STEP

SOFTWARE TOOLS ECOSYSTEM PROJECT

Introducing STEP:
A Presentation to the Facilities Software Task Force

January 8, 2024

Terry Jones, STEP Director

Introducing the Software Tools Ecosystem Project (STEP)

- STEP In A Nutshell
- The Tools Ecosystem and Why It’s Special
- The 5 Initial Tools under STEP stewardship
- A look toward next steps

STEP in a nutshell…

Lead PI: Terry Jones
Deputy PI: Phil Carns

Objectives and Scope

- Tools and supporting software for monitoring, analysis, and diagnosis of performance and behavior of codes on advanced computing systems.
- Examples: application profilers, tracing tools, system monitors, etc.
- Co-design with hardware vendors, application developers, facilities and tool developers.

Initial Funded Software Packages

Software Tool	PI
HPCToolkit	John Mellor-Crummey, Rice Univ.
PAPI	Heike Jagode, Univ. of Tennessee
Dyninst	Barton Miller, Univ. of Wisconsin
Tau	Sameer Shende, Univ. of Oregon
Darshan	Shane Snyder, Argonne

Challenges

- These tools rely on a deep understanding of hardware, and hardware is rapidly changing.
- Vendors are reluctant to divulge hardware details.
- Stakeholders are very diverse and currently have insufficient communication paths.

Opportunities

- STEP can be a vehicle to improve communication among tool developers, application teams, vendors and facilities.
- STEP can be a clearing-house for efficiently managing critical HPC technology.
- STEP can be a resource for workforce development.

Defining *Stewardship and Advancement*

- The "***ongoing*** support and enhancement needed for critical HPC tools to remain ***effective, efficient, and relevant*** in the rapidly evolving field of HPC."

- **In Scope**: ***Applied & targeted*** work on a software tool that is currently utilized or will be utilized in an HPC context. STEP relies on research to address specific needs.

- **Out of Scope**: Research and proof of principle work that is ***hypothesis-based & exploratory*** in nature. To build knowledge for the sake of knowing or investigating, rather than to solve a problem.

STEP
SOFTWARE TOOLS
ECOSYSTEM PROJECT

STEP Specifically Targets ***Tools***

WHAT ARE *TOOLS*?

- We define Tools to mean "the collection of *tools and utilities that can be applied to both understanding performance bottlenecks and facilitating run time mitigation of performance degrading phenomena.*"

- These tools interact with hardware, compilers, communication libraries, programming model runtimes, operating systems, middleware layers, as well as the many applications that utilize these tools.

What's ***So Important*** about Tools?

- *As computers have increased in complexity and scale, using them effectively has become much more difficult.*

- In addition to their role in enabling supercomputer performance (a decisive determinant of scientific discovery), these tools provide essential feedback to users, operations staff, and system and application software developers.

How Are Tools **Unique**?

- Tools are closely bound to architectures and system software in ways that other types of software, such as libraries and scientific applications, are not.
- For example, a tool that tracks how an application uses computing resources must be able to measure low-level architectural events and metrics and relate them to program progress and source code.
- The need for tools is most acute for understanding code performance on systems that push the boundaries of technology and scale, but these systems' novelty makes them extremely difficult for tool developers to support when first deployed.

STEP and Facilities

- STEP was designed with Facilities in mind: STEP facility representatives participated in the development of STEP as investigators

Richard Gerber, LBNL	Kevin Harms, Argonne	Matthew Legendre, LANL
Susanne Parete-Koon, ORNL	Galen Shipman, LANL	Veronica Melesse Vergara, ORNL

- Supporting the activities that are named in the initial Y1 plans that are heavily geared towards sustainability – See key activities for HPCToolkit, PAPI, Dyninst, TAU and Darshan.
- A convenient interface when Facilities needs to broadly reach the tools community with a high-level concern.
- A mechanism to address needs and requirements: give STEP input on things that are important for the tools portfolio for out years, STEP will seek software tool solutions.
- A source for tools information and assistance (e.g., software aspects of machine procurement).

HPCToolkit: Performance Measurement and Analysis of Complex Applications at Scale

Current Capabilities on GPU-accelerated Architectures

NVIDIA

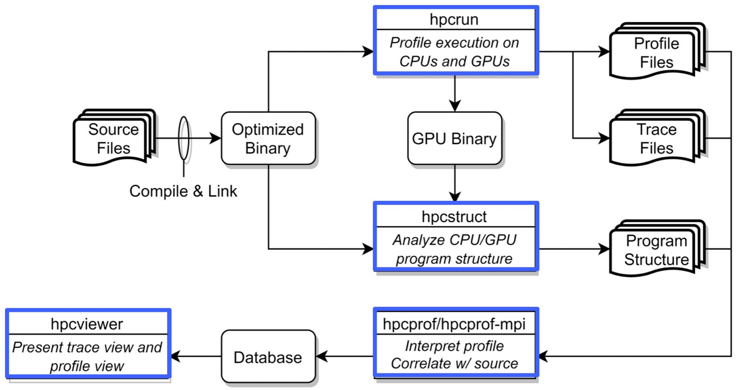
- heterogeneous profiles
- GPU instruction-level execution and stalls using PC sampling
- heterogeneous traces

AMD

- heterogeneous profiles
- no instruction-level measurements within GPU kernels
- measure OpenMP offloading using OMPT interface
- hardware counters to measure kernels
- heterogeneous traces

Intel

- heterogeneous profiles
- GPU instruction-level measurements with instrumentation; heuristic latency attribution to instructions
- measure OpenMP offloading using OMPT interface
- heterogeneous traces



Source Files

Compile & Link

Optimized Binary

hpcrun
Profile execution on CPUs and GPUs

GPU Binary

hpcstruct
Analyze CPU/GPU program structure

Profile Files

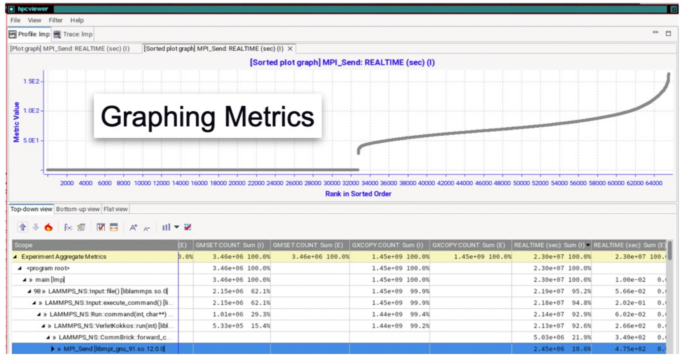
Trace Files

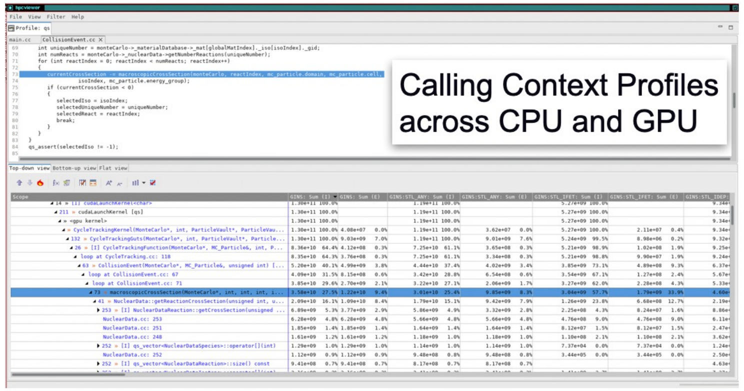
Program Structure

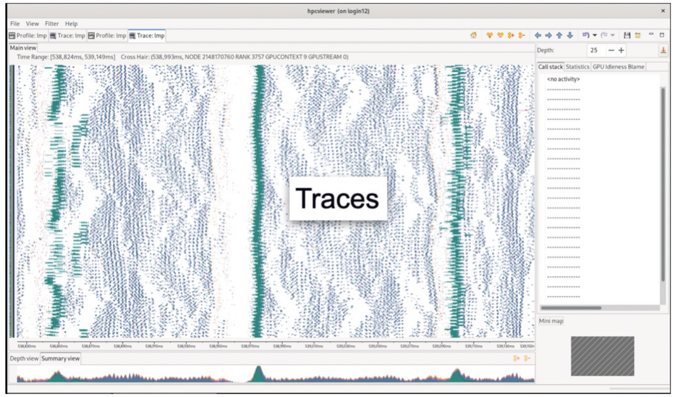
hpcprof/hpcprof-mpi
Interpret profile Correlate w/ source

