WEAVE

Newsletter #1

October 2024

Contents

WEAVE Essentials

IN THIS ISSUE

- **WEAVE environment 1.0 is now available and accessible to ALL LC users.**
- **Sina expands to engineering codes.**
- **HPC Amazon Tutorial Demontrates WEAVE's ecosystem**
- **Sina joins Axom**
- **Data management across codes is getting easier**
- **Summer Students**

WHAT YOU NEED TO KNOW

WEAVE is growing. Renee Olson and Xiao Chen joined WEAVE and are focusing on UQ/ML tools with a focus on integration with the other WEAVE tools (via Kosh Operators).

Gabriel Waegner joined us as part of the Academic Graduate program near the end of Q3. His focus is on the Sina common output format.

Barry Sly-Delgado joined us for a summer internship and looked into integrating TaskVine with Merlin, especially in regards to data movement.

WEAVE had a successful AWS-based tutorial, where about three dozen attendants joined efforts, sharing their individual AWS instances limited computing power, to successfully complete a UQ study.

ANNOUNCEMENTS

WEAVE is seeking UQ experts to identify packages useful to the community and help inform WSC/WPD UQ Strategy.

LLNL-BR-870669

Around WEAVE

WEAVE's AWS HPC Tutorial

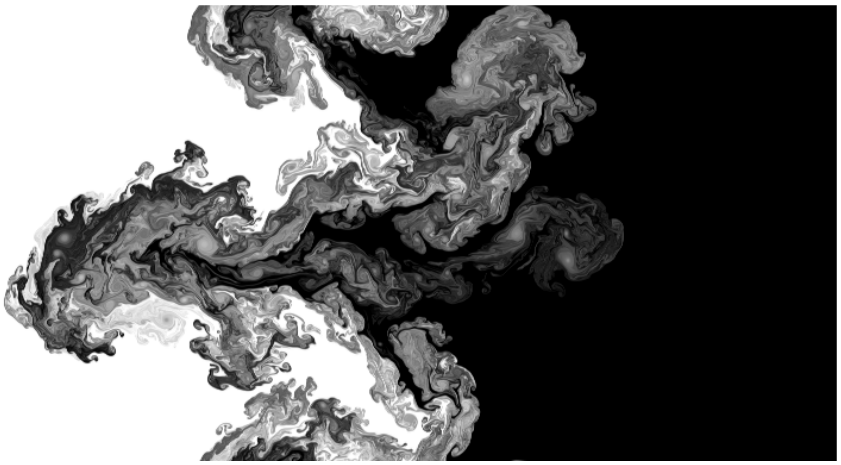

Figure. 1: Rayleigh-Taylor simulation two fluids with different densities and random small surface perturbation mix together.

WEAVE offered a tutorial showcasing all of our tools at the HPC innovation Center's summer tutorials series. About three dozen tutorial participants were taught how to use the Pyranda code to simulate a Rayleigh-Taylor problem and then went through a series of lessons, including:

1. Using Maestro to schedule a parameter study.
2. Using Trata to generate space-filling samples in the domain of interest.
3. Using Sina to track and catalog simulation results
4. Using Kosh to retrieve the necessary data for a simple Uncertainty Quantification Study.
5. Using IBIS to generate a Gaussian Process model predicting the mixing depth at time=60.

Merlin was used to spread compute across the AWS cloud distributing the work accross the participants.

The tutorial materials are available here: https://github.com/llnl/weave-demos under the CZ/pyranda_raleigh_taylor directory.

