SAND2017-13726PE

December 2017 ECP ST Project Review

ECP Project WBS 2.3.2.04 – SNL ATDM Tools

PM: Rob Hoekstra (SNL)

PIs: Ross Bartlett (SNL), Si Hammond (SNL)

December 20, 2017

EXASCALE COMPUTING PROJECT

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Guidance: 30 minute presentation + 10 min questions

Topics to cover (with suggested slide count)

- **Project Overview (1-2)**
  – Review the goal(s) of the project: team, funding level, impact goals and metrics, etc.

- **Project Plan (1-2)**
  – Describe your plan and roadmap: Focus on FY18 but also provide some details on FY19.

- **Self-assess your project in 5 categories (<15) Describe:**
  1. How the project is important or essential to achieving sustainable, production-level computing capabilities for ECP.
     - Focus on the new capabilities that ECP is funding, not the general capabilities represented your project.
     - Address what ECP would lack if your project could not deliver.
     - Discuss the project's clients and users, and the importance of your efforts to helping them achieve ECP goals.
  2. Project deliverables for 2019 and (assuming continuous funding) deliverables in the ECP timeframe (2021 - 2023).
  3. How the project will deliver its capabilities to users.
     - Discuss any project history of production software delivery.
     - Describe planned delivery paths, in particular: open source repositories with production source installation capabilities, third parties (vendors, OpenHPC), direct to facilities.
     - Describe means of documentation, testing, user support, licensing, etc.
  4. Progress toward the project's milestones.
     - List project impact goals and impact metrics.
     - Review and highlight completed milestones.
     - Discuss progress toward integration with applications, other ST projects, vendors, and facilities and platform readiness.
     - Discuss progress on software performance, scalability, and portability.
  5. Relationships with other software (from any source) and other ECP efforts.
     - List project dependencies (you depend on others and others depend on you) that are not ECP apps.
     - Discuss projects with similar capabilities, and distinguishing features you want to highlight.
     - Discuss collaborating projects and any relationships with ECP HI (co-design, PathForward, DSE) efforts.

- **Project costs (1)**
  – What are your project costs so far (total and by institution expressed as a percent of an annual allocation)? How do they compare to your spend plan? If different than the plan, why? What is your remaining carryover from FY17 (in $ and as a percentage of FY17 funding)?

- **Next steps: focus on FY18 plans and milestones (1-2)**

- **Project risks and issues (1-2)**
  – Discuss technical, programmatic, or personnel risks and issues and how you plan to address and manage them.
  – What can ECP (in particular ST) do better or differently to help you succeed?

Project Overview

- **Primarily focused on ”DevOps” and performance analysis for SNL’s ATDM codes, components and libraries**
- Leverage tools from ECP community when possible
- Deliver ”lessons learned” to community
- Participate in ECP productivity activities
- Resourcing and ATDM schedules will limit our ability to deploy tools to the larger ECP community…
- Budget $1.5M

DevOps

DevOps in the ATDM context is all of the software infrastructure development, testing support, integration, and deployment work in support of the ATDM software application and component development teams, which requires a broad knowledge base and strong skill set in software engineering, the ability to understand the requirements and needs of a diverse set of software development teams, and the ability to work in a complex environment with high demands and expectations**.**

Main project team responsibilities:

- Coordinate and prioritize tasks for the various teams that provide “DevOps” support for ATDM codes, applications, and customers
- Develop and help deploy shared build, test, and install infrastructure across the ATDM codes and projects
- Define and support development, testing, integration, and other related workflows for ATDM projects

Performance Analysis

- **Application Performance Analysis**
  - Profiling and performance analysis support to ATDM application/library teams
  - Tuning/optimization of build configurations and environments to support more efficient/faster code generation and execution
  - Testing on next-generation (ATS and CTS) platforms including scalability, performance *etc*
- **Codesign Collaboration**
  - Vendor engagement on compilers, profiling tools, environments, runtimes *etc* for performance/successful execution
- **Testbed Environment Development/Optimization**
  - Testbed software environment optimization utilizes direct feedback from ATDM activities
- **Compiler Research**
  - Vendor focused but growing engagement with ECP and NNSA LLVM activities

FY18

- Coordinate and prioritize DevOps tasks for SNL ATDM
- Stabilize Trilinos development and integration for ATDM usage
- Common build, test, and integration framework to bring together different ATDM projects and codes
- Continue extensions of Cmake/Ctest/CDash for ATDM priorities with Kitware contract
- Performance analysis of ATDM applications on ATS-1 and ATS-2 platforms
- Deployment of continuous performance monitoring connectivity for ASC platforms and testbeds via LDMS integration
- Analyze and improve our use of ATS-1 & 2 development environments (especially compilers) in support of ASC Tri-lab Co-design L2 Milestone

FY19

- Maintain and extend existing DevOps infrastructure
- Continue extensions of Cmake/Ctest/CDash for ATDM priorities with Kitware contract
- Address higher priority DevOps requests using Agile process
- Performance characterization on any early availability ATS-3 test systems
- Initial performance characterization of codes on Sandia Vanguard ARM system
- Continued performance analysis and optimization on ATS-1 and ATS-2 to support L1 milestone
- Leverage continuous performance monitoring (LDMS) systems for ATDM application reporting
- Support FY19 ASC Tri-lab Co-design L2 Milestone

