

Research, leadership, and progress
amid a pandemic

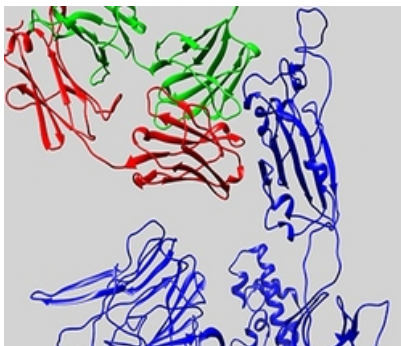


CASC Newsletter | Vol 9 June 2020

In This Issue:

- [CASC and LLNL's Response to the COVID-19 Pandemic](#)
- [Collaborations: CASC Leaders in the Scientific Community](#)
- [Lab Impact: Caliper, SPOT, and Hatchet: Enabling the Whys and Wherefores of Performance Analysis](#)
- [Advancing the Discipline: High-Order Matrix-Free Algorithms for the Next Generation of Supercomputers](#)
- [Path to Exascale: Umpire: Managing Heterogeneous Memory Spaces for Performance Portability](#)
- [Highlights](#)

CASC and LLNL's Response to the COVID-19 Pandemic



Visit LLNL's [COVID-19 website](#).

As the world reels from the effects of the COVID-19 pandemic, CASC researchers—like much of the world—are finding a new normal as they settle into their research and development activities from home offices. As video conferences replace daily personal interactions and walks down the hall replace sometimes grueling commutes through Bay Area traffic, LLNL, and CASC are still right out in front, working with scientific community to accelerate technologies to reduce—and hopefully to eliminate—the damaging health and economic effects of this pandemic. In CASC, we are contributing our knowledge of algorithms and HPC to the teams applying



the power of the world's most [powerful supercomputers](#) to help tackle COVID-19 and, thanks to some early [research and partnerships](#) going on at LLNL for years, were well positioned to make a big impact. A few examples include:

- CASC Researchers **Peer-Timo Bremer**, **Jayaraman Thiagarajan**, and **Rushil Anirudh** are part of a team providing decision support to FEMA for determining when to recommend transition between the four phases of [re-opening](#). With a large number of variables feeding into that determination, and no good data to fit models to yet, the multi-lab team is using the EpiCast agent-based model developed at Los Alamos National Laboratory to simulate people-level interactions—including commuting patterns, different types of industries, schools, etc. These simulations are used by the LLNL team to train surrogate (e.g., deep learning) models for different types of locales such urban, suburban, and rural. Optimization tools developed at Argonne National Laboratory then use these models to help answer questions such as, “Assuming this locale wants to move to the next phase and we have N active COVID-19 cases, how soon can we make that transition without overwhelming the healthcare system?”
- CASC researchers **Sam Jacobs**, **David Hysom**, **Brian Van Essen**, and the [LBANN](#) research team are working to support Computational Biology research and improve the quality of the [ATOM](#) project’s molecular optimizer’s design loop by training the generative molecular neural network on the entire 680-million-compound Enamine data set. With the increase in the number of compounds used to train the latent space, the goal is to improve the ability of the optimizer to identify novel compounds that increase efficacy in binding with the COVID-19 proteins while minimizing side effects. Additionally, by using large-scale tournament training, the team is looking to accelerate the time to discover, train, and tune neural network architectures for the molecular optimizer based on character-LSTMs, variational autoencoders, and sequence-to-sequence models.

No matter where we find our offices, our scientists at LLNL continue to bring the best minds to bear toward [solving the current threat of COVID-19](#) as well as a myriad of national security and open science problems that will make this world a better and safer place. And with a safety-conscious and deliberate approach, LLNL is slowly bringing more of the Lab population back onsite to perform work that simply cannot be performed at home. Future versions of this newsletter will continue to highlight successes in the projects mentioned above and in other efforts around the Lab that tap into CASC expertise. For now, enjoy some topics we had queued up for this newsletter before COVID-19 changed our lives.

