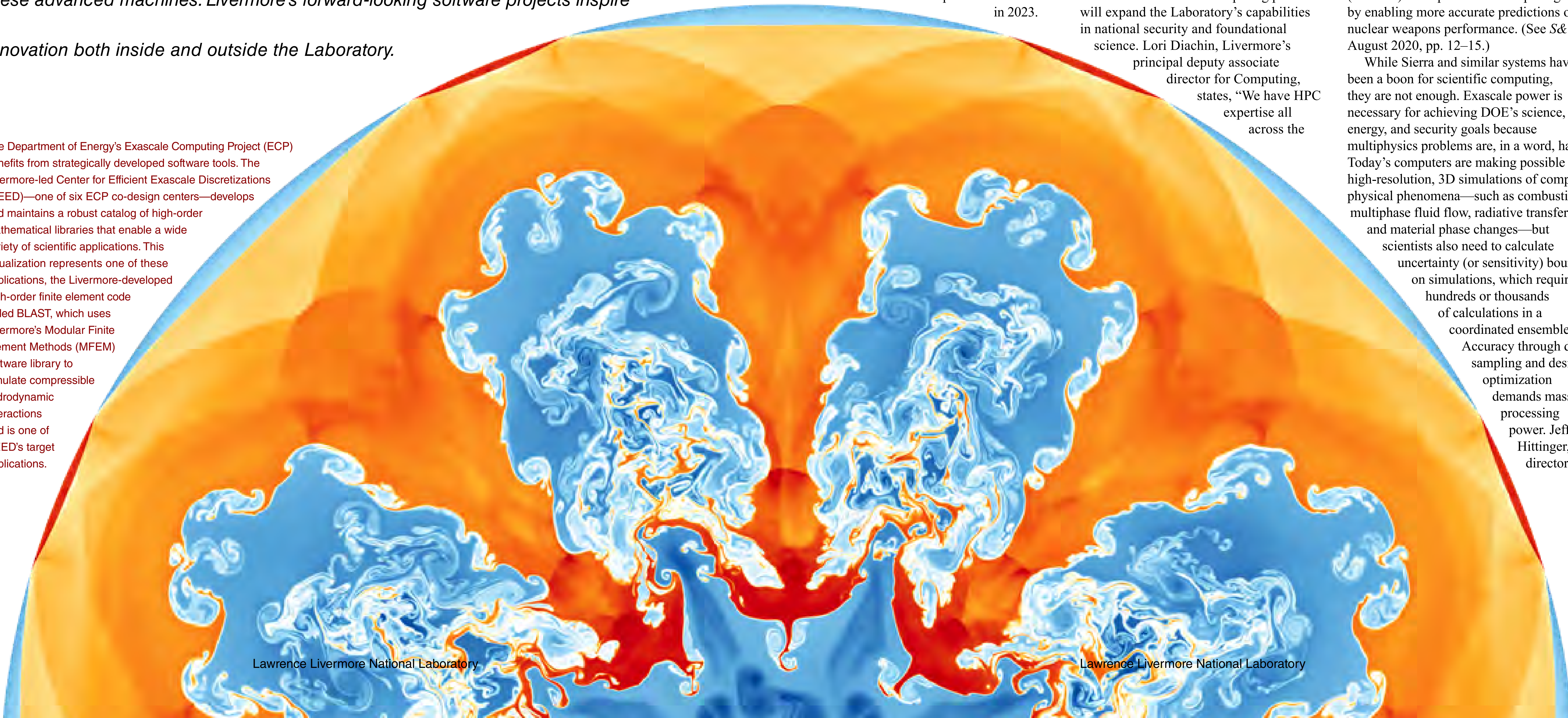
THE EXASCALE SOFTWARE PORTFOLIO

The exascale era will introduce the world to supercomputers, like the Laboratory's forthcoming El Capitan, that perform calculations faster than ever before. As high-performance computing hardware becomes more complex, so too does software design and development in support of these advanced machines. Livermore's forward-looking software projects inspire innovation both inside and outside the Laboratory.



The Department of Energy's Exascale Computing Project (ECP) benefits from strategically developed software tools. The Livermore-led Center for Efficient Exascale Discretizations (CEED)—one of six ECP co-design centers—develops and maintains a robust catalog of high-order mathematical libraries that enable a wide variety of scientific applications. This visualization represents one of these applications, the Livermore-developed high-order finite element code called BLAST, which uses Livermore's Modular Finite Element Methods (MFEM) software library to simulate compressible hydrodynamic interactions and is one of CEED's target applications.

EXASCALE supercomputers will process information a thousand times faster than the systems that introduced the possibilities of predictive simulation a decade ago. Lawrence Livermore will be among the world's first high-performance computing (HPC) centers to deploy an exascale-class system, capable of 10^{18} floating-point operations per second (flops), when El Capitan comes online in 2023.