Database

hpcviewer
Present trace view and profile view







Key Plans for 2024

- *Overhaul HPCToolkit to use emerging completely new AMD and Intel GPU APIs*
- *Extend HPCToolkit to exploit emerging support for instruction-level measurement of GPU computations using PC sampling on AMD and Intel GPUs*

STEP SOFTWARE TOOLS ECOSYSTEM PROJECT

HPCToolkit

RICE

https://ascr-step.org

Slide source: John Mellor-Crummey
presentation to the Facilities Software Task Force

PAPI: Performance API

What is PAPI?

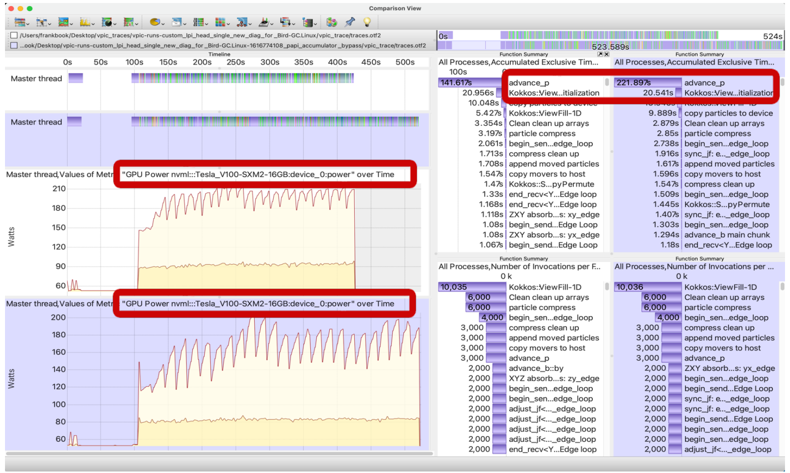
- PAPI provides a **universal interface** and methodology for **monitoring performance counters** and **managing power** across a diverse range of **hardware** and **software** components.

Monitoring Capabilities

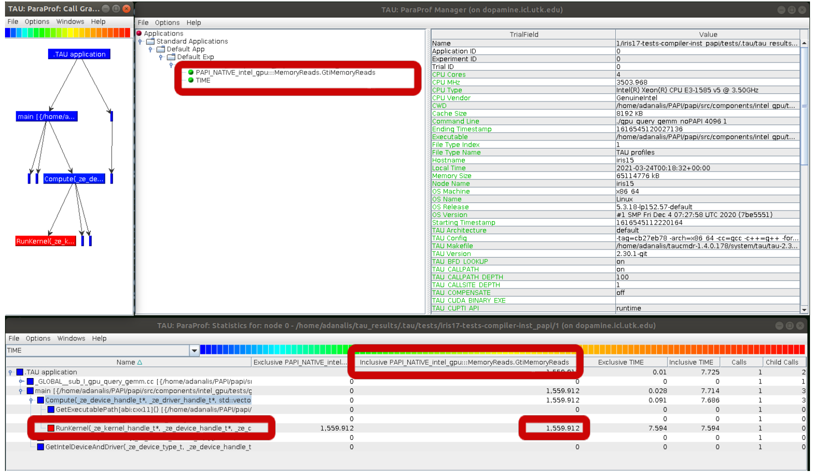
- Robust support for **advanced hardware** resources, which includes major CPUs, GPUs, accelerators, interconnects, I/O systems, power interfaces, and virtual cloud environments.
- **Finer-grain power management** for dynamic optimization of applications under power constraints.
- **Software-Defined Events** originating from software stack (e.g., communication and math libraries, runtime systems).
- **Semantic analysis** of hardware counters so that scientists can better make sense of the ever-growing list of events.

Community Impact

- PAPI channels the monitoring capabilities of **hardware counters**, **power usage**, and **software-defined events** into a robust software package for HPC systems.
- PAPI provides the ability to monitor these metrics through a **single, portable API**.
- Without PAPI, applications must use multiple APIs to access all available metrics, which can hinder productivity.


Master thread
Master thread, Values of Metric "GPU Power nvml:::Tesla_V100-SXM2-16GB:device_0:power" over Time
Watts
All Processes, Accumulated Exclusive Time
advance_p
Kokkos::View...itialization
All Processes, Number of Invocations per Function
Kokkos::ViewFill-1D
Clean clean up arrays
particle compress

PAPI performance and power measurements of a Kokkos application using Vampir / Score-P on NVIDIA GPUs.


TAU: ParaProf: Call Graph
TAU application
main
Compute
RunKernel
TAU: ParaProf Manager (on dopamine.icl.utk.edu)
Applications
Standard Applications
Default App
Default Exp
PAPI_NATIVE_intel_gpu:::MemoryReads.GtiMemoryReads
TIME
TrialField
Value
TAU: ParaProf: Statistics for: node 0
Name
Exclusive PAPI_NATIVE_intel...
Inclusive PAPI_NATIVE_intel_gpu:::MemoryReads.GtiMemoryReads
Exclusive TIME
Inclusive TIME
Calls
Child Calls
1,559.912

PAPI performance measurements of a math kernel using TAU on Intel GPUs.

Expected Future Impact from PAPI

- Monitoring capabilities of **new and advanced technologies** across the hardware-stack, e.g., GPU-to-GPU interconnect support for novel GPUs, rocprofiler v2 support for AMD GPUs.
- PAPI abstractions for expressing the **internal behavior of software** components and improved **interoperability** across the software-stack.