Collaborations | CASC Leaders in the Scientific Community

One of CASC’s roles is to connect the Laboratory to the greater external scientific computing community. Beyond attracting a talented workforce, we want to capture mindshare of the *academic research community* so that they seek out collaborations



with us, which increases the expertise and talent we can apply to our Laboratory's mission. We want to capture the mindshare of *government agencies* so that they recognize the Laboratory not only as an organization that can solve their problems but also as a partner who can help to define and shape the direction of their programs. Similarly, we want to capture the mindshare of *industry* and be seen as a trusted partner that can help them leverage the power of HPC. There are many ways by which we engage academia, industry, and government, but this month we highlight a few recent examples of technical excellence as recognized by the external scientific computing community.



The HPC community has two flagship events each year: The [International Conference for High Performance Computing, Networking, Storage, and Analysis](#) (i.e., SC or "Supercomputing") and [ISC High](#)

[Performance](#) (ISC). In 2019, CASC researchers were honored with the best research paper awards at both of these events. In June 2019, **Ignacio Laguna Peralta** and a team from Purdue University won the Hans Meuer Award at ISC for their paper, "[GPUMixer: Performance-Driven Floating-Point Tuning for GPU Scientific Applications](#)," describing a tool for performance-based, mixed-precision floating-point tuning on scientific GPU applications. CASC researchers were honored for their work at SC in November 2019, where their paper, "[A Massively Parallel Infrastructure for Adaptive Multiscale Simulations: Modeling RAS Initiation Pathway for Cancer](#)" was awarded [SC Best Paper](#).



Harsh Bhatia, Tom Scogland, and Timo Bremer were important contributors to this collaboration between Livermore Computing, CASC, LLNL's Physical and Life Sciences Directorate, IBM, LANL, San Diego State University, and Frederick National Laboratory for Cancer Research.



In the machine learning and artificial intelligence (ML/AI) Community, one of the flagship conferences is the [AAAI Conference on Artificial Intelligence](#) (AAAI AI), which was held in New York City in February

2020. This year, it received over 7,700 research paper submissions and accepted just 20.6%. One of these accepted papers, "[Building Calibrated Deep Models via Uncertainty Matching with Auxiliary Interval Predictors](#)," was co-authored by **Jay Thiagarajan** (first author) and **Timo Bremer** in collaboration with researchers at Arizona State University and IBM Research AI. To publish in one of the premier ML/AI conferences is quite an achievement, particularly since the DOE national laboratories are not yet widely recognized in the ML/AI community as significant drivers in ML/AI research. Only one other paper at AAAI AI was co-authored by authors from DOE laboratories (LLNL and ANL), which points to the strength and quality of CASC's machine learning experts.



Finally, in November 2018, DOE's Advanced Research Projects Agency-Energy (ARPA-E) launched the [Grid Optimization Competition](#) (GO Competition), a series of challenges to develop software management solutions for challenging power grid problems. The goal of the competition is to accelerate the development of transformational and disruptive methods ultimately to create a more reliable, resilient, and secure American electricity grid. In Challenge 1, which ended in mid-February, security-constrained (AC) optimal power flow (SCOPF) problems were solved, and algorithms were tested on complex, realistic power system models. CASC researcher **Cosmin Petra** led the LLNL team that won the overall competition, placing first in all four divisions and earning [special kudos](#) for their dominant performance. The LLNL team produced more than half of the best scenario scores: 816 first places out of a possible 1408 (58%), far exceeding the team with the second most firsts, which had only 282 (20%). Twenty-six teams competed, including teams from four other DOE national laboratories.

These are just a few of the many external recognitions CASC's researchers receive each year. We celebrate these successes not only because they help to attract more talent and sponsors but also because they indicate that we have a world-class research team contributing to the success of LLNL's programs. Congratulations to Ignacio, Harsh, Tom, Jay, Timo, Cosmin, and all of their collaborators!

Contact: [Jeff Hittinger](#)

Lab Impact | Caliper, SPOT, and Hatchet: Enabling the Whys and Wherefores of Performance Analysis

Performance analysis of large, complex applications is an activity often reserved for the experts. Between the complicated tools, the know-how required to even begin using them, the algorithm of "relying on the expert for what to look at next," and the diversity of cases where performance analysis is necessary, somebody (or a lot of somebodies) has job security. As a result, performance analysis is often deferred until a time when the problems are severe and obvious.