The Laboratory wields a large portion of the Department of Energy's (DOE's) HPC resources. These computing investments—hardware technology, software infrastructure, and scientific applications—have aided discoveries in nuclear and high-energy-density physics, materials science, climate change, energy efficiency, biological processes, and many other fields. Increased computing power will expand the Laboratory's capabilities in national security and foundational science. Lori Diachin, Livermore's principal deputy associate director for Computing, states, "We have HPC expertise all across the Laboratory. Contemplating the kinds of science we'll be able to do on exascale machines is exciting."

The heart of this effort is the predictive capability that comes from modeling, simulation, and visualization. For example, Sierra, one of the world's most powerful supercomputers, supports the National Nuclear Security Administration's (NNSA's) Stockpile Stewardship Program by enabling more accurate predictions of nuclear weapons performance. (See *S&TR*, August 2020, pp. 12–15.)

While Sierra and similar systems have been a boon for scientific computing, they are not enough. Exascale power is necessary for achieving DOE's science, energy, and security goals because multiphysics problems are, in a word, hard. Today's computers are making possible high-resolution, 3D simulations of complex physical phenomena—such as combustion, multiphase fluid flow, radiative transfer, and material phase changes—but scientists also need to calculate uncertainty (or sensitivity) bounds on simulations, which requires hundreds or thousands of calculations in a coordinated ensemble. Accuracy through data sampling and design optimization demands massive processing power. Jeffrey Hittinger, director of

Livermore's Lassen (above) and Sierra supercomputers exploit graphics processing units (GPUs) for increased computing power. With a peak performance topping 20 petaflops, the Lassen system is a smaller, unclassified version of Sierra. (Photo by Garry McLeod.)

Livermore's Center for Applied Scientific Computing, explains, "Efficient exploration of the solution space is a huge hill to climb."

As the exascale era begins, two major initiatives leverage and expand Livermore's HPC capabilities. The spotlight in this feature is software. The Exascale Computing Project (ECP) brings together many national laboratories to address many of the challenges inherent in their scientific and national security missions. At Livermore, the RADIUSS project—Rapid Application Development via an Institutional Universal Software Stack—aims to benefit scientific applications through a robust software infrastructure.

