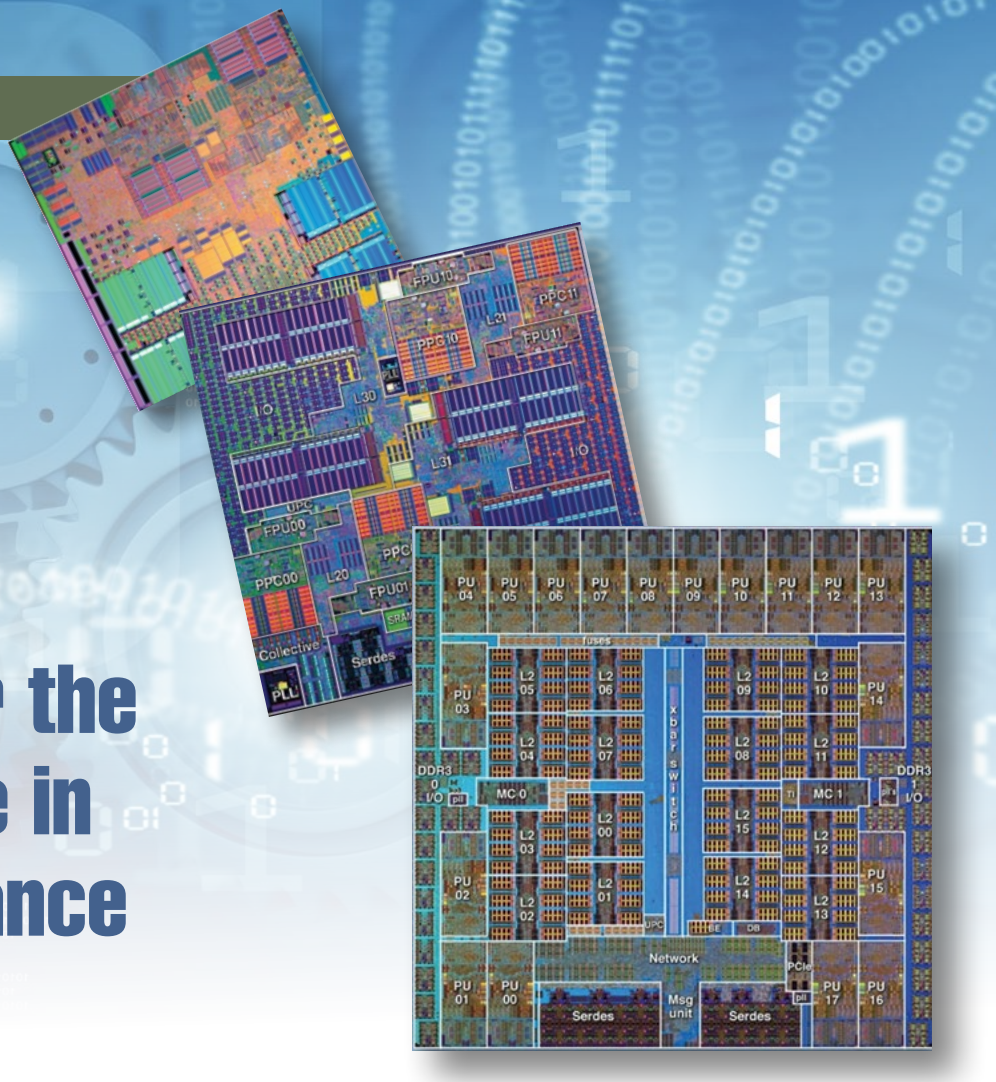


Gearing Up for the Next Challenge in High-Performance Computing

EVERYTHING changes. Nowhere is that maxim more apparent than in the world of computing. From smartphones and tablets to mainframes and supercomputers, the system architecture—how a machine's nodes and network are designed—evolves rapidly as new versions replace old. As home computer users know, systems can change dramatically between generations, especially in a field where five years is a long time. Computational scientists at Lawrence Livermore and other Department of Energy (DOE) national laboratories must continually prepare for the next increase in computational power so that the transition to a new machine does not arrest efforts to meet important national missions.