STEP SOFTWARE TOOLS ECOSYSTEM PROJECT

PAPI https://icl.utk.edu/papi/

ICL@UT INNOVATIVE COMPUTING LABORATORY

Slide source: Heike Jagode

https://ascr-step.org

presentation to the Facilities Software Task Force

DynInstAPI: Binary Code Analysis and Instrumentation Toolkit

Current Capabilities on CPU Architectures

Includes x86, ARM and Power

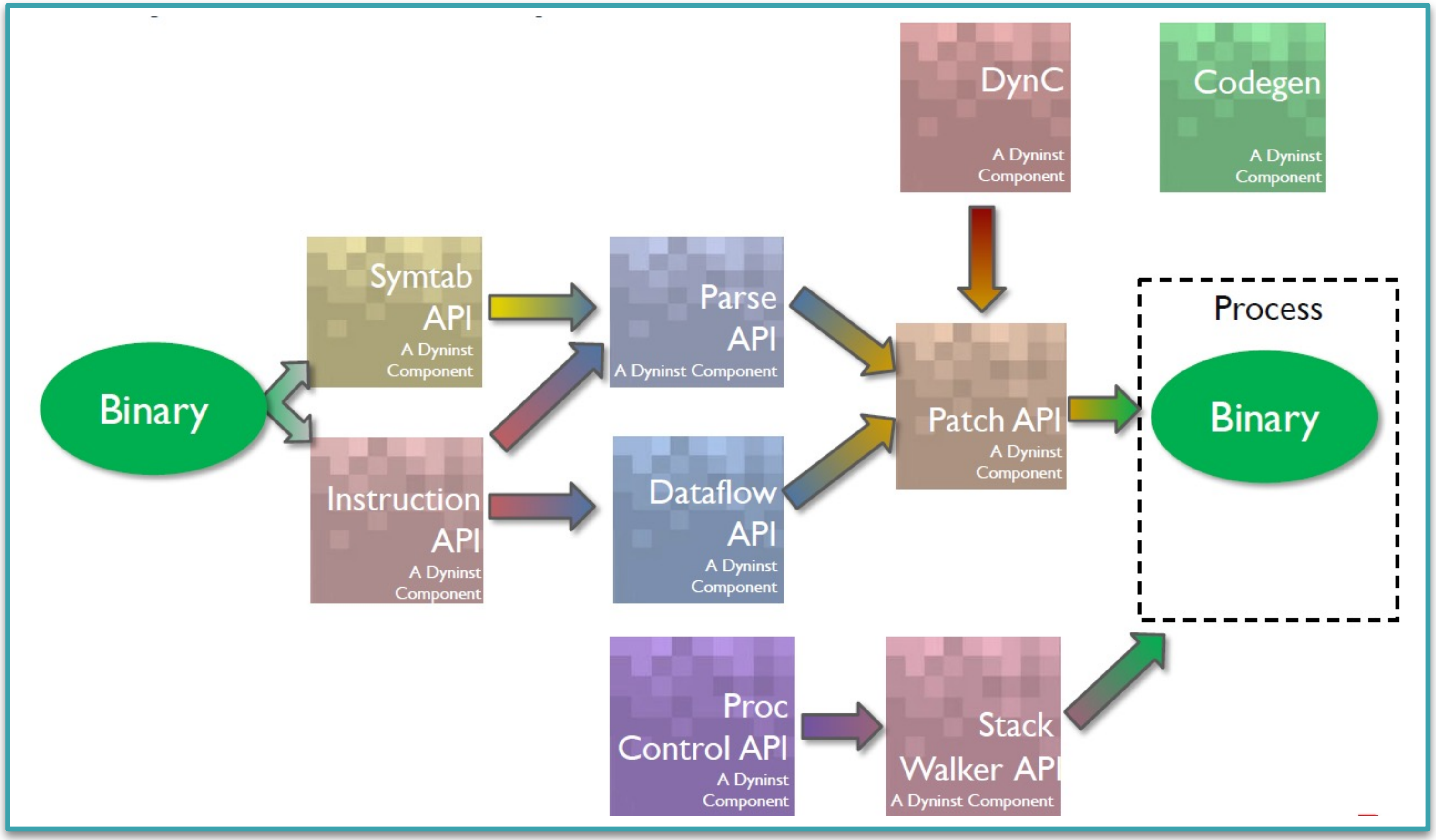
Binary Code Analysis

- Abstract program representation avoids the need for architecture specific knowledge
- Full control flow analysis of binary code to identify functions, loops, and basic blocks
- Dataflow analysis to support slicing, symbolic evaluation and pointer analysis
- Operates on stripped code

Binary Code Instrumentation

- Both static (modify executables and libraries) and dynamic (modify running applications)
- Supports dynamically and statically linked code
- High level: Can instrument function entry, exit and call; loop entry, exit and top;
- Portable: DyninstAPI tools can work across any of the supported architectures.
- Fine grained: Can instrument at any instruction
- Efficient

Is the instrumentation foundation for HPCToolkit, AMD Omnitrace, TAU, Cray ATP, Red Hat SystemTap, and others



Key Plans for 2024

- *Full Dyninst support for the newest x86 CPU families*
- *Extend Dyninst to support binary instrumentation of AMD GPU*

SOFTWARE TOOLS ECOSYSTEM PROJECT

Slide source: Barton Miller

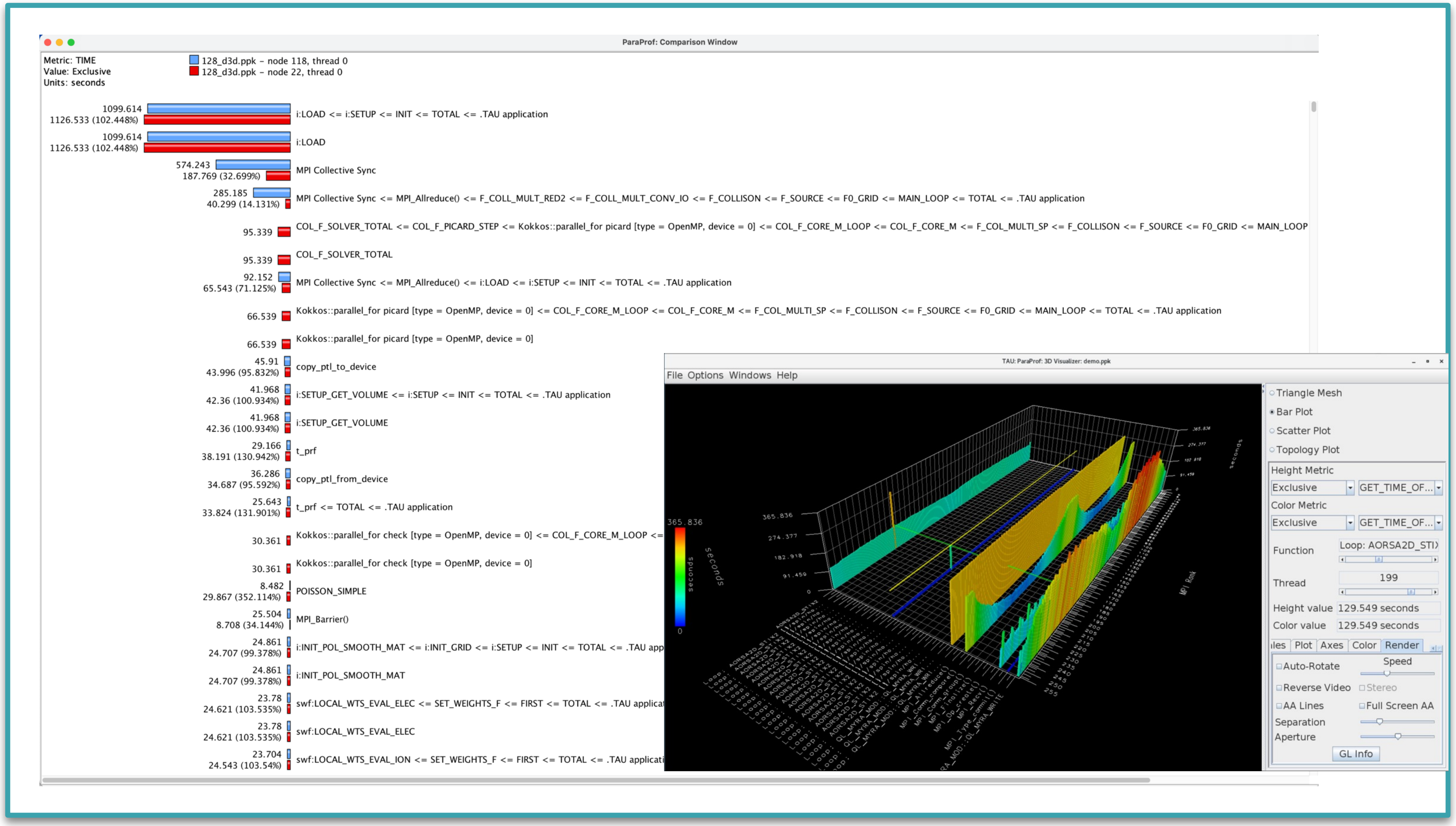
https://ascr-step.org

presentation to the Facilities Software Task Force

TAU Performance System®: Performance Evaluation Tool for GPU-accelerated Architectures