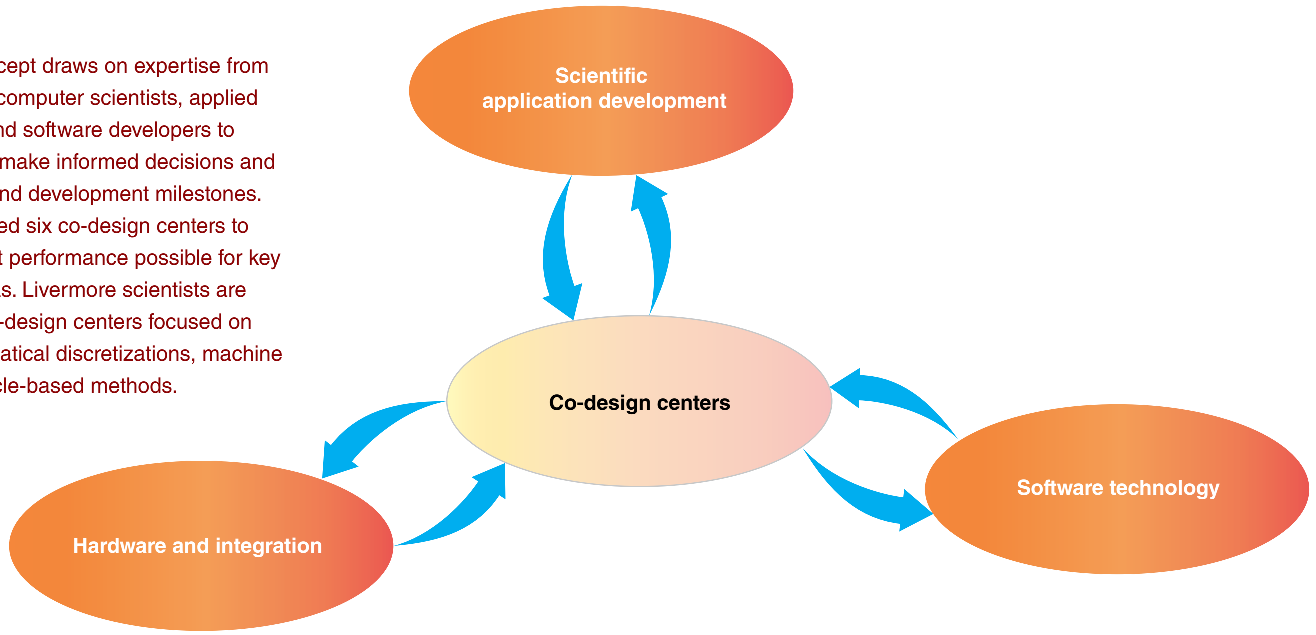
The Exascale Threshold

The exascale threshold—one thousand times faster than petascale—is incredibly difficult to reach. Although supercomputers are becoming more powerful, hardware manufacturers are nevertheless approaching limitations of processor speed and chip size. For decades, machines were built with a large but limited number of processors and memory modules, and each new version offered more capability for the same price, energy cost, and footprint. Now, physical constraints have resulted in increasing costs and energy consumption for small gains in computing performance. This situation has inspired new designs with the potential to improve the performance of scientific applications—but not without innovation in software.

The introduction of graphics processing units (GPUs) into HPC systems has opened the door to new computational possibilities. For certain types of calculations and applications, GPUs consume less energy and take up less space than central processing units (CPUs). Parallel processing capability thus increases, and a computer's workload can be balanced accordingly. Machine learning algorithms work well on GPUs, running faster at lower floating-point precision. Sierra's generation of computers is known as heterogeneous, or hybrid, because their architectures combine GPUs and CPUs. The DOE's first three exascale systems—El Capitan, Argonne National Laboratory's Aurora, and Oak Ridge National Laboratory's Frontier—will also take advantage of GPUs.

Diachin, who also serves as the ECP's deputy director, explains, "CPUs and GPUs are now physically closer together, but latencies exist due to the location of data in memory." During a calculation, data has to move near GPUs to take advantage of them, which means algorithms must be designed and executed with this in mind. Furthermore, HPC hardware varies by vendor, requiring customized software as well as interoperability solutions when switching platforms.

The HPC community quickly realized that monolithic code bases were no longer sustainable in this context, and software programming needed a new paradigm. Diachin states, "Transitioning to advanced architectures means investing in reusable software solutions that bridge the complexity between applications and diverse hardware. The problem involves a lot of exploration and hard work." Hittinger adds, "Heterogeneous computing architectures are complicated, especially at extreme scales. Multiphysics codes require significant computational resources and an HPC ecosystem that includes software libraries and tools designed to work on these machines. We are simplifying computer programming and improving software quality so scientists can focus on science."



The co-design concept draws on expertise from domain scientists, computer scientists, applied mathematicians, and software developers to help organizations make informed decisions and achieve research and development milestones. The ECP established six co-design centers to achieve the highest performance possible for key computational areas. Livermore scientists are involved in ECP co-design centers focused on high-order mathematical discretizations, machine learning, and particle-based methods.

A Sum Greater than the Parts

The ECP launched in 2016 as the U.S. government was already looking toward procuring exascale-capable computers. The 7-year, $1.8 billion effort is funded by the DOE's Office of Science and NNSA and includes most national laboratories and approximately 1,000 researchers. According to Diachin, Livermore was a natural fit for the project. "Our laboratory has a strong reputation for fielding world-class HPC systems," she says. "We bring decades of computing experience to bear on the most challenging scientific problems." (See *S&TR*, September 2016, pp. 4–11.) Like Diachin, several Livermore researchers hold ECP leadership roles.

All ECP research and development activities revolve around the delivery of a sustainable exascale computing ecosystem that supports mission-critical applications. By enabling higher fidelity solutions to scientific problems, the ECP aims to advance scientific discovery, strengthen national security, and improve industry competitiveness. Diachin states, "Collectively, we are creating tools and capabilities that the individual players would not otherwise be able to create."

