Software stacks and sustainability stewardship in DOE

ADAC Open Symposium Paris, France September 27, 2023 Todd Gamblin Livermore Computing Lawrence Livermore National Laboratory





The Exascale Computing Project is ending

- 7 years, \$1.7 billion
- Joint effort of 6 core DOE laboratories
- 3 core focus areas:
 - 24 Application Projects
 - 70 Software Technology Projects
 - 100+ packages
 - Hardware/Facility Integration
- 80 teams, 1,000 researchers



R&D funding under ECP ends in December 2023!





ECP established a hierarchical integration structure for software

- Related products integrate into SDKs (Software Dev Kits)
 - More aptly called "Product Communities"
 - Build, link, deploy, and test libraries together
- All products integrate into E4S: ECP's software stack
 - E4S is a subset of Spack packages
 - Plus lots of release, testing, and deployment work
- Applications leverage software technologies
- Integration ensures that products can be used together





E4S: Extreme Scale Scientific Software Stack



- 100 packages: Red boxes are the packages in it (about 100)
- 500 packages: Other dependencies needed to build E4S
- Over 600 total packages!

"Extreme-scale" refers to exascale machines, but could just as easily refer to the dependencies





Each application is essentially its own software stack



- This is LLNL's ARES; ECP codes are similar
- Nearly 3/4 of the packages are open source, nearly 2/3 are external open source

DOE's ecosystem relies heavily on the broader open source ecosystem





Spack has emerged as the de-facto tool for managing HPC software



Spack is successful because it includes the entire HPC ecosystem!







Spack enables software reuse at many levels



Spack leverages a great deal of infrastructure to keep the ecosystem working







Spack CI Architecture





Spack lowers the burden of maintaining many binary distributions



CI defines many stacks to build

- Lists of packages aimed at communities
 - E4S HPC distribution
 - Power, macOS, OneAPI versions
 - Various ML stacks
 - CPU
 - CUDA
 - ROCm
 - LLNL-specific stacks
 - AWS user stacks
- Easy to build same stack many different ways using versatile recipes
- We've built a OneAPI version by only changing the green part

17	packages:
18	all: Config parameters
19	target: [x86_64_v3]
20	variants: ~rocm+cuda cuda_arch=80
21	llvm:
22	<pre># https://github.com/spack/spack/issues/27999</pre>
23	require: ~cuda
24	
25	definitions:
26	- packages: LIST OT PACKAGES
27	# Horovod
28	- py-horovod
29	
30	# Hugging Face
31	- py-transformers
32	
33	# JAX
34	- py-jax
35	— py—jaxlib
36	
37	# Keras
38	- py-keras
39	- py-keras-applications
40	- py-keras-preprocessing
41	- py-keras2onnx
42	
43	# PyTorch
44	- py-botorch
45	- py-efficientnet-pytorch
46	- py-gpytorch
47	- py-kornia
48	- py-pytorch-gradual-warmup-lr
49	- py-pytorch-lightning
50	– py-segmentation-models-pytorch
1	github.com/spack/spack

Our CI system now does over 100,000 builds/week to support ECP and the broader ecosystem



Bursts come from users iterating on stacks in PRs



There has been considerable effort on Continuous Integration under ECP

Victories:

- Scaled out Spack CI to 100,000 builds/week
- Scaled integration CI for SDKs and E4S
- Jacamar runner allows HPC centers to run GitLab as a secure service for users
- NNSA using Jacamar for tri-lab project CI

Gaps:

- Most projects maintained on GitHub
- Jacamar only used internally at facilities
- Huge unmet demand for public project and user CI







ECP has brought the software ecosystem forward

Wins:

- Product support
- Unprecedented integration scale for scientific software
- Unprecedented portability across GPUs, CPUs
- Applications have started to rely heavily on dependency libraries!
- Common packaging and deployment tools
- Integration among products
- Gaps:
 - How do we sustain all of this!?





October 2020: ASCAC recommended a stewardship program

Transitioning ASCR after ECP



- Continue development
- Build broad engagement
- Continue to support applications

Recommendation A.1: Create a shared-software stewardship program within ASCR	27
Recommendation A.2: Engage current, and anticipate future, software needs	29
Recommendation A.3: Collaboratively support applications	32
Recommendation A.4: Broaden industry and academic engagement	33

Recommenda

Recommendation A.1: Create a shared-software stewardship program within ASCR

ASCR should create a comprehensive program that leverages the ECP ecosystem to support and curate shared software. This should incorporate ASCR program office oversight while delegating operational control to a software engineering team of laboratory and academic experts.

