that provide capabilities and guidance for others
Training	Collaborative content creation and curation, coordinated training events for domain users, deep, problem-focused solutions using multiple products	Portfolio installation and use, set up of build caches, turnkey and portable installations, container and cloud instances
Developer experience	Increased community interaction, increased overhead (some devs question value), improved R&D exploration, e.g., variable precision	Low-cost product visibility via doc portal, wide distribution via E4S as from-source/pre-installed/container environment
User experience	Improve multi-product use, better APIs through improved design, easier understanding of what to use when	Rapid access to latest stable feature sets, installation on almost any HPC system, leadership to laptop
Scientific Software R&D	Shared knowledge of new algorithmic advances, licensing, build tools, and more	Programmatic cultivation of scientific software R&D not possible at smaller scales
Community development	Attractive and collaborative community that attracts junior members to join, establishes multi-institutional friendships & careers	Programmatic cultivation of community through outreach and funded opportunities that expand the sustainable membership possibilities

Verifying that ECP KPPs have been completed is a critical activity for the next 18 months

- KPPs formally define the success or failure of ECP (a 413.3B Project)
- ECP KPPs were set in the fall of 2019 (at the Final Design Review)
  - KPP definitions for ECP vetted through many iterations with project teams, DOE sponsors, review teams
  - Each KPP has a direct line of sight to ECP goals and objectives
  - AD and ST subproject challenge problems and integration passing scores formally baselined as part of CD-2/3 approval (Feb 2020)
- ECP KPPs are unique and challenging
  - Built on complex code bases and expected to run in complex exascale environments
  - Deep scientific expertise needed to evaluate progress and completion for each subproject
  - PIs span a variety of experience and expertise bases; most have no previous experience with 413.3B requirements

ECP's KPPs: Quantified with Explicit Targets

KPP ID	Description of Scope	Threshold KPP	Objective KPP	Verification Action/Evidence
KPP-1	11 selected applications demonstrate performance improvement for mission-critical problems	6 of 11 applications demonstrate Figure of Merit improvement ≥50 on their base challenge problem	All 11 selected applications demonstrate their stretch challenge problem	Independent assessment of measured FOM results and base challenge problem demonstration evidence
KPP-2	14 selected applications broaden the reach of exascale science and mission capability	5 of 10 DOE Science and Applied Energy applications *and* 2 of 4 NNSA applications demonstrate their base challenge problem	All 14 selected applications demonstrate their stretch challenge problem	Independent assessment of base challenge problem demonstration evidence
KPP-3	76 software products selected to meet an aggregate capability integration score	Software products achieve an aggregate capability integration score of at least 34 out of a possible score of 68	Software products achieve the maximum aggregate capability integration score of 68	Independent assessment of each software product's capability integration score
KPP-4	Delivery of 267 vendor baselined milestones in the PathForward element	✓ Vendors meet 214 out of the total possible 267 PathForward milestones	✓ Vendors meet all 267 possible PathForward milestones	Independent review of the PathForward milestones to assure they meet the contract requirements; evidence is the final milestone deliverable

ECP has defined a rigorous process to verify L4 subprojects have met their KPP goals

Problem Definition

- **AD**: Minimum criteria for challenge problems set at baseline; refining the details of the specific problems to be run now
- **ST/Co-Design**: Passing number of integration capabilities set at baseline; defining particular targets and integration strategy now
- Proposed completion artifacts described by teams
- Reviewed and approved by SME panel
- Results in KPP contracts (AD) and strategy (ST/Co-Design) for each L4 subproject

Review process

- Rolling reviews as teams complete their challenge problem runs or integrate capabilities on exascale systems
- First reviewed by L3s and then passed to external SME team for consideration
- SMEs may approve the artifacts, request more information, or request a discussion with the team
- Once the SMEs approve, KPP submission report and artifacts, along with lead reviewer memo submitted to FPD for approval
- If FPD approves; the ECP KPP score is updated

Subject matter expert (SME) review panels assembled. For each L4 subproject:
AD: lead reviewer and at least two auxiliary reviewers
ST: panel of three SMEs, including facilities and applications communities

