

# Charting the Future of High Performance Software: HPSF Year in Review & What's Ahead

The past year was full of momentum for the High Performance Software Foundation (HPSF). HPSF continued its mission as a neutral hub for open source high performance software supporting portability, performance, and productivity across diverse hardware. Our portfolio spans key HPC and scientific computing projects, from package management and container runtimes to performance tooling, programming models, and scientific libraries.

**Here's a summary of what we accomplished in 2025 and what's on the horizon for 2026.**

## Membership Growth

This year HPSF was proud to welcome 11 new members to the HPSF family - including Microsoft as a Premier member and Arm and RIKEN Center for Computational Science as General members for a total of 27 members. Today, HPSF's members include:



Find out how your organization can [become a member of the HPSF](#).

## Community Growth

A major highlight of 2025 was HPSF's growing presence across the global high performance computing events. From Hamburg to St. Louis and many places in between, our community showed up, shared knowledge, and strengthened collaboration at some of the most high-profile gatherings in HPC. Throughout the year, HPSF participated in conferences such as FOSDEM'25, ISC High Performance, International Conference on Parallel Processing (ICPP), and SC'25.



The inaugural HPSF Conference launched in Chicago where developers, researchers, and industry stakeholders gathered for an exciting week in May that featured updates from HPSF projects, insights on processor and system trends, active discussions around the future of interoperability, and collaborative working group sessions. Check out our [HPSFCon recap](#) for more details.

HPSF established a presence on social media channels like [LinkedIn](#), [Bluesky](#), and [YouTube](#). Plus be sure to [sign up for the HPSF Newsletter](#) for a quarterly update on everything going on and how you can get involved.

## Project Growth

HPSF was pleased to welcome three new projects in 2025 including emerging project: OpenCHAMI and established projects: Modules, and Chapel. As we start a new year, [HPSF's full roster of projects](#) includes AMReX, Apptainer, Chapel, Charliecloud, E4S, HPCToolkit, Kokkos, Modules, OpenCHAMI, Spack, Trilinos, Viskores, and WarpX.

### AMReX

AMReX is an open-source high-performance software framework for building massively parallel, performance portable, block-structured adaptive mesh refinement (AMR) applications on a variety of architectures. Key features of AMReX include AMR, particles, embedded boundaries, linear solvers, parallel FFTs, and performance portability. AMReX is used by a wide range of multiphysics applications.





## AMReX (continued)

In 2025 we have extended our linear solver capabilities, generalized and optimized the AMReX particle classes, and expanded our CI testing to include AMD and Intel GPUs. We also held our first “AMReX-travaganza” at the inaugural HPSF Conference and are looking forward to another great session in 2026.

In 2026 we plan to continue supporting our broad user base, further expand our CI and regression testing to new architectures, add additional support for staggered mesh algorithms, explore reduced- and mixed-precision approaches, and continue to optimize existing features for current and future architectures. At the 2026 HPSFCon in Chicago, we plan to have a session focusing on python bindings, coupling to AI/ML workflows, and discussions of new directions for AMReX development.

## Apptainer

Apptainer (formerly Singularity) simplifies the creation and execution of containers, ensuring software components are encapsulated for portability and reproducibility.

In 2025 we released the 1.4.x series of updates which added new features, fixed bugs, and addressed a security vulnerability. New feature highlights include support for cluster-wide configuration of subid/subgid mappings, maximum compression options for the squashfs partition in SIF files, and a progress bar for the squashing step when building new container images.



In 2026 we plan to release the 1.5.x series. New feature highlights planned are support for Container Device Interface (CDI) descriptors for generic handling of new hardware devices, a new buildkit bootstrap for building Apptainer containers directly from Dockerfile descriptors, an option for building completely reproducible SIF containers, and the ability to download SIF files using an InterPlanetary File System (IPFS) gateway.

## Chapel

Chapel is a programming language for productive and portable parallel computing at any scale, from the desktop to the supercomputer.

