LSSw Meeting 3

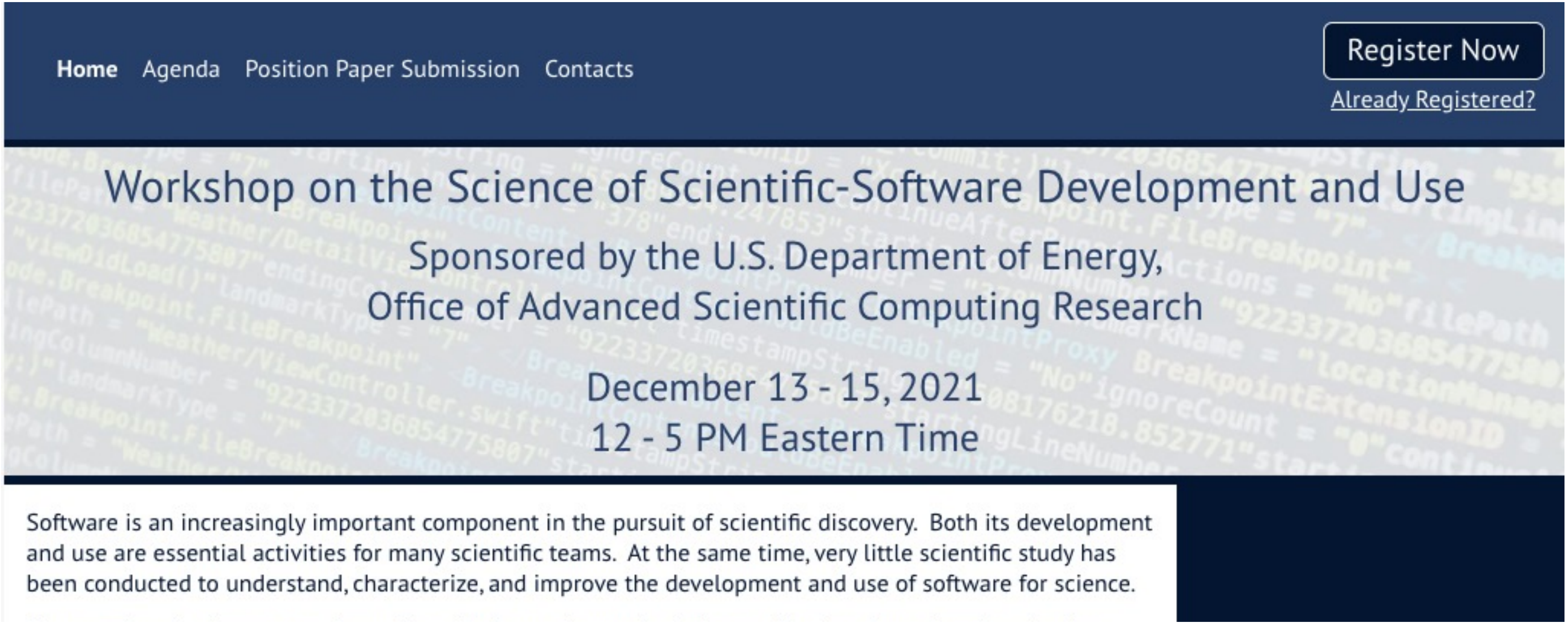
November 18, 2021

Announcements

Preview for LSSw Meeting 4: Dec 16, 2021

- Topic: Expanding the Leadership Scientific Software Developer and User Communities: A panel discussion
- Description: This month we have panelists representing the broader scientific software developer communities:
  - Panelists, TBD
- Prompts:
  - How has the traditional definition of leadership (HPC) scientific software developers limited who can be involved?
  - What is required to make the traditional definition more inclusive?
  - What do you see as the most urgent priority activities in planning for a holistic leadership software ecosystem over the next few years?
  - What is missing from the conversation about sustainable leadership scientific software?