KPP-1 and KPP-2 define the success for ECP applications teams

- KPP-1
  - KPP-1 is based on a Figure of Merit (FOM) defined individually for each project to capture the relevant **scientific work rate** for an application.
  - Each application measured a **baseline FOM value** at the inception of ECP
  - KPP-1 is calculated as the ratio of the FOM **on the exascale challenge problem** to the baseline
- KPP-2
  - KPP-2 is **based on developing new mission-critical capabilities at exascale** per the ECP mission needs statement to broaden the reach of exascale computing
  - To meet KPP-2 an application must successfully execute a capability demonstration of the challenge problem on an exascale platform.
  - Performance requirements for KPP-2
    - Must demonstrate parallel scalability on the exascale systems
    - Must sufficiently utilize hardware accelerators on a node
    - Must execute simulation using all necessary physics and algorithmic capabilities of the challenge problem

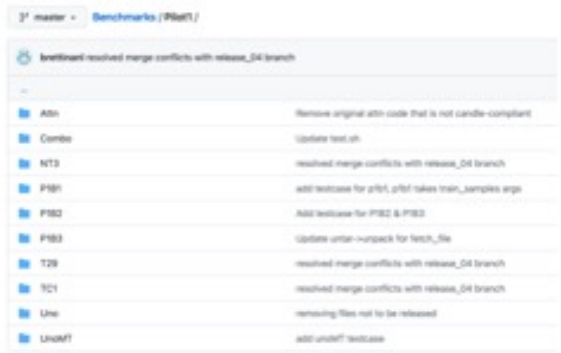
Verifying KPP-1 and KPP-2 completion

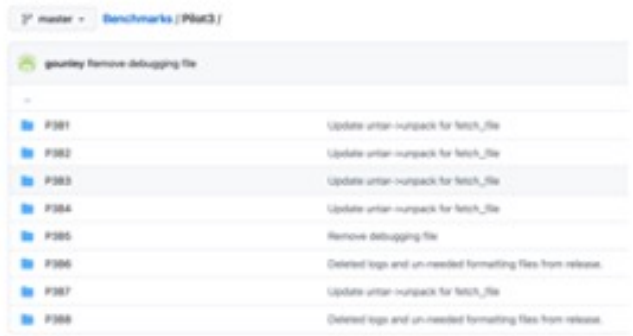
- KPP contracts give concrete descriptions of
  - The base challenge problem, including the specific physical phenomena, numerical approaches and minimum parameters (for KPP-1 this includes the FOM calculation)
  - Problem inputs, setup, resource estimates and runtime settings
  - Problem artifacts, e.g., output files and post-processed data
- SME reviewers will review artifacts provided by teams and confirm that
  - KPP-1: The FOM measurement met threshold (>50) and the executed problem met the challenge problem minimum criteria
  - KPP-2: The code utilized exascale resources and the executed problem met the challenge problem minimum criteria
- Runs must be fully documented and reproducible, including any caveats
- SMEs and reviewers iterated on contract completion to ensure no surprises
- Each team will be asked to provide a short report that describes the challenge problem, FOM, key steps needed to get performance

KPP-1 Verification Contract Example: CANDLE

Artifacts we will provide for our CP runs

- Run scripts (and logs) that will reproduce the 1-2-3 runs
- Timing logs demonstrating the achieved speedups
- Model outputs (training plots, convergence diagrams, diagnostics)
- Model code, training datasets, etc. on CANDLE github site
- Considering the feasibility of capturing in “containers”





KPP-2 Verification Contract Example: ExaWind

ExaWind challenge problem: Minimum criteria

- 2 x 2 array of M
  - NREL 5-MW
  - Simulation a
- Must resolve the
- Resolution must generation of ne vorticity; O(1) C in RANS region
- Simulation will r
- Demonstrate ca 500 s with one r

Stretch goal:

Question 1a: ExaWind KPP simulation: Example scenario

- If we use strong-s and assume 9000
  - 28% of Frontie **limits**
  - We will push t
- Above scenario is Wind to advance a
  - Either increase across fewer C
  - Current results to sub-optimal

