STEP

SOFTWARE TOOLS
ECOSYSTEM PROJECT

Annual Report
CY2024

U.S. DEPARTMENT OF
ENERGY

Office of Science | Advanced Scientific Computing Research

Mission Statement

The mission of the Software Tools Ecosystem Project (STEP) is to enable critical software tools to proactively adapt to emerging platform technologies (such as new accelerators, storage devices, network technologies, and smart devices) and emerging application use cases (such as advanced machine learning and workflow frameworks) so that they continue to meet the needs of scientific computing and provide a strong foundation for future Advanced Scientific Computing Research activities. Our challenges include the wide breadth of our stakeholders and rapidly evolving platform technology dependencies.

Project Overview and Consortium Activities

STEP is a component of the Next Generation Scientific Software Technology (NGSST) program and the Consortium for the Advancement of Scientific Software (CASS). STEP seeks to ensure the stewardship and advancement of tools and supporting software for monitoring, analysis, and diagnosis of performance and behavior. The leadership team consists of three funded roles:

- STEP Director: Terry Jones, Oak Ridge National Laboratory
- STEP Deputy Director: Philip Carns, Argonne National Laboratory
- STEP Events & Workforce Expertise: Suzanne Parete-Koon, Oak Ridge National Laboratory

STEP holds monthly all-hands meetings to report status, coordinate activities, and disseminate information. Meetings are open not only to representatives from the STEP software portfolio, but also program management and members of the STEP Oversight Council who respectively provide feedback from ASCR and community stakeholders.

At the consortium level, STEP played a key role in the formulation of the Consortium for the Advancement of Scientific Software. Terry Jones is a founding member of the steering committee, and Phil Carns served as one of three co-facilitators for governance bootstrapping. STE P also leads two CASS working groups (the vendor forum working group and the facilities liaison working group) and participates in several others in support of cross-cutting efforts within the consortium. In addition, STEP members and software packages pursue a number of activities to bolster the tools community including:

- Hackathons at ALCF, LBL, and OLCF
- Several SC24 events, including a tools panel, a closely aligned workshop (SQCS), and BoFs (Americas HPC Collaboration; Adaptive & Greener HPC)
- The 2024 Scalable Tools Workshop at Lake Tahoe
- Active participation in international collaborations including JLESC and DOE-MEXT.

Overall Project Goals and Opportunities

STEP remains focused on achieving the following project goals:

- Deliver software tools that achieve greater effectiveness and efficiency for applications running on a spectrum of computing systems within the DOE complex – from laptops to leadership class machines
- Grow workforce expertise and shared experience
- Gain leverage with partners who possess like requirements
- Be a nexus of communication and information for tools stakeholders

We seek to be an agile project that can work effectively across a broad range of funding levels. At present, we are pursuing strategies that accommodate very limited resources. We are prioritizing essential tasks, utilizing creative solutions, and maintaining clear communication with stakeholders to ensure we're delivering the most impactful results.

If given the opportunity, our long term vision is to grow the scope of STEP by fostering engagement between community stakeholders and vendors to codesign hardware monitoring interfaces and software that affect tools, with the aim of enhancing efficient utilization of low-level performance monitoring and analysis capabilities. Moreover, we seek to support work through ongoing collaborative efforts by cross-cutting teams to address shared concerns across multiple tools and enhance long-term sustainability. We wish to support incubation of new technology as well as agile short-duration activities that have high potential to address rapidly evolving

needs and opportunities. Finally, to the extent possible, we aim to support community-wide stewardship and advancement activities with broad impact on the tools ecosystem, e.g. standards, with the aim of enhancing performance portability and enabling efficient use of diverse hardware platforms.

Software Package Portfolio

- PAPI .....Software package lead: Heike Jagode, University of Tennessee Heike's webpage
- HPCToolkit....Software package lead: John Mellor-Crummey, Rice University John's webpage
- TAU .....Software package lead: Sameer Shende, University of Oregon Sameer's webpage
- Dyninst .....Software package lead: Bart Miller, University of Wisconsin Bart's webpage
- Darshan .....Software package lead: Shane Snyder, Argonne National Laboratory Shane's webpage

Each software package has detailed accomplishments and plans included later in this report:

STEP Team Members

NAME	PERSPECTIVE	NAME	ORGANIZATION
Lead PI	Tools	Terry Jones	Oak Ridge National Lab
Deputy PI	Tools	Philip Carns	Argonne National Lab
Co-PI	Tools	James Brandt	Sandia National Laboratories
Co-PI	Vendors	James Custer	Hewlett Packard Enterprise
Co-PI	Tools	Ann Gentile	Sandia National Laboratories
Co-PI	Facilities	Richard Gerber	Lawrence Berkeley National Lab
Co-PI	Facilities	Kevin Harms	Argonne National Lab
Co-PI	Tools	Heike Jagode	University of Tennessee
Co-PI	Tools	Mike Jantz	University of Tennessee
Co-PI	Tools	Matthew Legendre	Lawrence Livermore Natl Lab
Co-PI	Vendors	John Linford	NVIDIA
Co-PI	Vendors	Keith Lowery	Advanced Micro Devices
Co-PI	Facilities	Verónica Melesse Vergara	Oak Ridge National Lab
Co-PI	Tools	John Mellor-Crummey	Rice University
Co-PI	Tools	Barton Miller	University of Wisconsin
Co-PI	Vendors	José Moreira	IBM
Co-PI	Applications	Erdal Mutlu	Pacific Northwest Natl Lab
Co-PI	Facilities	Suzanne Parete-Koon	Oak Ridge National Lab
Co-PI	Tools	Sameer Shende	University Oregon
Co-PI	Tools	Shane Snyder	Argonne National Lab
Co-PI	Tools	Galen Shipman	Los Alamos National Laboratory
Co-PI	Tools	Devesh Tiwari	Northeastern University
Co-PI	Applications	Theresa Windus	Ames National Lab

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under the Exascale Computing Program *(17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration,* and the Next Generation Software Technologies portfolio (ERKJ409).

HPCToolkit

Description

HPCToolkit is an integrated suite of tools for measurement and analysis of application performance on computers ranging from multicore desktop systems to GPU-accelerated supercomputers running applications at exascale. On CPUs, HPCToolkit profiles and traces applications by using statistical sampling of timers and hardware performance counters. On GPUs, HPCToolkit leverages vendor software frameworks for monitoring and analysis of GPU operations and instruction-level monitoring within GPU kernels.

Context

HPCToolkit is dependent upon vendor provided interfaces for measuring code performance on GPUs. Over the last 12 months, the HPCToolkit project team has engaged with both AMD and Intel about bugs and missing capabilities in their measurement interfaces.

The project team provided feedback to AMD about pre-releases of their emerging ROCprofiler-SDK interface, which supports profiling and tracing of GPU operations, time-based PC sampling within GPU kernels, and support for using hardware performance counters to measure events as kernels execute. AMD has been receptive to feedback and the ROCprofiler-SDK interface is nearing completion for its first full release.

The project team engaged Intel about their tooling interfaces for in Level Zero and their emerging PTI-GPU measurement substrate for profiling and tracing GPU operations.

- Attempts to measure LAMMPS at large scales on Aurora revealed a serious problem with Level Zero: GPU kernels initiated sometimes don't report completion. A reproducer of the problem provided to ALCF that gathers call stacks across MPI ranks in a deadlocked execution to show that some ranks ar stuck awaiting kernel completion while others await them in a collective communication.
- Throughout 2024, there have been issues recording Intel GPU binaries for port-mortem analysis of instruction-level performance measurements. For much of the year, Intel GPU binaries were reported as relocatable binaries with no documentation about how to work with them. In the fall of 2024, Intel provided capabilities to recover GPU dynamically-linked binaries from GTPin and Level Zero; however, GPU binaries differ across MPI ranks because the relocation strategy used by Intel is based on the global state of the memory allocator in each rank. This last issue remains unaddressed, which makes it infeasible to measure and attribute instruction-level metrics in large scale executions on Aurora.
- Intel's PTI-GPU interface is roughly half finished. It can measure and trace the execution of GPU operations; however, it currently lacks an API that provides a callback when a GPU operation is initiated; this makes it difficult to attribute metrics for GPU operations back to the calling contexts where they are initiated. The project team contacted the developer of this interface but she has been directed to work on tool support for AI frameworks and hasn't had time to add the key missing capability.
- Currently, Intel's GTPin binary instrumentation tool fails for LAMMPS binaries compiled with certain options. Acceptance testing on Aurora began before a reproducer could be provided to Intel. One will be provided in 2025.