Workshop on Research Software Science: Dec 13 - 15



Information and registration at:

https://www.orau.gov/SSSDU2021

Whitepaper submission deadline: ~~Nov 19~~, Nov 22, 2021

LSSw Meeting 3

- Topic: Progress, impediments, priorities and gaps in leadership scientific software: A panel discussion
- Description: This month we have five panelists representing non-DOE institutions with a variety of relationships to leadership scientific software:

    - Shawn Brown, Pittsburgh Supercomputing Center
    - Jeff Durachta, NOAA
    - Alice Koniges, Maui HPC Center
    - Piyush Mehrotra, NASA
    - Andrew Wissink, US Army

Panelist Prompts

- How do you and your organization use third-party HPC software, from DOE and elsewhere?
- What is particularly valuable about the DOE-based HPC software you use?
- What can make DOE-based HPC software better?
- What are some issues in the HPC software community that are not being sufficiently addressed right now?

- Goal for the day: Identify and record the progress, impediments, priorities and gaps in leadership scientific software from other US institutions.

Shawn T. Brown, PhD.

- Director, Pittsburgh Supercomputing Center
  Vice Chancellor for Research Computing, University of Pittsburgh
- Experience related to leadership scientific software
  - *PI, Bridges-2 Research Computing Platform*
  - *Co-PI, Neocortex AI/ML Testbed Platform*
  - *Technical Lead on 8 Public Health decision support platforms for studying infectious disease spread, vaccine supply chain logistics and distributions, and economic and operational modeling.*
  - *Lead the development of the CBRAIN orchestration platform at McGill's Centre for Integrative Neuroscience*
  - *CTO of the Canadian Open Neuroscience Platform and the Neurohub Platform*
  - *Contributed to chemistry packages GAMESS, Q-Chem, and Dynamo*

Prompts

- How do you and your organization use third-party HPC software, from DOE and elsewhere?
  - The PSC heavily leverages 3rd party software for managing and operating our computational resources.
  - To support users, we deploy multiple 3rd party scientific software packages.
- What is particularly valuable about the DOE-based HPC software you use?
  - While we have not done this recently, the PSC in the past has had a strong relationship with DOE labs in deploying similar HPC systems and worked together to great success to grow together as a community.
- What can make DOE-based HPC software better?
  - Continue to invest in open community-based software platforms.
- What are some issues in the HPC software community that are not being sufficiently addressed right now?
  - Dealing with complexity of data and heterogenous computing.
  - Supporting large-scale AI and Data Analytics tools.

Jeff Durachta

- Lead for Modeling Systems Division, NOAA Geophysical Fluid Dynamics Laboratory, Princeton, NJ
- Experience related to leadership scientific software
  - *Earth System Modeling S/W Development and Optimization, NOAA GFDL, 1999 to present*
  - *Scientific HPC Application Developer and Performance Engineer, Independent Consultant, 1998 to 2016*
  - *Lead Application Performance Engineer, SiCortex, Maynard MA, 2007 – 2009*
  - *Junior / Senior Application Performance Engineer, IBM Corporation, 1990 - 1998*