Current Capabilities

- Integrated performance instrumentation, measurement, and analysis toolkit
- TAU is installed as a module at ALCF, LLNL, OLCF, LANL, Sandia, NERSC, and other supercomputing centers
- Supports DyninstAPI for instrumentation, PAPI for hardware performance counters based measurement, HPCToolkit and Darshan for profile display in TAU's 3D profile browser, ParaProf
- TAU supports Python instrumentation (TensorFlow, PyTorch, Horovod) and HPC runtimes (Kokkos, MPI, OpenMP)
- TAU supports tracing with OTF2 and visualization using Vampir (TU Dresden)
- tau_exec launcher requires no modification to the application binary!
- On AMD GPUs, TAU supports OMPT, ROCProfiler, ROCTracer, LLVM plugin for compiler-based instrumentation using hipcc, ROCTx
- On NVIDIA GPUs, TAU supports CUDA Profiling Tools Interface (CUPTI), NVTx
- On Intel GPUs, TAU supports OMPT, Intel Level Zero and OpenCL, Kokkos



ParaProf: Comparison Window

Metric: TIME
Value: Exclusive
Units: seconds

🟦 128_d3d.ppk – node 118, thread 0
🟥 128_d3d.ppk – node 22, thread 0

```
    1099.614
1126.533 (102.448%)   i:LOAD <= i:SETUP <= INIT <= TOTAL <= .TAU application
    1099.614
1126.533 (102.448%)   i:LOAD
     574.243
 187.769 (32.699%)    MPI Collective Sync
     285.185
  40.299 (14.131%)    MPI Collective Sync <= MPI_Allreduce() <= F_COLL_MULT_RED2 <= F_COLL_MULT_CONV_IO <= F_COLLISON <= F_SOURCE <= F0_GRID <= MAIN_LOOP <= TOTAL <= .TAU application
      95.339          COL_F_SOLVER_TOTAL <= COL_F_PICARD_STEP <= Kokkos::parallel_for picard [type = OpenMP, device = 0] <= COL_F_CORE_M_LOOP <= COL_F_CORE_M <= F_COL_MULTI_SP <= F_COLLISON <= F_SOURCE <= F0_GRID <= MAIN_LOOP
      95.339          COL_F_SOLVER_TOTAL
      92.152
  65.543 (71.125%)    MPI Collective Sync <= MPI_Allreduce() <= i:LOAD <= i:SETUP <= INIT <= TOTAL <= .TAU application
      66.539          Kokkos::parallel_for picard [type = OpenMP, device = 0] <= COL_F_CORE_M_LOOP <= COL_F_CORE_M <= F_COL_MULTI_SP <= F_COLLISON <= F_SOURCE <= F0_GRID <= MAIN_LOOP <= TOTAL <= .TAU application
      66.539          Kokkos::parallel_for picard [type = OpenMP, device = 0]
       45.91
  43.996 (95.832%)    copy_ptl_to_device
      41.968
   42.36 (100.934%)   i:SETUP_GET_VOLUME <= i:SETUP <= INIT <= TOTAL <= .TAU application
      41.968
   42.36 (100.934%)   i:SETUP_GET_VOLUME
      29.166
  38.191 (130.942%)   t_prf
      36.286
  34.687 (95.592%)    copy_ptl_from_device
      25.643
  33.824 (131.901%)   t_prf <= TOTAL <= .TAU application
      30.361          Kokkos::parallel_for check [type = OpenMP, device = 0] <= COL_F_CORE_M_LOOP <=
      30.361          Kokkos::parallel_for check [type = OpenMP, device = 0]
       8.482
  29.867 (352.114%)   POISSON_SIMPLE
      25.504
   8.708 (34.144%)    MPI_Barrier()
      24.861
  24.707 (99.378%)    i:INIT_POL_SMOOTH_MAT <= i:INIT_GRID <= i:SETUP <= INIT <= TOTAL <= .TAU app
      24.861
  24.707 (99.378%)    i:INIT_POL_SMOOTH_MAT
       23.78
  24.621 (103.535%)   swf:LOCAL_WTS_EVAL_ELEC <= SET_WEIGHTS_F <= FIRST <= TOTAL <= .TAU applicat
       23.78
  24.621 (103.535%)   swf:LOCAL_WTS_EVAL_ELEC
      23.704
  24.543 (103.54%)    swf:LOCAL_WTS_EVAL_ION <= SET_WEIGHTS_F <= FIRST <= TOTAL <= .TAU applicati
```

TAU: ParaProf: 3D Visualizer: demo.ppk

File Options Windows Help

○ Triangle Mesh
● Bar Plot
○ Scatter Plot
○ Topology Plot

Height Metric: Exclusive | GET_TIME_OF...
Color Metric: Exclusive | GET_TIME_OF...
Function: Loop: AORSA2D_STI)
Thread: 199
Height value: 129.549 seconds
Color value: 129.549 seconds

Key Plans for 2024

- *Expand TAU's support for rewriting binaries and Dynamic Shared Objects using DyninstAPI*
- *Expand CI/CD support for TAU using GPUs from Intel, AMD, and NVIDIA on Frank at UO*

Slide source: Sameer Shende

https://ascr-step.org

presentation to the Facilities Software Task Force

Darshan: Enabling Insights into I/O Behavior of HPC Applications

Darshan is an application I/O characterization tool commonly deployed by HPC facilities

- Lightweight instrumentation methods to avoid perturbing applications
- Transparently enabled, requiring no modifications to users' code
- Generates summary reports of applications' I/O behavior

❶ Darshan runtime library

Transparently intercept application I/O calls, capture I/O characterization data, and generate log file when application terminates

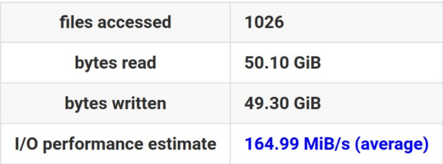
Existing instrumentation modules span the HPC I/O stack: HDF5, PnetCDF, MPI-IO, POSIX, Lustre, etc.