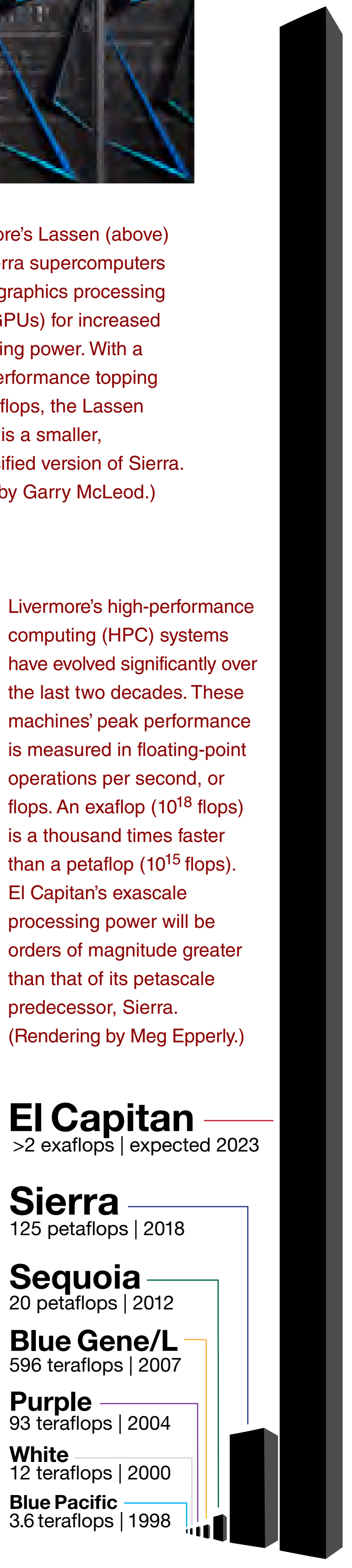
The project is organized into three focus areas: application development, software technologies, and hardware and integration. Scientific application development is the ECP's top priority, with two dozen teams working to demonstrate simulation capabilities at a large scale. For example, a Livermore-led team is refining multiple codes that simulate physics processes relevant to stockpile stewardship. Software technology teams concentrate on the underlying software infrastructure that helps applications run accurately, quickly, and reliably. Meanwhile, hardware and integration teams work with industry vendors and HPC facilities on power-efficient and affordable HPC designs, testbed support, and ECP software deployment logistics.

The software development effort includes evaluating the tools and features that will reduce exascale development costs. For instance, a flexible exascale ecosystem will need to integrate applications composed of independently developed parts, each with its own programming language and parallel programming model. Similarly, data management and visualization tools are essential to collecting, analyzing, moving, and storing simulation data. "We will accomplish exascale computing by working together on common solutions," notes Livermore computer scientist Rob Neely, deputy program director for Weapon Simulation and Computing, who oversaw ECP software technology developed within NNSA's Advanced Simulation and Computing (ASC) portfolio and at NNSA laboratories.

Coordinated Development

One of the ECP's most critical tasks is effective collaboration to avoid redundancy and ensure interoperability of the ecosystem's components. This coordinated development coalesces under the concept of *co-design*, which draws on expertise from domain scientists, computer scientists, applied mathematicians, and software developers. "Co-design centers centralize the effort on commonly recurring algorithms by working with multiple application teams to provide highly optimized solutions," states Tzanio Kolev, who leads the ECP's co-design Center for Efficient Exascale Discretizations (CEED).

Livermore researchers lead or participate in three of the ECP's six

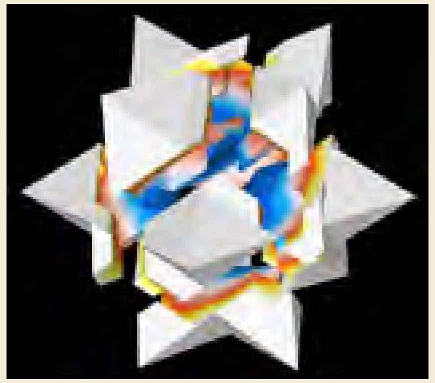


Livermore's high-performance computing (HPC) systems have evolved significantly over the last two decades. These machines' peak performance is measured in floating-point operations per second, or flops. An exaflop (10^{18} flops) is a thousand times faster than a petaflop (10^{15} flops). El Capitan's exascale processing power will be orders of magnitude greater than that of its petascale predecessor, Sierra. (Rendering by Meg Epperly.)

co-design centers. These interdisciplinary centers are organized around key scientific computing subjects, and their work helps achieve the ECP's research and development milestones. For example, CEED's purview is improving computational accuracy and efficiency of simulations via finely calibrated mathematical discretization libraries. (See the box below.) The ExaLearn co-design center is developing a scalable machine learning and artificial intelligence software framework for use in scientific applications and at experimental facilities. The Co-design Center for Particle Applications specializes in particle-based simulations of molecular dynamics and other particle interactions.

Mathematical Foundations

Large-scale, complex scientific applications cannot run on new computing architectures without a foundation of rigorous and efficient mathematical solutions, such as high-order discretization methods. Computational mathematician Tzanio Kolev explains, "Scientific applications in any field of study must incorporate robust mathematical calculations that are accurate and predictive." For example, finite element numerical methods naturally describe scientific phenomena relevant to Laboratory missions, such as compressible fluid flow, heat transfer, design optimization, and additive manufacturing. These methods provide efficient solvers for partial differential equations, which define many real-world processes in a rigorous mathematical form. The process of discretization transforms continuous mathematical functions into discrete components, making those functions understandable to a computer. Cumulatively, these sophisticated techniques exploit a supercomputer's data parallelism and memory access, improving performance by orders of magnitude over traditional low-order methods.