These challenges prompt the question *how do these problems impact ongoing work or the usability of the package on Frontier, El Capitan, and Aurora*? AMD has been adapting their rocprofiler-sdk in response to feedback from the HPCToolkit project, the Tau project, and the PAPI project. While important work is ongoing, STEP is largely satisfied with the functionality in AMD's rocprofiler-sdk at present. During the month of January, Mellor-Crummey tested and integrated support for HPCToolkit on LLNL's MI300A-based El Capitan system to make sure that everything worked appropriately with national security applications on El Capitan and related test-and-delivery systems at LLNL. A first of its kind system, Aurora has the most remaining work. HPCToolkit is collaborating closely with Intel to identify gaps and opportunities; this is a collaboration exemplar where a successful outcome is critical to not only HPCToolkit, but also the needs of the broad HPC community to realize computing's potential for scientific discovery.

CY2024 Activities

The project team developed a completely new implementation of HPCToolkit's monitoring substrate for AMD GPUs based on AMD's emerging ROCprofiler-SDK. The new measurement substrate in HPCToolkit supports profiling and tracing of GPU operations on AMD GPUs, measurement and attribution of activity within GPU kernels using time-based PC sampling, as well as measurement of GPU kernel executions with hardware

counters. Engagement between the project team and AMD yielded a uniquely powerful capability for tagging each PC sample with a correlation ID. The correlation ID on each PC sample enables HPCToolkit to precisely attribute metrics for the measured instruction back to the calling context where the kernel was launched, even when kernels execute concurrently.

The project team developed support in HPCToolkit for instruction-level performance measurement and attribution on Intel's Ponte Vecchio GPUs – the GPUs present in ALCF's Aurora supercomputer. The implementation uses the Metric Streamer capability provided by Intel's Level Zero runtime to collect PC samples. Unlike PC samples for AMD GPUs, PC samples on Intel GPUs lack a correlation ID for association with a kernel launch. To associate PC samples back to their kernel launch context, the initial implementation of PC sampling for Intel GPUs in HPCToolkit serializes kernel executions. Work is underway to explore an alternate approach for associating PC samples with launch context using statistical methods, trading precise attribution of samples for reduced measurement overhead and intrusion.

The code for the aforementioned capabilities is currently undergoing quality assurance testing in branches. Integration into the main branch of HPCToolkit's GitLab repository is expected in January 2025.

HPCToolkit associates PC samples collected on either AMD or Intel GPUs with detailed calling contexts inside GPU kernels that include device functions, inlined functions and templates, loops, and statements. This is a unique capability for both architectures.

The project team was also involved with both in-person and virtual GPU hackathons on Frontier and Aurora throughout the course of 2024. At the hackathons, the project team helped application developers employ HPCToolkit to measure and analyze performance of their applications.

Impact, Sustainability, and Quality

Property	State
Impact	**Goal:** Provide instruction-level measurement and analysis of GPU computations on Frontier and Aurora. **Accomplishments:** The project team extended HPCToolkit with support for instruction-level measurement using PC sampling on both AMD and Intel GPUs. HPCToolkit is the first tool to attribute instruction-level performance measurements within GPU kernels back to heterogeneous calling contexts (spanning both CPU and GPU) that include functions, call sites, inlined functions, loops, and statements.
Sustainability	**Goal:** Work with the High Performance Software Foundation (HPSF) to develop sustainability strategies. **Accomplishments:** The HPCToolkit project PI is a member of the technical advisory committee of HPSF. A member of the project team is participating in the HPSF working group on continuous integration (CI), which is looking to leverage Spack, Hubcast, and Jacamar to support testing of HPSF software (including HPCToolkit) at HPC facilities such as DOE labs. During 2024, the HPCToolkit project has implemented continuous deployment/delivery of HPCToolkit's hpcviewer graphical user interface.
Quality	**Goal:** use HPCToolkit's emerging Python API to develop support for validating measurement data **Accomplishments:** HPCToolkit's Python API has been integrated with the LLNL's Hatchet, which supports analysis of individual experiments, and LLNL's Thicket which supports analysis of multiple experiments. A poster about this work was nominated for best research poster at SC24

Software Quality Characteristics

Website	Documentation	Repository	Test Suite	Spack	E4S	Smoke Test.	Durability
HPCToolkit	Documentation	Repository	☑	Spack	☑	☑	☑