That next jump in power will be a big one, as new machines begin to approach exascale computing. Exascale systems will process 10^{18} floating-point operations per second (flops), making them 1,000 times faster than the petascale systems that arrived in the late 2000s. Computational scientists will need to address a number of high-performance computing (HPC) challenges to ensure that these systems can meet the rising performance demands and operate within strict power constraints.

Up the Supercomputer Highway

This is not the first sea change presented by advances in supercomputing. Since the first computers arrived in the 1950s, four eras have made an entrance, each with its advantages and challenges. In the mainframe era, large sequential processing machines executed computer code instructions one at a time, in serial fashion. Memory capacity (the amount of data that could be stored) was often an issue for mainframe computers, limiting the size of applications and requiring developers to find a balance between memory usage and application.

The vector era of the 1970s and 1980s offered a large performance boost. With vector processors, computers could gather sets of data elements scattered around the system's memory and align them into vector registers, where codes could efficiently operate on the data and send the results back into memory. This architecture mapped favorably to scientific programs, where arrays of data with different values to be computed by the same set of instructions could now be processed concurrently. Ultimately, researchers found they could vectorize only about 30 percent of the operations performed by Livermore's most complex national security multiphysics codes. Therefore, to improve overall runtimes, Laboratory scientists and computer architects at the partnering vendors worked together to improve the scalar performance of serial (one-at-a-time) operations that could not be vectorized. They also continued to work on vectorizing codes to improve performance even further.

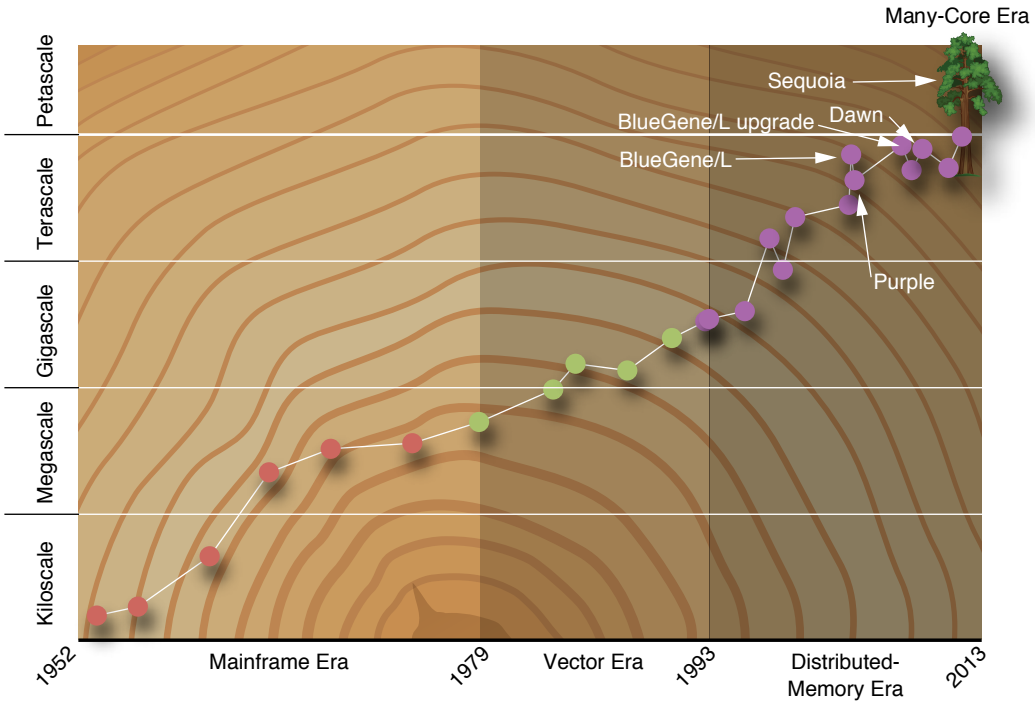
Vector processing gave way to the distributed-memory era in the 1990s, when commodity serial processors connected by fast networks proved to be a cost-effective architecture. Algorithms were again redesigned for parallel programming, using message-passing routines for efficient communication between nodes. The boost in performance came from parallelization across nodes and from increases in scalar performance on the processors.

