Parallel Supercomputers Move into Industry

FEATURE

by Eric J. Lerner

Massively parallel machines using microprocessors offer a new level of simulation

During the past few years, changes in the world of supercomputing have spread from the university and government laboratory to industrial research. The exotic supercomputer of old, based on a custom-designed, ultrafast, single processor, is being largely replaced with parallel processors—supercomputers in which many ordinary microprocessors are chained together to work on a single problem. Although these new computers are capable of enormous speeds and offer realistic simulations, they also bring with them significant problems in software design.

The shift to parallel processing stems from the investment needed to produce faster microprocessors. In the early days of supercomputer pioneers like Seymour Cray, a small firm could develop custom-built individual processors that greatly outpaced smaller computers produced for a more general market. But the popularization of personal computers in the 1980s generated enormous revenues for the big computer chipmakers, such as Intel and IBM, which poured money into making ever-faster chips. Eventually, the speed of individual microprocessors converged on that of the fastest single-processor supercomputers. At that point, the only way to achieve higher supercomputer speeds lay in linking many microprocessors together into parallel-processing units.

Parallel-processor basics

Today's commercial parallel computers link tens, hundreds, or even thousands of individual microprocessors, each identical to those routinely used in personal computers and workstations. The available models, including IBM's SP series, Origin from Silicon Graphics, Inc. (SGI), and Hewlett-Packard's Convex Exemplar, all use a similar structure. In each, microprocessors are given one section of a problem by a central controller and work on it with a speed of several hundred million arithmetic operations per second. Potentially, hundreds of such processors can achieve speeds of tens of billions of operations per second (and larger machines built for the government can approach trillions of operations per second); but their access to data forms a bottleneck. Data simply cannot be pulled out of memory and distributed to the processors—or outputted from them—fast enough .

Computer designers solve this problem with either shared memories or distributed memories. In shared-

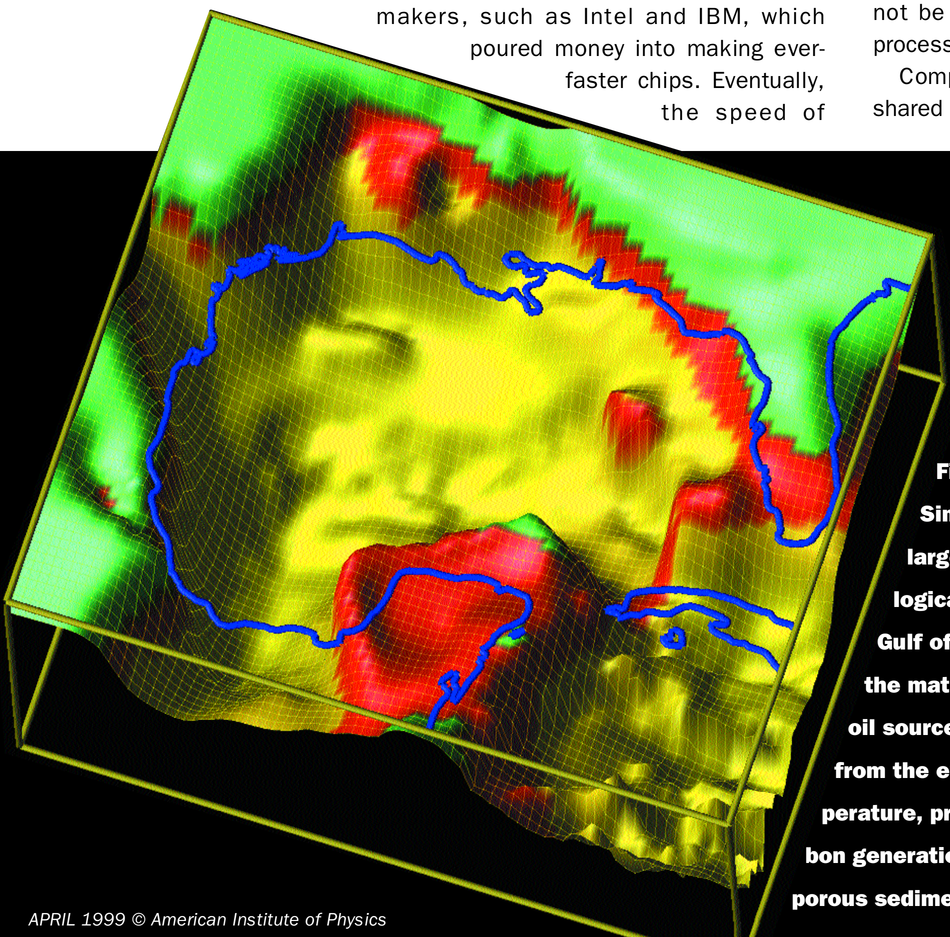
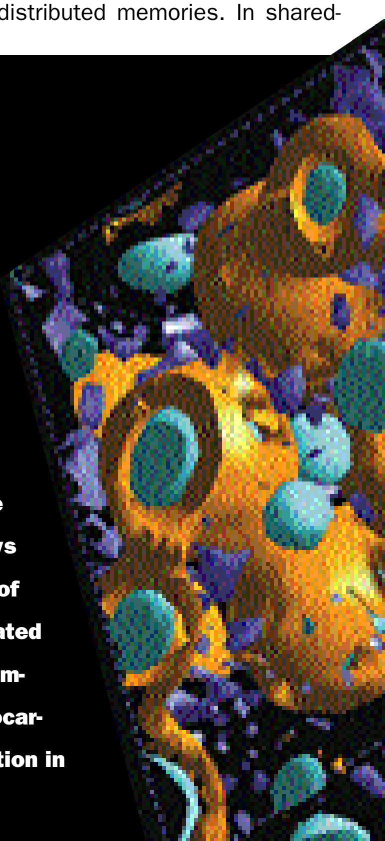