Code	CPU st limit or
AMR-Wind	1.5E4 g
Nalu-Wind	2.0E4 g

Question 1c: Confirmation of successful KPP simulation

- `job_name.o1234,nalu-wind.log,amr-wind.log,5MW_Land_BD_DLL_WTurb.log,` etc.
  - All inspected for sanity, successful completion of timesteps, and reporting of time per timestep
  - Data will inform prediction of time required on Frontier for scientifically meaningful simulation
- `amr-wind-image-0.png, amr-wind-image-10.png`
  - Ascent images for AMR-Wind and selected quantities viewed for expected transition from initial condition to final condition
- `output/nalu-wind-mesh-nalu_instance_num.exo.rank_num`
  - Loaded into Paraview for visualization of pertinent physical quantities and expected transition from initial condition to final condition
- If the ExaWind team has access to Frontier node hours for science-based runs (e.g., via INCITE), the team will provide, e.g., results isocontour snapshots, turbine power results, and flow movies

Details and analysis from KPP-simulation artifacts will be documented in a final report

ST and Co-Design projects use KPP-3 to measure integration and drive creation of a productive and sustainable ecosystem

KPP-3 Basics

- **Integration Goal:**
  A statement of impact on the ECP ecosystem, consequential and sustainable use by client.
- **Metric: Capability integration**
  - **ST:** Use of the product for the first time or a significant feature set recently developed, representing an FTE-year or more worth of effort.
  - **CD**: Number of applications using the co-design center's technologies in a sustained way.
- **Threshold/Objective:**
  50%/100% of the weighted (stretch) impact goals are met.

KPP-3 Details

- Weights correlate with scope of impact. Examples:
  - OpenMP, MPICH, AMReX – Weight of 2.
  - Most – Weight of 1.
  - Legion, ParSEC, ExaGraph – Weight of 0.5.
- Integration must represent sustainable progress, not just "tried it" or "considering it".
- Not looking for hero-level integration score counts. Integration is hard work.
  - Typical threshold goals: 4 integrations. A few are higher.

KPP-3 Integration Clients & Artifacts Overview

AD or ST Client

- ST product in use by an AD or ST client, demonstrated on exascale platform
  - May include multiple linked products
- Example:
  - MFIX-Exa + AMReX + ALPINE Catalyst + ALPINE statistical feature detection algorithm + VTK-m + Cinema
- Artifacts:
  - Merge requests/Change logs
  - Run and output logs
  - Journal papers, technical report, milestone report
  - Client Letter
  - Demos or visualizations

Tool Usage

- Utility/Library used in client workflow; pre-exascale or exascale
- Examples:
  - HPCToolKit
  - Darshan
- Artifacts:
  - Merge requests/Change logs
  - Client Letter
  - Performance studies (plots) demonstrating impact on client
  - Technical report, journal paper, milestone report

Facilities Deployment

- Utility/library deployed on exascale machine for general use
- Examples:
  - Performance toolkits
  - ParaView & VisIt visualization applications
- Artifacts:
  - Merge requests/Change logs
  - Module load screenshots
  - Log files from unit tests
  - Tutorial slides, documentation or other user-support activities
  - Milestone report

Community Ecosystem & Vendor Deployment

- Integration into sustainable community software environment or adopted by vendor
- Examples:
  - LLVM
  - OpenMP, OpenACC
- Artifacts:
  - Merge requests/Change logs
  - Meeting notes
  - Proposal to standards or vendor
  - Code review summary
  - Documentation
  - Milestone report

All ST and Co-Design teams have defined KPP-3 integration strategy

Integration	Capability Description	Integration Goals	Target Environment	Target Environment Needs	Verification as Part of Other Activities
What is the JIRA INT issue => defines producer & client	Paragraph describing the capability developed by the L4 product	Why does the client need this? Data reduction, code library, use of tool to drive design decisions, etc.	❑ Frontier ❑ Aurora ❑ El Capitan ❑ Pre-Exascale ❑ Community	List of pre-installed software needed for KPP verification	❑ KPP-1 ❑ KPP-2 ❑ KPP-3 If KPP-3 run: estimate compute cycles needed