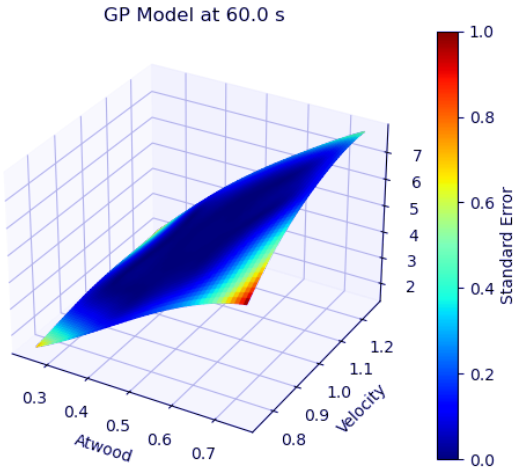

Figure. 2: Rayleigh-Taylor mixing width at time=60 as predicted by a GP trained on results generated from multiple AWS instances

Step Name	Step Workspace	Workers
run-pyranda_ATWOOD.0.429.VEL.0.967	run-pyranda/A1	single_worker.%ip-172-31-7-206.us-west-1.compute
run-pyranda_ATWOOD.0.429.VEL.0.967	run-pyranda/A1	single_worker.%ip-172-31-7-71.us-west-1.compute
run-pyranda_ATWOOD.0.419.VEL.1.105	run-pyranda/A1	single_worker.%ip-172-31-13-156.us-west-1.compute
run-pyranda_ATWOOD.0.419.VEL.1.105	run-pyranda/A1	single_worker.%ip-172-31-10-50.us-west-1.compute
run-pyranda_ATWOOD.0.496.VEL.0.945	run-pyranda/A1	single_worker.%ip-172-31-15-203.us-west-1.compute
run-pyranda_ATWOOD.0.496.VEL.0.945	run-pyranda/A1	single_worker.%ip-172-31-6-185.us-west-1.compute
run-pyranda_ATWOOD.0.356.VEL.0.869	run-pyranda/A1	single_worker.%ip-172-31-5-90.us-west-1.compute
run-pyranda_ATWOOD.0.356.VEL.0.869	run-pyranda/A1	single_worker.%ip-172-31-13-156.us-west-1.compute
run-pyranda_ATWOOD.0.313.VEL.0.886	run-pyranda/A1	single_worker.%ip-172-31-7-71.us-west-1.compute
run-pyranda_ATWOOD.0.313.VEL.0.886	run-pyranda/A1	single_worker.%ip-172-31-0-93.us-west-1.compute
run-pyranda_ATWOOD.0.589.VEL.1.015	run-pyranda/A1	single_worker.%ip-172-31-1-234.us-west-1.compute

Figure. 3: Merlin was used to harness AWS cloud power during an AWS WEAVE UQ tutorial

Successful Summer Internships

Taskvine

This summer Barry Sly-Delgado worked on integrating the TaskVine task scheduler into Merlin. TaskVine is a task scheduler with built-in capabilities to utilize in-cluster resources (bandwidth and disk), to effectively and automatically move data during workflow execution. This serves as an alternative to the Celery task queue that is currently available in Merlin. This integration involved the creation of a DAG manager, the Vine STEM, which converts Merlin specification steps to TaskVine tasks.

A key TaskVine feature is that it maintains the provenance of individual tasks allowing for the recomputation of results in the case of faults. As provenance is maintained, intermediate results can be maintained across workflow executions, reducing end-to-end runtime latency for separate workflow executions that share common task dependencies. The TaskVine scheduler should be available in an upcoming Merlin release and will require only minimal changes to current specifications to enable.