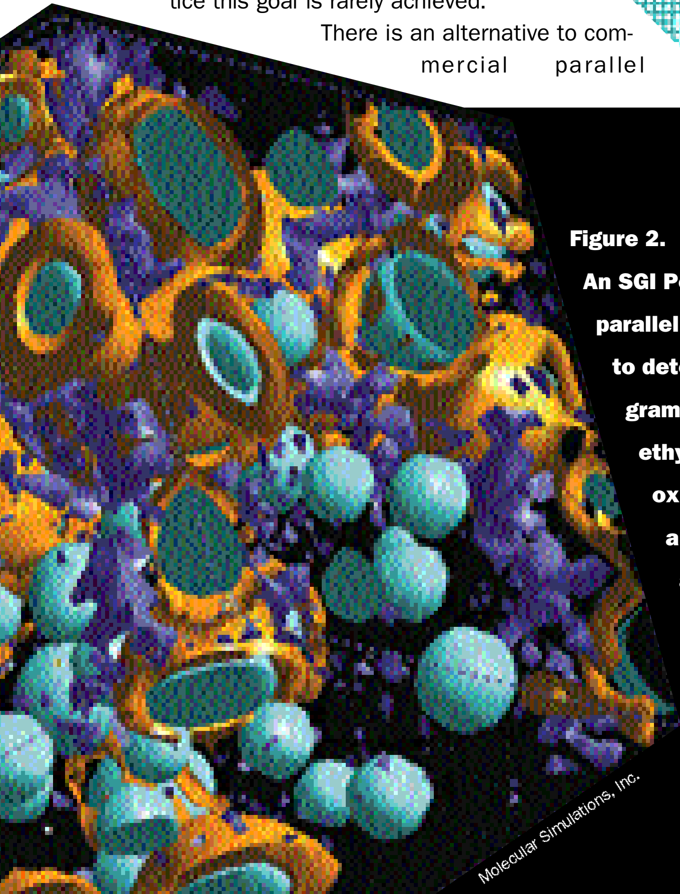
Figure 1. Simulation of a large-scale geological basin in the Gulf of Mexico shows the maturation level of oil source rock, calculated from the evolution of temperature, pressure, hydrocarbon generation, and migration in porous sediments.



memory machines, such as SGI's Origin, each processor can access the entire memory. In distributed-memory machines, like IBM's SP series, each processor has its own separate memory. Shared-memory machines are more versatile for users who need to run both parallel and nonparallel applications. Distributed memory prevents potential conflicts when two processors try to access the same piece of information. In practice, distributed-memory programs can run on either type of machine, with the shared memory being divided up by the software, and the overall performance of the two types of memory is quite comparable.