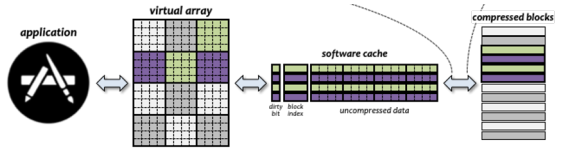
Integration POC Producer	Integration POC Consumer	Planned or Backup Activity	Expected Artifacts — depends on use case	Status
@ tag username	@ tag username	❑ Planned ❑ Backup	❑ Run logs ❑ Publications, conference presentations, videos, etc. ❑ Output visualizations ❑ Screen shot from module load ❑ Merge requests ❑ Documentation or user support activities ❑ Client letters or client analysis showing impact ❑ Post hoc analysis workflow	❑ ST L4 Draft Memo Completed ❑ ST L3 Manager Approval ❑ Consumer L4 Signoff ❑ SME Approval

KPP-3 Project/Product Strategy: 4 planned + 2 backup integrations Example for Data/Viz project: ALPINE / ZFP

ALPINE: In situ visualization and analysis algorithms and infrastructure

Ascent ParaView VISIT ParaView Catalyst LibSim

ZFP: Compressed representation for floating-point and integer arrays



ALPINE – 4 planned + 2 backup

Tentative Score: 4

- **ALPINE (Ascent Replay) <> WarpX**
- **ALPINE (Catalyst + feature detection algorithm) <> MFiX-Exa**
- **ALPINE (ParaView deployment) <> Frontier**
- **ALPINE (VisIt deployment) <> Frontier**
- ALPINE (Ascent) <> ExaLearn <> PeleC
- ALPINE (Ascent + sampling algorithm) <> ExaSky:Nyx

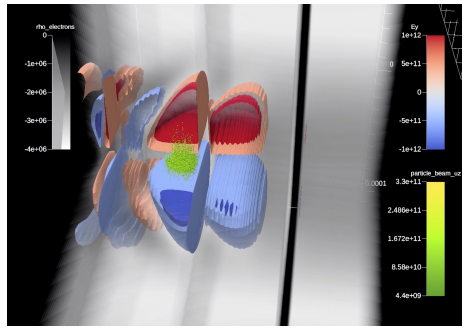
ZFP – 4 planned + 2 backup

Tentative Score: 4

- **ZFP <> HDF5 <> SW4**
- **ZFP <> ADIOS <> WarpX for data reduction**
- **ZFP <> QMCPACK:RMG for in-memory compression**
- **ZFP <> CODAR for integration into Zchecker**
- ZFP <> QMCPACK for in-memory compression
- ZFP <> ADIOS <> WarpX for in-memory compression

KPP-3 Integration Evidence: ALPINE and WarpX Plasma Accelerator

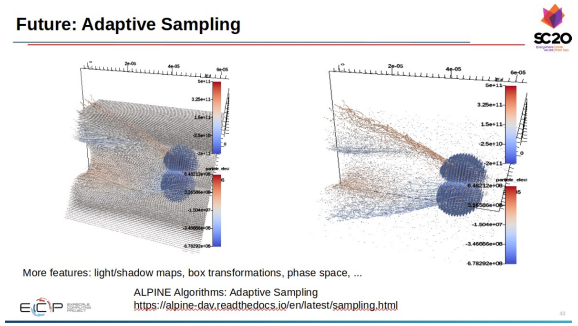
In situ visualization and analysis

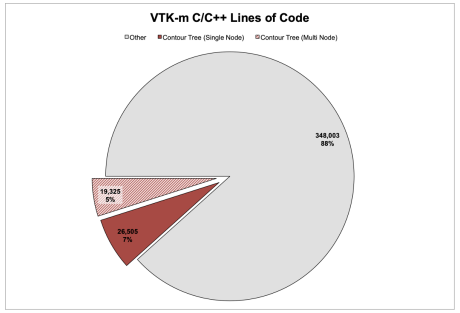


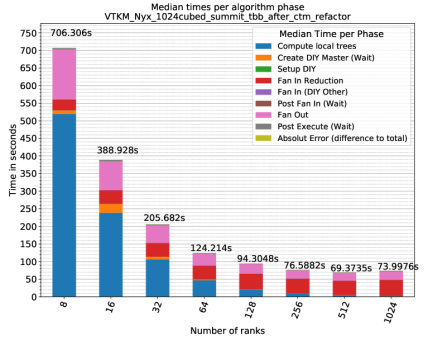
[INT-825](#) ALPINE <> WarpX + Piscar
[INT-827](#) VTK-m <> ALPINE
[INT-829](#) ALPINE <> Cinema
[INT-311](#) ALPINE <> AMReX
[INT-842](#) AMReX <> WarpX + Piscar
[INT-826](#) Cinema <> WarpX + Piscar
[INT-1444](#) VTK-m <> WarpX + Piscar