Durachta, GFDL

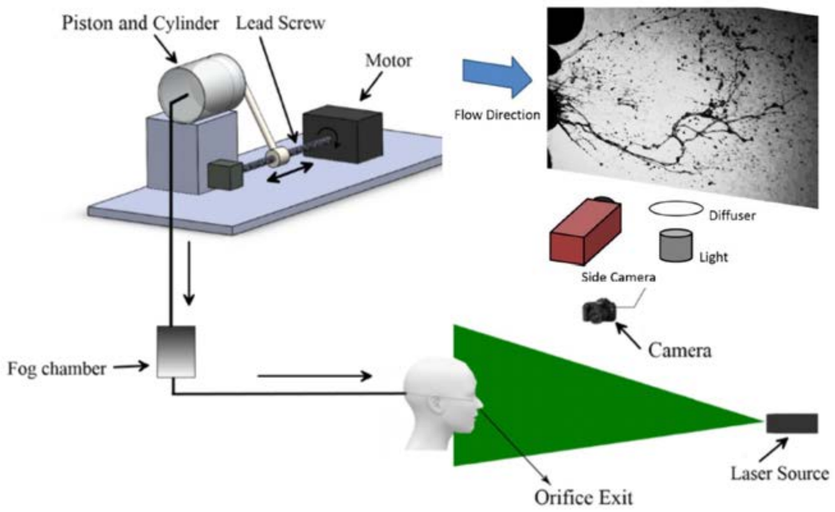
- How do you and your organization use third-party HPC software, from DOE and elsewhere?
  - GFDL (and OAR and NWS) are primarily RYO shops. Thus the primary 3rd party HPC S/W stack is comprised of OpenMP enabled Fortran and C compilers, parallel debuggers, performance analysis tools (such as HPC Toolkit) and MPI libraries. Unfortunately, there is little math library use within my environment. The scientist s/w developers feel that mathematical stencils they produce are part of the art of their science. Further as this is a research environment, the code is often rapidly changing.
- What is particularly valuable about the DOE-based HPC software you use?
  - The 2 HPC software stack tools we use are Spack and HPC Tooklit. Spack has been invaluable as a tool for unifying scientific analysis software builds and thus environments across our disparate versions of Linux. In the area of performance analysis, I have used HPC Toolkit on and off over the years practically since its inception. One of HPC Toolkit’s key value propositions is that it is cross-platform. Analyzing and understanding performance features from the perspective of multiple platforms can be key to getting the big picture. Ubiquity also cuts down on the overheads of learning performance analysis on a new platform.

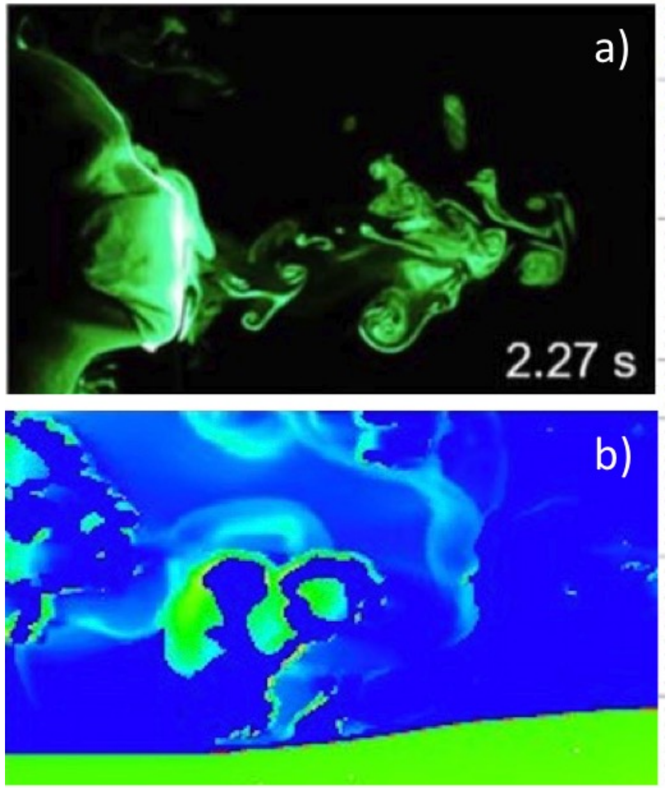
Durachta, GFDL

- What can make DOE-based HPC software better?
  - I think our answer is mostly tied to the question below.
- What are some issues in the HPC software community that are not being sufficiently addressed right now?
  - For us, the landscape of viable programming models still seems rather muddled. At least it seems to be guided by the personal tastes of the developers rather than a set of rules (I'd settle for heuristics myself) that guide the implementation choices. And given that NOAA is very resource limited, we will really only get one shot. More particularly, there is a distinct lack of clarity on a sustainable, Fortran based programming model going forward. The (apparent) HPC software community support for FLANG is encouraging. But from our perspective, the non-convergence of OpenMP and OpenACC is highly problematic even IF one could achieve reasonable Fortran code performance on a given platform with either. For example, the rather different memory characteristics across vendors is but one very problematic area. Maintaining 2 sets of directives is untenable. How do we move the mountains that must be moved to unify OpenMP and OpenACC?