Joining HPSF was arguably the major highlight for Chapel in 2025, and as part of that process, we opened up many aspects of our project such as [weekly meetings](#), [project governance](#), and the repositories for our [brand-new project website](#) and blog. Other 2025 highlights included community outreach: [22 new blog articles](#) spanning diverse topics



like [memory safety comparisons](#) with Rust, Python, and C++, [AI-generated Chapel programs](#), and [user interviews](#); talks, demos, and tutorials in [dozens of settings](#) including LUMI, LANL, and KAUST; and [ChapelCon '25](#), expanded to a four-day format this year to support more submitted talks along with tutorials, and coding sessions. On the technical side, we made [four quarterly releases](#), whose highlights included better [Python interoperability](#), a Rust-inspired [editions](#) feature for language versioning, and a scalable sort routine [published at PAW-ATM 2025](#) in comparison with other HPC programming models.

This year we plan to improve our support for tooling, compilation times, error messages, and debugging, particularly for distributed settings; to support dynamically loaded Chapel modules; and to deploy our first major packages using the Mason package manager. On the project governance side, we aspire to expand Chapel's Technical Steering Committee beyond its initial set of members to improve our institutional diversity.

## Charliecloud

Charliecloud is a lightweight, fully unprivileged, open source container implementation for HPC applications. It can handle the entire container workflow, including building images, pushing/pulling to registries, and dealing with accelerators like GPUs. We take a unique approach to containerization while remaining compatible with the broader container ecosystem where it matters.





## Charliecloud (continued)

We are proud of what we've accomplished in 2025. On-boarding into HPSF has yielded multiple improvements for Charliecloud logistics, for example moving development from GitHub to GitLab.com, a better workflow, newbie documentation, and strengthened governance. Technical improvements include Container Device Interface (CDI) standard support for injecting host resources into a container at runtime, optional garbage collection for our C code using libgc (yes, you can do that in C!), a new ch-image modify command for interactive or shell script modification of container images, refactoring of significant portions of the C code as well as continuous integration (CI) tests, CMD and ENTRYPOINT instruction support for Dockerfiles built with ch-image build, and many other bug fixes and enhancements.

We have ambitious plans for 2026. Two items in particular would be nearly impossible without HPSF: we want to expand CI to a wide variety of architectures and software environments, and we want to increase the institutional diversity of our technical steering committee.

Container build also remains a technical focus, for example strengthening unprivileged build, build in restrictive environments, and improving the build cache. We also hope to improve CDI support, to include proposing revisions to the upstream standard, and add the ability to directly run foreign images such as Singularity/Aptainer and OCI bundles. We'd like to better document our API with an eye to providing a libcharliecloud for other applications to use. Finally, we'd like to make Charliecloud compatible with lightweight virtual machine shims emerging on MacOS and Windows to run containers on those operating systems. Please help! We are friendly.

## E4S

E4S is an effort to provide open source software packages for developing, deploying and running scientific applications on HPC and AI platforms.

In 2025, we supported new GPU platforms including NVIDIA Blackwell with x86\_64 and aarch64 (Grace) hosts. E4S containers and cloud images (ParaTools Pro for E4STM) now support Ubuntu 24.04 and Rocky Linux 9.7 OSes. New AI tools (Google ADK, Gemini, VLLM, NVIDIA NeMo, NVIDIA BioNeMo, Huggingface) were integrated in the E4S distribution.



## E4S (continued)

We released two versions of E4S (25.05 and 25.11) with a performant HPC-AI software stack.

This year we plan to improve the integration with updated Spack features of splicing, an improved binary cache, containers for CI on Frank@UO, and coverage of agentic and generative AI packages in E4S.

## HPCToolkit

HPCToolkit is an integrated suite of tools for measurement and analysis of program performance on computers ranging from multicore desktop systems to GPU-accelerated supercomputers

In 2025 we developed and deployed new capabilities for instruction-based performance measurement using PC sampling on AMD and Intel GPUs.



This year we plan to improve the scaling and performance of instruction-based analysis within GPU kernels, release a Python-based API for analyzing performance data, and improve support for analysis of AI and ML applications.

## Kokkos

The Kokkos C++ Performance Portability Ecosystem is a production level solution for writing modern C++ applications in a hardware agnostic way.

2025 was dominated by our preparation for and release of Kokkos 5, a new major version of Kokkos that represents a significant modernization of the code with a rebase on C++20 and the move to leveraging std::mdspan. This not only makes Kokkos more maintainable for core developers, it also further enhances the robustness of interfaces for users with an increased ability to detect and prevent erroneous usage of Kokkos. As part of our quest for higher quality, we also significantly improved documentation.