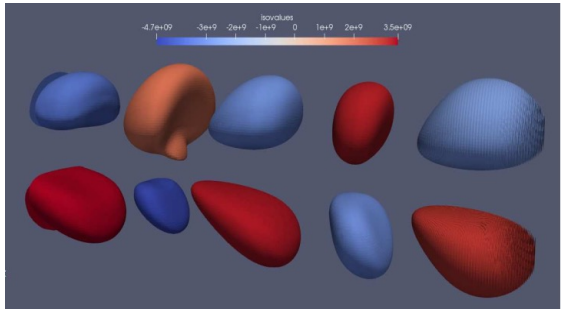
Functionality: ALPINE and WarpX have several integrations points to address WarpX analysis needs

- **Ascent: Replay** to develop in situ visualization pipeline
  - Recent work on productizing Ascent for Frontier deployment
  - Improved functionality such as annotations, data binning
  - Added support for WarpX requirements such as CMake version and adding Ascent to WarpX CI process
- **Ascent: Adaptive Sampling**
  - **Off-the-shelf** application of sampling algorithm to reduce uninteresting particle background
- **Contour tree topological analysis** to identify features such as isocontours
  - Recent work includes metric computation for distributed contour tree representation; prototype has been ported to VTK-m with scaling studies on Summit
  - Port to Spock identified issues with Kokkos `sort`; working with VTK-m and Kokkos to resolve
  - Contour tree represents 12% of VTK-m code base
  - FY22 task is to re-integrate with WarpX









Identified contours are saved in Cinema database for post hoc analysis

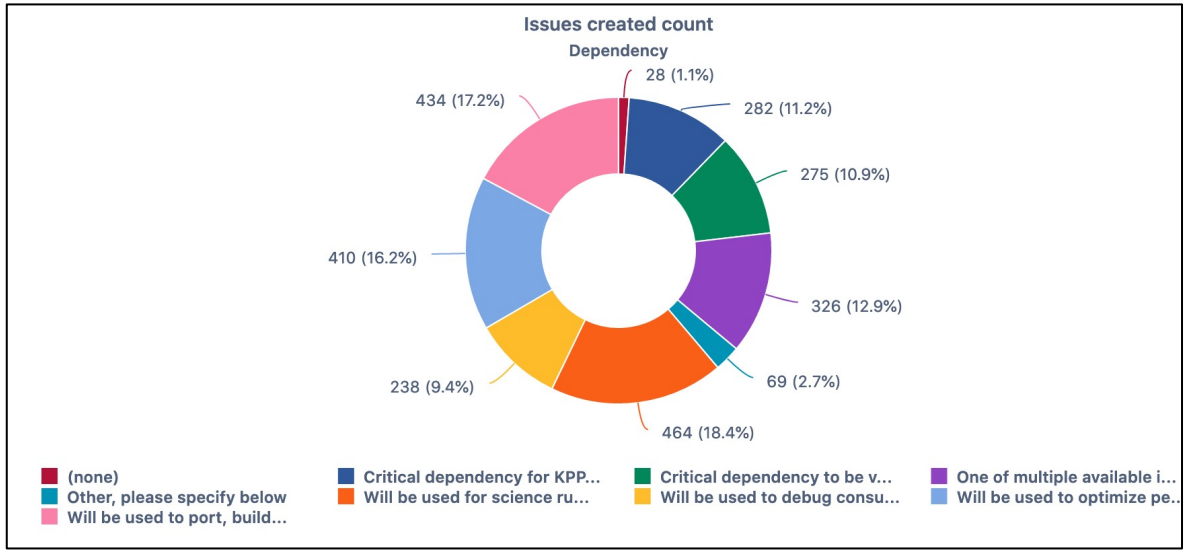
ECP's living dependency database provides a key source of information for managing critical integration points

