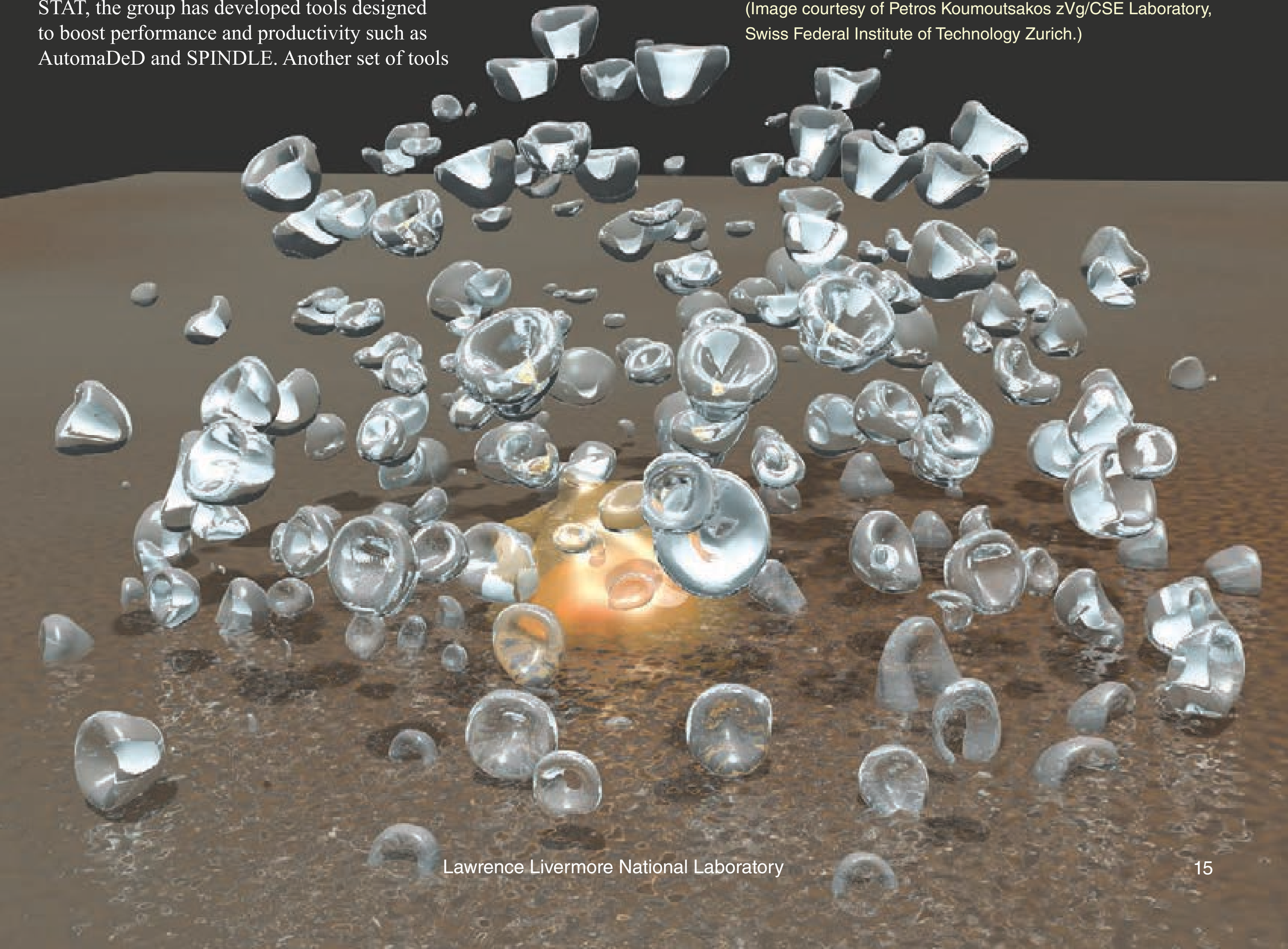
Research Highlights

Supercomputing Tools Speed Simulations

IN late 2013, an international team of scientists was in the midst of simulating a collapsing cloud of 15,000 bubbles using Lawrence Livermore's Sequoia supercomputer, when the calculations suddenly stopped. In quick response, the team turned to a Livermore software tool called STAT (Stack Trace Analysis Tool) to locate which of more than 6 million computing threads (calculations) was causing the problem. Within a few minutes, STAT traced the hangup to a particular microprocessor core (computing engine). The team went on to complete the pioneering simulation and win the 2013 Golden Bell Prize for outstanding achievement in high-performance computing (HPC). (See *S&TR*, January/February 2014, p. 20.)

