***The government seeks individual input; attendees/participants may provide individual advice only*.**

Middleware and Grid Interagency Coordination (MAGIC) Meeting Minutes[1]
February 2, 2022, 12-2 pm ET

Virtual

Participants

Alan Sill (TTU)	Keith Beattie (LBNL)
Alison Derbenwick Miller (Oracle)	Lisa Arafune (CASC)
Bill Miller (DOE)	Ludovico Bianchi (LBNL)
Bill Spotz (DOE)	Mallory Hinks (NCO)
David Bernholdt (ORNL)	Mike Heroux (Sandia)
David Martin (Northwestern)	Miron Livny (Wisconsin)
Donald Petravick (Illinois)	Rich Carlson (DOE/SC)
Duncan Poole (NVIDIA)	Ryan Adamson (ORNL)
Eric Lancon (BNL)	Salman Habib (ANL)
Birali Runesha (U of Chicago)	Sharon Broude Geva (U of Michigan)
Hall Finkel (DOE/SC)	Stefan Robila (NSF)
Jack Wells (NVIDIA)	Tevfik Kosar (NSF)

Introductions: This meeting was chaired by Rich Carlson (DOE/SC) and Tevfik Kosar (NSF)

Toward a Leadership Software Center

Mike Heroux, Sandia National Laboratories

- Mike gave a disclaimer that the following description is notional. They have shared it with stakeholders. The goal is to be ready to execute a plan for post-ECP sustainability.
- Background
    - DOE Exascale computing Project (ECP) initiated the Extreme-scale Scientific Software Stack (E4S)
    - E4S development will continue under ECP for 2 more years
    - A brief reminder of E4S and SDKs
- E4S
    - Spack-based distribution of software tested for interoperability and portability to multiple architectures
    - Available from source, containers, cloud, binary caches
    - Leverages and enhances SDK interoperability thrust
    - Not a commercial product – an open resource for all

[1] Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program.

    - Growing functionality: Nov 2021: E4S 21.11 – 91 full release products
- E4S delivers products needed now and into the future
    - Gave examples of products and engagement
- Delivering an Open, Hierarchical Software Ecosystem
- Gave examples of E4S and SDKs as platforms are providing tremendous value
- Moving Forward
    - Exploring ideas now to better orient E4S efforts toward the post-ECP era
    - Engaging key US agencies and international institutions is essential to the longevity
    - Proposed a plan for:
        - DOE ASCR Leadership Software Center
        - A leadership and stewardship role in sustaining and growing E4S through LSC
- Mission: Deliver a robust, reliable, high-quality and sustainable software stack that enables the rapid development of DOE scientific applications for the pursuit of scientific discovery in leading-edge computing environments
- Vision: Build the best leadership computing scientific software stack in the world that enables innovative computational and data science solutions to global challenges and breakthrough problems on leadership computing platforms
- Values: Software that addressed leading edge and emerging application needs on leading edge and emerging computing platforms, and unambiguously adds value to the software ecosystem because of its usefulness, quality, sustainability, and complementarity
- Mike described the goals, requirements, and LSC motivation
- ECP experience: Software sustainability requires a new kind of organization
- Key Elements of LSC Structure
    - LSC Project Leadership
    - Technical area SDKs
    - Product development teams
- LSC Execution Approach
    - Plan, Execute, Track, Assess Lifecycle
    - Change Management Process
- Capability Integration Strategy
    - DOE software products have four primary integration targets
        - Vendors
        - Community SW
        - Facilities
        - Direct to apps
- Frank – designed for Libs and Tools Developers
    - Prep system for ECP libs and tools
    - Access to latest non-NDA HW/SW
    - Shared file system – 1 copy of SW
    - Port to many device types at once
    - Porting support from E4S team
    - CI testing workhorse (500k builds)

