Hypercube Breaks a Programming Barrier

A deceptively simple approach allows a massively parallel computer to approach the maximum theoretical speedup

USING a commercially available "hypercube" computer that puts 1024 processors to work simultaneously, researchers from Sandia National Laboratories have broken through a major psychological barrier in designing algorithms for such machines. Whereas most computer scientists had assumed that distributing a problem over so many processors would produce much less than a 1024-fold speedup in its solution, due to inherent bottlenecks in the data flow, the Sandia group has demonstrated on three classes of problems that speedups can indeed approach the theoretical limit.

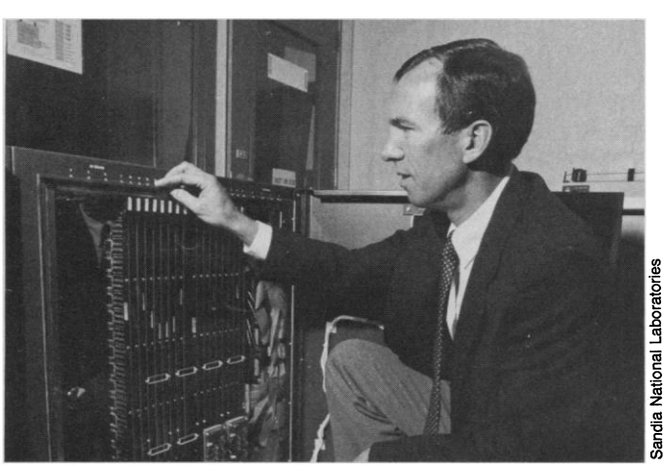
The three classes were nonlinear wave propagation, structural mechanics, and fluid flow. "Two years ago the scientific community thought a 200 times speedup on practical problems like these was impossible," says Sandia researcher John Gustafson, who developed the algorithms in collaboration with Robert E. Benner, Gary R. Montry, and David Womble. Yet in all three cases, he says, "we've gotten over 1000."

The community's skepticism about large-scale parallelism goes back to an argument first put forward in 1967 by Gene Amdahl, one of the pioneers in mainframe computer design. Yes, said Amdahl, putting more processors to work can indeed speed things up, just as hiring more typists can speed things up in an office. On the other hand, just as every letter and every report has to be drafted by someone before it can be typed, every computation contains at least a few steps that have to be carried out sequentially. A computer cannot even add quantity x to quantity y, for example, unless it first calculates the value of those quantities. So Amdahl maintained that there is a limit to how much faster a multiprocessor computer can be: eventually, the sequential bottlenecks will dominate.

The Sandia researchers agree that Amdahl's analysis is quite correct, so far as it goes. However, they also point out that his limit tacitly assumes that the size of the computation is fixed. So they get around that limit by taking an approach that is, in retrospect, almost absurdly obvious: they scale up the size of their problems in proportion to the number of processors available.

To carry on with the office analogy, Amdahl's tacit assumption corresponds to taking a single report and dividing it among hundreds of typists, one per page. The Sandia approach, however, corresponds to giving each typist a full-sized report of his or her own; the sequential bottlenecks are still there, but the office as a whole is now handling such a massive throughput that the delays are comparatively trivial.

As Sandia's director of Computer Science and Mathematics, Edwin H. Barsis, puts it, "If something runs [on a single processor machine] in a few minutes, we don't try to run it in microseconds. Instead we ran one structure problem in a week that would have taken 20 years to run on a single processor."



Sandia National Laboratories

Hypercube in a box

Gilbert G. Weigand, director of Sandia's Parallel Processing Division, kneels before the open cabinet of the NCUBE/ten computer. The device's 1024 processors are mounted on 16 processor boards inside the box.

In practice, of course, achieving the full benefit of massive parallelism was not quite as straightforward as it sounds in principle. "We went back to the basic engineering and physics, and restructured the problems so that they would run efficiently on a parallel machine," says Barsis. In particular, the researchers rewrote their software so as to minimize the amount of time that individual processors spend in communicating among themselves and in synchronizing their activities, two chores that are just as time-consuming in parallel computer systems as they are among people in corporate offices.

The resulting algorithms are in the public domain, and have been accepted for publication in the July 1988 issue of the Society for Industrial and Applied Mathematics' (SIAM's) *Journal on Scientific and Statistical Computing*. It is unfortunately true, says Barsis, that the algorithms are highly specialized, and are not yet generalizable to other classes of problems. On the other hand, he says, "things look promising. I anticipate that in the future we'll have a better understanding of what makes a particular class of problems amenable to massively parallel solutions."

Barsis says that Sandia was able to take the lead in this kind of research in part because it has the need—the laboratory regularly uses conventional supercomputers in evaluating the design of nuclear weapons systems, electronic packaging, and nuclear waste fuel canisters—and in part because it now owns one of the most massively parallel computers anywhere: the $2-million NCUBE/ten, made by the NCUBE Corporation of Beaverton, Oregon.

From the outside, the core of the NCUBE machine simply appears to be a cube 1 meter on a side. Internally, however, it has 1024 individual processors linked in a pattern equivalent to a ten-dimensional hypercube. (Four processors linked in a square would be a two-dimensional hypercube, eight processors linked along the edges of a cube would be a three-dimensional hypercube, and so on.) Each of the individual processors, in turn, has roughly the computational power of a VAX 780, which has long been a familiar standard for mid-level computing on campuses.

Using the Sandia algorithms, says Barsis, the NCUBE machine can now run problems at roughly the same speed as a top-of-the-line supercomputer from Cray Research, Incorporated—at about one-tenth the cost. In his opinion, however, the real promise of parallelism is not just in doing the same things more cheaply. "The potential lies in the future," he says, "when each processor will itself be equivalent to a Cray instead of a VAX." ■ M. MITCHELL WALDROP