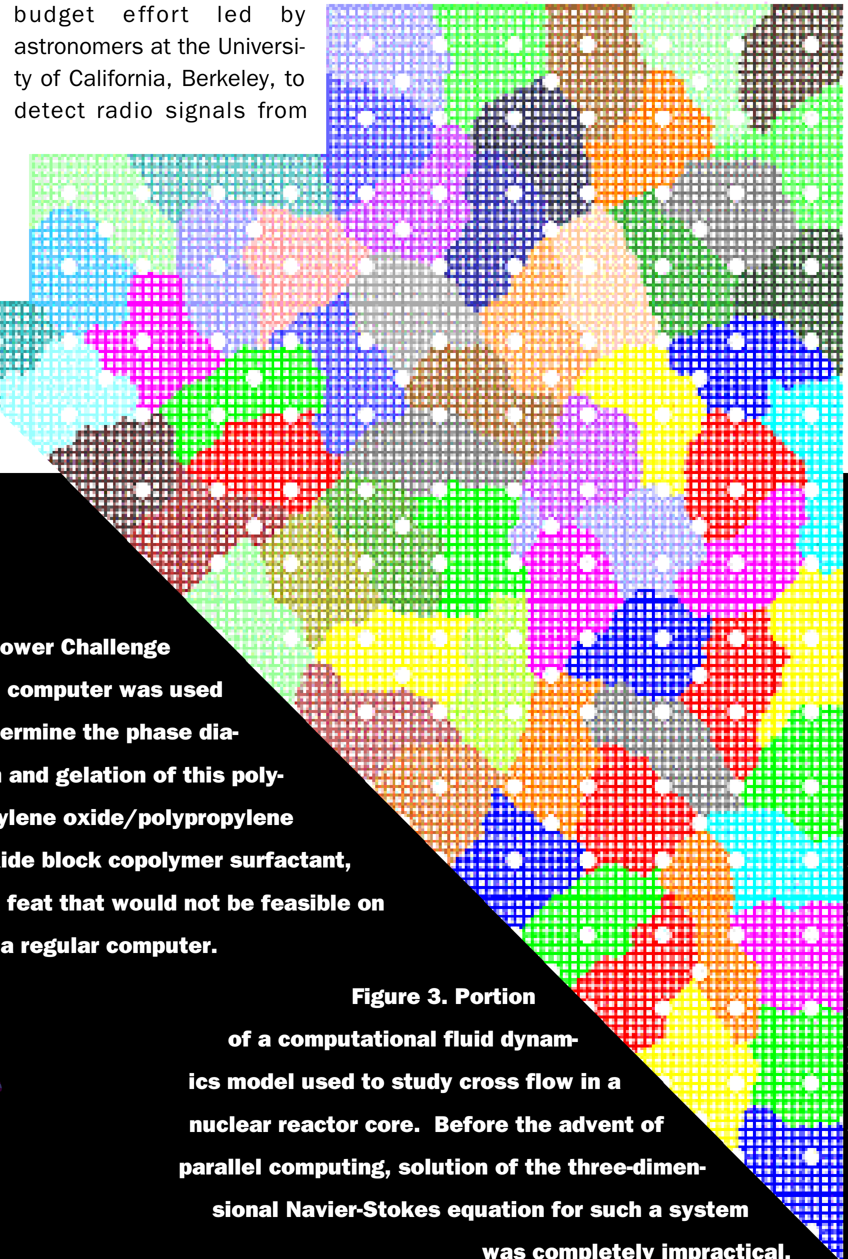
In either case, fast cache memories are essential as a buffer between the processor and the main memories. While a processor is calculating with one batch of data, the cache fills up with another batch of data. The processor withdraws individual pieces of data from the fast, small cache, not from the much slower main memory. How efficiently this process works largely determines how well computer speed scales with the number of processors. Ideally, speedup should grow linearly with the number of processors, but in practice this goal is rarely achieved.

There is an alternative to commercial parallel processors: computer clusters, which are microprocessors purchased as ordinary PCs or workstations and then linked into a network. Several groups, including one at Pratt & Whitney, have combined basic off-the-shelf units that make up a local area network into do-it-yourself parallel supercomputers. More ambitious advocates of this approach talk about connecting PCs through the Internet and using the idle computing capabilities of thousands of PCs to work large problems whose solutions are not urgent. This approach is being attempted as part of a low-budget effort led by astronomers at the University of California, Berkeley, to detect radio signals from



Molecular Simulations, Inc.

Figure 2. An SGI Power Challenge parallel computer was used to determine the phase diagram and gelation of this polyethylene oxide/polypropylene oxide block copolymer surfactant, a feat that would not be feasible on a regular computer.



Argonne National Laboratory, IBM, and adap-

Figure 3. Portion of a computational fluid dynamics model used to study cross flow in a nuclear reactor core. Before the advent of parallel computing, solution of the three-dimensional Navier-Stokes equation for such a system was completely impractical.

extraterrestrial civilizations.