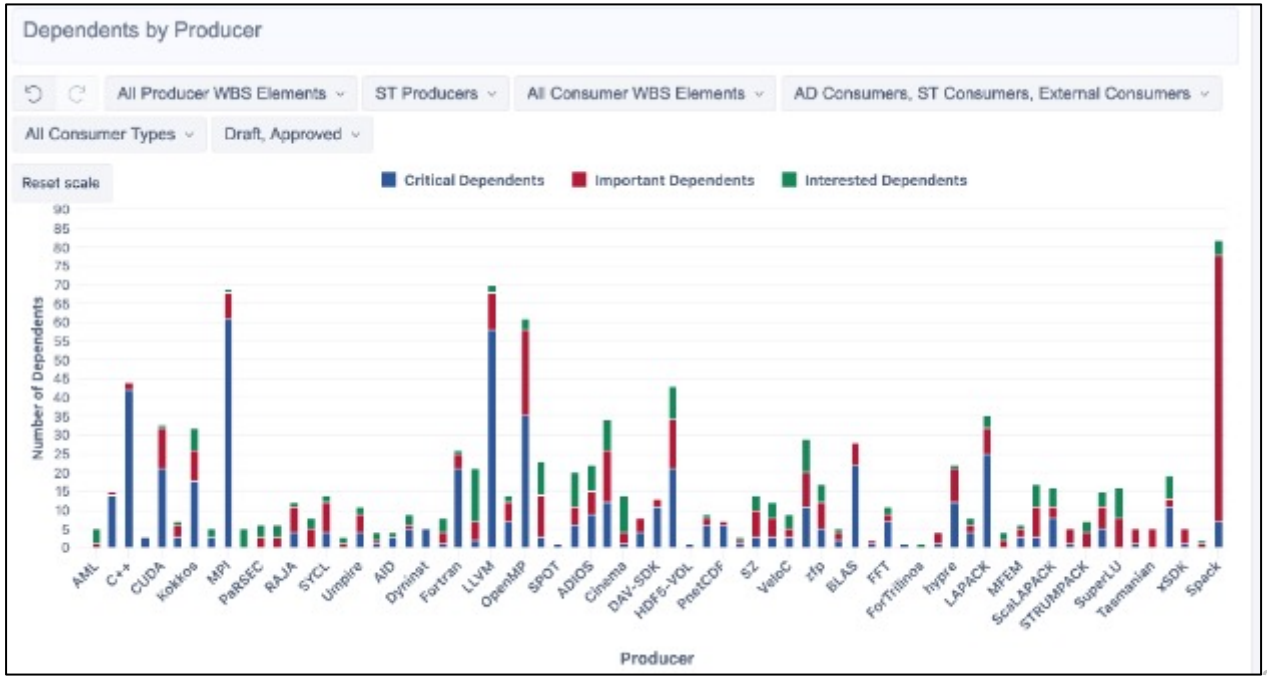
Dependencies updated in KPP-3 planning process

- 1161 issues in the database
- Corrected existing issues and added new ones as needed
- All KPP-3 target integrations are now included in the database; confirmed by ST/Co-design L3s