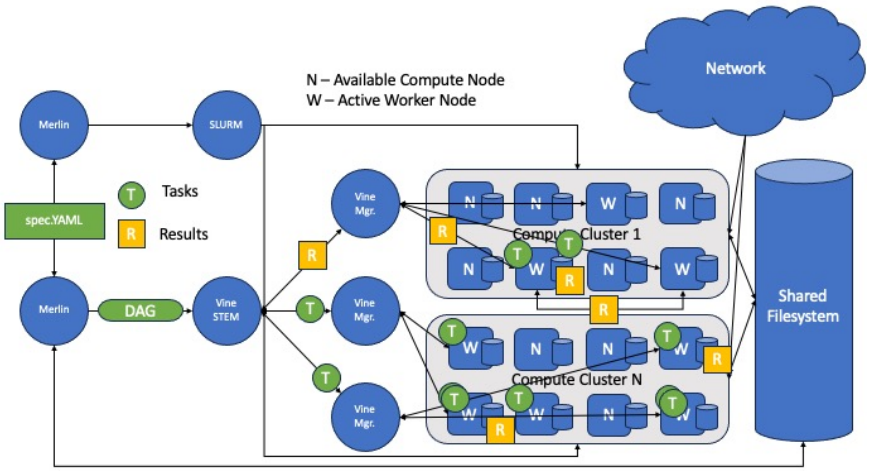

Figure. 4: Merlin using TaskVine Architecture

```
merlin:
    resources:
      task_server: taskvine
      workers:
       default:
        manager: hello_samples_manager
        steps: [step_1, step_2]
      managers:
       hello_samples_manager:
    samples:
        generate:
            cmd: python3 $(SPECROOT)/make_samples.py --filepath=$(MERLIN_INFO)/samples.csv --number=$(N_SAMPLES)
        file: $(MERLIN_INFO)/samples.csv
        column_labels: [WORLD]
```

Figure. 5: New specifications to take advantage of TaskVine in Merlin

The GWC Early Internship Program Comes to WEAVE

This summer, the Lab welcomed two returning students from the Girls Who Code (GWC) Early Internship Program (EIP), one of whom contributed to WEAVE's own Sina! The program (now in its second year) connects the Lab's GWC outreach efforts to the hiring pipeline. As part of her second year with the program, Harshitha "Hershey" Sarathy spent half her time on a challenging "learning project" curriculum, and the other half meeting, learning about, and working with LLNL teams on real problems, building to a new feature for Sina: **the ability to export data to Pandas dataframes.** This will streamline a number of analysis and visualization workflows, as well as pave the way for some exciting future expansion. The Sina team is thrilled with her work and wishes her similar resounding successes for the upcoming academic year!

If you'd like to learn more about the EIP program, please reach out to Jennie Wright or Jamie Lewis.

FEATURED ITEM: WEAVE Environment

To enhance the user experience with workflow tools, WEAVE is providing a world-readable, Spack-based Python environment for all LC users across all zones (CZ/RZ/SCF). This environment includes all WEAVE tools, commonly used Python packages, and, most importantly, several commonly used but difficult-to-build machine learning and AI packages. It aims to offer a stable, well-tested environment that users can activate directly or use as a foundation for custom Python environments.

Many users are familiar with the /usr/tce deployment on LC systems. While the WEAVE environment relies on some components of this deployment (e.g., MPI), it aims to provide a richer Python ecosystem with the capability to add packages, such as the WEAVE tools. In the future, WEAVE-badged tools will also be available (stay tuned for more information on the WEAVE badging program in an upcoming newsletter). To learn more about the WEAVE environment, visit: https://lc.llnl.gov/weave/llnl/environment.html.

What is the WEAVE environment?

At its core, the WEAVE environment is a Spack-based Python environment that includes all WEAVE tools and a wide variety of commonly used Python packages (e.g., PyTorch, Matplotlib, Scikit-learn, etc.). All WEAVE test suites are run against this environment, and if everything passes, the environment is deployed for our LC users. The WEAVE documentation, available at https://lc.llnl.gov/weave, is also updated at that time.

For the full list of packages available in the environment see:

- blueos (gpu-enabled)
- toss (cpu-only)

It's worth noting that a similar process occurs every night using the development branches of the WEAVE tools. This nightly environment is also made available to our users. However, despite our best testing efforts, we cannot guarantee that new features will be stable in this environment. Additionally, due to its nightly nature, the environment may experience temporary glitches during (re)deployment. Therefore, we do NOT recommend using the development version for production purposes. Instead, it should be used as a way to evaluate WEAVE's newest features and bug fixes.