Yet, CASC developers have been hard at work to bring ubiquitous performance analysis to the masses in close collaboration with Livermore Computing (LC) and application teams. To start, the [PAVE](#) team in CASC including **David Boehme**, **Todd Gamblin**, and CASC alum **Martin Schulz** developed [Caliper](#) [1], a toolbox relying on a simple common API written into your application to identify the sections of code of interest. Instrument your application once and study all the aspects of the application performance on all of the platforms with multiple tools—now that's a bargain!

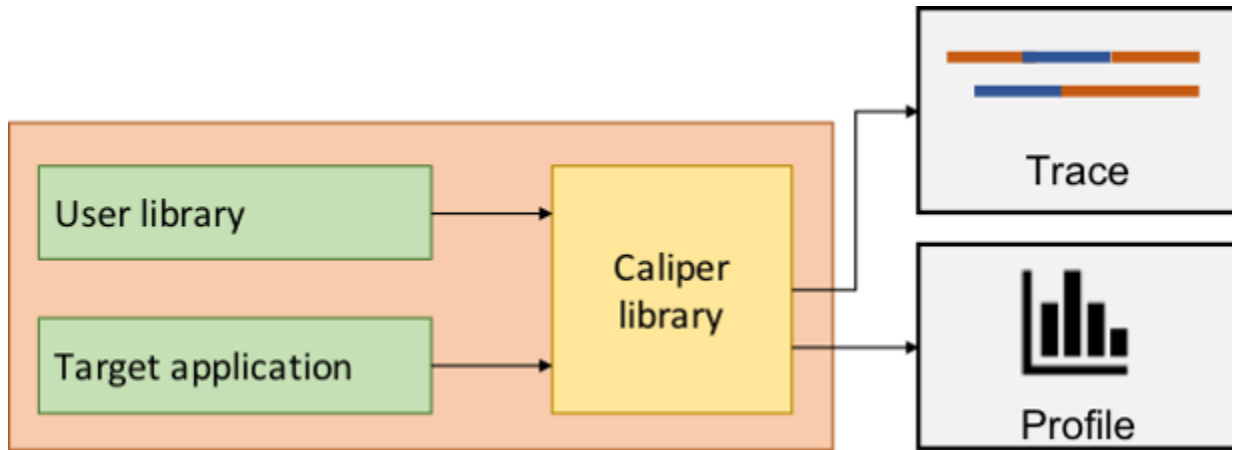


Figure 1: Caliper collects data in instrumented applications and libraries and produces raw data for traces and profiles.

Little by little, applications at LLNL are being instrumented with Caliper—in particular many of our largest multi-physics applications and libraries running on systems like [Sierra](#). Nearly 5 years of effort with a broad set of collaborators has gone into designing and developing the prototype, gathering initial use cases, iterating with early adopters based on feedback, and convincing others to get on board. The result is that we have enabled a standard way to instrument applications, use performance analysis tools, and importantly—standardized performance data output.

The second phase of performance analysis for the masses, Spot, leverages this standardized performance output, and provides the most commonly requested performance analysis in a browser. Dreamed up in the [PAVE](#) project and productized by our ProTools team in Livermore Computing, Spot requires modest additional setup in the application, and has quickly become the tool of choice to view the oodles of performance data produced by Caliper-instrumented nightly tests.

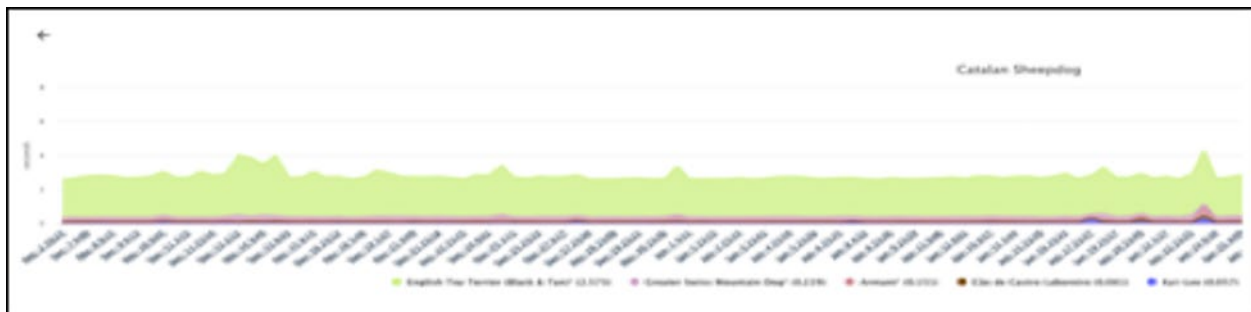


Figure 2: Spot takes data collected nightly by Caliper and displays time spent in different parts of the application over time. Spot v1 “dog” test data shown.

