The Do-It-Yourself Supercomputer

By William W. Hargrove, Forrest M. Hoffman and Thomas Sterling

Photographs by Kay Chernush

CLUSTER OF PCs at the Oak Ridge National Laboratory in Tennessee has been dubbed the Stone SouperComputer.

This article is the second in a two-part series. The first part, "How to Build a Hypercomputer," by Thomas Sterling, appeared in the July 2001 issue.

puter

Scientists have found a cheaper way to solve tremendously difficult computational problems: connect ordinary PCs so that they can work together

IN THE WELL-KNOWN STONE SOUP FABLE, a wandering soldier stops at a poor village and says he will make soup by boiling a cauldron of water containing only a shiny stone. The townspeople are skeptical at first but soon bring small offerings: a head of cabbage, a bunch of carrots, a bit of beef. In the end, the cauldron is filled with enough hearty soup to feed everyone. The moral: cooperation can produce significant achievements, even from meager, seemingly insignificant contributions.

Researchers are now using a similar cooperative strategy to build supercomputers, the powerful machines that can perform billions of calculations in a second. Most conventional supercomputers employ parallel processing: they contain arrays of ultrafast microprocessors that work in tandem to solve complex problems such as forecasting the weather or simulating a nuclear explosion. Made by IBM, Cray and other computer vendors, the machines typically cost tens of millions of dollars—far too much for a research team with a modest budget. So over the past few years, scientists at national laboratories and universities have learned how to construct their own supercomputers by linking inexpensive PCs and writing software that allows these ordinary computers to tackle extraordinary problems.

In 1996 two of us (Hargrove and Hoffman) encountered such a problem in our work at Oak Ridge National Laboratory (ORNL) in Tennessee. We were trying to draw a national map of ecoregions, which are defined by environmental conditions: all areas with the same climate, landforms and soil characteristics fall into the same ecoregion. To create a high-resolution map of the continental U.S., we divided the country into 7.8 million square cells, each with an area of one square kilometer. For each cell we had to consider as many as 25 variables, ranging from average monthly precipitation to the nitrogen content of the soil. A single PC or workstation could not accomplish the task. We needed a parallel-processing supercomputer—and one that we could afford!

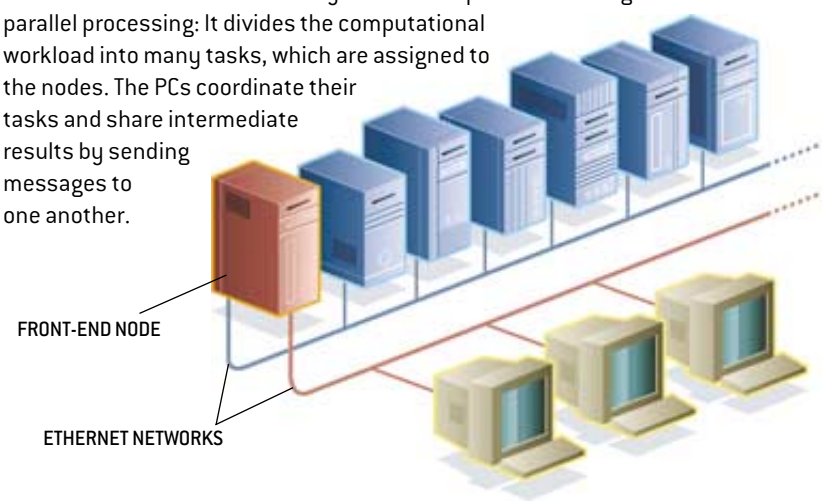
Our solution was to construct a computing cluster using obsolete PCs that ORNL would have otherwise discarded. Dubbed the Stone SouperComputer because it was built essentially at no cost, our cluster of PCs was powerful enough to produce ecoregion maps of unprecedented detail. Other research groups have devised even more capable clusters that rival the performance of the world's best supercomputers at a mere fraction of their cost. This advantageous price-to-performance ratio has already attracted the attention of some corporations, which plan to use the clusters for such complex tasks as deciphering the human genome. In fact, the cluster concept promises to revolutionize the computing field by offering tremendous processing power to any research group, school or business that wants it.