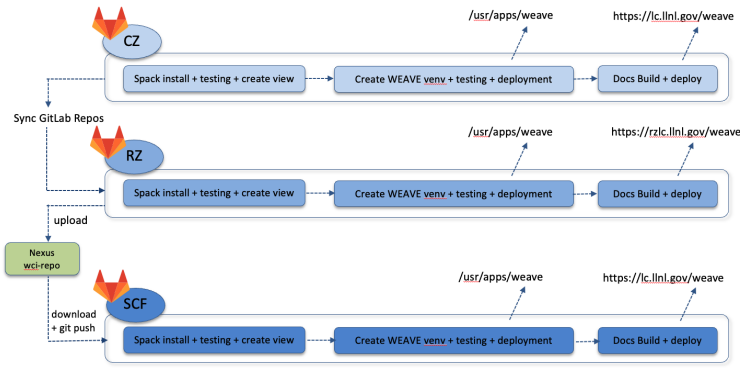

Figure. 6: Spack-based WEAVE CI/CD in CZ, RZ and SCF

Activating

You can use the WEAVE environment on any toss_4_x86_64_ib and blueos_3_ppc64le_ib_p9 machines on LC. Since the WEAVE virtual environment is Spack-based, the install is OS-specific. However this is transparent to you as a user, in order to use the WEAVE virtual environment, simply follow the instructions bellow:

bash

```
# On clusters with no gpu, activate latest 'production' version (1.0)
source /usr/apps/weave/weave-prod-cpu/bin/activate
# On clusters with gpu, activate latest 'production' version (1.0)
source /usr/apps/weave/weave-prod-gpu/bin/activate
```

[t]csh

```
# On clusters with no gpu, activate latest 'production' version (1.0)
source /usr/apps/weave/weave-prod-cpu/bin/activate.csh
# On clusters with gpu, activate latest 'production' version (1.0)
source /usr/apps/weave/weave-prod-gpu/bin/activate.csh
```

In order to access our **development** version replace *prod* with *develop*.

LLNL-BR-870669

FEATURED ITEM: Sina Library

Standardization EFFORT

While Sina has been integrated across WSC codes, the level of integration varies a lot between codes. WEAVE is aggressively reviewing existing integrations with a goal of harmonizing how users can configure and customize their Sina output. In addition, the team is working to ensure that each code provides a useful set of default meta-data in its Sina output. The team is currently working with ARES to support the expected semantics for that code, with MARBL and ALE3D being reviewed in parallel. Among the features being explored with ARES are intermediate dumps and subsets dumps.

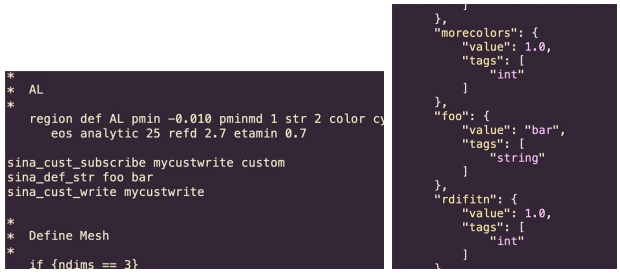

Figure. 7: Excerpts from ares deck and the resulting "custom" sina output json file dump

AXOM

The C++/FORTRAN Sina library is joining the AXOM shared code infrastructure project in order to simplify its integration with WSC codes and integrate into AXOM's build and test environment. After this integration, the Sina C++/Fortran library will be maintained in the AXOM repository structure, while the Sina Python package will remain in the WEAVE project structure.

Diablo Integration

A FORTRAN interface was added to the Sina library enabling WEAVE to integrate with a wider set of WSC codes, including the engineering analysis codes Diablo and Paradyn. The first Fortran code to see Sina integrated was Diablo. The latest release of Diablo is able to produce Sina common output files.