LSSw Sustainability Town Hall

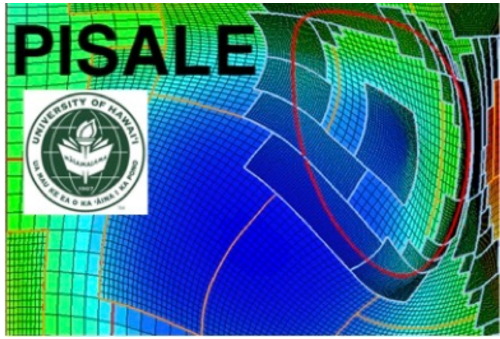
Alice Koniges, koniges@Hawaii.edu
Graduate Faculty and Hawaiʻi Data Science Institute
University of Hawaiʻi





HAWAIʻI DATA SCIENCE

Current UH grants/applications using DOE software and the PISALE code



Elements: ALE-AMR Framework and the PISALE Codebase

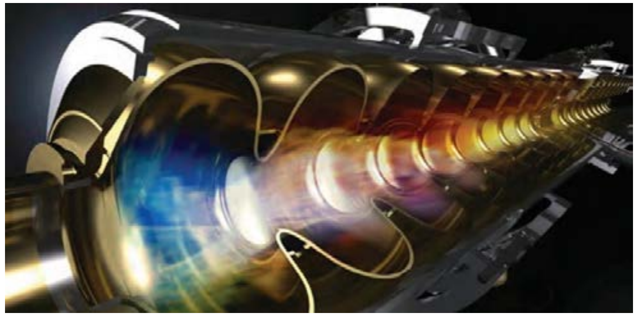
This project will apply the code for simulations of complex groundwater flow processes in Hawaiian islands characterized by highly heterogeneous volcanic rocks and dynamic interaction between freshwater and seawater.

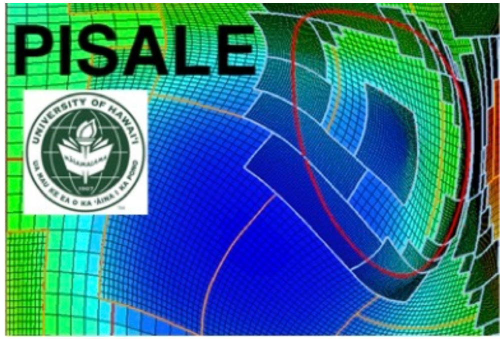
MURI: Faster than the speed of sound

We use the code to study effects of rain, ice, and aerosols on hypersonic vehicles. This is a multi-university effort led by the University of Minnesota entitled Particulate and Precipitation Effects on High-speed Flight Vehicles.