Software Features Implemented In CY2024

				
2024.01.1 Release of HPCToolkit. Installed on systems at ALCF, OLCF, LLNL.	☐Plan ☐Code	■Build ■Test	■Release ■Deploy	■Operate ■Monitor
Refined hpcanalysis Python interface to performance data	■Plan ■Code	■Build ■Test	☐Release ☐Deploy	☐Operate ☐Monitor
Extended HPCToolkit with support for PC sampling on Intel GPUs. Overhauled HPCToolkit's support for AMD GPUs, including profiling, tracing, hardware counters and PC sampling.	■Plan ■Code	■Build ■Test	☐Release ☐Deploy	☐Operate ☐Monitor

Workforce Status

At the end of 2024, the HPCToolkit project team includes the PI (John Mellor-Crummey), two male PhD research staff, and three PhD students – two of whom are women. During 2024, the project hosted two student academic visitors – one for most of the year and another for the summer. During 2024, the HPCToolkit project had to let go an external contractor and one staff member because STEP funding was 2/3 less than ECP funding for HPCToolkit. With the emergence of ML/AI on HPC systems, members of the project team will need additional exposure to AI/ML frameworks and computations.

Consortium and Project Engagement

The STEP project provides partial support for development of HPCToolkit, Dyninst, PAPI, Tau, and Darshan. Common funding facilitates coordinated development. The HPCToolkit team has worked closely with members of the Dyninst team to analyze and remediate root causes of serialization when analyzing the machine code for Aurora's default MPI library; this work is ongoing. The PAPI team has addressed the inability to concurrently measure multiple hardware counter metrics for kernels executing on Intel GPUs – a problem identified by the HPCToolkit team. The HPCToolkit team provided code to the Tau project team to add support for PC sampling to Tau.

HPCToolkit provides support for performance measurement and analysis on the DOE's exascale supercomputers. This is well-aligned with the mission and needs of STEP and CASS. In the spring of 2024, the ExaWind project team used HPCToolkit to measure ExaWind on over 1K nodes at a Frontier hackathon. Insights gained by the project team using HPCToolkit combined with guidance from hackathon mentors led to a 28x improvement in ExaWind at scale.

CY2025 Goals

(1) Work with AMD to finalize ROCprofiler-SDK support for MI300. Implement HPCToolkit support for PC sampling on MI300, including collection and attribution of information about instruction stalls.
(2) Work with Intel to resolve showstoppers that present post-mortem analysis of instruction-level measurements on Aurora.
(3) Deploy new HPCToolkit capabilities for instruction-level performance monitoring at scale on Aurora, Frontier, and El Capitan.
(4) Improve HPCviewer user interface to better support analysis of large-scale performance data.
(5) Mentor developers in hackathons and trainings at DOE laboratories.

PAPI

Description

The Performance API (PAPI) is a software library that offers a universal interface for accessing hardware counters, enabling the measurement and optimization of software performance across various computing architectures. Specifically, PAPI enables monitoring of low-level performance metrics and offers power management support, including reading and capping, for the latest CPUs and GPUs from various hardware vendors. It also provides insights into network congestion for different interconnects and supports monitoring of I/O activities. Lastly, PAPI's software-defined event support helps uncover critical software events from various libraries and runtimes.

Context

By integrating vendor-specific, low-level monitoring capabilities for the latest architectures into PAPI, the team collaborates closely with various hardware vendors on a regular basis. For example, the team provided extensive feedback to AMD on the design and functionality of the new AMD ROCprofiler-SDK. This collaboration led to updates in the AMD ROCm software to provide more useful and targeted insights for performance evaluations on the latest GPUs/APUs, such as MI300.

CY2024 Activities

In 2024, PAPI was enhanced with a new component, "rocp_sdk", which supports AMD GPUs/APUs through the ROCprofiler-SDK interface. This provides users with early access to preliminary support for AMD ROCprofiler-SDK events. PAPI has also been enhanced with monitoring capabilities for the AMD Zen5 core and L3 subsystems; basic support for the RISC-V architecture has been developed; and initial support for heterogeneous CPUs has been implemented (E.g., Intel Alderlake and Raptorlake can now enumerate events for both Power (P) and Efficiency (E) Cores on heterogeneous systems. Other features that have been added are: core support for Intel Alderlake Gracemont (E-Core) and Goldencove (P-Core), as well as Raptorlake (P and S), and GraniteRapids. Additionally, uncore support has been implemented for SapphireRapids and Icelake-X.