Beowulf and Grendel

THE NOTION OF LINKING computers together is not new. In the 1950s and 1960s the U.S. Air Force established a network of vacuum-tube computers called SAGE to guard against a Soviet nuclear attack. In the mid-1980s Digital Equipment Corporation coined the term "cluster" when it integrated its mid-range VAX minicomputers into larger systems. Networks of workstations—generally

A COMPUTING CLUSTER

The Stone SouperComputer at Oak Ridge National Laboratory consists of more than 130 PCs linked in a computing cluster. One of the machines serves as the front-end node for the cluster; it has two Ethernet cards, one for communicating with users and outside networks, and the other for talking with the rest of the nodes in the cluster. The system solves problems through parallel processing: It divides the computational workload into many tasks, which are assigned to the nodes. The PCs coordinate their tasks and share intermediate results by sending messages to one another.



SAMUEL VELASCO

"CRASH CART" with a monitor and keyboard diagnoses problems with the Stone SouperComputer.

less powerful than minicomputers but faster than PCs—soon became common at research institutions. By the early 1990s scientists began to consider building clusters of PCs, partly because their mass-produced microprocessors had become so inexpensive. What made the idea even more appealing was the falling cost of Ethernet, the dominant technology for connecting computers in local-area networks.

Advances in software also paved the way for PC clusters. In the 1980s Unix emerged as the dominant operating system for scientific and technical computing. Unfortunately, the operating systems for PCs lacked the power and flexibility of Unix. But in 1991 Finnish college student Linus Torvalds created Linux, a Unix-like operating system that ran on a PC. Torvalds made Linux available free of charge on the Internet, and soon hundreds of programmers began contributing improvements. Now wildly popular as an operating system for stand-alone computers, Linux is also ideal for clustered PCs.

The first PC cluster was born in 1994 at the NASA Goddard Space Flight Center. NASA had been searching for a cheaper way to solve the knotty computational problems typically encountered in earth and space science. The space agency needed a machine that could achieve one gigaflops—that is, perform a billion floating-point operations per second. (A floating-point operation is equivalent to a simple calculation such as addition or multiplication.) At the time, however, commercial supercomputers with that level of performance cost about $1 million, which was too expensive to be dedicated to a single group of researchers.

One of us (Sterling) decided to pursue the then radical concept of building a computing cluster from PCs. Sterling and his Goddard colleague Donald J. Becker connected 16 PCs, each containing an Intel 486 microprocessor, using Linux and a standard Ethernet network. For scientific applications, the PC cluster delivered sustained performance of 70 megaflops—that is, 70 million floating-point operations per second. Though modest by today's standards, this speed was not much lower than that of some smaller commercial supercomputers available at the time. And the cluster was built for only $40,000, or about one tenth the price of a comparable commercial machine in 1994.

NASA researchers named their cluster Beowulf, after the lean, mean hero of medieval legend who defeated the giant monster Grendel by ripping off one of the creature's arms. Since then, the name has been widely adopted to refer to any low-cost cluster constructed from commercially available PCs. In 1996 two successors to the original Beowulf cluster appeared: Hyglac (built by researchers at the California Institute of Technology and the Jet Propulsion Laboratory) and Loki (constructed at Los Alamos National Laboratory). Each cluster integrated 16 Intel Pentium Pro microprocessors and showed sustained performance of over one gigaflops at a cost of less than $50,000, thus satisfying NASA's original goal.