STAT is the product of a small team of computer scientists comprising the Development Environment Group (DEG) in Livermore's Computation Directorate. In addition to STAT, the group has developed tools designed to boost performance and productivity such as AutomaDeD and SPINDLE. Another set of tools is being developed under the PRUNER project to help with the reproducibility of large simulations. The group works closely with the Laboratory's Center for Applied Scientific Computing, which supports the demanding computing requirements of Livermore scientists.

The group's tools are designed to work on massively parallel machines such as Sequoia, one of the world's most powerful supercomputers. Sequoia has 1,572,864 processing units, or cores,

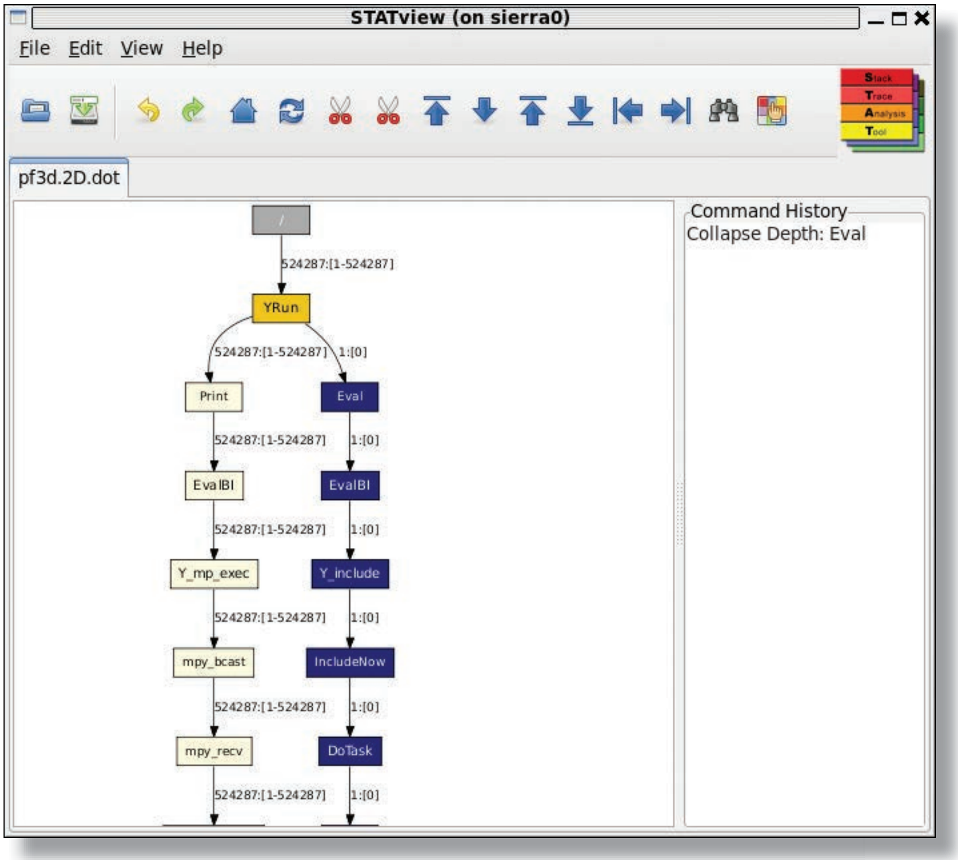


Lawrence Livermore scientists and collaborators set a new supercomputing record in fluid dynamics by resolving unique phenomena associated with a cloud of collapsing bubbles. The work earned the team the 2013 Gordon Bell Prize. See the simulation at http://www.youtube.com/watch?v=zfG9soIC6_Y. (Image courtesy of Petros Koumoutsakos zVg/CSE Laboratory, Swiss Federal Institute of Technology Zurich.)

and a peak performance speed of 20 petaflops (quadrillion floating-point operations per second). This processing power is compressed into 96 racks, each the size of a large refrigerator. (See *S&TR,* July/August 2013, pp. 4–13.)

Massively parallel supercomputers break a problem into tiny parts that are solved simultaneously. For many applications, such as simulating complex physical phenomena, this computer architecture has replaced vastly slower serial processing, in which tasks are performed sequentially by a single processing element. The Laboratory has been a leader in using parallel supercomputers since their inception. As a result, advanced simulations at Livermore have become as important to scientific exploration as theory and experiment.

According to computer scientist and DEG member Dong Ahn, developing tools for supercomputers requires expertise that resides in just a few research centers worldwide. This expertise includes skills in programming and debugging massively parallel supercomputers as well as anticipating the tools needed for next-generation machines, on which applications are expected to routinely use millions of processors. As parallel supercomputers become more powerful, Livermore computer scientists develop new methods to maximize the potential of such machines. Says Ahn, “Livermore is an applied laboratory, and our research must have practical value to users.”



This computer screen shot shows how the STAT tool analyzed 542,288 calculations. STAT formed a graphical “tree” representing a snapshot of an application that was hung up. The tree revealed the problem was in a single calculation (on the right, in blue) and resulted from a programming error.