Versatile, high-order math libraries give scientific applications a boost in performance and accuracy when run on HPC systems. The MFEM software library and GLVis visualization tool produced this image of a heat diffusion simulation on a 3D unstructured tetrahedral mesh and its parallel decomposition.

In the exascale era, researchers must modify such algorithms so scientific codes can make the most of graphics processing units (GPUs). Accordingly, math libraries play a significant role in the Exascale Computing Project's (ECP's) software portfolio. High-order solution algorithms are well suited for GPU-based architectures but difficult to execute. Kolev states, "Controlling the arithmetic intensity and ensuring the accuracy of these methods is a mathematically big challenge. We are working on different ways to accomplish this, such as by developing novel matrix-free algorithms and solvers." One of the ECP's co-design centers, the Center for Efficient Exascale Discretizations (CEED), is dedicated to making high-order methods as practical and efficient as possible so scientists do not need to reinvent or optimize these parts of their code.

Led by Kolev, CEED combines experts from Lawrence Livermore and Argonne national laboratories and five universities. The Center's goals include developing a comprehensive software suite of libraries, solvers, application programming interfaces, and programming models; improving the tools that transition low-order applications to using high-order methods; and defining community standards such as format specifications for high-order data and operators. Since CEED's inception, the team has published nearly 50 scientific papers and given more than a dozen presentations on these technologies.

Among the Center's projects are "mini-apps" that capture key physics properties and are used to benchmark scientific applications' performance. In true co-design spirit, ECP teams and vendors use mini-apps such as Laghos (Lagrangian High-Order Solver) to model compressible gas dynamics and fluid flow. Laghos solves ordinary differential equation systems through novel use of mass and force matrices, resulting in less data storage and fewer memory transfers.

Another important tool in CEED's software suite is the Modular Finite Element Methods (MFEM) library, which provides building blocks for developing finite element algorithms. Researchers use MFEM to run simulations on a wide variety of machines—from personal laptops to the largest GPU-powered supercomputers. MFEM's development under CEED directly benefits next-generation codes for the National Nuclear Security Administration's Advanced Simulation and Computing program.

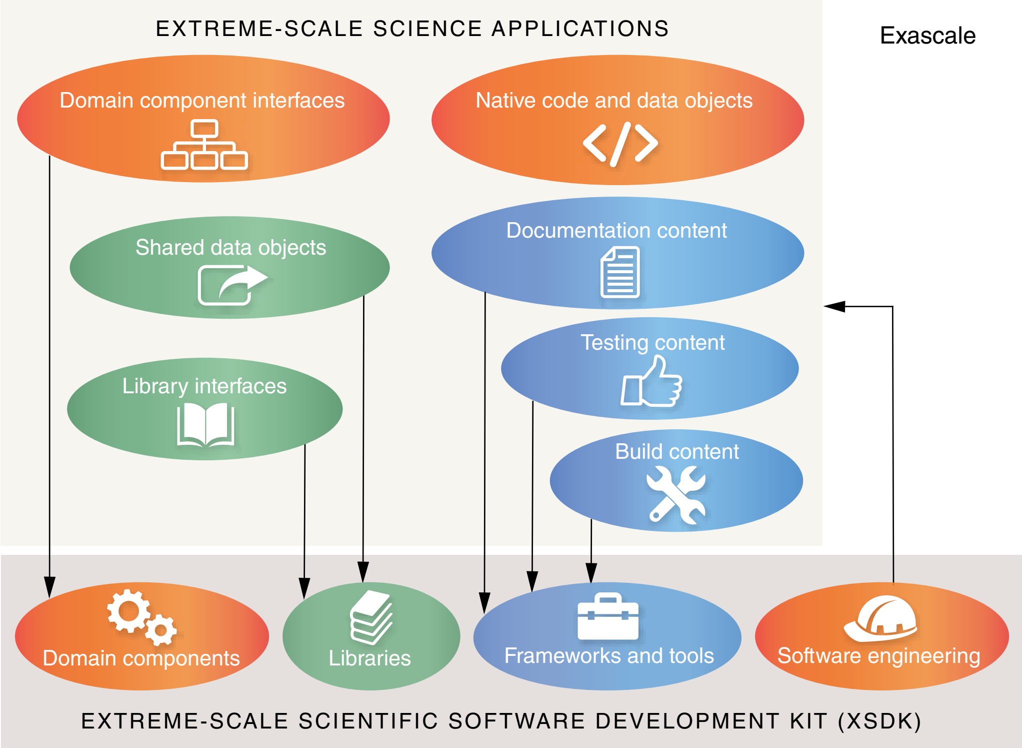
With its version 4.0 release in 2019, MFEM leapt from a solution optimized for central processing units (CPUs) to one that also supports GPUs. The upgrades were tested on Lassen, one of Livermore's GPU-based supercomputers. A more recent incremental release offers optimized support for the specific type of GPUs that will be used in the DOE's first exascale systems. Kolev describes this effort as the most difficult task the CEED team has accomplished to date. "Imagine you replaced a car's internal combustion engine with an electric engine. From the driver's perspective, the car still runs, but everything is different under the hood," he says. GPUs are not as "smart" as CPUs, so MFEM's algorithms were revised to express tasks simply and independently. The library was also refactored to accommodate memory movement between GPUs and CPUs and to support many different hardware platforms while also remaining compatible with machines that use only CPUs. The result is a highly flexible math library that will be adaptable to exascale demands.