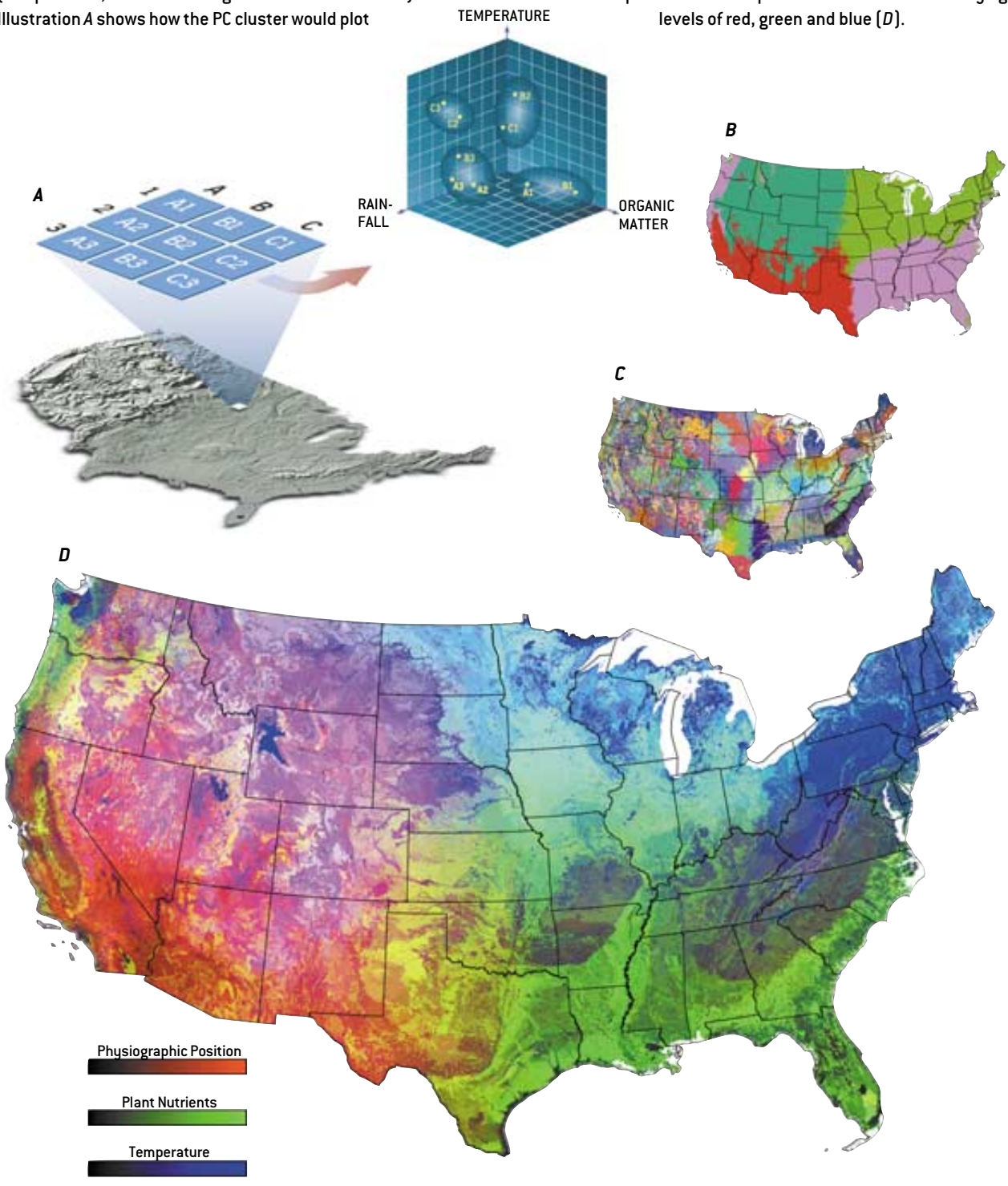
The Beowulf approach seemed to be the perfect computational solution to our problem of mapping the ecoregions of the U.S. A single workstation could handle the data for only a few states at most, and we couldn't assign different regions of the country to separate workstations—the en-

THE AUTHORS

WILLIAM W. HARGROVE, FORREST M. HOFFMAN and *THOMAS STERLING* are pioneers of Beowulf computing. Hargrove, who works in the computational physics and engineering division at Oak Ridge National Laboratory in Tennessee, is a landscape ecologist with big problems and too much data. Hoffman, a computer specialist in the environmental sciences division at ORNL, spends his spare time building supercomputers in his basement. Sterling, who created the first Beowulf cluster while at the NASA Goddard Space Flight Center, is at the California Institute of Technology's Center for Advanced Computing Research and is a principal scientist at the Jet Propulsion Laboratory.