Impact, Sustainability, and Quality

Property	State
Impact	**Goal:** PAPI integrates new capabilities for emerging and advanced technologies, ensuring it is well-positioned to address the evolving needs of the HPC community and potentially expand its user base. **Accomplishments:** PAPI integrated AMD’s new ROCprofiler-SDK for MI300, monitoring for AMD Zen5, basic RISC-V support, and initial heterogeneous CPU support for Intel Aldelake and Raptorlake. It also added core support for different Alderlake and Raptorlake chips, GraniteRapids, and uncore support for SapphireRapids and Icelake-X.
Sustainability	**Goal:** PAPI's development follows sustainable software practices, including open collaboration via pull requests, CI testing, integration into Spack and E4S, and attentiveness to stakeholder needs. **Accomplishments:** For each 2024 activity, PAPI’s sustainable software practices have been achieved, which can be tracked in the PAPI Git repository.
Quality	**Goal:** PAPI's quality will be assessed by its integration of monitoring for emerging technologies, timely availability, and ease of use. **Accomplishments:** Support for AMD MI300 was prioritized and achieved through NDA access to AMD's ROCprofiler-SDK. Intel core and uncore support was promptly integrated upon access to new hardware. AMD Zen5 support was developed without direct hardware access, relying on industry partners to test and validate the pre-release PAPI development.

Software Quality Characteristics

Website	Documentation	Repository	Test Suite	Spack	E4S	Smoke Test.	Durability
PAPI	Documentation	Repository	☑	Spack	☑	☑	

Software Features Implemented In CY2024

				
Development of a new PAPI component, "rocp_sdk", which supports AMD GPUs/APUs (e.g., MI300) through the ROCprofiler-SDK interface.	■Plan ■Code	■Build ■Test	■Release ☐Deploy	☐Operate ☐Monitor
PAPI integration of (a) AMD Zen5, (b) core support for numerous Intel CPUs: Alderlake, Raptorlake, GraniteRapids, and (c) uncore support for SapphireRapids and Icelake-X.	■Plan ■Code	■Build ■Test	■Release ■Deploy	■Operate ☐Monitor
PAPI development for basic RISC-V support, and initial heterogeneous CPU support for Intel Aldelake and Raptorlake.	■Plan ■Code	■Build ■Test	■Release ☐Deploy	☐Operate ☐Monitor

Workforce Status

The PAPI team benefits from a diverse mix of senior and early-career researchers, as well as grad students, offering both experience and fresh perspectives. However, funding instability leads to the loss of experienced senior researchers, which poses a much greater challenge than student turnover, especially for a software library like PAPI that the HPC community relies on and, sadly, expects to be updated, even without dedicated funding.

Consortium and Project Engagement

All newly developed and integrated performance monitoring capabilities in PAPI are automatically available to high-level performance toolkits that rely and utilize PAPI under the covers (e.g., TAU). Apart from STEP, PAPI currently has limited integration with other CASS ecosystems. I believe PAPI aligns well with STEP's mission and needs; otherwise, I trust we would receive timely feedback from our leadership if that were not the case.

CY2025 Goals

(1) Continue collaborating with AMD to address the remaining issues with the AMD ROCprofiler-SDK for both dispatch and sampling modes, ensuring all monitoring capabilities are fully available to users on the MI300.
(2) Finalize the development and integration of predefined events, known as "PAPI Presets," for GPU components, with an initial focus on NVIDIA GPUs. Current Hopper architectures offer approximately 200,000 events per GPU. The goal of “PAPI Presets for GPUs” is to organize this large number of raw performance counter events into a more user-friendly and functional format.
(3) Develop integration of PAPI's new ROCprofiler-SDK component into Spack and E4S, including "smoke tests" to quickly verify the basic functionality of all new PAPI features.

Dyninst

Description

Dyninst (and its related toolkits) is a binary code and instrumentation facility. It provides the ability to analyze binary code (both control and dataflow analyses) instrument the code efficiently. Dyninst also allows for modification of the executable. It supports such use cases as performance profiling, debugging, testing, tracing, security auditing and hardening, and architectural simulation. Its primary CPU platforms are x86, ARM, and Power.