An Extensible High Energy Density Modeling Tool for Extreme Regime

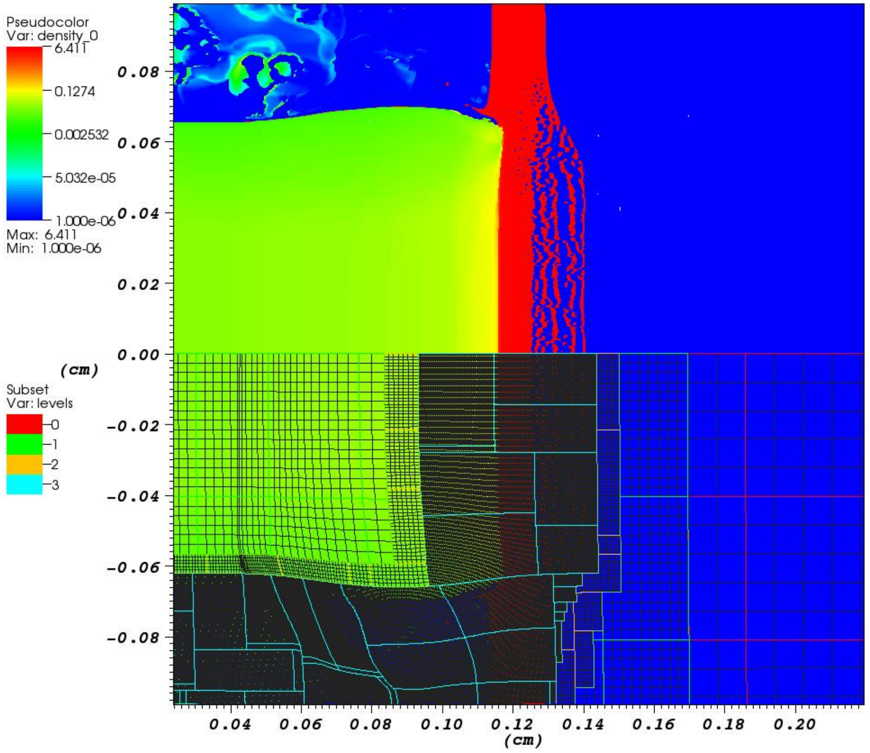
High Energy Density (HED) Physics implies the study of systems at very high pressures and temperatures. Our simulations will address a critical need to understand the interaction between HED material and surrounding liquid material for experiments at the X-ray Free Electron Laser (XFEL) located at the Stanford Linear Accelerator Center.



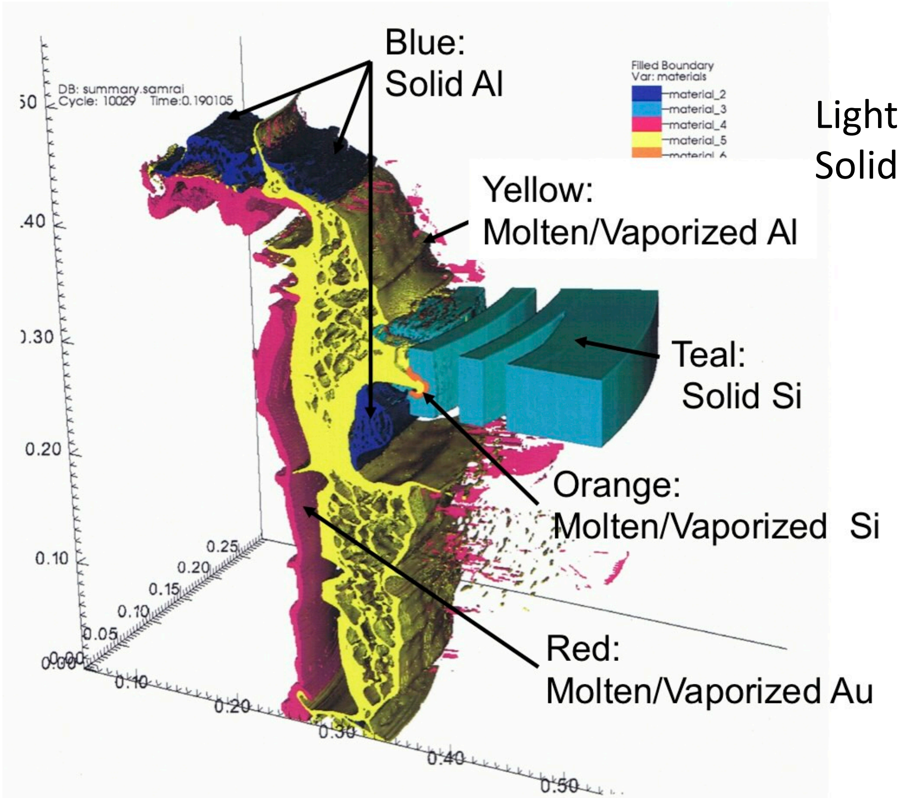


Pacific Island Structured-AMR with ALE (PISALE) code effort makes extensive use of the DOE's VisIt visualization tool