To further improve performance and overcome a growing gap between compute and memory speeds, developers added a small amount of fast memory (called a cache) inside each processor. Cache keeps data close to the central processing unit (CPU) and available for reuse, eliminating extra operations to store and fetch data from main memory. Unfortunately, the memory capacity per core and the memory bandwidth between cores and local memory have not kept pace with increases in peak floating-point performance, creating ever more serious choke points for applications.

Attempts to address this issue require innovation in both hardware and software, leading to the fourth HPC era: many-core computing. This architecture is typified by either a very large number of CPU cores on a node, or an accelerator—often graphics processing units (GPUs) such as those originally developed for three-dimensional (3D) rendering in video games. The node design can also include complex memory hierarchies. For example, one section of main memory can be fast and small, and the other is large but slow. Livermore's Sequoia supercomputer is a harbinger of such advanced architectures, built with a large number of low-powered cores, yet retaining a "flat" memory hierarchy within a node. An identifying trait of the many-core era is a requirement to shift to threaded processes, again requiring radical algorithm redesigns for the codes and continued innovations in languages and compilers.

Data Movement and Parallelism

Livermore computational physicist Bert Still explains how the next-generation HPC systems will affect the current situation. "In the past, applications were developed on systems where the main work of computing—floating-point operations—took place on the CPU," says Still, the deputy project leader for the Advanced Architecture Software Development project funded by the National Nuclear Security Administration's Advanced Simulation and Computing (ASC) Program. "We and our industrial partners focused on streamlining this work in both applications and computing architectures." As a result, data packets and streams were often directed around the computer system—in and out of memory and various subsystems—with little regard for the electricity required to move that memory around the machine. Now that more data must be stored, handled, and manipulated, the electrical cost of moving data could prove prohibitive. Thus, the first challenge is to reduce data motion, either by designing algorithms and applications that perform as many calculations as possible on a piece of data before returning it to main memory, or by minimizing the communication required with neighboring nodes.



Each era of high-performance computing brings its own challenges along with increased performance capabilities. The first three—the mainframe, vector, and distributed-memory eras—have passed. In the current many-core era, node designs deploy many central processing units (CPU) cores or a graphics processing unit (GPU) accelerator with various memory configurations. Livermore's Sequoia stands on the threshold of this era.

Livermore's Sequoia represents another leap in the evolution of supercomputer architectures—a many-core configuration combined with a node design featuring a "flat" memory hierarchy. Even with its large number of cores, Sequoia is significantly more energy efficient than a conventional computer system thanks to the low power required by these cores. Ranked as the world's fastest supercomputer for a time, this machine can run suites of uncertainty calculations used to increase confidence in the predictions of computer models. (Photograph by Bob Hirschfeld.)

Still notes that although Sequoia is significantly more energy efficient than a conventional computer system, it consumes 9.6 megawatts at peak speed. "If 1 megawatt costs $1 million per year, you can see how the costs push us toward energy-efficient advanced architectures," says Still. "If the architecture and codes stayed the same and we just pushed to a bigger system, the power requirements would be prohibitive. The annual electric bill for running that system could be several hundred million dollars—far more than the cost of the capital equipment."

The second challenge involves the increased parallelism in the system as computer architects design machines for yet more performance. In the past, performance gains were accomplished by pushing the clock speed (the rate at which each microprocessor executes instructions) and adding power-hungry complexity (more transistors) to CPUs to automatically exploit low-level parallelism. "The 'good old days' of increasing clock rates ended nearly a decade ago," says Livermore scientist Rob Neely, who leads the Advanced Architecture Software Development project. "We now redeploy those extra transistors in multicore CPUs to boost overall performance."