## Kokkos (continued)

Particularly the KokkosKernels subproject invested significant resources to produce comprehensive coverage of its entire API. The subefforts KokkosFFT and KokkosComm matured and reached a state where applications can start to rely on them. 2025 was also the year in which the French CEA emerged fully as the third major contributing institution after Sandia and Oak Ridge National Laboratories. This significantly expands the developer base of the Kokkos ecosystem and provides a much strengthened foundation for the community.

The upcoming year will bring new capabilities, improved performance and maturity as well as more architecture support. Building on the modernization of the code base, the team is reinvestigating performance considerations across the major backends, updating them for the latest and upcoming hardware. A highlight for the year will be the upstreaming of support for NextSilicon's Maverick 2 accelerator. The fruit of a multi-year collaboration with NextSilicon, this milestone will present the first time that Kokkos's hardware support officially goes beyond CPU and GPU architectures. We are also looking forward to the upcoming community events the Kokkos team is participating in: a tutorial at SCAsia/HPCAsia conference in Japan, the HPSF Community Summit in Germany, and the Kokkos User Group Meeting at the HPSF Conference in Chicago being the next ones.

## Modules

Modules is a tool designed to help users dynamically modify their shell environment.

In 2025, when we joined the foundation, we placed a strong focus on clarifying the project's organization in order to lower the barrier to entry for newcomers. On the technical side, version 5.6 was released last summer, introducing a [module hierarchy mechanism](#). This feature makes it possible to organize modulefiles around a primary module path: loading certain modules dynamically enables additional module paths, while preserving a dependency relationship between the activated module and the modulefiles it exposes.





## Modules (continued)

This year, we plan to expand the project's Technical Steering Committee and hold meetings with the community to discuss and shape the roadmap. We are also excited to participate in the HPSF Conference in March, where we will host our first project session with the module community at large. As always, new releases are planned. Early in the year, we will focus on improving performance when evaluating large numbers of modulefiles. In addition, we intend to collaborate more closely with projects such as Spack and EESSI so they can benefit from our latest feature additions.

## OpenCHAMI

OpenCHAMI (Open Composable Heterogeneous Adaptable Management Infrastructure) is a system management platform designed to bring cloud-like flexibility and security to HPC environments.

In 2025 we added Dell to the governing board. Both Dell and HPE have added OpenCHAMI to their roadmaps for future systems showing the value of open collaboration. Independent HPC operators and large HPC integrators can both benefit from the improvements we've made in provisioning speed, hardware discovery, and post-boot configuration.



This year we plan to continue improving the existing microservices, especially around code-quality and troubleshooting. Partners are adding advanced inventory features for handling hardware changes over time and more robust power and console management. We'll host a community day at the HPSF Conference in March as well as a developer summit in the UK in May.

## Spack

Spack is a package manager for supercomputers, Linux, and macOS.

2025 was a big year for Spack. The long-awaited Spack 1.0 release in July included the broadest changes to Spack since the project's inception. Some of the most exciting changes centered around full dependency management for compilers and separating the packages repo from the code for the core tool.





## Spack (continued)

Spack 1.1, released in November, included substantial usability and performance improvements to go with the capabilities of version 1.0.

Looking forward into 2026, the Spack team is excited about upcoming changes to the installation UI and to provide better support to teams sharing a Spack instance.

## Trilinos

[Trilinos](#) is a high-performance software framework designed to support the development of scientific applications across diverse computational tasks, including linear algebra, optimization, differential equations, and mesh generation. With a focus on scalability and efficiency, Trilinos empowers researchers and engineers to address complex challenges in engineering, physics, and applied mathematics. Its modular architecture allows for customization and extension to meet the evolving needs of the scientific community.



In 2025, [Trilinos 16.2](#) officially retired its legacy sparse linear algebra package, Epetra, along with its associated capabilities, in favor of Tpetra, which utilizes templated types and leverages [Kokkos](#) for performance portability across diverse hardware architectures.