❷ Darshan log analysis tools

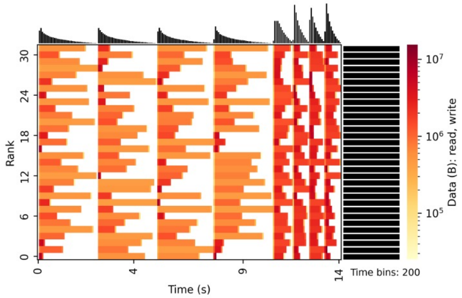
New PyDarshan framework to enable extraction and analysis of Darshan log data using popular Python packages like pandas, matplotlib, etc

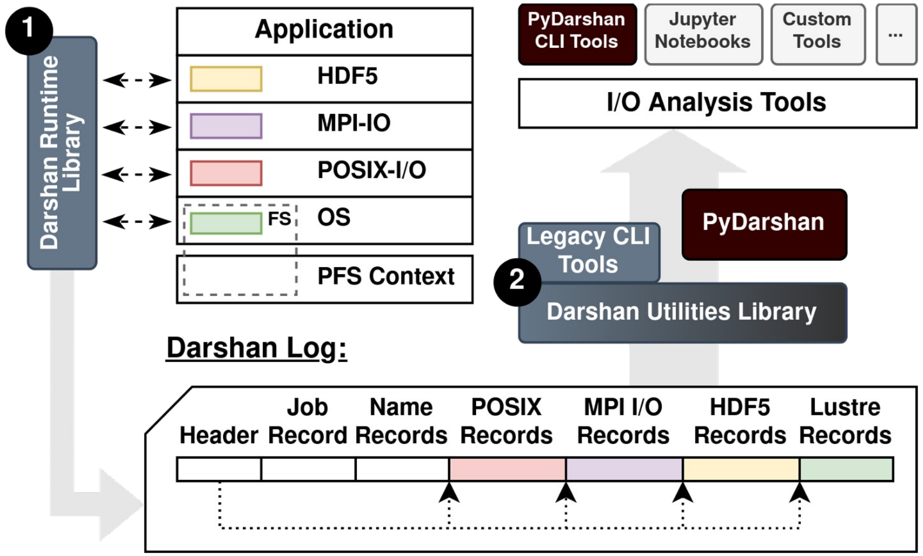
PyDarshan includes a job summary report tool to provide I/O overviews for users' logs

- E.g., I/O performance estimates (left) and I/O activity heatmaps (right)



files accessed	1026
bytes read	50.10 GiB
bytes written	49.30 GiB
I/O performance estimate	164.99 MiB/s (average)





Key Plans for 2024

- *Extend Lustre FS instrumentation to account for new progressive file layout (PFL) feature*
- *Improve design and capabilities of PyDarshan analysis tools*
- *Enhance CI testing capabilities using facility resources*

Slide source: Shane Snyder

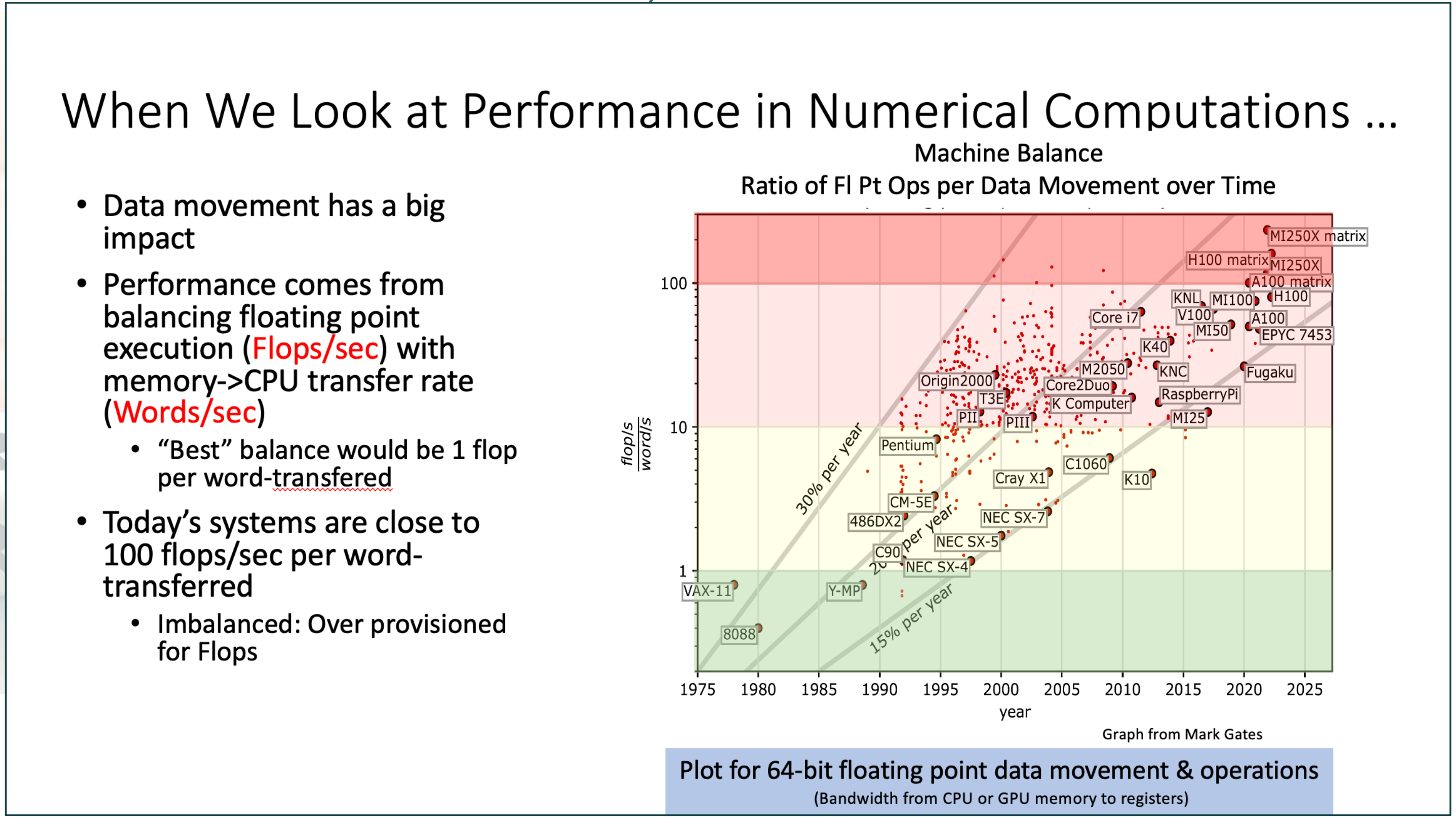
A Look Forward

Can Tools Transition from Reactive to Proactive?

A Look Forward

Can Tools Transition from Reactive to Proactive?

Can Tools Address the Memory Imbalance?



When We Look at Performance in Numerical Computations …

- Data movement has a big impact
- Performance comes from balancing floating point execution (Flops/sec) with memory->CPU transfer rate (Words/sec)
  - “Best” balance would be 1 flop per word-transfered
- Today’s systems are close to 100 flops/sec per word-transferred
  - Imbalanced: Over provisioned for Flops