Possible Answers

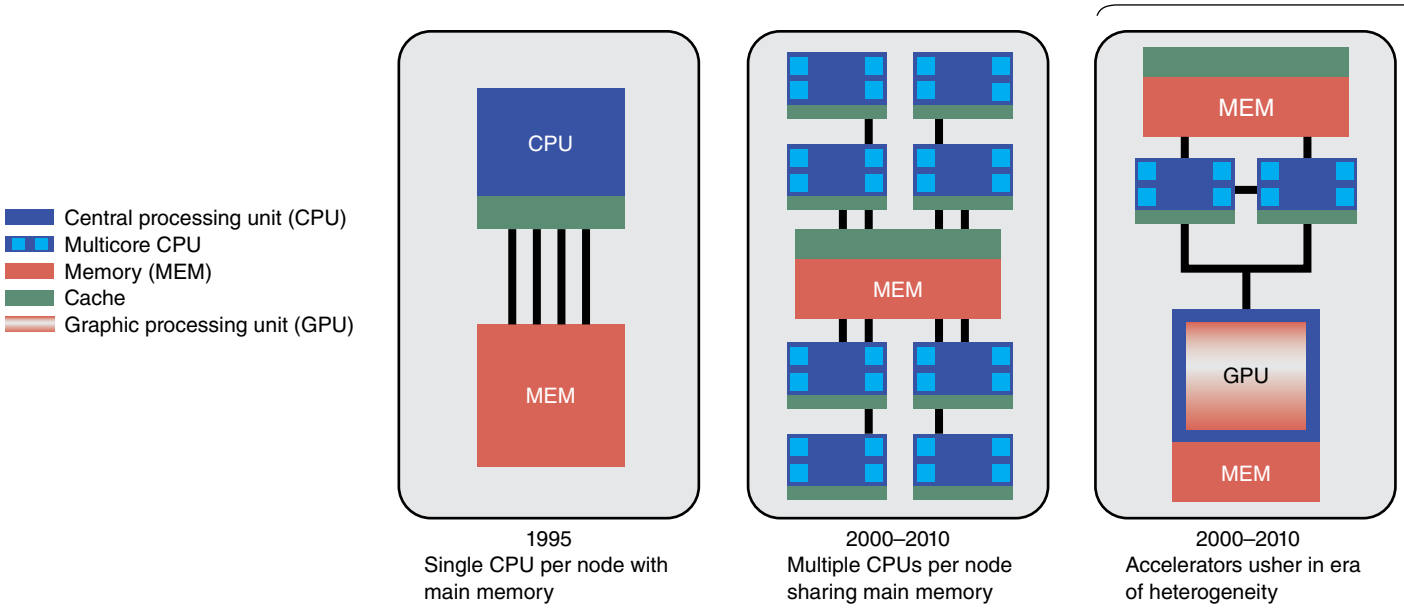
According to Still, a radical shift in architectures is required to minimize data motion and further reduce computational time. One approach is to design cores and memory within each node in a way that increases parallelism and concurrency. "We already see this trend in successive generations of the IBM BlueGene architectures over the last decade," says Still. "In 2005, the BlueGene/L machine had 196,608 cores in 98,304 nodes. By 2012, Sequoia had 1.6 million cores in the same number of nodes."

The BlueGene architecture relied on a homogeneous node consisting of multiple, identical cores. A competing architecture uses a heterogeneous node that combines GPUs with commercially available high-performance CPUs. GPUs have hundreds of cores that handle thousands of software threads simultaneously. They can take gigabytes of data and repeat the same operations very quickly by using thousands of streaming processors. Calculations that cannot effectively use GPUs are processed by CPUs instead.

Future heterogeneous designs will lessen the burden on the programmer by allowing the distinct memory between CPU and GPU to appear as a single unified memory. Explicitly managing data movement between CPU and GPU will no longer be required. However, to gain the best performance, developers will need to optimize the application by providing ample "hints" to the compilers indicating where data should be placed.

Another advanced architecture is one that is similar to the BlueGene supercomputers but works with both fast and slow memory in a configuration called nonuniform memory access (NUMA). The small, fast memory with high bandwidth is located on a many-core package. The large, slow memory is farther away and accessed by a slower link. In the NUMA configuration, each core has an instruction stream and fetches its own data but may share a cache with others cores on the chips.

The processing-in-memory architecture, which adds a simple arithmetic unit in or near main memory, is yet another design being considered. This approach would eliminate some traditional data motion, such as transferring data arrays to CPUs for calculations and returning the results back to memory for storage. Instead, a CPU could simply issue an instruction to the memory subsystem



to return the sum of that array. "In this design," says Neely, "a subset of the operations is offloaded to the memory processor, further reducing data motion and memory bandwidth requirements between the main CPU and memory."