MAKING MAPS WITH THE STONE SOUPERCOMPUTER

TO DRAW A MAP of the ecoregions in the continental U.S., the Stone SouperComputer compared 25 environmental characteristics of 7.8 million one-square-kilometer cells. As a simple example, consider the classification of nine cells based on only three characteristics (temperature, rainfall and organic matter in the soil). Illustration *A* shows how the PC cluster would plot the cells in a three-dimensional data space and group them into four ecoregions. The four-region map divides the U.S. into recognizable zones (*illustration B*); a map dividing the country into 1,000 ecoregions provides far more detail (*C*). Another approach is to represent three composite characteristics with varying levels of red, green and blue (*D*).



ECOREGION MAPS COURTESY OF OAK RIDGE NATIONAL LABORATORY; SAMUEL VELASCO (*illustrations*)

vironmental data for every section of the country had to be compared and processed simultaneously. In other words, we needed a parallel-processing system. So in 1996 we wrote a proposal to buy 64 new PCs containing Pentium II microprocessors and construct a Beowulf-class supercomputer. Alas, this idea sounded implausible to the reviewers at ORNL, who turned down our proposal.

Undeterred, we devised an alternative plan. We knew that obsolete PCs at the U.S. Department of Energy complex at Oak Ridge were frequently replaced with newer models. The old PCs were advertised on an internal Web site and auctioned off as surplus equipment. A quick check revealed hundreds of outdated computers waiting to be discarded this way. Perhaps we could build our Beowulf cluster from machines that we could collect and recycle free of charge. We commandeered a room at ORNL that had previously housed an ancient mainframe computer. Then we began collecting surplus PCs to create the Stone SouperComputer.

A Digital Chop Shop

THE STRATEGY BEHIND parallel computing is "divide and conquer." A parallel-processing system divides a complex problem into smaller component tasks. The tasks are then assigned to the system's nodes—for example, the PCs in a Beowulf cluster—which tackle the components simultaneously. The efficiency of parallel processing depends largely on the nature of the problem. An important consideration is how often the nodes must communicate to coordinate their work and to share intermediate results. Some problems must be divided into myriad minuscule tasks; because these fine-grained problems require frequent internode communication, they are not well suited for parallel processing. Coarse-grained problems, in contrast, can be divided into relatively large chunks. These problems do not require much communication among the nodes and therefore can be solved very quickly by parallel-processing systems.

Anyone building a Beowulf cluster must make several decisions in designing the system. To connect the PCs, researchers can use either standard Ethernet networks or faster, specialized networks, such as Myrinet. Our lack of a budget dictated that we use Ethernet, which is free. We chose one PC to be the front-end node of the cluster and installed two Ethernet cards into the machine. One card was for communicating with outside users, and the other was for talking with the rest of the nodes, which would be linked in their own private network. The PCs coordinate their tasks by sending messages to one another. The two most popular message-passing libraries are message-passing interface (MPI) and parallel virtual machine (PVM), which are both available at no cost on the Internet. We use both systems in the Stone SouperComputer.

Many Beowulf clusters are homogeneous, with all the PCs containing identical components and microprocessors. This uniformity simplifies the management and use of the cluster but is not an absolute requirement. Our Stone SouperComputer would have a mix of processor types and speeds because we intended to use whatever surplus equipment we could find. We began with PCs containing Intel 486 processors but later added only Pentium-based machines with at least 32 megabytes of RAM and 200 megabytes of hard-disk storage.