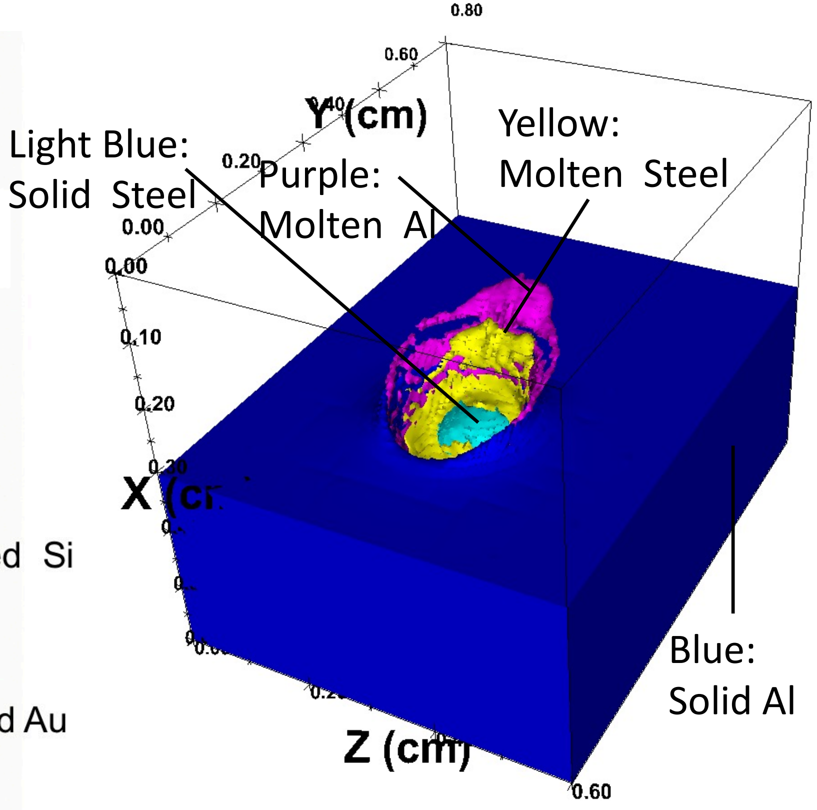
Also uses solvers, SAMRAI



Laser heated plate with blowoff and rear spall. Mesh using AMR is shown in lower half of image.



A complex multimaterial simulation with use of void to create fragments. Model 1/32 of target.



Impact of steel sphere at 45^{o} on Al plate.

Mana User Survey: Some of the tools installed on the UH Mana system are: slurm, hdf5, mpich, openmpi, papi, parallel netcdf, petsc, paraview, scalapack, superlu, swig, and trilinos

- How do you and your organization use third-party HPC software, from DOE and elsewhere?
  - MPICH/OpenMPI for use in parallel CFD computing with openfoam
  - Use conda to pip install python packages e.g. pytorch and TensorFlow
  - DOE software on our central HPC cluster for, job scheduling, library dependencies for other 3rd party software applications, visualization, as well as profiling code for optimization
  - Machine learning and for analyzing data
- What is particularly valuable about the DOE-based HPC software you use?
  - Parallel implementation of OpenFoam
  - A flexible HPC job scheduler that is useful for all scale of HPC systems
- What are some issues in the HPC software community that are not being sufficiently addressed right now?
  - Support/documentation/help
  - Support for containers
  - Documentation of settings

345 compute nodes
120 GPUs
8,452 cores
62.9 TB of memory
1 PB of long term storage
61 TB of flash scratch storage
150 TB of standard scratch storage