Machine Balance

Ratio of Fl Pt Ops per Data Movement over Time

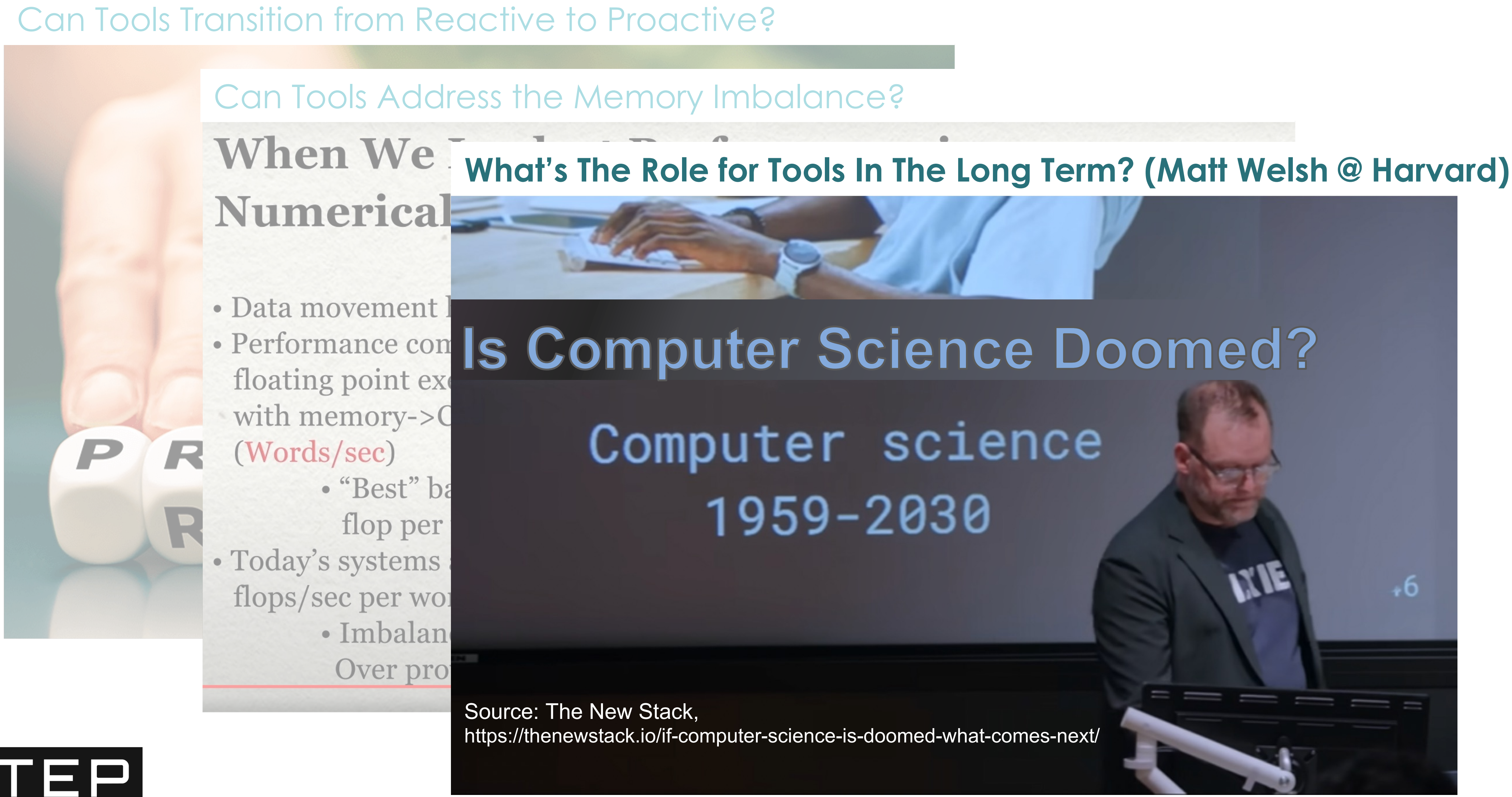
Graph from Mark Gates

Plot for 64-bit floating point data movement & operations

(Bandwidth from CPU or GPU memory to registers)

*Source: Jack Dongarra, https://new.nsf.gov/events/overview-high-performance-computing-future

A Look Forward



Can Tools Transition from Reactive to Proactive?

Can Tools Address the Memory Imbalance?

When We

Numerical

- Data movement l
- Performance con floating point exe with memory->C (Words/sec)
  - “Best” ba flop per
- Today’s systems a flops/sec per wor
  - Imbalanc Over pro

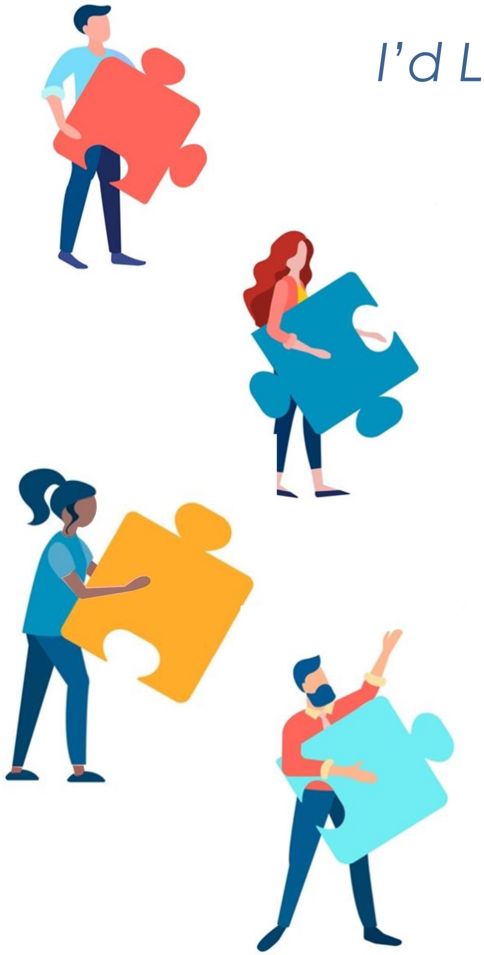
What’s The Role for Tools In The Long Term? (Matt Welsh @ Harvard)

Is Computer Science Doomed?

Computer science 1959-2030

Source: The New Stack, https://thenewstack.io/if-computer-science-is-doomed-what-comes-next/

QUESTIONS



I'd Like to acknowledge the full Software Tools Ecosystem Project (STEP) Team

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

STEP
SOFTWARE TOOLS
ECOSYSTEM PROJECT

Extra Slides

STEP Coverage Matrix – *Software Packages Fostered Through STEP*

	Current Machines	Future Machines (STEP)
Meta Tool used by other tools	4, 6, 10	4, 6, 10
CPU Examination & Performance	2,7,12	7
GPU Examination & Performance	2,7,12	7
Memory Examination & Perf	9, 11	
Storage I/O Examination & Perf	4, 5, 8	4
Debugger & Correctness	13	
Parallel load-balance / scaling	7, 12	7
Machine Level Exam & Perf	1, 4, 7, 12	7, 12
Network/Workflow level Exam & Perf	3	

Legend (green=funded):

1=ArgoNRM,
2=Bliss,
3=Chimbuko,
4=Darshan,
5=Drishti,
6=Dyninst,
7=HPCToolKit,
8=LDMS Conn,
9=Memgaze,
10=PAPI,
11=SICM,
12=TAU,
13=Thapi

