STEP
SUSTAINABLE TOOLS ECOSYSTEM PROJECT

Breakouts Session 1
Challenge 2: Exploding Use Cases

August 16, 2023

James Custer, HPE

Outline

- Tool Outcomes
- New Capabilities
- User Interaction
- Gaps

STEP
SUSTAINABLE TOOLS
ECOSYSTEM PROJECT

Tool Outcomes - Questions

- An HPC tool is only as useful as the outcome it produces. What are the most important outcomes to consider from tools to support current and evolving compute infrastructure? Consider the perspectives of the greater community (e.g., facilities, system administrators, platform procurers and architects, users, and developers)
- What new design requirements does this place on tool developers?
- What interactions and interfaces among tools and system software is needed to enable these outcomes?
- What does sustainability look like for this topic?

Tool Outcomes

- Expand the “user base” and “interaction base” of tools
  - Create machine ingestible output to enable ML analysis or interoperability of tools
  - Target tools to specific humans, e.g. expand the output format to create tools for different use cases (e.g. app users vs facilities)
- Create a flexible, modular tool ecosystem (avoiding monolith tools)
- Increased development and knowledge-base is being required, so we should:
  - Create STEP Tools Portability Layer(s) – creates a common base and interfaces to increase interoperability of tools
  - Provide expert advisors for tool developers – provides common advice so tool development evolves in similar ways and reduces the global knowledge needed for tool development

STEP SUSTAINABLE TOOLS ECOSYSTEM PROJECT

New Capabilities - Questions

- What new capabilities for processing closer to architectural resources do we anticipate?
- How do they provide new opportunities for tools to not only provide insights into application efficiencies, but also to dynamically remap and reconfigure applications and resources?
- How do we address latency and interoperability challenges in order to realize this potential?
- What does sustainability look like for this case?

New Capabilities

- Need to incentivize vendors to expose performance related information and provide insight into what data they expose the from exploding ecosystem of hardware
- Monitoring will need to use functional units to collect performance measurements
- Burst to cloud provides performance opportunities but need to figure out how tools incorporate remote performance characteristics
- Security needs to be integral in the development of new tools
  - Encryption and role-based access needs to be everywhere, even within a compute node
- Enable new insights using AI

STEP SUSTAINABLE TOOLS ECOSYSTEM PROJECT

User Interaction - Questions

- How can we improve bidirectional interaction between application teams and tool developers?
- What is the right vehicle to make this sustainable across more than just pair-wise discussions between a single application team and a single tool team?

User Interaction

- Need to drive interactions using hackathons and training workshops
- Need to be aware of the distinction between users and developers of applications. Developers are often unaware of the performance bottlenecks until the users report it. Developers may not have run the code at scale
- Key to tool usage: no change to source code, build, or even the application binary. Just modifications (*with examples*) for launching the tool on un-modified binary
- Make tool usage more intuitive, including automating the analysis
- Bridge the gap between performance data collected and actionable intelligence and present it in a clearly understandable way

Gaps - Questions

- What are the biggest perceived gaps in HPC tool support for the variety of runtime systems and programming models that applications wish to use?
- What are new programming paradigms and workflows (e.g., Python-based AI/ML and Julia-enabled parallelism) that are becoming more relevant to HPC?
- How must tools adapt to support these new use cases?
- Do they require fundamentally distinct tools, or is it better to expand the breadth of existing tools?
- Are users comfortable with "the right tool for the job, as needed" or do we need a more coherent integrated approach with tools that fit together?

Gaps

- Communication and collaboration problems between vendors/tool developers/application developers
- Tools lack user-friendliness, especially for novices.
- Varied support for runtime systems and programming models
- Performance analysis of workflows and cloud computing
- There is a lack of support for modern languages like Rust, Lua, Go, and Julia. This is coupled with gaps in tooling for control flow/data flow programming models, task-based parallelism, and distributed task-based parallelism.
- The profiling infrastructure in a language like Python is not readily amenable to be used effectively by our tools.
- Data flow is typically not done in C/C++/Python, they're done in DSLs. How do we measure performance?
- There is a high barrier to entry and tool complexity
- Vendors should allow open-source tools more opportunity to implement features early on and provide performance tools libraries that help interpret the data
- Users' understanding of which tools are good for specific tasks needs to be improved