Currently, Dyninst provides instrumentation support for HPCToolkit and (in some modes) for TAU. It is also the foundation for the LLNL STAT scalable debugger (in production and demonstrated effective at 5 million cores). Cray/HPE tools, including the ATP (abnormal termination processing debugger) and differential debugger are based on our toolkits. In addition, the Red Hat SystemTap monitoring tool is based on Dyninst, as is the new AMD Omnitrace. Currently in flight is support for the AMD GPUs allowing all current Dyninst tools to move to this platform. We also have code analysis (but not yet instrumentation) support in progress for NVIDIA and Intel GPUs.

Context

During this past year, work on Dyninst has focused on three areas: (1) support analyzing and instrumenting evolving CPU architectures, (2) development of new support for GPU code analysis and instrumentation (with a focus on AMD GPUs), and (3) foundational work to improve Dyninst's modularity, extensibility, efficiency, and reliability.

CY2024 Activities

CPU code analysis and instrumentation: support the majority of leadership class systems.

GPU code analysis and instrumentation: finish code analysis for AMD GPU families and develop code instrumentation for these GPUs.

Dyninst infrastructure: Continue full CI, testing, and release process. Start planning for moving to the Capstone instruction decoding framework to try to reduce ongoing overhead of tracking new architectures.

Impact, Sustainability, and Quality

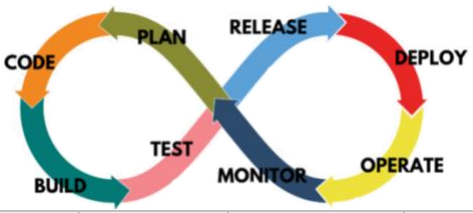
Property	State
Impact	**Goal:** Continue to support the variety of tools that depend on Dyninst and support the leadership class systems. **Accomplishments:** We work closely with STEP tool groups such as HPCToolkit and TAU, planning new features and responding to bug reports in a timely fashion. For support of leadership class systems, we currently handle the following CPU architectures: AMD 2nd and 3rd gen EPYC; Intel Sapphire Rapids, Cooper Lake, Raptor Lake Skylake; Arm v8 and v9; Power 9 and 10. This covers El Capitan and Frontier.
Sustainability	**Goal:** To maintain a level of funding that allows for Dyninst to continue as an ongoing venture that supports active tool projects on the relevant leadership class systems. **Accomplishments:** • Dyninst is an open source project based in Github with contributors from academia, industry and government labs. New features and fixes have been accepted from across the HPC tool developer and vendor communities (with at least a dozen organizations and individuals making contributions). Version 13.0 (the first with GPU features) was released March 2024. • We continue working to build a diverse funding base. In addition to STEP, we all have funding from the Tri-Labs (LLNL) and industry (AMD), with new funding in progress from the European Union (for RISC-V support).

Property	State
Quality	**Goal:** Maintain release-quality code for all of our **Accomplishments**: We track issues and features, conduct cross-team code reviews, provide extensive test suits, have documentation on all features of our toolkits, and integrate in a CI workflow. Our repository site provides concrete statistics on our effectiveness in addressing issues, and our test suite dashboard provides a public view of the status of the Dyninst toolkits.

Software Quality Characteristics

Website	Documentation	Repository	Test Suite	Spack	E4S	Smoke Test.	Durability
Dyninst	Documentation	Repository	☑	Spack	☑	☑	☑

Software Features Implemented In CY2024

				
Code analysis for AMD GPUs (GFX908, GFX90A, and GFX940)	■Plan ■Code	■Build ■Test	☐Release ☐Deploy	☐Operate ☐Monitor
Basic instrumentation demonstrated (pre-release) for AMD GPUs.	■Plan ■Code	■Build Test	Release Deploy	Operate ☐Monitor
Design and initial coding of integrating the Capstone instruction decoding framework.	■Plan ■Code	■Build Test	Release Deploy	Operate ☐Monitor

Workforce Status

The Dyninst effort under STEP currently supports only part (43%) of one full time staff member and no graduate research assistants. The staff member helps with continuity of development, but the current level of funding is less than needed to properly support the full space of users and leadership class systems.

Consortium and Project Engagement

We work closely with other tool projects in STEP, especially the HPCToolkit and TAU efforts.

CY2025 Goals

1. Have full static binary instrumentation (binary rewriting) of AMD GPU code.
2. Integrate the Capstone instruction decoding framework.
3. Complete refactoring of Dyninst to more effectively support cross-platform instrumentation (needed for GPU code).