Common Output Effort

Due to concerns related to file sizes and readability, Gabriel has been working on identifying potential alternate formats for the Sina common output file (currently JSON). His main focus so far is on HDF5 which shows some promising performance improvements in both size and speed for large files. Other output formats are also being considered. Feedback about the common output file, especially in regards to data storage (curves, etc.) are welcome and should be addressed to the WEAVE team either via the MS-TEAM or by email at: weave-support@llnl.gov

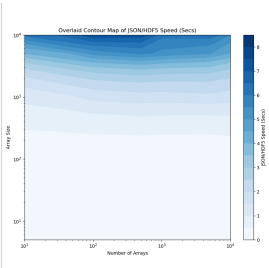

Figure. 8: Comparing JSON vs HDF5 write speed for a Sina record

LLNL-BR-870669

WEAVE Project Updates

Maestro (white242@llnl.gov)

Maestro 2.0 is taking shape. Maestro has better flux support for flux-native machines. In the works:

- Text-based User Interface for statuses
- Python API for study workspaces/metadata
- Database for status
- Enhanced Restarts

Maestro 2.0 will be a breaking change to the user-facing-YAML-based study specification and in addition will have several backend improvements. We are taking requests and comments on things that work or don't work well in the spec (Merlin or Maestro), and missing features or capabilities you would find useful. For example, take a look at and contribute to our step level control flow discussion at: `https://github.com/LLNL/maestrowf/discussions/440`

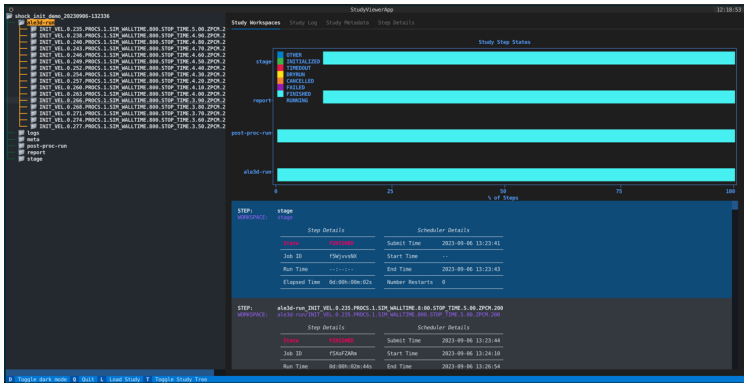

Figure. 9: Upcoming Maestro feature: Text-based User Interface for Statusing Studies

Merlin (gunnarson1@llnl.gov)

Merlin has new **detailed-status** and **queue-info** commands to enable easier management and monitoring of workflows. We're working on improved examples and a more complete test suite as well as increasing the resiliency for worker jobs. Barry worked on integrating with TaskVine to enable more flexible data movement capabilities. This will be available in an upcoming Merlin release.

Sina (haluska2@llnl.gov)

The Sina Python package now has the notion of aliases when searching through the data section of the Sina records. Using aliases you can search through records that describe the same thing via different metadata names. For example one code or user might store the number of processors as "processors" while another stores it as "num_procs". Using the alias feature to link them, you can now find both when search for records where "processors" is defined.

In additon, there is an ongoing effort to allow users to use Sina records with regular Python ORM tools. This will allow teams building custom workflow environments to include Sina in their use-case-specific data management infrastructures.

Kosh (moreno45@llnl.gov)

Decorators were added along with a notebook to help users avoid locks when using Kosh in a distributed fashion. Aliases are available in Kosh as well and can be used when searching for features. A notebook was added to help users better understand how to use Kosh in the context of multiple simulation runs (ensembles).

Users can now leverage Pandas for analysis: Pandas readers are now available through Kosh (e.g pandas/excel, etc.). There is now a store.to_dataframe function that will export result to a Pandas dataframe for additional analysis in the Pandas ecosystem.

Kosh store and dataset attributes can now be exported to pandas dataframes.