Software Sustainability

In a software development context, *sustainability* means ensuring that software remains relevant and useful, works correctly, and is regularly updated to meet users' evolving goals. "In the ECP, software sustainability includes support for more applications and larger scale machines. Software tools must be interoperable in this context," explains Ulrike Meier Yang, Mathematical Algorithms and Computing group leader at Livermore and head of the ECP's Extreme-Scale Scientific Software Development Kit (xSDK) effort.

Focusing on numerical libraries, the xSDK team works toward the seamless integration of software packages needed by ECP applications. According to Yang, math libraries are usually independently developed with their own software strategies on different platforms and built with different compilers, which can lead to a variety of issues when they are used in combination. She notes, "Our team achieves consistency across libraries through activities such as normalizing build processes and avoiding namespace conflicts. We vet new libraries for inclusion in the Kit and resolve any incompatibilities." The xSDK provides both the turnkey aggregation of a range of mathematical software as well as a set of community policies and documentation that encourage standardization.

So far, the xSDK contains 23 math libraries including several developed by Livermore scientists, such as the HYPRE library of high-performance preconditioners and solvers, the SUNDIALS collection of nonlinear and differential equation solvers, and the Modular Finite Element Methods (MFEM) library. Most scientific applications use various subsets of the full catalog, which are tested on different operating systems at ECP partner sites. "Interoperability means we serve a range of user scenarios," states Yang. "The xSDK software suite and its accompanying policies are valuable because they are versatile, reliable, and cost-effective."



The ECP's Extreme-Scale Scientific Software Development Kit (xSDK) provides a standardized aggregation of math libraries for use by scientific applications on exascale-capable machines. The xSDK team's methodical approach to software sustainability helps the ECP deliver dependable software to scientific applications.

Institutional Improvements

Livermore's RADIUSS project benefits from the ECP's productivity and insights. Hittinger notes, "The ECP has invested a lot in software sustainability and maintenance. We don't want to lose that momentum at the Laboratory." Neely, who in addition to his ECP role serves as the RADIUSS project lead, adds, "RADIUSS is inspired by the ECP's efforts to make software highly dependable for users."

An additional motivator is NNSA's ASC program, which for years has funded applications that simulate and predict the performance and safety of nuclear weapons—as well as the software to execute these codes on supercomputers. According to Neely, this investment naturally dovetails with RADIUSS objectives. He says, "The ASC program has always recognized the need for software sustainability to ensure the most reliable simulations. RADIUSS leverages the tools developed via ASC, allowing us to build complicated software out of simpler, modular components."

The project aims to strengthen versatile HPC software and broaden its usage at Livermore and across the scientific application community. Neely explains, "Scientific computing is at an intersection of evolving architectures, increasingly complex simulations, and the need to change software approaches accordingly. Ultimately, we are advocating for adoption of Livermore's scalable, stable open-source software in the broader community. RADIUSS encompasses a production-quality set of tools for every scientific application developer to use, including users outside of the programs that fund development of these software products."

RADIUSS builds on expertise from computer scientists and software developers all over the Laboratory to encourage common development standards and provides another venue for developers to encourage adoption of their products. The team tackles software sustainability within the entire HPC ecosystem—computing infrastructure, code installation, application integration, data management and visualization, testing guidelines, documentation, and more. The project's software tools individually provide solutions for specific use cases and collectively offer a flexible "shopping list" for HPC users. (Many of these tools are also part of the ECP's software portfolio.) Neely emphasizes, "We are adopting, creating, and promoting best practices and a standard way of developing and releasing open-source software at the Laboratory."

For instance, performance and workflow optimization are key areas that RADIUSS promotes. Running a scientific application on a supercomputer is not as simple as clicking a button. Codes contain scripts that perform calculations, pull in or generate data, or execute other tasks. How these actions are completed depends on their interaction with interconnected computing nodes, and coordination of individual jobs that run during large-scale simulations is tricky. ASC software tools like Flux allow users to schedule and manage HPC resources, while the Caliper library lets users customize performance measurements for their applications.