Leveraging DOE Software at NASA

Dr. Piyush Mehrotra

Division Chief

piyush.mehrotra@nasa.gov

http://www.nas.nasa.gov

NASA Ames Research Center, Moffett Field, Calif., USA

November 2021

Supercomputing Center@ NASA Ames

NASA's Premier Supercomputer Center

Resources have broad mission impact across all of NASA's Missions

Over 600 science & engineering projects with more than 1,600 users

Hardware Assets

Computing Systems

- **Pleiades** – 7.25 PF peak
  - 241,324 cores, 11,207 nodes
  - InfiniBand Interconnect, hypercube topology
  - GPU racks – NVIDIA V100: 83 nodes – 0.65 PF peak
- **Electra** – 8.32 PF peak
  - 124,416 cores; 3456 nodes
  - Modularized container-based approach – PUE ~1.03
- **Aitken** – 8.41 PF peak
  - 177,152 cores; 2176 nodes
  - Modularized container-based approach – PUE ~1.05

Visualization: hyperwall

- 128-node GPU-based cluster
- 8x16 LCD displays; 245 million pixels

Storage – Global Shared File Systems

- 7 Lustre File systems: ~65 PB
- Archive tape system capacity:1 EB

Services

- ***Systems*** – Customized solutions including compute and storage solutions to meet specific project or mission requirements.
- ***Cloud*** access for immediate or non-standard computing.
- *Network* – End-to-end network performance enhancements for user communities throughout the world.
- ***Application Performance and Productivity*** – Software solutions provided to research/engineering teams to better exploit installed systems.
- ***Visualization and Data Analysis*** – Custom visualization during traditional post-processing or concurrent during simulation to understand complex interactions of data.
- ***Data Analytics & Machine Learning*** – Exploitation of data sets through neural nets and emerging new techniques.
- ***Custom Data Gateways*** – Custom data portals to support diverse programs and projects

Leveraging DOE Software

- E4S – Extreme Scale Software Stack
  - Many packages available, some will be useful for HECC
  - Will help reduce work for HECC staff and users
    - Migrating from using pkgsrc to spack-based E4S packages
    - Reducing the # of user installing packages in their own directories
  - Full stack too large for our purposes – working with Sameer Shende

- TOSS – Tri-lab Operating System Stack
  - Working collaboratively to assess the possibility of adapting TOSS to run in NASA's high end computing environment.
    - NASA has some area of expertise that may improve the overall coverage of hardware support that TOSS supports (e.g. lustre, RDMA drivers, MOFED).
  - Successful prototype implementation across 9 different architectures.
  - At a Go/No-Go decision point, need a NASA/DOE formal agreement on the collaboration.

Benefits of DOE-based HPC software

- Tremendous amount of effort required to maintain software environments. Leveraging open-source collaboration towards more commonality across systems and environments is beneficial.