All of these architectures include new memory technologies, and the field is evolving rapidly still. ASC leaders are evaluating candidate architectures with the goal of acquiring the best performance gain possible with the fewest modifications to the million-plus lines of code in the multiphysics packages. "We need computer programs that can express the actions we want and a system to perform in languages such as C++, Python, and FORTRAN," says Neely. "To get the necessary performance gains, we must focus on the whole picture: hardware, software, and applications."

As Still points out, complex science questions are looming, and they involve calculations that current machines cannot handle. Whether it's simulating the interactions of intense laser beams with plasmas, the atomic-level behavior of metals under extreme stress and strain, or the effects of local weather variability on global climate systems, the more accurately simulations can mimic and predict natural processes, the better. Improving predictive capability involves more data, more processing power, and more complex calculations. Given the current flux in computer architecture design, scientists face the challenge of rethinking or even rewriting codes to ensure confidence in the modeled predictions.

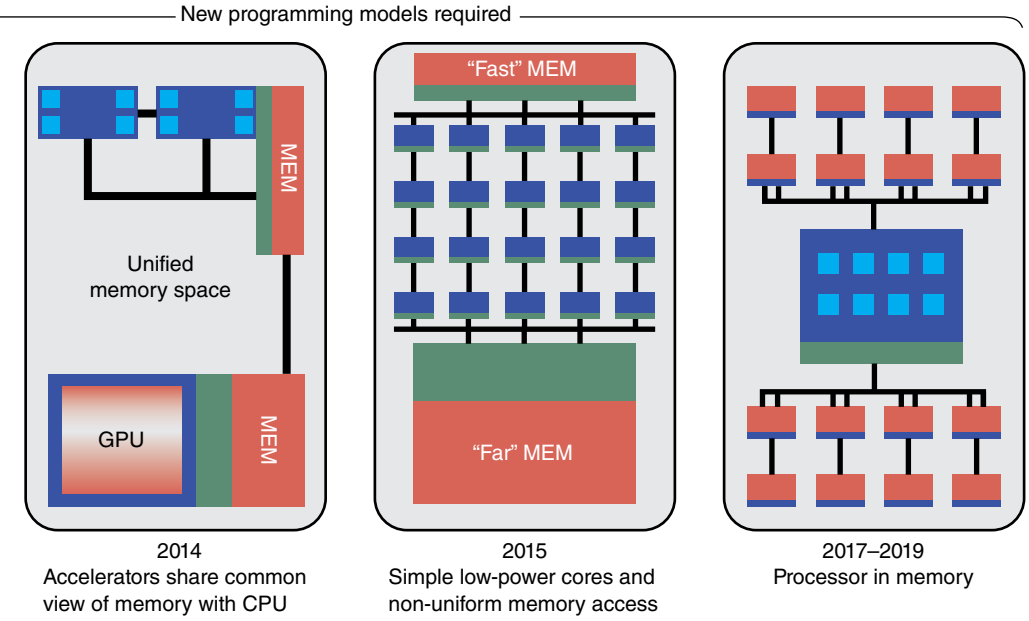
Working Together for Success

Even as the experts peer into the future, the Collaboration of Oak Ridge, Argonne, and Lawrence Livermore national laboratories (CORAL) is focusing on the next big near-term system. In January 2014, CORAL announced a joint request for proposals for next-generation supercomputers that offer peak performance of at least 100 quadrillion flops (petaflops), about 5 times the capability of Sequoia but only 10 percent of the exascale mark. Under CORAL, scientists at the three laboratories are working with vendors to develop computer systems that will be deployed in 2017 and 2018. Livermore's system will be used for national security calculations to support nuclear stockpile stewardship under the ASC Program. Oak Ridge and Argonne will use their supercomputers to perform missions for DOE's Office of Science, under the Advanced Scientific Computing Research Program.

Bronis de Supinski, chief technology officer for Livermore Computing, explains, "Our collaborative goal was to choose two systems that, as a set, offer the best overall value to DOE. We want diversity of technologies and vendors as well as systems that will provide value to the Office of Science laboratories."