    - Next: bare metal, BIOS-changing support for low-level software work
- Comment from Alan Sill: One of my concerns with ECP up to now, despite the great progress and tremendous amount of work being done is that if you're not in the ECP project, it's sometimes difficult to access the builds and the tools and therefore to contribute. Question from Alan: Going back to the slide on bias level support, do you have a solution for that? Because in our CAC center, in partnership with industry, we have done a lot of work on direct control with the redfish protocol of the baseboard management controllers, and we'd love a chance to work with you on this.
    - Mike: This is a conversation we've just really started in earnest with some of our Argonne team members and they need this like low level support.
    - Alan: We can do that. We've been working in advance of the standards, actually testing the standards before they're implemented by the vendors.
    - Mike: I don't know if you know Kate Keahey. (Alan: yes)
- Software Support Challenges
    - Support for DOE libraries and tools has been challenging in the past:
        - Support funding is a tax on incoming R&D funds
        - External user support ad hoc and cautious
    - Impact: Uptake is risky
        - Support is contingent on:
            - Continued R&D funding (DOE users)
            - Minimal incompatibility with DOE user needs (external users0
    - App teams (even DOE) reluctant to adopt – concerned about long-term support
- Working toward software sustainability: ASCR Task force, community events
- ASCR RFI Response
    - Broad call, Dec 13, 2021, deadline
    - 10 prompt sets
    - 37 community responses – from brief and long
    - ECP response:
        - 48 pages, addressing all prompts
        - Foundation for 2012 IPR response
- 2021 IPR Recommendation Response
    - Revision of RFI response plus:
        - A proposed transition timeline and rationale document for the relevant timeline scenarios
        - Public release of the dependency database, or some releasable subset
        - Release date: March 1, 2022

Q&A

- Alison Derbenwick Miller: We'd love to chat with you more about the industry partnerships you're talking about and also to get involved in LSC, so we'll go sign up and do that.
- Keith Beattie: I want to add one thing to this, which is the impact that an organization like this would have on the careers that people can experience while working in research in terms of retention, recruitment, getting underrepresented groups involved.

<u>HPC: The Case for Dynamic and Sustainable LSSW</u>

Salman Habib, Argonne National Laboratory

- Changing Roles of Supercomputers
    - Modern Supercomputing
        - HPC is a broad term encompassing the extremes of compute-intensive, data-intensive, and high-throughput regimes, with a wide variety of use cases
        - Traditionally, in the DOE context, HPC meant supercomputing
        - Supercomputing occupies its own special corner in the HPC world
        - Supercomputers are evolving too
        - Evolving towards key roles as compute/data hubs in a larger-scale hierarchical environment
- Two Complexity Axes
    - Codes
    - Systems
- What's needed
    - Personal Example
        - System: Thinking Machines CM-5
        - Programming Model: Data-Parallel Language – CMF and C*
        - Software: CMSSL (Lennart Johnsson)
        - Opportunities created:
            - Quantum Field Theory
            - Beam dynamics for accelerators
            - Quantum dynamical systems
            - Computational cosmology
            - Nonlinear dynamics
            - Stochastic PDEs
- Imperfect solutions
    - Science Teams: Hard Choices
        - Hardware issues
        - Software issues
        - Two responses
            - Large-team response: control your own destiny – have your own developers
            - Small-team response: minimize all external dependencies
- Lessons from the ECP experience
    - ECP and LSSW
        - Domain science teams are properly resourced
        - Software teams are incentivized to work with domain science partners
        - "Shared fate" integrated effort driven by project success
        - Enough time and resources to develop longer-term solutions
        - Teaming to understand problems and develop solutions jointly by applications and software technology efforts

        - Larger software environment also associated with ECP (software management, libraries, etc.)
- What happens after ECP
    - LSSW: Dynamics and Sustainability
        - Hardware evolution will be a continuous process, software must evolve on a matching timescale, or it will not be used
        - Performance portability remains a serious concern
        - Scientific application codes need to be also included under the LSSW umbrella
        - Workforce development and planning is an essential component, there are too many single point of failure possibilities in many development efforts
        - Software lifecycle management — retire the old, bring in the new as determined by demand — software providers and users should have well-defined and well-supported relationships
        - Target a larger stable software environment for the supercomputing community (as exists for the cloud-based data-intensive community)