Now even the non-performance-experts are able to view performance data, and because performance analysis is a ubiquitous part of the developer workflow, it is easy to quickly spot (!) performance changes in standard regression test suites that are run as part of every commit or nightly test.

The Spot project is now tackling the ever-growing list of requests for features, and the best part is that these features rarely require additional changes to the setup code in the applications thanks to the standard API that abstracts details of the underlying tools and hardware.

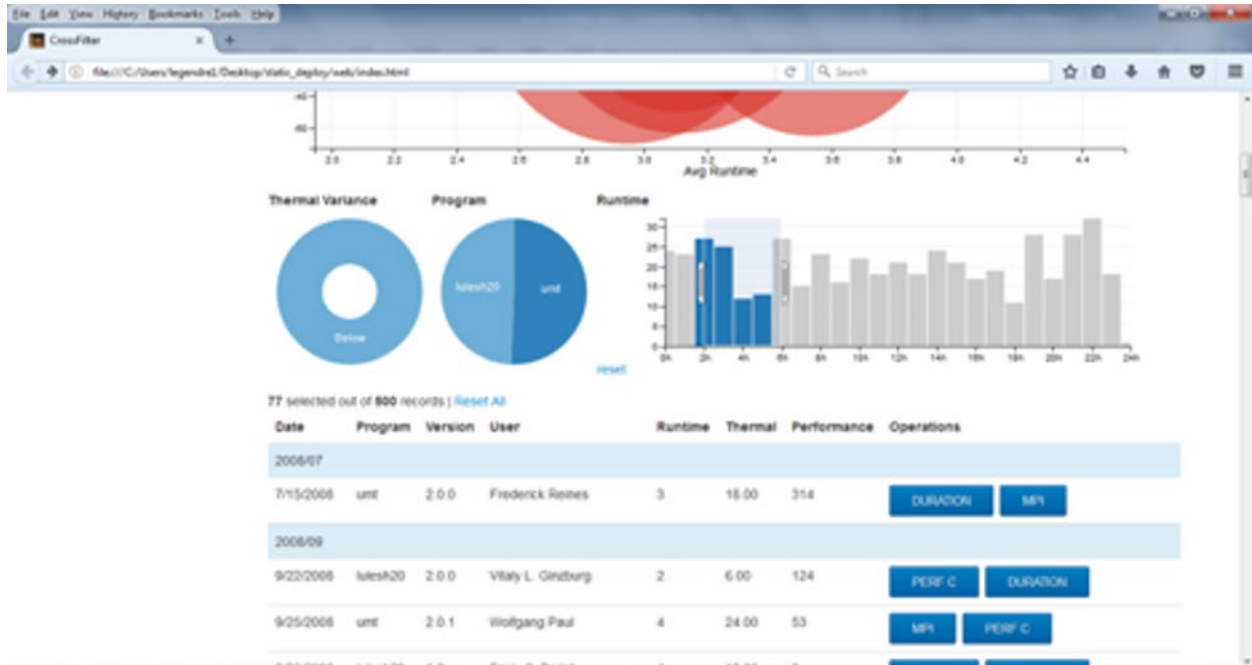


Figure 3: SPOT v2 showing a table of application runs.

But when presented with data, users are naturally curious to dive into the next level. They either want to get answers to their questions faster than capabilities can be added to Spot, or their questions are quite specific. For these cases, the standardized performance output from Caliper is proving extremely enabling.

Coupling the hierarchical call tree data from Caliper and the timing information, [Hatchet](#) [2] was born. Wielding this sharp Jupyter-based weapon, the curious user can now pose the questions she has and see the answers immediately. For example: What speedup am I getting from using the GPUs? Which portions of my code scale poorly? What difference does it make to use one MPI implementation over another? It's all a matter of 'simple' math on call trees. Hatchet is yet another example of how powerful the intersection of application and tool developers working in collaboration can be.