The RADIUSS portfolio also includes portability and memory management tools that tackle hardware-related challenges. Application codes written for CPU-based systems need to work on newer GPU-based systems. Moreover, heterogeneous architectures vary, so codes also need to adapt to any available supercomputer. This challenge was anticipated in Livermore's weapons program a decade ago, motivating the development of tools like RAJA Portability Suite, which provides abstractions of calculation loops to target machine-specific programming models and constructs, as well as memory allocation and movement decisions. (See *S&TR*, August 2020, pp. 12–15.) "Most people want to take advantage of the GPU revolution," says Neely. "RADIUSS is reducing overheads for application teams, providing a pathway to next-generation architectures, and building a knowledge repository of local expertise." (See the box on p. 10.)

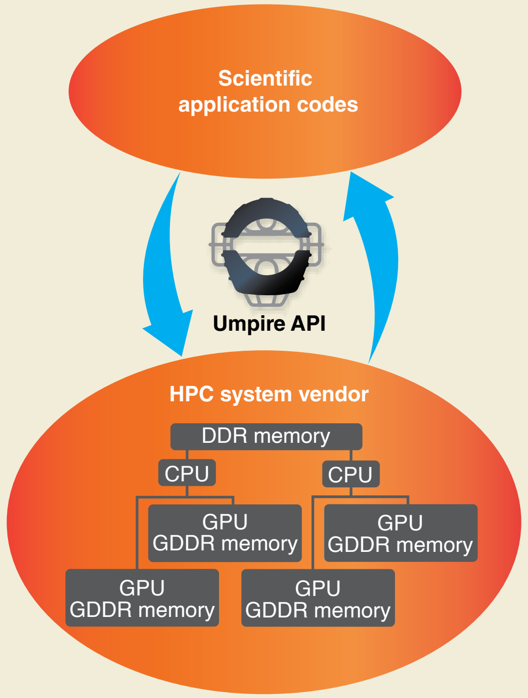
Intelligent Memory Allocation

The scenario is familiar to high-performance computing centers that run large scientific codes on heterogeneous machines—limited memory resources require strategic memory management, but multiphysics codes and the hardware they run on can vary widely. Researchers must consider where data will be stored as the simulation is processed, and how best to move data to and from available compute nodes for optimal performance. Central processing units (CPUs) store more data, but graphics processing units (GPUs) are faster.

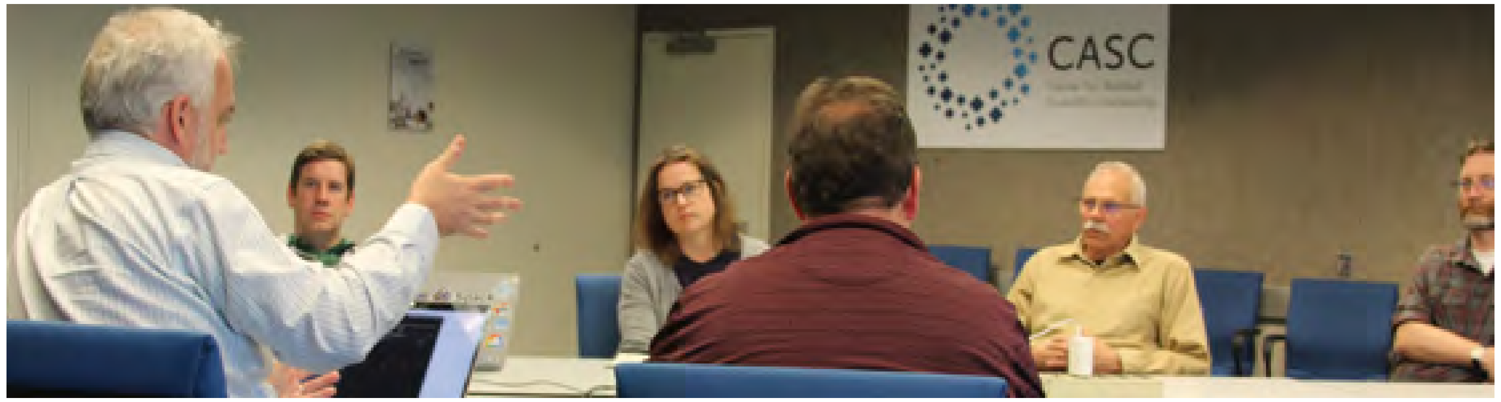
Lawrence Livermore's RADIUSS project—Rapid Application Development via an Institutional Universal Software Stack—addresses such execution challenges with a suite of advanced software tools that ultimately improve a code's performance. For instance, Umpire is a memory management solution developed by researchers David Beckingsale, Marty McFadden, Kristi Belcher, and Rich Hornung. Like an umpire makes decisions on a baseball diamond, Umpire determines how to allocate data among a supercomputer's complex memory resources and accommodates a range of device specifics and programming models. Principal investigator Beckingsale notes, "Instead of forcing users to commit to one technology-specific implementation, Umpire creates a memory resource for everything it detects on each system. Users do not have to know anything about the hardware or decide how to manage memory while their codes run."