STEP
SOFTWARE TOOLS
ECOSYSTEM PROJECT

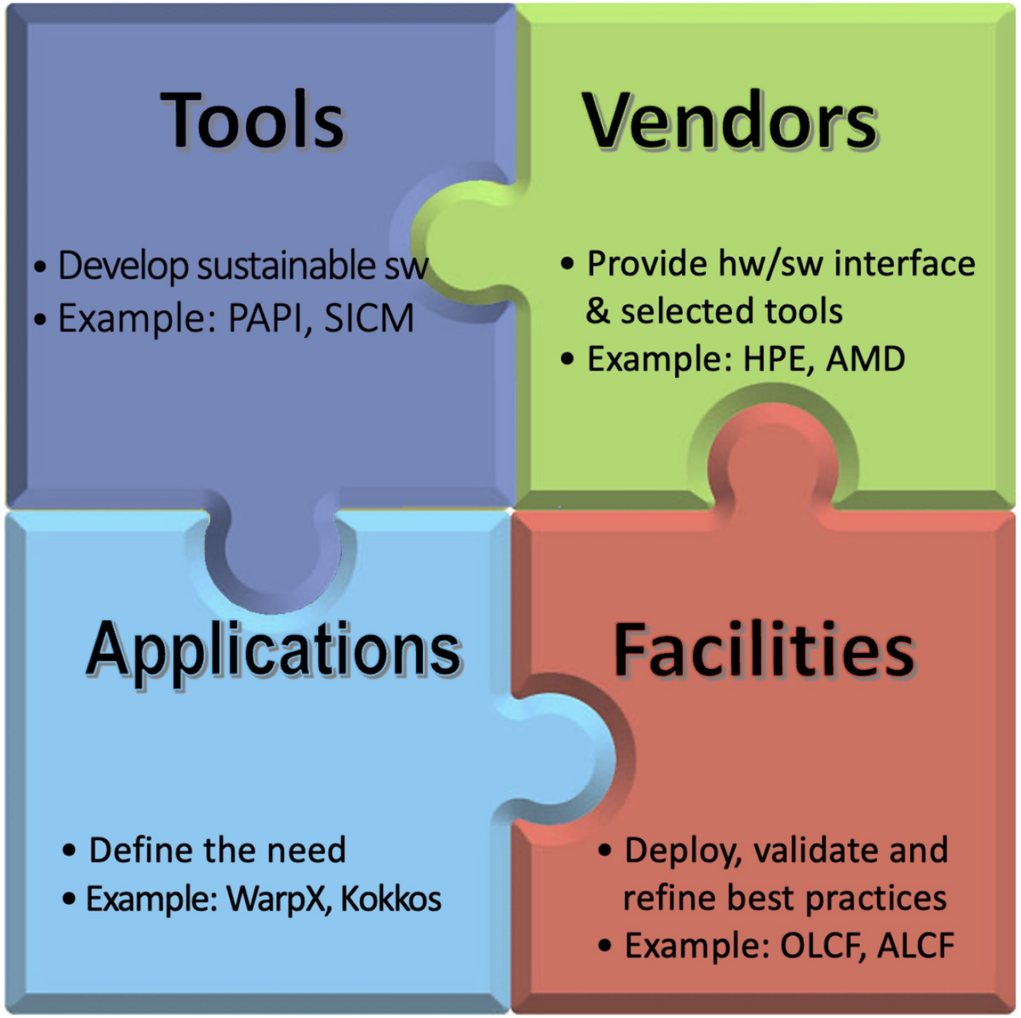
CY2024 STEP Portfolio

Tool	Tool Description	Tool PI	Lead Institution	STEP Funding	Former ECP Funded	DOE Deployed	Maturity[1]
HPCToolkit	Performance analysis toolkit for HPC	John Mellor-Crummey	Rice University	Yes	Yes	Yes	M
PAPI	Perf monitoring library & toolset	Heike Jagode	Univ. of Tennessee	Yes	Yes	Yes	M
Dynist	Binary code and instrumentation	Barton Miller	Univ. of Wisconsin	Yes	Yes	Yes	M
TAU	Portable profiling and tracing toolkit	Sameer Shende	Univ. of Oregon	Yes	Yes	Yes	M
Darshan	I/O instrumentation tool	Shane Snyder	Argonne Natl Lab	Yes	Yes	Yes	M
Additional critical software tools that STEP would support with a larger budget ceiling.							
SICM	Memory tool for multi-tier memories	Terry Jones	Oak Ridge Natl Lab	tbd	Yes	Yes	M
Drishti	I/O Analysis and visualization tool	Suren Byna	Ohio State Univ.	tbd	Yes	Yes	M
Chimbuko	Tool for extreme scale machines	Christopher Kelly	Brookhaven Natl Lab	tbd	Yes	Yes	I
LDMSinspect	Darshan/LDMS integration	Devesh Tiwari	Northeastern Univ	tbd		Yes	I
Argo NRM	Node resource management tool	Swann Perarnau	Argonne Natl Lab	tbd	Yes	Yes	I
Thapi	Heterogeneous API instrumentation	Brice Videau	Argonne Natl Lab	tbd	Yes	Yes	I
Memgaze	Memory analysis toolset	Nathan Tallent	Pacific NW Natl Lab	tbd			I
Qraft	Quantum toolkit	Devesh Tiwari	Northeastern Univ	tbd			I

[1] STEP intends to support both mature software products and newer emerging tools. Maturity is denoted by either M=mature or I=Incubator.

STEP Aspirations

- Support Tools Community
- Utilize Co-design
- A proactive stance on the rapidly evolving hardware landscape: new (completely different) tool APIs from AMD and Intel that include support for GPU PC sampling.



Tools

- Develop sustainable sw
- Example: PAPI, SICM

Vendors

- Provide hw/sw interface & selected tools
- Example: HPE, AMD

Applications

- Define the need
- Example: WarpX, Kokkos

Facilities

- Deploy, validate and refine best practices
- Example: OLCF, ALCF

Tools Stakeholders And Their Concerns

Applications

Ease of use,
Actionable feedback

Vendors

Maximize impact,
Product success

Tool Devs

Novel insights,
Broad applicability

Facilities

Interoperability,
Ease of deployment

Slide source: Phil Carns

Several significant attributes…

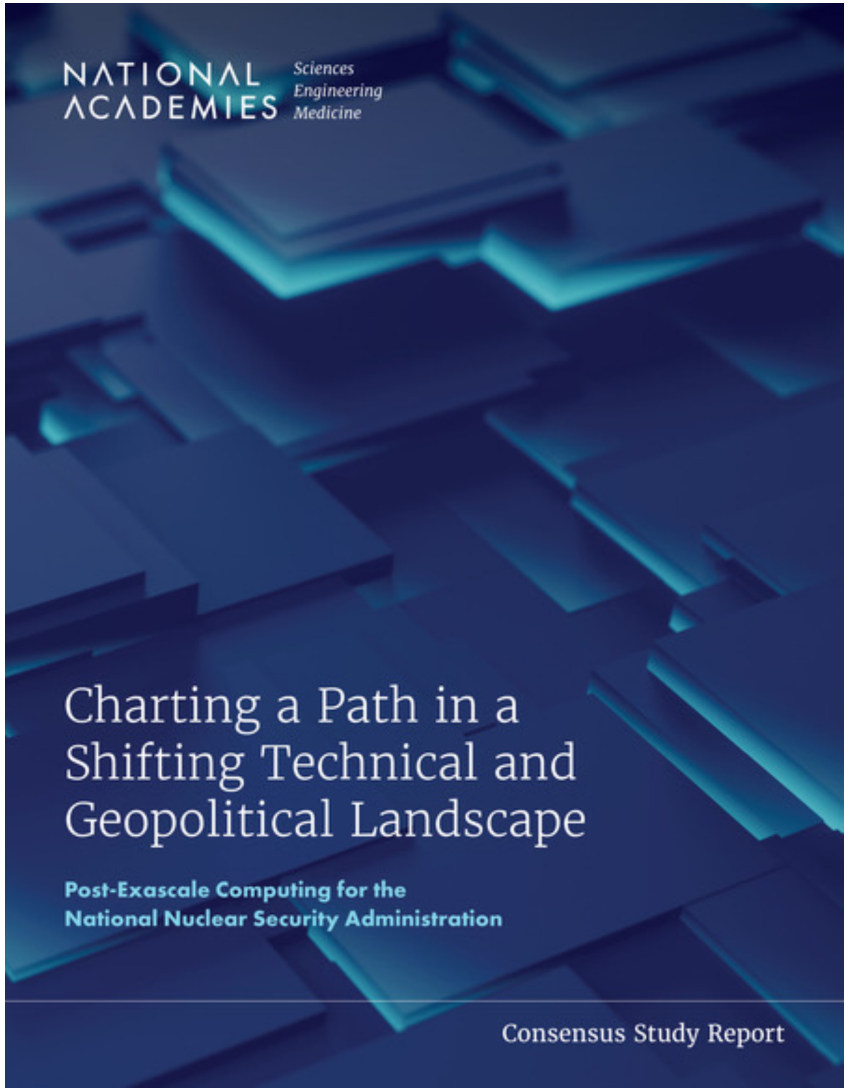
1. Not _**just**_ replacing ECP
2. Focus on _**stewardship**_
3. Requires _**strong partnership**_ with facilities, vendors, and application teams.