There is now a Kosh LNorm that Interpolates Two 2-D Arrays [x0, y0], [x1, y1] or four 1-D arrays x0, y0, x1, y1 to a shared domain and calculates: *LNorm = np.sum(abs(y0-y1)**power)**1/power*

Trata (olson59/chen73@llnl.gov)

IBIS (olson59/chen73@llnl.gov)

- Trata and IBIS both have a new 1.1.0 release. This release includes Kosh wrappers for Trata and IBIS modules that allow users with Kosh stores to more easily use Adaptive Sampling, MCMC, one-at-a-time effects, and sensitivity plots on their Kosh datasets.
- We integrated contributions from a user to make Morris effect computation robust to missing data.
- Sobol indices are now available as an option for sensitivity studies in the IBIS 1.1.0 release.
- We are looking into PSUADE for additional surrogate modeling and other UQ capabilities.

PyDV (moreno45@llnl.gov)

PyDV had numerous bug fixes, but none more important than the convolution functions **convolb** and **convolc** which now match and outperform Ultra. Other new features include:

- Labels now get updated according to curves present
- Reinserted underscores to curve names
- Files not ending in valid data should now load correctly
- Menu now has ‘menulength‘ option
- Added ‘sum‘ function, ‘cumsum‘ function, ‘area‘ function
- Updated ‘stats‘ function to show more information
- Updated ‘savecsv‘ and ‘readcsv‘ to allow curves with different lengths
- Updated ‘read‘ to allow x-columns to be x-tick labels
- You can now control the format for the ‘disp’ command
- Curves read from Sina common output file understand multiple independent curves.

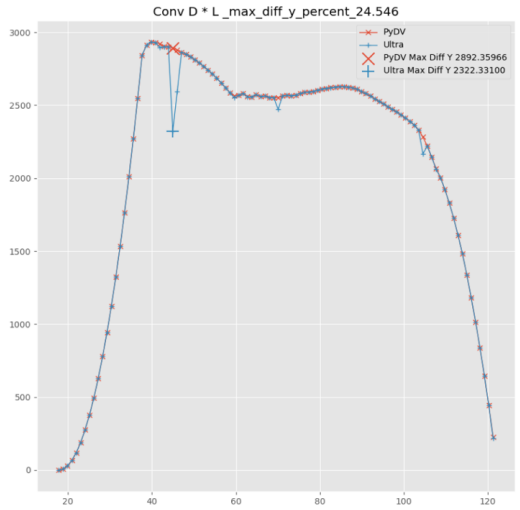

Figure. 10: PyDV fixes issues in convolb and convolc, and now outperforms Ultra. In the plot, the blue lines and plusses show an artifact in Ultra’s convolutions that does not occur with the PyDV implementation.

WEAVE Resources

WEAVE Community

You can reach out to the WEAVE team or the WEAVE community on our MS Team or via email at: `weave-support@llnl.gov`

WEAVE Documentation

The WEAVE documentation can be found on readthedocs at: `https://llnl-weave.readthedocs.io/en/latest/index.html`

On LC you can access our development documentation on CZ, RZ, SCF at:

`https://lc.llnl.gov/weave` and `https://rzlc.llnl.gov/weave`

A set of tutorials (some accessible only from their allowed zone) can be found under the tutorials section of the above site.

WEAVE Environment

All WEAVE packages are available via pip. Most of them are also available with Spack or Conda.

On LC you can access WEAVE on CZ, RZ, SCF at:

/usr/apps/weave

e.g: **source /usr/apps/weave/weave-prod-cpu/bin/activate**

See instructions at: `https://lc.llnl.gov/weave/llnl/environment.html`

Note: If you prefer or need to create your own virtual environment you can do so from our Spack-based distribution:

/usr/apps/weave/tools/create_venv.sh -p cpu -e venv -v latest-stable

or to create from our development version:

/usr/apps/weave/tools/create_venv.sh -p cpu -e venv -v latest-develop

Pro Tips

Merlin Samples Tip

Want to run a workflow with a different samples file but don't want to change the spec? No problem, simply use the –**samplesfile** option with **merlin run**. For example:

```
merlin run --samplesfile new_samples.csv
```

Exporting Kosh to Pandas

Want to export the content of a Kosh dataset (or store) to a Pandas dataframes? Use their **.to_dataframe()** method.

```
df = dataset.to_dataframe()
```