In the case of Pratt & Whitney, hundreds of workstations in a network, which normally stand idle after quitting time, are linked by software that allows them to process in parallel during off-hours. Although such cluster computers are similar to distributed-memory parallel processors, the network connecting them is not special-purpose, and its speed limits the speed with which the cluster can function. Such cluster computers are far less costly than a parallel supercomputer, but they are no panacea. “Depending on the speed of the network, such cluster computers can sometimes be great, or sometimes be a disaster,” says Thomas Marinaccio, an expert in parallel processing at Adaptco (Melville, NY), a software and engineering service firm.

Applications

Industrial problems most suited to parallel-processor supercomputers are those that are computationally intensive and in which each part of the problem depends only on conditions in one local region. This basically means problems involving continuous media—either a solid, liquid, or gas, or combinations of the three—amenable to mathematical simulation. In most such problems, engineering, computer science, and physics principles have to be taken into account to obtain a useful simulation.

One major area of supercomputer application is petroleum geology (Figure 1). Looking for oil is an expensive process. Although huge reserves of cheap-to-produce petroleum still exist in the Middle East and elsewhere, they have already been claimed, and few easily exploitable regions remain on land. Offshore wells in deeper and deeper water cost $20 to $50 million apiece, and on average, companies drill seven dry wells for every producer.

New simulation tools can ease the discovery and production process, cutting costs. Simulations can show how oil-bearing sediment is laid down and help determine where heat, pressure, and time have “cooked” sediment into petroleum. Other simulations can model how oil flows through the reservoir as it is pumped out, and predict where isolated pockets can be found and recovered.

The complexity of geological processes requires solving computational problems that were insolvable before the advent of supercomputers. One key complication is faulting, which causes sedimentary blocks to slip past each other. This process is particularly difficult to model because it requires changes in the topology of a model, that is, in the number and configuration of regions. Generally, simulations involve variables that change smoothly and continuously over time with fixed boundaries, such as the wing of an airplane. Allowing boundaries to move, or allowing new boundaries to be created discontinuously, adds levels of complexity.

Ulisses T. Mello, a researcher at IBM’s Thomas J. Watson Research Center, is a pioneer in this field. With IBM colleague Michael Henderson, he produced the first topologically flexible geological model that could simulate discontinuous processes like faulting. The basic unit of this model is a geological block that is bounded by defined surfaces such as faults or boundaries between different sedimentary layers. The geological block incorporates both the topological and geological characteristics of the model and can deform in response to the modeled geological forces that act on it. A grid, generally triangular, is superimposed on each block to track its motions and compression.

To avoid the problem of creating new topological blocks when a fault comes into existence, Mello’s models have blocks for all the geological units that exist at present in a given region and that have been inferred to exist at an earlier time but are now eroded away. At the start of the simulation, before faults that now exist had formed, the boundaries are “inactive”—the model keeps track of them, but physical conditions are identical on both sides of the boundary and the blocks on both sides are locked together. When geological forces are sufficient to initiate a fault, the model activates the fault boundary and the blocks on the two sides of the fault are free to move separately. Using this technique, Mello has successfully modeled the formation of salt domes, where oil accumulates. To do so, he used an IBM SP computer with up to 200 processors, each of which followed a given chunk of sediment.

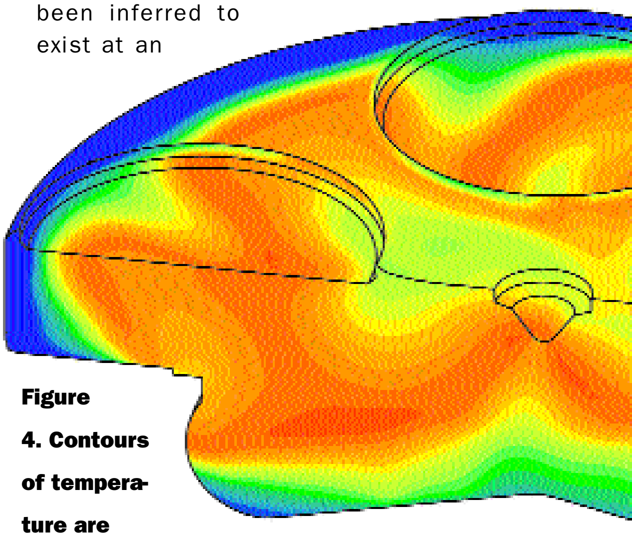
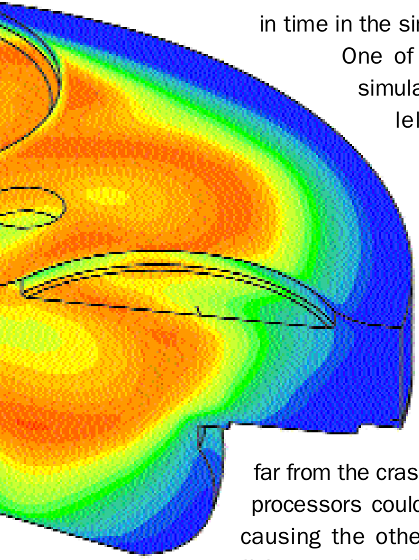


Figure 4. Contours of temperature are shown in the combustion chamber of a direct-injection engine. A full cycle of intake, compression, combustion, and exhaust can now be calculated overnight on a 32-processor machine, instead of taking weeks or even months on a nonparallel computer.