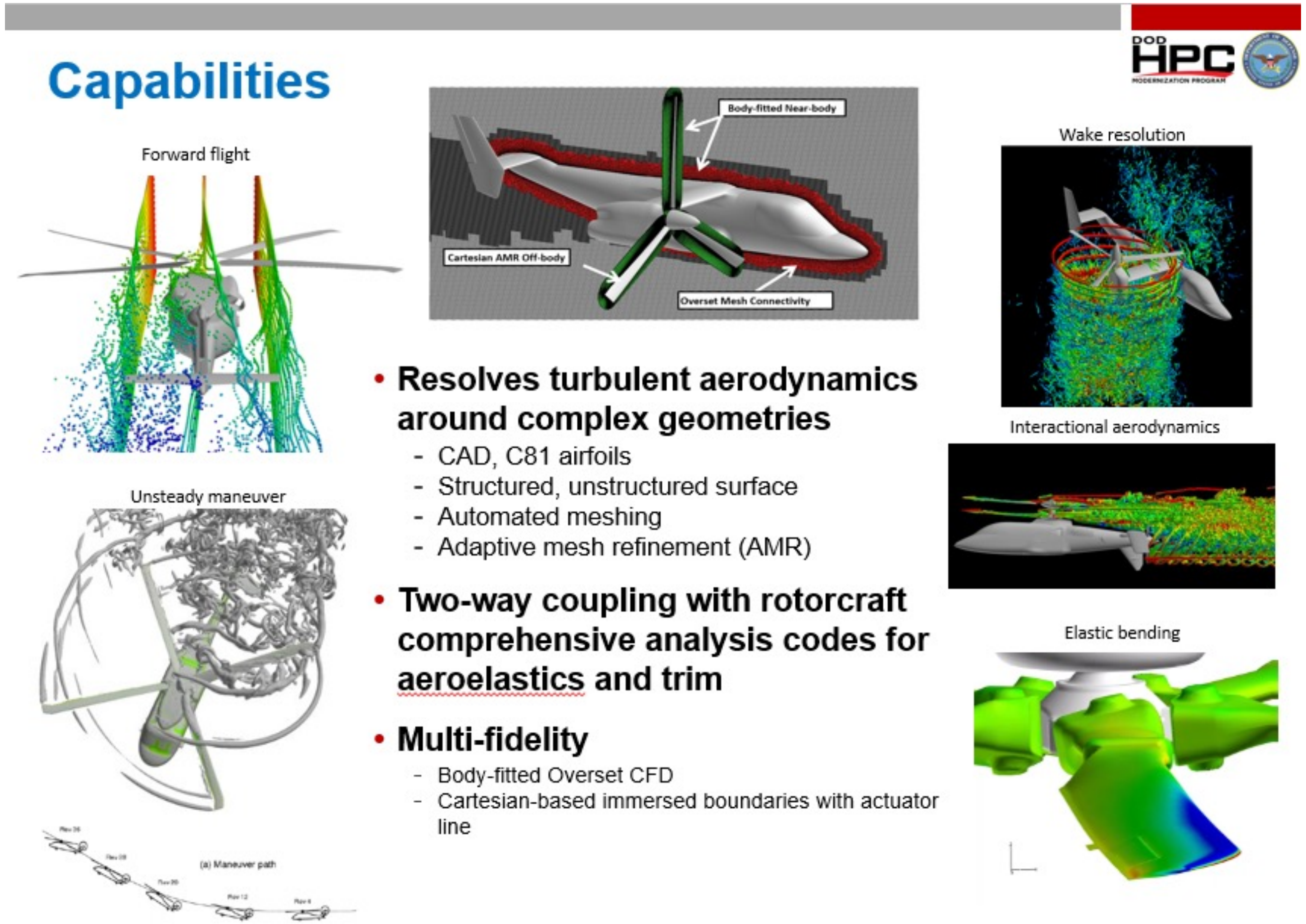
Challenges

- Performance portability/scalability across heterogeneous architectures/environments.
- Application-level fault-tolerance/resilience in the face of hardware/software failures.
- …

Andrew Wissink

- Aerospace Engineer, US Army
- Principal Developer of Helios vertical lift simulation software

- Experience related to leadership scientific software
  - *Former developer of DOE SAMRAI AMR software*
  - *Over the past decade, focused on coupled multi-physics using Python linked with underlying C/C++/Fortran*
  - *In-situ visualization*



Prompts

- How do you and your organization use third-party HPC software, from DOE and elsewhere?
  - We use SAMRAI, Trilinos, SIERRA, hypre, PETSc, TAU
  - Starting to consider Kokkos, RAJA as we move to GPUs
- What is particularly valuable about the DOE-based HPC software you use?
  - Tested and validated
  - Well supported
  - License that supports restriction-free distribution to US govt and industry
- What can make DOE-based HPC software better?
  - DOE are heavy users of templated C++, we don't use these features
  - Often difficult to integrate with our codes
- What are some issues in the HPC software community that are not being sufficiently addressed right now?
  - Mesh-based GPU support (many of the support infrastructures are solver oriented)