National Academies on Post-Exascale for NNSA…

"On the technology front, the single-thread performance of microprocessors continues to be relatively flat, and improvements in transistor density are slowing. Processing elements increasingly rely on more abundant, finer-grained parallelism and increasingly specialized hardware features that can improve performance by tailoring to a given computational domain. These trends are creating a significant disruption in available hardware components, with commercial interests focused largely on AI, embedded systems, and cloud services. **There is considerable uncertainty as to whether the processors emerging in response to these trends, with their lower precision and limited high-speed memory, can be productively applied to NNSA applications**." (page 4 [emphasis mine])

"**FINDING 3.6**: Co-design of hardware and systems for high-performance scientific computing applications has been a modest success to date and will be more important in the future and need to be deeper. Technological and market trends are likely to shift the balance of co-design to the laboratories, requiring more innovation and engineering in the areas of hardware design, system integration, and system software." (page 7)

"**RECOMMENDATION 2.2**: NNSA should strengthen efforts in computer science research and development to build a substantial, sustained, and broad-based intramural research program that is positioned to address the technological challenges associated with post-exascale systems and co-design of those systems to ensure that the laboratories are positioned for leadership in computing breakthroughs relevant to NNSA mission problems." (page 8)



NATIONAL ACADEMIES Sciences Engineering Medicine

Charting a Path in a Shifting Technical and Geopolitical Landscape

Post-Exascale Computing for the National Nuclear Security Administration

Consensus Study Report

Source: National Academies of Sciences, Engineering, and Medicine. 2023. Charting a Path in a Shifting Technical and Geopolitical Landscape: Post-Exascale Computing for the National Nuclear Security Administration. Washington, DC: The National Academies Press. https://doi.org/10.17226/26916.

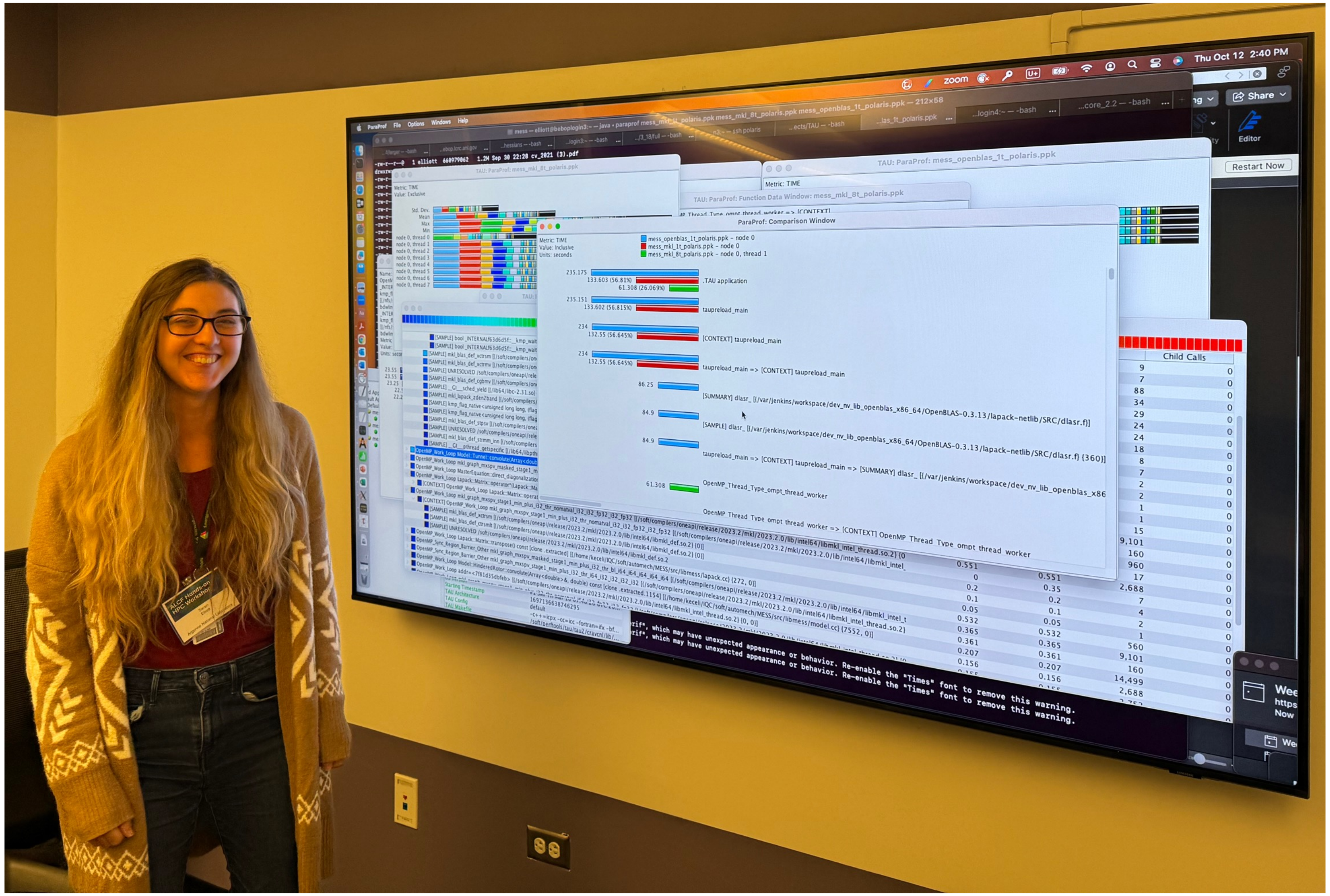
Tools Impact: *A Recent TAU Example*

- Sarah Elliott’s code is called MESS
  https://tcg.cse.anl.gov/papr/codes/mess.html

- Together with TAU and working the ALCF catalyst team led by Murat Keceli and Tim Williams, Sarah reduced the total runtime of her application by a factor of 3.8x in one day - on Oct 12, 2023.

- Sarah is a domain scientist in the Chemical Sciences and Engineering (CSE) group (not affiliated with ALCF).

Source: Sameer Shende


ParaProf: Comparison Window
TAU: ParaProf: mess_openblas_1t_polaris.ppk
Child Calls
Thu Oct 12 2:40 PM

Sarah Elliott from Argonne – https://tcg.cse.anl.gov/papr/codes/automech/contributor/elliott.html.