Debugging Six Million Calculations

Many of DEG’s tools focus on finding bugs. In the world of parallel computing, debugging has become a difficult and complex task. A massively parallel application is a big search space in which errors can reside. Sequoia has nearly 1.6 million cores, each running four threads of execution (calculations). Often bugs emerge only at large scales, overwhelming users with the complexity of the task to isolate the problem. “If something breaks, we need to know quickly what went wrong in one or more of 6 million threads,” says HPC systems engineer Adam Bertsch. “We also need to know why it went wrong.”

“Many traditional debugging techniques are precluded by the sheer amount of resources that must be examined,” says computer scientist Scott Futral, DEG leader and along with Bertsch, member of the Gordon Bell Prize–winning simulation team. Because Livermore applications are continually refined, developers can spend 25 percent or more of their time debugging and optimizing codes, a practice that has become increasingly costly.

In response to the need for an advanced debugging tool aimed at machines such as Sequoia, DEG members collaborated with researchers from the University of Wisconsin and the University of New Mexico to design STAT. This highly scalable tool is capable of identifying errors in computer codes running on machines with more than 1 million processor cores. In 2011, STAT won an R&D 100 Award as one of the year’s top innovations. (See *S&TR*, October/November 2011, pp. 14–15.)

STAT is used throughout the Department of Energy’s supercomputer community. It is most effective for diagnosing calculations that are “hung up,” although the tool has also proved useful for isolating other problems. STAT indicates where in the code all of the processes are at a given point in time, giving the user insight into where the bug may lie. With a strong graphical user interface, STAT produces two-dimensional (2D) and 3D graphs in the form of treelike structures. The 2D tree represents a single snapshot of the entire application, while the 3D tree presents a series of snapshots from the application captured over time.

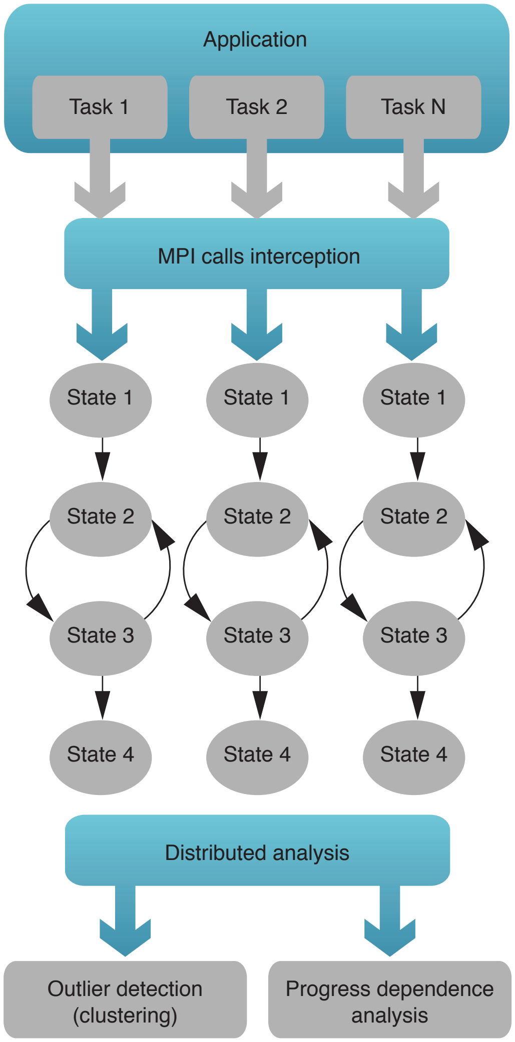
STAT played a significant role in validating Sequoia as racks of nodes were added over several months. "As we added racks, we had the opportunity to prove STAT's scalability," says Ahn. The tool helped both early users and system integrators of Sequoia to quickly isolate errors, including issues that manifested only at extremely large scales. In one case, STAT rapidly diagnosed a deadlock in a simulation using over 500,000 cores, allowing the user, who had tried unsuccessfully for weeks to solve the problem using a traditional method, to complete his project on schedule.

For the simulation project that earned the Gordon Bell Prize, STAT helped researchers achieve an ultrahigh-resolution simulation of cloud cavitation collapse. That work set a simulation record in fluid dynamics with 14.4 petaflops of sustained performance. When the calculation suddenly stopped, recalls Bertsch, STAT quickly scanned all 6 million calculations and isolated a problem in one of the processor cores. The team replaced the processor that contained the identified core, and the application was able to proceed. The resulting simulation represented a 150-fold improvement over previous simulations for this type of application and a 20-fold reduction in time to complete the task.