Figure 4: HATCHET’s ‘division’ operator enables calculation of speedup in each function in the application via a single line in a Python script, quickly pinpointing bottlenecks. HATCHET trees shown demonstrate Lulesh execution on (a) 1 core, (b) 27 cores, and (c) the speedup of 1/27 cores.

The growing ecosystem of interoperable tools built on a common API (Caliper) is a testament to the value of standard abstractions and output formats. Thanks to Caliper, Spot, and HATCHET—our code teams have more information than ever about the performance of their codes, and can treat performance analysis as an integrated part of their daily developer workflow instead of a once-in-awhile specialist activity. And because these tools are available as open source (or soon will be), we hope to encourage a broader community to adopt and contribute.

[1] David Boehme, Todd Gamblin, David Beckingsale, Peer-Timo Bremer, Alfredo Gimenez, Matthew LeGendre, Olga Pearce, and Martin Schulz. 2016. Caliper: Performance Introspection for HPC Software Stacks. In *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC ‘16)*. Article 47, 11 pages.

[2] Abhinav Bhatele, Stephanie Brink, Todd Gamblin, 2019. HATCHET: pruning the overgrowth in parallel profiles. *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC ‘19)*. Article 20, 11 pages.

Contact: [Olga Pearce](#)

Advancing the Discipline | High-Order Matrix-Free Algorithms for the Next Generation of Supercomputers

Exascale supercomputers are just around the corner: The DOE’s first computers capable of performing more than 10^{18} floating point operations per second are anticipated to arrive in 2021. Lawrence Livermore’s [El Capitan](#), expected in 2022, will be the NNSA’s first supercomputer whose performance will exceed one exaflop. These



machines, as well as [Summit](#) and [Sierra](#), which are currently the world's #1 and #2 fastest supercomputers, rely heavily on heterogeneous computing architectures, powered largely by graphics processing units (GPUs). Taking advantage of the over one quintillion (or a billion billion) calculations that these exascale computers will perform every second requires algorithms designed from the ground up for GPU-based architectures.

As CASC's [Sidney Fernbach Postdoctoral Fellow](#), I had the opportunity to focus on the development of computational physics methods that are specifically designed for these architectures. One of the main challenges posed by these numerical methods is the solution to the large systems of equations that arise from the simulation of large-scale physics problems. These systems require large amounts of memory, and while GPUs excel at performing huge numbers of calculations quickly, they typically have relatively little memory. Furthermore, the amount of time spent reading and writing the data associated with these systems can dwarf the amount of time actually performing computational operations.

To address this challenge, together with the [MFEM team](#), I have been pursuing a high-order and matrix-free approach. This approach minimizes the amount of data required to describe the system of equations, at the expense of re-computing certain quantities more than once. Luckily, these types of computations are very well suited for GPUs, and the savings on data transfer more than pays off. Using these new algorithms and reducing the memory requirements also enables the use of higher fidelity methods that were previously considered too computationally expensive. These methods provide higher accuracy and more detailed physics than traditional low-order methods. The latest version of [MFEM](#) features high-performance GPU-enabled matrix-free high-order capabilities, vastly outperforming the traditional matrix-based algorithms.