It was rare that machines met our minimum criteria on arrival; usually we had to combine the best components from several PCs. We set up the digital equivalent of an automobile thief's chop shop for converting surplus computers into nodes for our cluster. Whenever we opened a machine, we felt the same anticipation that a child feels when opening a birthday present: Would the computer have a big disk, lots of memory or (best of all) an upgraded motherboard donated to us by accident? Often all we found was a tired old veteran with a fan choked with dust.

Our room at Oak Ridge turned into a morgue filled with the picked-over carcasses of dead PCs. Once we opened a machine, we recorded its contents on a "toe tag" to facilitate the extraction of its parts later on. We developed favorite and least favorite brands, models and cases and became adept at thwarting passwords left by previous owners. On average, we had to collect and process about five PCs to make one good node.

As each new node joined the cluster, we loaded the Linux operating system onto the machine. We soon figured out how to eliminate the need to install a keyboard or monitor for each node. We created mobile "crash carts" that could be wheeled over and plugged into an ailing node to determine what was wrong with it. Eventually someone who wanted space in our room bought us shelves to consolidate our collection of hardware. The Stone SouperComputer ran its first code in early 1997, and by May 2001 it contained 133 nodes, including 75 PCs with Intel 486 microprocessors, 53 faster Pentium-based machines and five still faster Alpha workstations, made by Compaq.

Upgrades to the Stone SouperComputer are straightforward: we replace the slowest nodes first. Each node runs a simple speed test every hour as part of the cluster's routine housekeeping tasks. The ranking of the nodes by speed helps us to fine-tune our cluster. Unlike commercial machines, the performance of the Stone SouperComputer continually improves, because we have an endless supply of free upgrades.

Our room at Oak Ridge TURNED INTO A MORGUE filled with the picked-over carcasses of dead PCs.

Parallel Problem Solving

PARALLEL PROGRAMMING requires skill and creativity and may be more challenging than assembling the hardware of a Beowulf system. The most common

COMPUTING CLUSTER at the American Museum of Natural History in New York City contains 560 Pentium III microprocessors. Researchers use the system to study evolution and star formation.

model for programming Beowulf clusters is a master-slave arrangement. In this model, one node acts as the master, directing the computations performed by one or more tiers of slave nodes. We run the same software on all the machines in the Stone SouperComputer, with separate sections of code devoted to the master and slave nodes. Each microprocessor in the cluster executes only the appropriate section. Programming errors can have dramatic effects, resulting in a digital train wreck as the crash of one node derails the others. Sorting through the wreckage to find the error can be difficult.

Another challenge is balancing the processing workload among the cluster's PCs. Because the Stone SouperComputer contains a variety of microprocessors with very different speeds, we cannot divide the workload evenly among the nodes: if we did so, the faster machines would sit idle for long periods as they waited for the slower machines to finish processing. Instead we developed a programming algorithm that allows the master node to send more data to the faster slave nodes as they complete their tasks. In this load-balancing arrangement, the faster PCs do most of the work, but the slower machines still contribute to the system's performance.

Our first step in solving the ecoregion mapping problem was to organize the enormous amount of data—the 25 environmental characteristics of the 7.8 million cells of the continental U.S. We created a 25-dimensional data space in which each dimension represented one of the variables (average temperature, precipitation, soil characteristics and so on). Then we identified each cell with the appropriate point in the data space [*see illustration A on page 76*]. Two points close to each other in this data space have, by definition, similar characteristics and thus are classified in the same ecoregion. Geographic proximity is not a factor in this kind of classification; for example, if two mountaintops have very similar environments, their points in the data space are very close to each other, even if the mountaintops are actually thousands of miles apart.

Once we organized the data, we had to specify the number of ecoregions that would be shown on the national map. The cluster of PCs gives each ecoregion an initial "seed position" in the data space. For each of the 7.8 million data points, the system determines the closest seed position and assigns the point to the corresponding ecoregion. Then the cluster finds the centroid for each ecoregion—the average position of all the points assigned to the region. This centroid replaces the seed position as the defining point for the ecoregion. The cluster then repeats the procedure, reassigning the data points to ecoregions depending on their distances from the centroids. At the end of each iteration, new centroid positions are calculated for each ecoregion. The process continues until fewer than a specified number of data points change their ecoregion assignments. Then the classification is complete.