On November 14, 2014, Secretary of Energy Ernest Moniz announced that IBM, working closely with OpenPOWER Foundation partners NVIDIA and Mellanox, was chosen to design and develop systems for Lawrence Livermore and Oak Ridge. The design uses IBM Power architecture processors connected by NVLink to NVIDIA Volta GPUs. NVLink is an interconnect bus that provides higher performance than the traditional peripheral component interconnect for attaching hardware devices in a computer, allowing coherent direct access to GPU and memory. The machine will be connected with a Mellanox InfiniBand network using a fat-tree topology—a versatile network design that can be tailored to work efficiently with available bandwidth.

IBM will initiate delivery of the Livermore machine, called Sierra, in 2017. Sierra will provide more than 100 petaflops of capability. "We estimate that the peak power required to run this machine will be about 10 megawatts—just slightly higher than



HPC nodes have evolved over the decades, becoming ever more complex and densely packed. In 1995, machine architecture featured a simple node with a single CPU and a small cache for storing copies of frequently used data from the main memory. Current machines have multicore CPU–cache units that share a common main memory.

Sequoia," says de Supinski. A small, early-access system scheduled for delivery in 2016 will have an earlier generation of the IBM Power processor architecture, NVIDIA Pascal GPUs, and a version of NVLink. "It will be a complete precursor system," de Supinski adds, "so we can explore the capabilities and begin to deploy some early software and applications on the machine."

Before Sierra arrives, scientists in the Computation Directorate will work with the vendors to ensure that "no code is left behind when Sierra goes live," says Michel McCoy, the ASC program director at Livermore. "Having the hardware on the floor is only part of the challenge. We also need system software that boosts the machine's usability so that applications and key libraries will run efficiently and effectively—not only on Sierra's massively parallel, accelerator-based nodes but also on alternative architectures and future systems, as well."

As part of this collaboration, code developers will analyze and modify algorithms, investigate new data structures and layouts, and map workflows onto the new system. Once the early-access system is live, vendors will provide customized training to the Laboratory's applications scientists, working onsite to share their expertise. "This kind of collaboration allows us to tune our mission-critical application codes and quickly resolve issues as they arise," says Neely.

McCoy notes that efforts to get the weapons codes ready for Sierra will also benefit the codes that run on Livermore's unclassified systems. "It's not just stockpile stewardship that depends on HPC capabilities," he says. "We have a wide array of projects that rely on our supercomputing resources, from biomedical research to climate modeling and energy production."

The Laboratory's Multiprogrammatic and Institutional Computing (M&IC) Program, led by Brian Carnes, brings tailored, cost-effective unclassified computing services to all Livermore programs and scientists. "Through M&IC, we buy a smaller version, or 'clone,' of the larger system purchased for the ASC Program," says Carnes. "This strategy ensures that all of the Laboratory's science and technology areas have up-to-date computational resources. It's also more efficient if researchers across the Laboratory can work in a homogeneous computing environment, whether their projects are classified or unclassified."

Stepping into the Future

Still and others are looking forward to the increased capability that Sierra will bring. "On Sequoia, we can run suites of large 2D or small 3D uncertainty calculations, which are used to validate the computer models," says Still. "Sierra will allow us to do moderate to large 3D uncertainty calculations. It's another step up in our capabilities to run these complex problems."

DOE's support for HPC brings together the people who build the machines, those who write the codes, and those who use the software and hardware to explore important questions in science. The speed with which computing technology changes presents exciting opportunities while introducing challenges. "The problems may seem daunting, but they can be solved," says Still. "We know exascale won't be the end, and we want the Laboratory to be ready to address those issues when they arise."

—Ann Parker

Key Words: Advanced Simulation and Computing (ASC) Program, BlueGene, Collaboration of Oak Ridge, Argonne, and Lawrence Livermore national laboratories (CORAL), central processing unit (CPU), graphics processing unit (GPU), high-performance computing (HPC), nonuniform memory access (NUMA) configuration, processing-in-memory supercomputer architecture, Sequoia, Sierra.

For further information contact Bert Still (925) 423-7875 (still1@llnl.gov).