Software Stewardship Vision

The focus of the proposed stewardship program is to support and extend DOE shared software products, starting with the ECP software stack and extending over time to other software that may be "productized," that is, made available to wide, shared use and "hardened" for reliable use by diverse users. The hub will provide a context for support of software packages in a coherent



Report to the DOE Office of Science, Advanced Scientific Computing Research Program

October 2020

Lawrence Livermore National Laboratory

October 2021: ASCR RFI on stewardship of software

- Potential scope of stewardship activities, including:
 - Training on software dev, use
 - Workforce support
 - Infrastructure for common development needs
 - Curation and governance processes
 - Maintaining situational awareness
 - Shared engineering resources
 - Project support



Software dependencies and requirements: What software packages and standardized languages or

ASCR Software-Stewardship: Next Steps

Next steps for ASCR regarding software stewardship:

- 1. Finalize the targeted scope of potential software-stewardship activities for FY23.
- 2. Define the relationship between those software-stewardship activities and synergistic activities in the Facilities, Research, and Advanced Computing Technologies (ACT) Divisions.
- 3. Pursue the definition and release of a funding opportunity, or funding opportunities, covering the targeted scope.
- 4. Work with ECP, ASCR facilities, and other stakeholders to enable a common understanding of how all stakeholders will contribute to the overall process.

https://bit.ly/stewardship-rfi-summary





October 2022: CHIPS and Science Act adds sustainability language

"It is the sense of Congress that *the Exascale Computing Project has successfully created a broad ecosystem* that provides shared software packages, novel evaluation systems, and applications relevant to the science and engineering requirements of the Department, and that *such products must be maintained and improved in order that the full potential of the deployed systems can be continuously realized*."





October 2022 – Feb 2023: Seedlings for sustainability call

- ASCR solicits seedling proposals for sustainability programs
- Call says applicants should:
 - Articulate a long-term vision for sustaining a substantial part of the ecosystem
 - Describe how their efforts might be rapidly scaled up in the future
 - Request no more than \$125,000 for a one-year award
 - Prepare to scale up rapidly in the future
- 6 pilot projects selected
 - Workshops and planning have been going on since March 2023







Phase 2 of Seedlings is a Stewardship program for "Next-Generation Scientific Software Technologies" (NGSST)

- Invited 7 projects to join forces for a combined stewardship organization:
 - PESO Activities focused on evolving a cohesive ecosystem for DOE software E4S, Spack, Integration
 - OASISProject support, shared engineering resources, training
Math libraries, data, vis, products not covered by other teams
 - **SWAS** Workflow community
 - **STEP** Tools community
 - **S4PST** On-node and off-node Programming Models (Kokkos, LLVM, Fortran, C++, MPI)
 - **COLABS** Workforce, training, best practices, and coordination for RSE community
 - **OSSF** Partnering with foundations to advance stewardship

Some details about the NGSST program

- Focus on *stewardship* as opposed to *sustainability*
 - Implies advancement, not "mere" sustainability
- Teams must develop a *shared governance model* with:
 - Leadership sharing (e.g., rotating chair, elections, etc.)
 - Model for expanding in the future
- Should look (according to DOE) to the outside like one organization
- Ceiling is \$14M total for the organization
 - We are hopeful that this number will increase
 - ECP software technologies had a budget of ~\$50M
- DOE strongly encourages contribution from external stakeholders:

"Stewarding next-generation software technologies requires investment from DOE combined with agile strategies that <u>enable other stakeholders to provide</u> <u>additional funding commensurate with their additional needs</u>."





What are software foundations

- Nonprofit corporations, typically one of:
 - 501(c)(3): Charitable (religious/educational/scientific) organization
 - Serves the public
 - Cannot lobby or influence legislation
 - Exempt from federal, state, and local taxes

501(c)(6): Organization of people with common business interests

- Serves its members' common interest
- Seeks to improve business conditions
- Exempt from only federal income taxes
- Can lobby (but funds used for lobbying are taxed)



There are many existing models











How big are foundations?