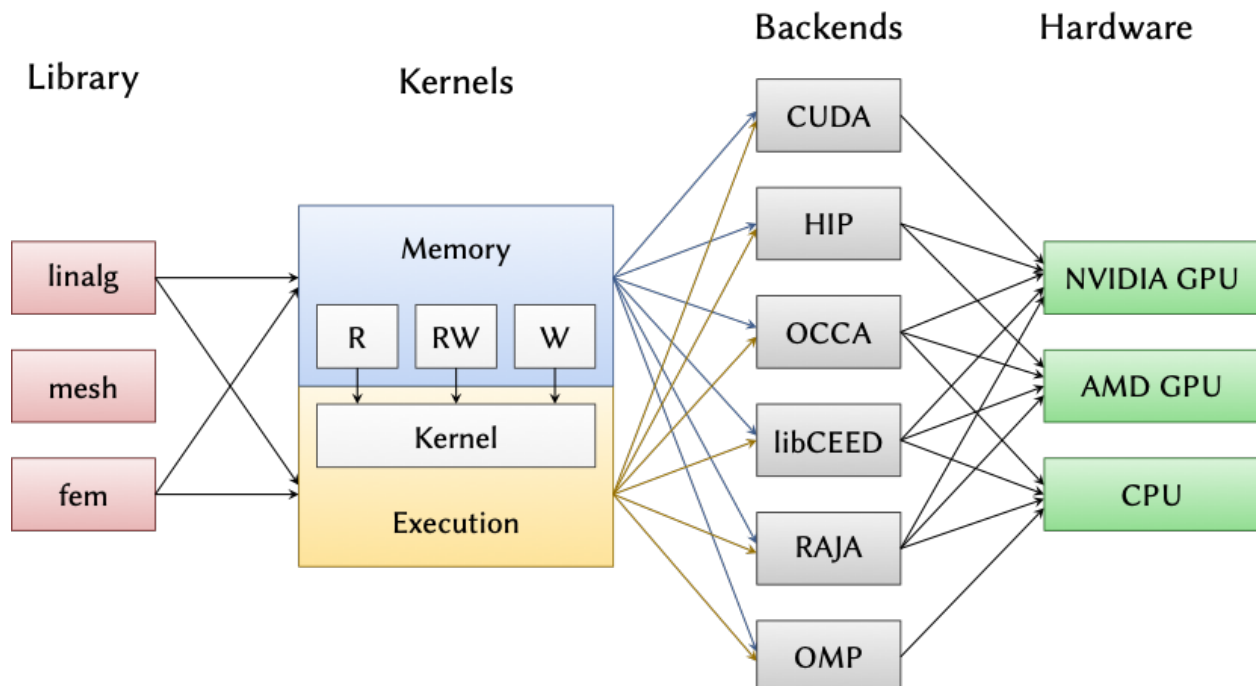




Figure 5: Schematic of MFEM's GPU capabilities.

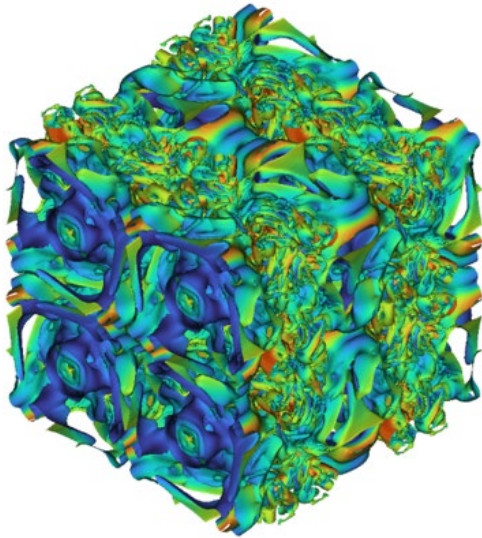


Figure 6: High-order matrix-free incompressible fluid flow simulation.

However, transitioning to a matrix-free approach also introduces some new challenges. Without an explicitly formed matrix available to describe the system of equations, traditional solvers and preconditioners such as algebraic multigrid are no longer straightforward to apply. This necessitates research and development on solvers designed for the matrix-free context. [One approach](#) that I pursued relates the large system of equations to an auxiliary system of equations based on a reduced-accuracy discretization that is much sparser. Although this auxiliary discretization is significantly less accurate, it is amenable to solver techniques such as multigrid. It turns out that this less accurate method

can be used to efficiently accelerate the convergence of the original high-accuracy system. We have applied this technique to the [large-scale fluid dynamics simulations](#), demonstrating both the efficiency of the solver and accuracy of the underlying discretization.

Much future work is still required in order for our computational physics simulations to achieve the highest possible performance on the next generation of supercomputers. For this reason, intensive research into high-order matrix-free solvers continues, both as part of the Optimal High-Order Solvers LDRD (PI: Andrew Barker, CASC) and within the [CEED exascale co-design center](#).