Questions

- Mike Heroux: What if you were to take today's codes and have to put them on the CM5? Would that somehow be simpler, or would that also be just equally challenging?
    - Salman: I think it would be very difficult actually. If you had to do the equivalent of CMSSL now on today's machines, it would be much harder because the problem space is so much more diverse and so much more complicated.
- Rich Carlson: The domain science community is developing codes that you're writing and then there's the libraries and functionalities that the vendors provide. Is there a difference in the sustainability models in those two communities?
    - Salman: I've seen this happening, mostly on the DOE and the NSF side. And I think the models, there are any number of models, and I would say the vast majority of them are broken. So, one model is, we give you money to write software and put it on GitHub. People will add to it or not and it will survive or not, depending on whether people like it or not. I think this model is really bad because high-quality software, especially leadership software cannot be developed that way. You need to do something much more serious than that. On the domain said, it's been a struggle for a long time. For certain classes of scientific software, it's not as much of a struggle, but for others it is. So, I think in a national security lab, there are certain classes of problems that are seen as very important, and they are quite nicely supported. On the other hand, in the Office of Science, that's not the case. If you're doing a high energy physics experiment for example, people can argue as to whether they get the appropriate resources.
    - Salman: On the side of computer science and applied math, the tendency is to do more R&D and not enough hardening of the R&D into production area so people can actually use it. On the science side, it's a similar situation, where our computing hasn't gotten it's due in the sense of support and there's a lot of

reports and studies and surveys that people have done, but the model hasn't changed.
    - Mike Heroux: I understand you would want additional funding for sustainable application software. What other things would help sustainability?
        - Salman: The big question is often things that are good in principle, but don't work in practice. So, what has to happen is that the design of these things, the interfaces, has to be sufficiently flexible.
- Rich Carlson: Are the development tools sufficient to do this, you know continuous integration, IDC? Is there more work that needs to be done there or are they up to a rate that your grad student could use that and have the software be sustainable?
    - Salman: I think those are actually pretty reasonable. If you're a good software person you can probably do it. The problem is student training. Students know one language and their view of programming is very different. If you had students trained at the appropriate level, then the toolkit is not a problem, but that's where this workforce development question comes into play.
- There was a lot of discussion on the needs for student training.
- Alan Sill commented that cloud computing companies are building, but not passing on the savings to the scientific community. There was a lot of discussion based around that comment.

<u>Software Deployment at Facilities</u>

Ryan Adamson, Oak Ridge National Laboratory

- Exascale Computing Project (ECP) Software Deployment Overview
- Complexity is the enemy of scale – separation of duties helps
    - Roles and Responsibilities within the ECP Ecosystem
        - Software integration interfaces with the Testing Task Force, E4S, and ST to adapt builds, tests, and deployments to facility systems
        - Continuous Integration Infrastructure supports the automation of all activities on facility systems
        - Shasta and HPCM testing provides insight into facility hardware from a system functionality perspective
        - The Testing Task Force is the vertical integration of all these components
- Facility-Specific Takeaways
    - Common themes
        - Facility users are the primary focus
        - Facilities prefer to deploy and integrate stable software that is easy to build, test, and support
        - Operational best practices and various regulations are highly values
    - Opportunities for Improvement
        - Facilities need Level 2 support for scientific software
        - Software teams can help maintain facility-specific tests
        - Vendor ecosystem integration
- Software Deployment at Facilities' path to convergence
    - Step 1: Ideation