Looking ahead, Trilinos 17.0, scheduled for release in early 2026, will adopt Kokkos 5.0 and C++20 standards. Additionally, the March 2025 paper "[Trilinos: Enabling Scientific Computing Across Diverse Hardware Architectures at Scale](#)" highlights key architectural updates, advancements in performance portability, and community-driven collaboration.

In 2026, efforts will prioritize enhancing solver performance across physics disciplines and architectures, including GPU systems (e.g., [El Capitan](#)) and others (e.g., [Crossroads](#)). Support will continue for transitioning applications to the Kokkos-based Tpetra stack. CI/CD improvements will focus on containerization, [Spack](#) enhancements, and performance testing to reduce regressions. Development containers will also be provided to the community to simplify replication of CI builds and external contributions.



## Viskores

Viskores is a toolkit of scientific visualization algorithms for emerging processor architectures. It is designed to particularly excel on GPU processors and similar accelerators. It is designed with portability in mind by providing algorithm building blocks and abstractions that favor highly threaded environments. This design allows Viskores to be performant across diverse hardware.

In 2025 we transitioned our code from its predecessor library to Viskores. We released Viskores version 1.0, our first production release under the HPSF, and have since followed with Viskores 1.1. The new version adopts the C++17 standard and provides fixes to work well with C++20. More flexibility has also been added to the array management allowing Viskores to adapt to more memory layouts.

This year we plan to increase the rendering capabilities and make them available to more applications. We have made progress on implementing an ANARI rendering device, which would allow applications to leverage Viskores' rendering capabilities through a general API. The benefit will be an assurance that visualization applications using ANARI for 3D rendering can be assured that a device will always be available on HPC systems.

## WarpX

WarpX is an advanced Particle-In-Cell (PIC) code. It supports many geometries and numerical solvers to study kinetic multi-physics problems in particle accelerator, fusion, laser, plasma, space and astrophysics. As a highly-parallel and optimized code, WarpX runs on all major brands of GPU and CPUs and is used by scientists and industry from laptop to datacenter scale.

In 2025, we have generalized our particle data structures and upstreamed general methods to AMReX. We coordinated, reviewed and incorporated community contributions from diverse backgrounds and projects, enhancing numerical stability and increasing time-to-solution for many science cases.





## WarpX (continued)

We also further improved our usability with dedicated hackathon and "documentathon" sprints. We contributed to the inaugural HPSFcon and look forward to engaging even more in 2026.

This year, we plan to continue supporting the broad scientific user community of WarpX, expand our CI testing to GPUs and more architectures, continue to grow our packaging and add containerization support, add even more Python control on our C++ core functionality, and we continue to optimize WarpX performance on desktop, cloud and HPC hardware.

## Recognition & Impact

HPSF is an active and growing community shaping the future of HPC software. 2025 was a foundational year for HPSF with its debut conference, expanding membership and project roster, growing community engagement, and external recognition. Thank you to the HPSF [Governing Board](#), HPSF [Technical Advisory Council \(TAC\)](#), and community for making the past year such a success.



In November 2025, HPSF was honored as "[Best HPC Collaboration](#)" by the global HPC community in the HPCwire Readers' and Editors' Choice Awards, a strong external validation of HPSF's community-driven, open source mission. The recognition reflects growing adoption and trust in HPSF's role as a unifying force across labs, industry, and academia.



# Year Ahead

Looking ahead to 2026, the challenge (and opportunity) will be in scaling not just codebases, but collaboration, participation, and impact. With growing demand from AI, cloud HPC, and scientific computing, HPSF is poised to deliver.

We hope you will plan on joining us March 16-20, 2026 for the second edition of our [HPSF Conference 2026](#) back in Chicago and [submit to speak by January 11th](#). You'll also find us at lots of other events throughout the year. From SCA / HPCAsia 2026 in Osaka, Japan to HPSF Community Summit 2026 in Braunschweig, Germany. Keep up to date on where we'll be on our [events calendar](#) and be sure to sign up for the latest updates in your inbox with the [HPSF newsletter](#).

Whether you're a seasoned HPC user, a newcomer interested in performance-portable development, or simply curious about where high performance computing is headed, there's a place for you in HPSF. You are invited to [get involved in HPSF](#) by contributing to projects, joining working groups, having your organization become a member of the foundation, submitting to calls for proposals, finding us at events, and helping spread the word.

Cheers to a great year ahead!