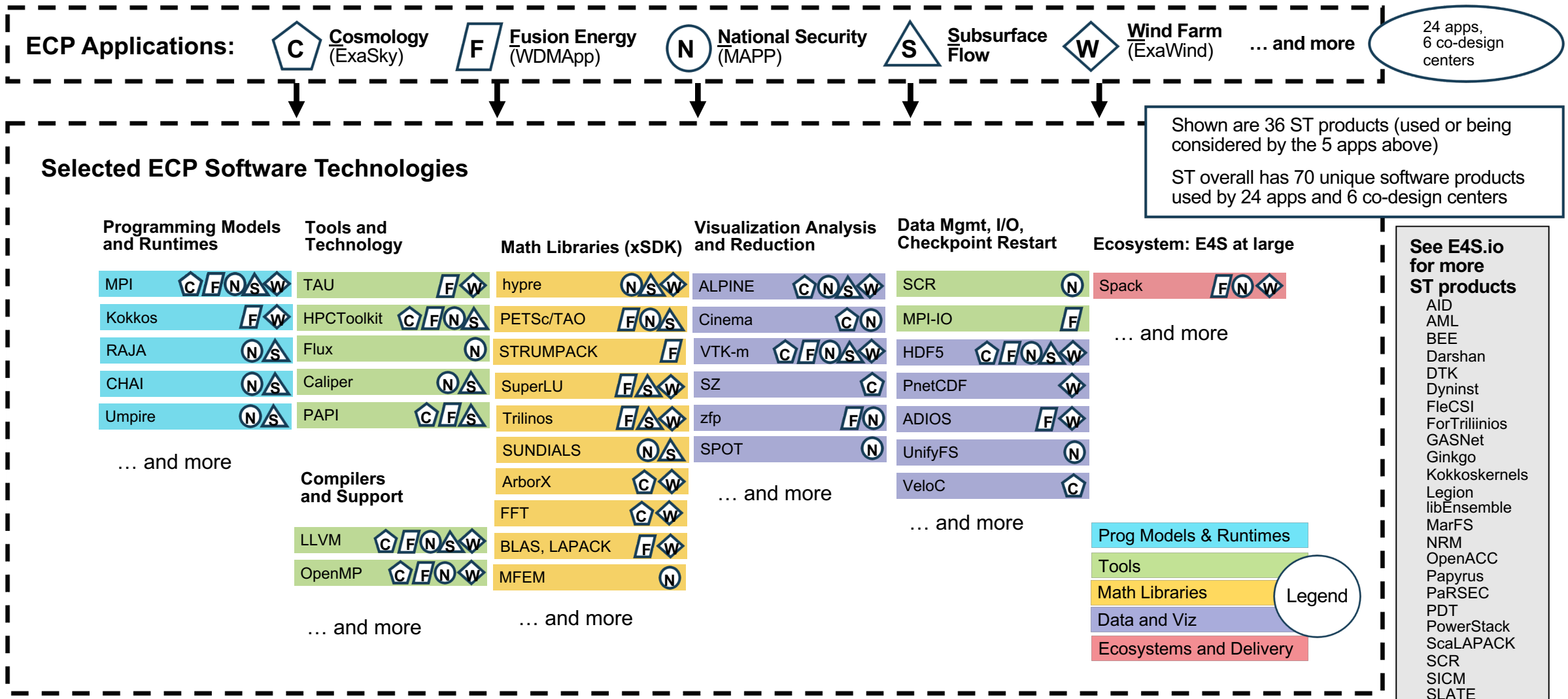
Used in tracking critical dependencies as issues arise

- Answering questions such as "who depends on X?"
- Motivated deep dive discussion between critical high-use producers and consumers
  - Programming models
  - Sparse linear solver technologies





AD-ST Dependency Example: ST Products Consumed by 5 Apps



ECP apps rely on multiple software technologies; some software products contribute to multiple distinctly developed components of a multiphysics app (such as fusion energy modeling) that must run within a single executable.

Risks We're Watching and/or Actively Mitigating

Emerging risks (fresh from recent annual AD and ST reviews)

- Compiler/runtime bugs, optimization; HPC library performance (DGEMM, FFT, etc.)
- Performance of OpenMP Offload on AMD and Intel; Sparse linear solvers
- I/O maturity; SlingShot 11 Interconnect
- Staff retention through ECP end; excessive exascale resource usage

Actively Mitigating over Last Year

- **AD**: app-specific solvers, GPU port/perf, algorithm challenges, physical models
- **ST**: OpenMP, Spack, Solvers, VTK-m, I/O (HDF5, DataLib), FFT benchmarking
- **HI**: Facility support, broader Facility engagement, SlingShot testing (v10 now moving to v11)

Next Steps: Proof Testing 5 years of focused RD&D

Frontier arrival means it's time to stress (KPP) test our tools & technologies with a sense of urgency

- We are implementing a quick pace of management and execution - exascale system state during this period will be more dynamic
- ECP must maintain pace and if appropriate help steer. ECP leadership needs to be tuned in and "at the table"
- PIs to report their status bi-weekly: What happened recently? What's planned? What are the blocking items?
- Management of ECP's exascale resource allocation and use: what to move up/down in priority, monitor usage, sync with Facility leadership