The WEAVE Team

Charles Doutriaux (Team Lead)

Charles Doutriaux

Charles has nearly 30 years of experience at the lab. He obtained his post-graduate degree in Climate and Physico-Chemistry of the Atmosphere at the University Joseph Fourier in Grenoble, France, in 1996. He subsequently joined LLNL's Program for Climate and Model Diagnostics where he became the technical lead for the Community Data Analysis Tools (CDAT) and was a contributing author to the 2007 Nobel Peace Prize Winning Intergovernmental Panel on Climate Change Fourth Annual Report (IPCC AR4). He later led the Earth System Grid's Compute Working Team international group. ESGF. ESGF and CDAT received multiple Federal Laboratory Consortium for Technology Transfer awards and ESGF received the 2017 R&D 100 award. In 2019, Charles joined WSC's to become the Advanced Machine Learning group (aka Vidya) deputy lead. As part of Vidya Charles created Kosh an open-source tools for data management. In 2022, Charles assumed the role of team leader for the Workflow Enablement and Advanced Environments (WEAVE), with a focus on developing tools that empower users' workflows and enhance data management solutions.

Xiao Chen (UQ/ML)

Xiao Chen

Dr. Xiao Chen has over a decade of experience at LLNL, holding a Ph.D. in Applied and Computational Mathematics from Florida State University. Since 2011, he has served as Principal Investigator for a $2.1M LDRD ER project on HPC-enabled stochastic inversion and for a $1M DOE-OE UQ/ML project focused on hybrid learning and optimization for power grid resilience (2020-2024). Additionally, he has been Co-PI for DOE-OE UQ/ML projects, including a $700k adaptive sampling initiative (2019-2020) and a $1M real-time stochastic decision-making tool for power grids (2016-2019). Dr. Chen's research has resulted in approximately 50 peer-reviewed publications and a U.S. patent. He has also contributed to projects that received the R&D 100 and PLS awards. Committed to mentoring and fostering innovation, he collaborates across various LLNL departments to advance multidisciplinary scientific endeavors.

Brian Gunnarson (Merlin Lead, WF Orchestration)

Brian Gunnarson

Brian Gunnarson graduated Magna Cum Laude from the University of Oregon (UO) in 2022 with a bachelor's degree in Computer Science and minors in Mathematics and Computer Information Technology. While at UO, he co-founded Inquire, a website designed to improve communication between professors and students during the COVID-19 pandemic, which was implemented in multiple classes. Shortly after earning his degree, Brian joined the Workflow Enablement and AdVanced Environment (WEAVE) team as the software lead for Merlin, a workflow orchestration tool. During his two years at the lab, he received a WSC Silver Award for his contributions to Merlin and co-authored a paper with the Inertial Confinement on El Capitan (ICECap) team that was published in Physics of Plasmas.

Rebecca Haluska (Sina Lead, Data Management POC)

Rebecca Haluska

Rebecca 'Becky' Haluska has been at the lab for 7 years, working on projects from CSR and performance monitoring to ML-driven antibody design, plus outreach through GWC and the EIP. Under the auspices of WEAVE, she works with users and code teams to increase the availability of critical simulation data, as well as designing query and visualization tooling modules to help users test, build, and automate workflow data management. Most of this finds a home in Sina, an open-source tool hosted and supported by WEAVE that's seen integration with many major codes around the lab.

Jorge Moreno (Kosh and PyDV Lead, Data Management)

Jorge Moreno is the lead developer for Kosh and PyDV. He attended the University of California, Berkeley where he obtained an M.S. in Mechanical Engineering in 2017 while researching within the Combustion Modeling Lab. He joined Sandia National Laboratories as part of the Systems team in 2018 where he was the lead for multiple tests and the Modeling and Simulation liaison lead. Later on, he joined the Thermal Simulation team where he was the PI for a Verification, Validation and Uncertainty Quantification (VV&UQ) project and the lead for system level thermal analyses. He joined LLNL in August 2022. Jorge split his time between WEAVE and is embedded with other projects to help improve and develop WEAVE tools. He originally worked with WPD Suites and moved to the Vidya Advanced Machine Learning group in April 2024.