Crashing in silicon

Mello’s approach requires knowing the end state, which geologists do; but it may be applied to fields in which things crack or break, such as automobile crashes. This is a second major area of supercomputer application in industry. Crash testing, from a simulation point of view, has a lot in common with geology. In both cases, solid materials are distorted by large forces. As in geological modeling, a triangular or quadrilateral grid is imposed on an object—in this case a car—and the grid deforms along with the vehicle. Because vehicles contain different materials, the mechanical properties of the materials are modeled by equations linked to each node (or given point) in the grid. These equations often involve nonlinear processes. So the processors handling a given set of nodes must rework the solution over and over many times, changing the distortion until a self-consistent solution is arrived at for each instant in time in the simulation.



One of the key problems in simulating crashes on parallel processors lies in balancing the loads among the processors. If one set of processors handles many nodes near a point of collision, with strong and rapid distortions, and another set has mostly static nodes far from the crash point, one or several processors could become overloaded, causing the other processors to wait until it completes its tasks. Crash programs have to be designed to anticipate such load imbalances and minimize them by allocating fewer nodes per processor in the regions of greatest strain and motion.

With distributed-memory programs, researchers have substantially speeded up simulations that run on parallel processors. For example, in the simulation of a BMW crash at 15 km/h, using 6,000 nodes, an SGI Power Challenge with a distributed-memory program cut processing time from 22 hours with a single processor to 1.5 hours with 16 processors, a 14-fold speedup.

Electronic wind tunnels

Automakers also use parallel processors for one of the oldest applications of large-scale simulations: computational fluid dynamics (CFD). Minor design modifications can have a great impact on improving fuel efficiency. Changes in drag of 10%, for example, can change fuel consumption by 6%. Simulations allow designers to quickly fine-tune designs without repeated costly and time-consuming wind tunnel testing using full size mock-ups (see *The Industrial Physicist*, 9/97, pp. 50-51).

Simulations also address other automotive design problems that affect engine efficiency. Cooling the motor is critical today because of temperature-sensitive computer chips, and simply using large fans and big inlets reduces the aerodynamic efficacy of the overall design. Simulations of chemical reactions inside moving cylinders (see *The Industrial Physicist*, 12/98, pp. 29, 31-33), and even of oil flowing through pumps, can significantly affect design and influence final auto performance (Figure 4).

For all these applications, the advent of parallel-processor supercomputers has been critical to making simulation a practical alternative to wind tunnel and other experimental tests. To achieve adequate accuracy, simulations must be very large scale. For example, a 1-million-node simulation of an entire car’s exterior aerodynamics will produce a drag measurement in error by 50% as compared with wind tunnel testing, which is not a useful result. But with 9 million nodes, the simulation and wind tunnel measurement of drag agree to within the 10% accuracy usually obtained in the tunnel tests. To get turnaround times down to the one or two days required for reasonable design cycles, 8 processors are needed for 9 million nodes.

Automotive CFD simulations pose particular problems because of moving parts, such as pumps and, especially, cylinders. Commercially available software packages, such as STAR-CD, automate the task of both setting up the grid and dynamically changing it in response to the motion of the parts. Such packages include subroutines geared to particular automotive needs, such as fans, pumps, and manifolds. Again, nearly linear speedup can be obtained: using a Hewlett-Packard Exemplar S-class system, a model of air flow in a Mercedes-Benz engine ran 12 times faster on 16 processors than on 1.

At present, simulations reduce but do not eliminate the need for wind tunnel testing. In the next five years, however, as design turnaround times drop to about 1 hour, extensive simulation may relegate wind tunnel testing to a final check of the finished design. Indeed, as parallel processor speeds increase, supercomputer simulation may replace experimental testing more and more in a variety of industrial fields (Figures 2, 3).

BIOGRAPHY

Plasma physicist Eric J. Lerner (elerner@igc.apc.org), author of *The Big Bang Never Happened*, is a free-lance writer based in Lawrenceville, New Jersey. His articles have appeared in *IBM Research*, *Laser Focus World*, *Discover*, and other publications.