2.2.1 Chemistry and Materials Applications	Current system (Crusher or Frontier)	Current status	Status Details

2.2.1.01 LatticeQCD	Crusher	• **Status**: optimizing performance
		• **Main obstacles**: compilers, system software immaturity
			• **CPS/DWF**:
			  • CPS/Grid/HIP: Multinode Dslash, linalg performance benchmarked on Crusher up to 16 nodes, 16 nodes maintains ~65% of single node performance for weak scaling.
			  • CPS/QUDA/HIP: Inverter performance on 16 crusher nodes compared with 16 Summit nodes, crusher ~30% better performance than summit
			• **MILC/HISQ**: We have built and tested a scaled-down version of both our HMC and analysis challenge problems, but can't proceed further until system problems are corrected. (Large multinode jobs fail because of system problems.) Global reductions in GRID are slow. We are investigating. Volume scaling study: We find encouraging performance on up to 16 nodes.
			• **Chroma/Wilson-Clover**: status needed.

2.2.1.02 NWChemEx	Crusher	• **Status**: porting code from CUDA to HIP, optimizing performance • **Main obstacles**: HIP expertise, vendor/OLCF assistance needed	• Could use help from someone with HIP expertise to port and optimize TAMM and TA tensor library, DFT code, and other needs as appropriate • Working to improve the overall scalability of the Hartree-Fock implementation on GPUs • Significant optimization of intermediates, individual sparsity kernels, and Hadamard contractions on GPUs for the DLPNO algorithm. The final issue has a dependency on TAL-SH and on GPU tensor operations. • Need to create a stand-alone test code for most used one-sided communications with GA for TAMM.

Biweekly checks of application subproject progress and status on Crusher / Frontier

WBS/Name	PI	PC	3/3/22	3/3/22 Status	2/17/22	2/17/22 Status
	@Barton Miller	@Tim Haines @Xiaodan Liang	☑ No issues ☐ Ongoing issues ☐ New issues ☐ Resolved issues		☑ No issues ☐ Ongoing issues ☐ New issues ☐ Resolved issues	All issues discovered during SDK work are reported to the respective L4 project and reported there.
	@Jack Dongarra	@Heike Jagode @Anthony Danalis	☐ No issues ☐ Ongoing issues ☐ New issues ☐ Resolved issues		☐ No issues ☑ Ongoing issues ☐ New issues ☐ Resolved issues	Ongoing issues with AMD GPUs: (1) Rocprofiler intercept mode does not allow to reassign queue callbacks. Reassigning callbacks at runtime would be a desirable feature for PAPI so that users can define multiple eventsets for different parts of the code (kernels). Currently, not being able to reassign queue callbacks means that a PAPI user has to stick to only one eventset per application run. [Jan 24, 2022] We created a Rocprofiler issue #71 to report the problem to AMD: https://github.com/ROCm-Developer-Tools/rocprofiler/issues/71 (2) On Crusher (as well as Tulip), the "sampling" mode works but the "intercept mode" triggers a core dump inside librocprofiler64.so. "Intercept mode" only works if rocprofiler_start_queue_callbacks is called after any HIP memory-related calls (e.g., hipMalloc, hipMemcpy, hipStreamCreate, etc.) are made; and, equivalently, if rocprofiler_stop_queue_callbacks is called before any HIP memory-related calls. [Feb 16, 2022] We created a Rocprofiler issue #74 to report the problem to AMD: https://github.com/ROCm-Developer-Tools/rocprofiler/issues/74

Biweekly checks of software technology subproject progress and status on Crusher / Frontier

Questions?

https://www.exascaleproject.org/contact-us/

For more info

- Alexander F. et al. *Exascale Applications: Skin in the Game,* Phil. Trans. R. Soc. A 378: 20190056 (2020) (http://dx.doi.org/10.1098/rsta.2019.0056).
- Douglas Kothe, Stephen Lee, and Irene Qualters, *Exascale Computing in the United States,* Computing in Science and Engineering 21(1), 17-29 (2019).