TAU

Description

TAU is performance evaluation tools that supports parallel profiling and tracing on HPC systems. It requires no changes to the application source code, build system, or binary. It also supports instrumentation using a plugin for the LLVM compiler, source instrumentation using the Program Database Toolkit (PDT), and binary rewriting using DyninstAPI. It supports automatic instrumentation for parallel programs written in C++, C, Fortran, Python, Java, Chapel, and UPC. Instrumentation can also be inserted using third party timing libraries such as CAMTIMERS, PETSc, Perfstubs, Caliper, PhiProf, ROCTx, NVTx, and Score-P are supported. It supports instrumentation of OpenMP using OMPT including target offload directives, ROCm using RocProfiler and RocTracer, CUDA using CUPTI, Kokkos, OpenCL, and Python. TAU interfaces with PAPI to access hardware performance counter data and supports direct instrumentation as well as event-based sampling.

Context

TAU is a widely used performance evaluation tool that has evolved over time to support most profiling and tracing packages on DOE computing systems at ALCF, OLCF, NERSC, Sandia, LLNL, and LANL. TAU supports instrumentation of application binaries at the routine level. Support for advanced instrumentation and Continuous Integration/Continuous Deployment (CI/CD) is important for sustaining TAU. Support for GPUs from Intel, AMD, and NVIDIA for CI/CD is key to sustain the TAU project.

CY2024 Activities

To advance the capabilities of dynamic instrumentation in TAU using DyninstAPI for targeted instrumentation of dynamic shared objects (DSOs) and initial support for CI/CD for GPUs are CY2024 activities.

Impact, Sustainability, and Quality

Property	State
Impact	**Goal:** To improve the instrumentation capabilities of TAU using library level instrumentation of un-modified DSOs. **Accomplishments:** TAU now supports rewriting of DSOs using tau_run, a tool that is developed using the DyninstAPI package. This is important for instrumentation of third-party libraries like HDF5 that are linked with an un-modified application binary. It helps generate performance data for the library without any modification to the source code, build system, or executable.
Sustainability	**Goal:** To improve TAU's support for GPUs. **Accomplishments:** TAU's support for Intel, AMD, and NVIDIA GPUs has been updated. New versions of TAU (v2.32.1, v2.32.2, and v2.33) have been released in CY24.
Quality	**Goal:** To improve TAU's support for CI/CD on Intel, AMD, and NVIDIA GPUs. **Accomplishments:** TAU uses Frank at the University of Oregon for CI/CD. In CY24, support for Intel, AMD, and NVIDIA GPUs was added with tests run on 10 GPU architectures daily. These tests may be viewed at https://gitlab.e4s.io/uo-public/tau/-/pipelines.

Software Quality Characteristics

Website	Documentation	Repository	Test Suite	Spack	E4S	Smoke Test.	Durability
TAU	Documentation	Repository	☑	Spack	☑	☑	☑

Software Features Implemented In CY2024

				
Support for instrumentation of DSOs in TAU.	■Plan ■Code	■Build ■Test	■Release ☐Deploy	☐Operate ☐Monitor
Support for CI/CD on Intel GPUs.	■Plan ■Code	■Build ■Test	■Release ■Deploy	■Operate ☐Monitor
Support for CI/CD on AMD GPUs.	■Plan ■Code	■Build ■Test	■Release ■Deploy	■Operate ☐Monitor
Support for CI/CD on NVIDIA GPUs.	■Plan ■Code	■Build ■Test	■Release ■Deploy	■Operate ☐Monitor
Integration of TAU in E4S and Spack.	■Plan ■Code	■Build ■Test	■Release ■Deploy	■Operate ☐Monitor

Workforce Status

There were no changes in personnel in CY24.

Consortium and Project Engagement

The TAU team has worked closely with the E4S and Spack teams under the CASS ecosystem and is well aligned with the mission and needs of STEP and CASS. Further integration with DyninstAPI, PAPI, and joint tutorials with HPCToolkit are planned.

CY2025 Goals

In CY2025, TAU will improve on the instrumentation capabilities with DyninstAPI to support selective instrumentation with DSOs and improve on extent of CI/CD testing with GPUs from three vendors.

1. Support for selective instrumentation in rewriting DSOs with DyninstAPI at the routine level.
2. Support for MPI in nightly testing of TAU with Intel GPUs.
3. Support for MPI in nightly testing of TAU with AMD GPUs.
4. Support for MPI in nightly testing of TAU with NVIDIA GPUs.