Umpire works through an application programming interface (API) that abstracts and unifies memory allocation. The API allows for multiple complementary tasks such as querying compute nodes for availability, adjusting memory-pooling methods to speed up allocations, transferring simulation data between GPUs or between GPUs and CPUs, and tracking which data is stored where. "For large-scale physics applications, not enough memory exists on GPUs alone. Data must move around dynamically," explains Beckingsale.

Many of the Laboratory's production codes—from stockpile stewardship to seismic monitoring—rely on Umpire. It leverages other RADIUSS tools and can work alone or in tandem with Livermore's RAJA portability software, which helps move codes from one type of computing architecture to another. According to Beckingsale, the work to adapt codes to the current generation of GPU-based machines will pay off when the next generation arrives. He states, "We will take this same approach to memory management on exascale computers."



Like an umpire makes decisions on a baseball diamond, Livermore's Umpire software determines how to allocate a supercomputer's complex memory resources such as double data rate (DDR) and graphics double data rate (GDDR) integrated circuits. Multiphysics codes and the hardware they run on can vary widely, so Umpire accommodates a range of device specifics and programming models, ultimately improving a code's performance.



Rob Neely (far left) leads Livermore's RADIUSS team—Rapid Application Development via an Institutional Universal Software Stack—in tackling software sustainability within the entire HPC ecosystem. (Photo by Meg Epperly.)

Open for Collaboration

Though much of Livermore's scientific application portfolio is necessarily classified, a culture of openness in unclassified software development has taken root at the Laboratory. Open-source software (OSS)—the practice of releasing licensed code and inviting outside feedback and contributions—is valuable to many projects and essential when external collaborators are involved, as in the ECP. (See *S&TR*, January/February 2018, pp. 4–11.) For example, the xSDK team consists of developers at five laboratories and six universities, so their software efforts must be accessible to all participants. Hittinger emphasizes the importance of open-source development in projects large and small, stating, "More community input makes for better software. External contributors help us identify bugs, evolve features, and attain broader usage."

Although RADIUSS primarily benefits the Laboratory, Neely points out that input from the open-source community is constructive. He says, "All of the products under the RADIUSS banner are open source. We hope to share our work and experiences with, and learn from, other national laboratories and HPC centers." RADIUSS team member David Beckingsale adds, "Our software benefits greatly from engagement with the HPC community, vendors, and university collaborators. Public-facing development helps give users confidence in our projects and see that they are actively maintained."

OSS can quickly build momentum among users and developers, as recent successes with the Spack package manager and the Scalable Checkpoint/Restart framework have shown. Both of these Livermore-led OSS projects won 2019 R&D 100 Awards for innovation. (See *S&TR*, July 2020, pp. 8–11.) Both are also part of the ECP's software portfolio and promoted by RADIUSS. In fact, all software tools and libraries mentioned by name in this article are open source.

Beyond Exascale

As the exascale era dawns, Livermore researchers and software developers take a holistic view of the supercomputing landscape, where versatility and scalability are crucial to high performance—regardless of machine. Diachin states, "Interest in deploying the ECP's software is global, so our software must be compatible with many different computing architectures and be performance portable." Hittinger adds, "The best software stack insulates scientific codes against future architecture changes."

Exascale-capable systems like El Capitan will come online in the next few years. The first exaflop calculation will be run, and global supercomputer rankings will shift accordingly. However, Hittinger points out, "An exaflop is a milestone, not the finish line." A more meaningful moment will come when scientists can solve a problem with exascale computing capability that they could not have solved previously. He continues, "Computing constantly evolves. We won't simply stop at the next breakthrough."

—Holly Auten

Key Words: Advanced Simulation and Computing (ASC) program, application programming interface (API), Center for Efficient Exascale Discretizations (CEED), central processing unit (CPU), co-design, Department of Energy (DOE), El Capitan, exascale, Exascale Computing Project (ECP), Extreme-Scale Scientific Software Development Kit (xSDK), floating-point operations per second (flops), graphics processing unit (GPU), hardware, high-performance computing (HPC), Modular Finite Element Methods (MFEM), National Nuclear Security Administration (NNSA), open-source software (OSS), Rapid Application Development via an Institutional Universal Software Stack (RADIUSS), software sustainability, RAJA Portability Suite, Umpire.

For more information contact Lori Diachin (925) 422-7130 (diachin2@llnl.gov) or Rob Neely (925) 423-4243 (neely4@llnl.gov).