Although it has proven itself many times, STAT is considered a "lightweight" tool that may not always locate a bug if the problem is something other than a hung calculation. For this reason, the group has extended STAT's debugging features with the DysectAPI tool. Still in early testing, DysectAPI is designed to enable users to "program their intuition" so as to construct various higher level debug queries. The tool represents a new approach to debugging a computer program that runs on more than 100,000 processors. The method screens out unnecessary information to allow the user to rapidly zero in on the cause of a crash, fault, or other bug. Ahn says that one could use STAT to first perform a "triage" to locate the general area of the problem and then apply the DysectAPI tool to pinpoint the problem.

Increasing Computational Efficiency

DEG experts have also developed AutomaDeD, a tool that uses artificial intelligence to automate the debugging process for massive simulations. AutomaDeD has two major functions: identifying abnormal computational tasks and regions of code, and finding the least-progressed task. The first function is accomplished by detecting outliers and the second by ordering processes according to their relative progress. This work involves developing and rapidly detecting problems when system performance deviates statistically from the model. AutomaDeD creates probabilistic behavioral models of how simulations should



The AutomaDeD tool uses artificial intelligence to go beyond STAT and automate debugging when hundreds of thousands of tasks are being performed simultaneously. Message passing interface (MPI) is used by Livermore's Sequoia and other massively parallel supercomputers to pass information among the machine's hundreds of thousands of computing engines or cores. AutomaDeD identifies abnormal computational tasks and malfunctioning regions of code by detecting outliers (bottom left) as well as the least-progressed tasks (bottom right).

work. When a failure occurs, these models are analyzed to find the origin of the failure.

SPINDLE, another tool from DEG, addresses problems that can occur when millions of cores simultaneously open an application consisting of thousands of shared libraries. The tool builds a cache server to quickly send data from the libraries to the compute nodes. Ahn explains that many applications retrieve libraries of code and data that are shared by every processor, which can greatly slow down processing. SPINDLE's novel approach to loading coordinates simultaneous file system operations so that the file system does not become a bottleneck. This tool is an example of middleware infrastructure, which sits "on top" of system software. SPINDLE has proven to be highly scalable. In one test, system performance at 64 nodes without SPINDLE was similar to system performance at 1,280 nodes with SPINDLE—a 20-fold improvement.

DEG scientists, in collaboration with the University of Utah, are also studying the reproducibility of large simulations under a project called PRUNER. This work is focused on obtaining a fundamental understanding of simulation failures that occur only occasionally, or seemingly without a pattern, and then developing tools to detect and remedy them. Ahn says it may seem counterintuitive, but when a large supercomputer duplicates the same long string of calculations, it can occasionally give slightly different results, or failures can occur such as a crash. This so-called nondeterminism is often the bane of parallel software development, and it can be costly to fix. Many sources of nondeterminism exist such as the sheer scale of computing, a programmer's assumptions, and the order in which calculations are performed. Under the PRUNER project, tools are being developed to detect, control, and eliminate sources of nondeterminism. "These tools would be helpful in validating programs," says Futral.

Anticipating CORAL

The group is already anticipating the next generation of massively parallel supercomputers, scheduled to appear in 2017. A collaboration of Oak Ridge, Argonne, and Lawrence Livermore (CORAL) national laboratories will deliver these machines. Livermore's system will join Sequoia in serving the National Nuclear Security Administration's Advanced Simulation and Computing Program in support of nuclear stockpile stewardship. The next-generation system will perform up to 200 peak petaflops, about 10 times faster than Sequoia's 20 petaflops.

CORAL represents an important step toward the long-awaited exascale (extreme scale) systems. Ahn says that although supercomputer simulations are used in virtually every research area at Lawrence Livermore, many scientific challenges require computing at the exascale (10^{18} flops). These exascale systems, which are likely to debut at Livermore and other Department of Energy national laboratories early in the next decade, will deploy millions of processing elements or cores. Because of their size, simulations run on exascale machines will present challenges in diagnosing both software and hardware faults, problems to which traditional methods and tools are unsuited.

Ahn emphasizes the role played by academic partners, including the University of Wisconsin and the University of New Mexico for STAT; the Technical University of Denmark for DysectAPI; Purdue University for AutomaDeD; the Jülich Supercomputing Centre in Germany for SPINDLE; and the University of Utah for PRUNER. In the same collaborative spirit, all supercomputing tools developed at Livermore are open source, meaning anyone can use them and are invited to improve them.

With an eye on the fast-changing supercomputer future, Livermore computer scientists are preparing for new generations of giant machines. In particular, the onset of extreme computing may require equally extreme software tools, but Ahn and his colleagues are confident those tools will be in hand.

—Arnie Heller

Key Words: AutomaDeD, debugging, DysectAPI, exascale, Gordon Bell Prize, high-performance computing (HPC), PRUNER, Sequoia, SPINDLE, STAT (Stack Trace Analysis Tool).

For further information contact Dong Ahn (925) 422-1939 (ahn1@llnl.gov).