Note that while TAU is a mature toolkit supporting GPUs from three vendors, they are making changes to their GPU runtimes and introducing new APIs. AMD has introduced the rocprofiler-SDK in ROCm 6.3.2 and this has necessitated a re-engineering of TAU's measurement layer significantly. Binary rewriting of shared libraries and executables to improve TAU's usability requires additional effort. Integration with CI systems for improving TAU's software quality is key to its usage. Porting TAU to evolving platforms and runtime systems remains an important goal to sustain the tool.

Darshan

Description

Darshan is a lightweight I/O characterization tool designed for HPC applications. It captures a concise summary of I/O behavior for every job it instruments, detailing application usage of various layers of the I/O stack. Darshan is popular among HPC application and I/O library developers, system administrators, and other I/O practitioners as it's able to seamlessly quantify I/O characteristics of their jobs.

Context

To provide comprehensive understanding of HPC I/O behavior, tools like Darshan must capture details about I/O workloads across the entire storage stack (including high-level libraries, low-level system interfaces, and file system-specific interfaces) and summarize/present this information to users. The focus of this year's work has been on extending Darshan to capture and summarize detailed I/O characteristics related to application usage of Lustre and DAOS storage systems. These are two of the most popular HPC storage systems deployed at ASCR facilities, and understanding application usage of these systems is key to extracting high performance and efficiency.

CY2024 Activities

Improve design and capabilities of PyDarshan analysis tools, enhance Darshan instrumentation of Lustre to account for new features (e.g., PFL), and add new corresponding CI testing capabilities. We will work with the STEP leadership to refine our metrics and identify methods to clearly demonstrate how our efforts lead to measurable increases.

Impact, Sustainability, and Quality

Property	State
Impact	**Goal:** Improved actionable feedback provided by Darshan analysis tools **Accomplishments:** • Enhanced Dashan analysis tools to summarize key details on applications' usage of Lustre and DAOS storage systems.
Sustainability	**Goal:** CI testing to ensure ongoing correctness **Accomplishments:** • Updated Darshan's Github development workflow to run extensive regression tests on proposed code changes. • Added regular CI testing on ALCF Polaris system to ensure Darshan functionality in production environments.
Quality	**Goal:** instrumentation support for newest HPC storage technologies (e.g., recent Lustre updates) **Accomplishments:** • Added comprehensive instrumentation of Lustre and DAOS storage APIs to better understand application usage of these systems in production.

Software Quality Characteristics

Website	Documentation	Repository	Test Suite	Spack	E4S	Smoke Test	Durability
Darshan	Documentation	Repository	☑	Spack	☑	☑	

Software Features Implemented In CY2024

				
Enhanced instrumentation of new Lustre file striping features (e.g, progressive file layouts).	■Plan ■Code	■Build ■Test	■Release Deploy	Operate Monitor
Developed new instrumentation modules for DAOS's native object API, to understand usage of this new low-level storage API.	■Plan ■Code	■Build ■Test	Release Deploy	Operate Monitor
Extended Darshan job analysis tools to summarize and present key information related to application usage of DAOS APIs.	■Plan ■Code	■Build ■Test	Release Deploy	Operate Monitor

Workforce Status

Funding levels for Darshan in the STEP project are only sufficient for providing partial support for a single developer. However, the Darshan team provides mentoring to junior staff, hosts student interns when possible, and makes sure to promote diversity and inclusivity when possible.

Consortium and Project Engagement

Darshan has integrated well into the STEP and CASS ecosystems, providing the tooling needed for understanding and improving HPC storage access, a critical bottleneck for data-intensive applications. Collaboration between STEP software projects and CASS enables collective efforts in uncovering and addressing issues related to sustaining scientific software, particularly from the perspective of software tools. A key missing piece in these sustainability efforts is the availability of comprehensive CI resources that would enable testing of diverse data management software environments.

CY2025 Goals

Impact: Develop processes to anonymize/publish Darshan log data captured in production at the ALCF (and potentially other ASCR facilities) for sharing with the broader HPC I/O research community

Sustainability: Modernize and improve Darshan's documentation (i.e., migrate to Read the Docs and provide up-to-date details on how to install/deploy Darshan and analyze its data) and release processes (i.e., using GitHub automation)

Quality: Productize Darshan mechanisms for instrumenting I/O workloads at thread-level granularity