Jorge Moreno

Lina Muryanto (CI/CD Lead)

Lina received a B.S in Mathematics with Computer Science from Massachussetts Institute of Technology, Cambridge and a M.S in Computer Science from Stanford University. She was a senior software engineer on the Solaris Cluster team at Oracle from 2000 until 2012 and the principal software engineer for the Distributed Resource Allocation Manager (DReAM) team at oracle from 2012 until 2017. In 2018 she joined LLNL as the lead computer scientist for continuous integration and development (CI/CD) on the Earth System Grid Federation Project. In 2021 Lina joined the strategic deterrent program. She joined WEAVE in 2022 and has built their CI/CD from scratch. Lina is passionate about achieving high software quality and reliability through software test development, automation of software integration and release process. Lina Muryanto is the lead for WEAVE's CI/CD and Documentation

Lina Muryanto

Renee Olson (IBIS and Trata Lead, UQ/ML Integration Lead)

Renee has been at the lab for nearly three years working on various projects. She is currently the lead developer for WEAVE's UQ tools, Trata and IBIS, and is working to make more lab-created UQ and ML tools available in the WEAVE virtual environment. Her research interests include clustering, data reduction, Bayesian inference, and computer vision. During graduate school, Renee completed two summer internships at LLNL working with the DTED Flight Sciences team on a multi-fidelity Gaussian process model and a hierarchical Bayesian Inference with mentors including Ana Kupresanin, Kevin Quinlan, and Jeremy Thornock. Renee graduated with a M.S. degree in Statistics/Data Science in 2021 from Cal State East Bay, and a B.S. from Brigham Young University.

Renee Olson

Barry Sly-Delgado (Summer Intern)

Barry Sly-Delgado is a 4th-year Ph.D. student studying computer science at the University of Notre Dame.At Notre Dame, Barry works under Professor Douglas Thain within the Cooperative Computing Lab. His research focuses on integrating data management capabilities for large-scale scientific workflows. Over the summer Barry worked with the WEAVE team on integrating TaskVine, a task executor that manages data during workflow execution, within Merlin.

Barry Sly-Delgado

LLNL-BR-870669

Gabriel Waegner (Graduate Program, Sina)

Gabriel Waegner is a recent Computer Science graduate from Arizona State University, working as an Academic Graduate Appointee at LLNL formerly with the ACES team as a front end developer and API tester, and is now working with the WEAVE team to optimize file management.

Gabriel Waegner

Jeremy White (Maestro Lead, WF Orchestration POC)

Jeremy White is a Physicist and Software Developer at Lawrence Livermore National Laboratory. His research interests include algorithm development for multi-physics simulations and modeling, computational workflow automation and infrastructure, and software engineering. He holds a Ph.D. in Engineering Mechanics from the University of Wisconsin-Madison.

Jeremy White

Dan Laney (WSC CP Workflow Project Lead)

The WSC/CP Workflow project develops tools for meshing, visualization, analysis, mesh-to-mesh linking and workflow & data management. In his role as workflow project leader, Daniel has collaborated with teams in Strategic Deterrence and has been active in forming cross laboratory collaborations, co-chairing the first Workflow Workshop and Hackathon (WoWoHa) held at Sandia Laboratories in 2019 and served on the program committee for the first two DOE Data Days. As a member of the Center for Applied Scientific Computing in Computing, Daniel has been a researcher in scientific visualization, data compression and analysis, and related areas. Daniel also spent a decade on the KULL computer science team, working mainly on Silo outputs, Tracer particles, and simulated diagnostics for NIF experiments. Daniel received his Ph.D. in Engineering from the U. C. Davis in 2001 and a B.S. in Physics from U. C. Santa Barbara in 1996.

Dan Laney

Auspices

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-BR-870669