(mix of 2020-2022 numbers)

LLVM Foundation

Python Software Foundation

Apache Software Foundation

NumFOCUS

Eclipse Foundation

Linux Foundation (including CNCF)

~\$800,000 / yr ~\$2,100,000 / yr ~\$2,200,000 / yr ~\$5,600,000 / yr ~\$26,100,000 / yr ~\$248,000,000 / yr

Unclear how creating a new foundation can match these influence and leverage



What does the money go towards?

• Depends, but here is an example for Linux Foundation:



- Project Support 59.2%
- Community Infrastructure 9.8%
- Corporate Operations **7.4%**
- Community Training 7.3%
- Community Events 3.1%
- Community Tooling 5.4%
- Linux Kernel Support **3.2%**
- International Operations 1.5%

What's project support?

- Infrastructure (CI, testing, Slack, etc.)
- Project metric tracking (adoption, users, etc.)
- Web development
- Meeting Facilitation
- Documentation
- Legal services
- Marketing
- Certifications and Training
- Outreach, Events, Inclusivity initiatives

LF has a very large event organization team Events like KubeCon attract 10k+ people



What do Foundations NOT fund?

Most major feature development!

- Foundations are not typically developers for contract
- Could subcontract to other organizations for focused development
- They're typically experts at community organizing

Typically, the work of projects is done through in-kind effort

- Foundation members are contributors
- Aim is to coordinate efforts and effectively govern and grow project
- Sustainment comes from growth





So what can foundations do for DOE?

Growth



- Open governance allows new stakeholders to help steer theproject
 - Feel some ownership of the project
- Neutral home allows more companies to engage
- 2. Growing contributors
 - Outreach, training, and marketing lead to more users
 - Broader user base leads to more contributors
- 3. More adoption drives more interest and investment

Broader adoption can drive interest from big players

- Adoption has value
 - Customers ask for adopted/used software
- Our software benefits vendor customers.
 - Cost of reinvention is high
- The bigger community you can build, the more likely cloud providers will take interest



Myth: "There are only a few people in the world who can contribute to our software"

- The open source community is huge, and casting a wide net can *find* you more contributors
- One of the libraries I've heard this claim for is MFEM
 - MFEM had 100+ attendees at a user group meeting
 - Community continues to grow
- AWS recently built PALACE, an open source quantum code, on top of our MFEM library
 - AWS developers have contributed some of their work back to MFEM



February 21, 2023

In an interesting twist on quantum-inspired work making its way into traditional HPC – and in this case a step further into cloudbased HPC – AWS today introduced Palace, short for PArallel, Large-scale Computational Electromagnetics, a parallel finite element code for full-wave electromagnetics simulations. Palace was first developed at the <u>AWS Center for Quantum Computing</u> to perform large-scale 3D simulations of complex electromagnetics models in the design of quantum computing hardware. While Palace can be used in quantum hardware design, AWS expects it to be used in a wide range of simulations.

Announced in a <u>blog</u> today, AWS researchers[i] wrote, "We are making Palace[ii] <u>freely available on GitHub</u> as an open-source project for electromagnetic modeling workloads, not limited to





Lawrence Livermore National Laboratory

Foundations can exist alongside DOE's stewardship organization as an outreach and community building arm



Contribute to and influence project communities

Drive industry and community interest

Manage engagement w/CoPs

- HPC in the cloud is growing at 17% CAGR
 - Compare to HPC CAGR of 7%
- Cloud HPC is a growth area for industry
- We have the expertise in HPC
 - Accelerated computing
 - "100X" performance
- Making it work *in the cloud* grows our communities and contributor base

- CNCF's mission is to create a portable reference stack for cloudnative services
- Orgs like OpenSSF get visits to the white house: benefit from one voice
- We could be the portable reference stack for cloud HPC



HPSF: the High Performance Software Foundation

- We are planning to launch an umbrella foundation in Linux Foundation called HPSF
 - Initial Projects:
 - Spack
 - Kokkos
 - MPICH
 - HPCToolkit

- Possible Initial Members:
 - AWS, HPE, NVIDIA, CEA, DOE/SC, DOE/NNSA labs, Kitware
 - Talking to others!
- Mission is broader than just DOE products: focus on whole ecosystem
- Coordinate closely with DOE and other stewardship programs
- Coordinate closely with industry HPC players to *define the stack*

Get in touch if you are interested in joining!







Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.