Contact: [Will Pazner](#)

Path to Exascale | Umpire: Managing Heterogeneous Memory Spaces for Performance Portability

Current and future high-performance computing (HPC) systems feature very fast processors, with both the CPU and any attached accelerator like a graphics processing unit (GPU) being able to complete a huge number of operations per second. The challenge then becomes keeping these processors “fed” with data. Memory technology continues to improve in performance, and high-bandwidth memory is a key feature of exascale platforms. However, this faster memory often comes in a limited capacity, and typically requires access through a vendor-specific programming interface.



[Umpire](#) is an open-source application-focused library for fast, flexible, and portable memory management on HPC architectures. It provides concepts for allocating, modifying, and inspecting data in all the memory resources available on the system. At LLNL, Umpire is used on a wide variety of systems with a number of different hardware configurations, and it supports many key simulation codes, from stockpile stewardship to seismic simulation.

A substantial fraction of the LLNL Advanced Simulation and Computing portfolio relies on Umpire for fast and portable memory management.

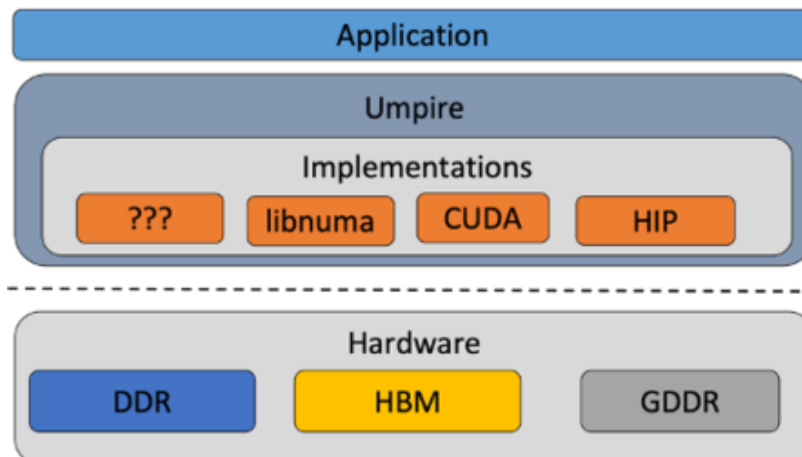


Figure 7: Umpire “sits” between applications and hardware to determine how to allocate complex memory resources.

Umpire’s *allocation* process provides a common interface, regardless of where the memory is being allocated. The algorithms used to efficiently allocate data can be modified using *strategies*. Strategies provide different memory-pooling methods to speed up allocations and can be used to provide hints to help the system put memory in the appropriate place in hardware. *Operations* provide an abstract interface for modifying and moving data, allowing application developers to write portable code to move data between different memory resources in the system.

Current work is focused on preparing Umpire for the upcoming DOE exascale systems: [Frontier](#) at Oak Ridge National Laboratory, [Aurora](#) at Argonne National Laboratory, and [El Capitan](#) at LLNL. Each system requires back-end support for the particular vendor hardware that the system provides. Once an appropriate back-end is developed, applications using Umpire will be able to run on any of these systems without needing to use platform-specific solutions.

Umpire development is funded by the [Exascale Computing Project](#), [RADIUSS](#), and the LLNL Advanced Technology Development and Mitigation program.

Contact: [David Beckingsale](#)



CASC Highlights

A lot has happened since our last newsletter.

New Postdocs (Since July 2019)

- Nathan Hanford (9/19)
- Ketan Mittal (10/19)
- Segei Shudler (1/20)
- Irene Kim (1/20)
- Konstantinos Parasyris (2/20)
- James Diffenderfer (5/20)
- Yu-Ting (“Tim”) Hsu

New Staff Hires (Since July 2019)

- Timothy La Fond (7/19)
- Wesam “Sam” Sakla (8/19)
- Genia Vogman (8/19)
- Sookyung Kim (9/19)
- Andrew Gillette (10/19)
- Indrasis Chakraborty (12/19)
- Jean-Paul Watson (1/20)
- Bo “Ivy” Peng (2/20)
- Will Pazner (2/20)
- Giorgis Georgakoudis (2/20)
- Abhik Sarkar (3/20)
- Youngsoo Choi (5/20)
- Tristan Vanderbruggen (5/20)

CASC Newsletter Sign-up

Was this newsletter link passed along to you? Or did you happen to find it on social media? [Sign up](#) to be notified of future newsletters, which we release approximately every 3 months.

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-WEB-811231