Assessment

1. How the project is important or essential to achieving sustainable, production-level computing capabilities for ECP.
   - Focus on the new capabilities that ECP is funding, not the general capabilities represented your project.
   - Address what ECP would lack if your project could not deliver.
   - Discuss the project’s clients and users, and the importance of your efforts to helping them achieve ECP goals

This project is focused on deployment, testing and performance analysis on our ATDM codes on pre-Exascale and Exascale platforms. A number of ECP tools/libraries are intended to be leveraged (Trilinos, SPACK, HPCToolkit, TAU, ExaPAPI, LLVM/Flang, …) and the resultant maturation through use with ASC codes benefits the broader ECP community. We will be limited by budgets and schedule with regard to developing and deploying tools more broadly. We are also working with the ECP productivity team to impact the broader community.

Assessment

2. Project deliverables for 2019 and (assuming continuous funding) deliverables in the ECP timeframe (2021 - 2023).

This project will continue to by focused on deployment, testing and performance analysis on our ATDM codes for Exascale platforms. In these later years, we hope that the ECP software tools for this scope of activities will have begun to mature so we can more effectively leverage their capabilities.

- Deployment and testing of ATDM tools and codes on pre-Exascale platforms and Exascale testbeds.
- Performance analysis in improvement using ECP and vendor tools
- Shared “lessons learned” for tools and analysis
- Support ATDM L1 & L2 milestones

Assessment

3. How the project will deliver its capabilities to users.
   - Discuss any project history of production software delivery.
   - Describe planned delivery paths, in particular: open source repositories with production source installation capabilities, third parties (vendors, OpenHPC), direct to facilities.
   - Describe means of documentation, testing, user support, licensing, etc.

The Tribial Build Integration and Test system (TriBITS) is the production build and test system for Trilinos (since 2008), the CASL VERA project (since 2011), and other related projects. It is available from GitHub, has comprehensive documentation, tutorials and testing and is available under a BSD license. In addition, we intend to improve performance analysis tools (vendor and community) including the LDMS-based performance monitoring so they are valuable to the broader community.

Assessment

4. Progress toward the project's milestones.
   - List project impact goals and impact metrics.
   - Review and highlight completed milestones.
   - Discuss progress toward integration with applications, other ST projects, vendors, and facilities and platform readiness.
   - Discuss progress on software performance, scalability, and portability.

- Prototype App+Trilinos builds and testing using full TriBITS and submitting to Cdash
- Set up ATDM project and issue tracking utilizing JIRA. Set up comprehensive JIRA and JIRA Portfolio site and system for all SNL ATDM projects on Sandia Secure Network. Created system for defining and managing dependencies between SNL ATDM projects, shared work, and collaboration between projects with full visibility on changes and delays to work items.
- Contract with Kitware to improve Cmake/Ctest/Cdash: Fortran support in Ninja, performance, enhanced Cdash.
- Demonstrate optimized build parameters and configurations for application/library execution on next generation architectures including Intel Haswell/Broadwell/Knights Landing processors, NVIDIA GPUs and IBM POWER8/POWER8+ systems
- Provide initial performance comparison across computing architectures with areas of concern/ weaknesses identified and fed back to application/library developers
  - Provide initial assessment of vendor supplied performance tools/best practices to application developers
- [ECP-ST] Provide initial demonstration of Kokkos/C++ runtime hooks to vendor provided profiling and/or debugging tools for assessment of compute kernels.

Assessment

5. Relationships with other software (from any source) and other ECP efforts.
   - List project dependencies (you depend on others and others depend on you) that are not ECP apps.
   - Discuss projects with similar capabilities, and distinguishing features you want to highlight.
   - Discuss collaborating projects and any relationships with ECP HI (co-design, PathForward, DSE) efforts.

– Strong dependencies within the ATDM portfolio. In addition, dependencies on Spack, ExaPAPI, TAU, HPCToolkit, LLVM/Flang, Trilinos, …

– Similar activities in LANL/LLNL ATDM portfolios

– The performance analysis activities are very strongly aligned with ECP HI with multiple individuals who are on the Proxy Apps, Co-design, Pathforward and Hardware Evaluation projects.

Next Steps

- Build out and mature ATDM software infrastructure
- Analyze code performance and make improvements for multiple architectural paths
- Support deployment of pre-Exascale and Exascale platforms
- Share “lessons learned” with ECP community
- Leverage and improve ECP software and performance tools as possible
- Participate in the ECP Productivity activities

Project Costs

Budget: $1.5M (DevOps: $500k; Performance: $1M)

No project level carryover in ASC, managed at the program level.

High level breakdown;

$1400k – labor & travel

$100k – contracts (Kitware, TechX)

Risks and Issues

- Insufficient resources to deliver what the SNL ATDM Applications really need (a full solution). Instead, we can only address a few issues and make the best impact that we can. Insufficient stability and availability of ECP, community and vendor tools for development and performance analysis
- Supporting overall project management for ATDM and ECP: The JIRA project management and issue tracking activities continue to consume a large amount of project staff time.
- Supporting tools and infrastructure for multiple architectural paths. Diversity is critical to ultimate success but it is difficult to do "equal justice" to all paths.
- Inadequate performance analysis capabilities from community and vendors.
- Lack of resources to engage as broadly as we would like with the ECP community.