    - Step 2: Validation
    - Step 3: Planning
    - Step 4: Execution
- Aligning Responsibilities, Ability, and Authority
- Roles and Responsibilities within the ECP Ecosystem
    - ST Developers
    - E4S Software Curators
    - SI/Facilities
- There are three main use cases for CI on facility systems
    - Software Development Testing
    - Ecosystem Integration
    - Facility Operations Assurance
- 5 Types and 6 layers of testing for the ECP Software Ecosystem
    - Types: Build, Validation, Integration, Regression Performance
    - Layers
        - ST teams drive implementation of all types of tests on developer, cloud, or product specific build and test CI pipelines
        - ST teams may run selected, high-value tests of all types of tests on facility CI platforms
        - Spack CI pipelines test builds of packages in the cloud and at some facilities
        - E4S team performs build, validation, and integration test on facility systems and elsewhere
        - SI teams run build, validation, and integration tests on facility systems
        - Facility users run validation tests such as "spack test" and the E4S test suite
- E4S/Facility support model could be a seed crystal for LSSw
    - The E4S Level 2 support plan has funding and facility buy-in
        - The E4S issue tracker on GitHub is the single point of collaboration
        - A new-issue template identifies facility submitted issues
        - E4S L2 support will address facility-submitted E4S issues by E4S L2 support staff within 3 business days
        - Facilities can raise priority of major issues
        - E4S issues will be updated by E4S product teams as progress is being made
        - E4S L2 support will provide regular reporting of E4S support metrics to facility staff
        - We expect the support model to evolve as needs change
- Security is HARD and can't be an afterthought
    - E4S and Spack are well-positioned to provide assurance

Questions

- Mike Heroux: Regarding the security, other than traceability, what other things would you see as enhancements to what we have? The providence is excellent, but there's more than just providence. There's also protecting the pipeline.
    - Ryan: That's right. There's securing the build infrastructure. The process of building a software. And then there's secure design principles and coding

standards. There's training for how to design code securely and there are also tools that exist for doing a dynamic and static analysis of software. I think those are really good to get into, but that can lead you down rabbit holes of what are false positives or not. But ultimately what Spack does in terms of its cryptographic check summing and things, you know that if you get something from Spack, it was built, and it still has the same hash. So, I have some assurance that this is the software that was built on this system. I think that's why I say I think Spack is actually really well-positioned to claim that we're meeting some of the spirit of this executive order and we have been for 5 years.

- Mike Heroux: I'd like to make one more mention of things that I think are in our space in the leadership area. We could look at versions of our software, let's say we found a bug at scale. We could actually scan our software base and see and detect and report things like scale and performance and correctness at scale.
- Rich Carlson: So, are there things that are missing that are going to evolve or appear as our workflow changes?
    - Ryan: Maybe there are two answers. The first one is there are absolutely some things that we could be doing to identify the pieces of software that aren't targeted yet to help adapt them to that new ecosystem that to bring in the ML and AI codes right into E4S to help write Spack packages so that we can deploy them with the tools that we're familiar with, to integrate them within our ecosystem. The other answer is that we should at the very least make sure we stay flexible in terms of having infrastructure that is scalable that we can add things to flexibly.
    - Alan Sill: I just want to make two quick points. One is sustainability and I'm really glad to see the focus on this from the outset. And also, scalability. Thinking about grids, how did we get that? We built a system in which contributing parties who wanted to see progress and could jump in and contribute. Not taking anything away from the tremendous accomplishments of ECP, which are really very impressive. We can't just approach this as a technical problem. It has to be a sociological problem at the outset.
    - Mike Heroux: Yeah, I think we're very aware of that. We have to have first, something that can be scaled. So, what ECP has allowed us is an incubation area where we can do that. But absolutely agree with everything you said.
    - Mike Heroux (in response to Rich's question): That's one of the main drivers for why we want to expand the definition of leadership to be more than just HPC. I think it's a very important part of what we're trying to do.

Roundtable

- Alan Sill: CASC annual survey of ROI practices and research computing and data is a little bit delayed from past years. It will be going out probably within the next month.

Next Meeting

March 2 (12 pm ET)