The mapping task is well suited for parallel processing because different nodes in the cluster can work independently on subsets of the 7.8 million data points. After each iteration the slave nodes send the results of their calculations to the master node, which averages the numbers from all the subsets to determine the new centroid positions for each ecoregion. The master node then sends this information back to the slave nodes for the next round of calculations. Parallel processing is also useful for selecting the best seed positions for the ecoregions at the very beginning of the procedure. We devised an algorithm that allows the nodes in the Stone SouperComputer to determine collectively the most widely dispersed data points, which are then chosen as the seed positions. If the cluster starts with well-dispersed seed

DENIS FINNIN *American Museum of Natural History*

positions, fewer iterations are needed to map the ecoregions.

The result of all our work was a series of maps of the continental U.S. showing each ecoregion in a different color [*see illustrations B and C on page 76*]. We produced maps showing the country divided into as few as four ecoregions and as many as 5,000. The maps with fewer ecoregions divided the country into recognizable zones—for example, the Rocky Mountain states and the desert Southwest. In contrast, the maps with thousands of ecoregions are far more complex than any previous classification of the country's environments. Because many plants and animals live in only one or two ecoregions, our maps may be useful to ecologists who study endangered species.

In our first maps the colors of the ecoregions were randomly assigned, but we later produced maps in which the colors of the ecoregions reflect the similarity of their respective environments. We statistically combined nine of the environmental variables into three composite characteristics, which we represented on the map with varying levels of red, green and blue. When the map is drawn this way, it shows gradations of color instead of sharp borders: the lush Southeast is mostly green, the cold Northeast is mainly blue, and the arid West is primarily red [*see illustration D on page 76*].

MORE TO EXPLORE

Cluster Computing: Linux Taken to the Extreme. F. M. Hoffman and W. W. Hargrove in *Linux Magazine,* Vol. 1, No. 1, pages 56–59; Spring 1999.

Using Multivariate Clustering to Characterize Ecoregion Borders. W. W. Hargrove and F. M. Hoffman in *Computers in Science and Engineering,* Vol. 1, No. 4, pages 18–25; July/August 1999.

How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters. Edited by T. Sterling, J. Salmon, D. J. Becker and D. F. Savarese. MIT Press, 1999.

More information about Beowulf computing can be found at the following Web sites:

stonesoup.esd.ornl.gov/
extremelinux.esd.ornl.gov/
www.beowulf.org/
www.cacr.caltech.edu/research/beowulf/
beowulf-underground.org/

Moreover, the Stone SouperComputer was able to show how the ecoregions in the U.S. would shift if there were nationwide changes in environmental conditions as a result of global warming. Using two projected climate scenarios developed by other research groups, we compared the current ecoregion map with the maps predicted for the year 2099. According to these projections, by the end of this century the environment in Pittsburgh will be more like that of present-day Atlanta, and conditions in Minneapolis will resemble those in present-day St. Louis.

The Future of Clusters

THE TRADITIONAL MEASURE of supercomputer performance is benchmark speed: how fast the system runs a standard program. As scientists, however, we prefer to focus on how well the system can handle practical applications. To evaluate the Stone SouperComputer, we fed the same ecoregion mapping problem to ORNL's Intel Paragon supercomputer shortly before it was retired. At one time, this machine was the laboratory's fastest, with a peak performance of 150 gigaflops. On a per-processor basis, the run time on the Paragon was essentially the same as that on the Stone SouperComputer. We have never officially clocked our cluster (we are loath to steal computing cycles from real work), but the system has a theoretical peak performance of about 1.2 gigaflops. Ingenuity in parallel algorithm design is more important than raw speed or capacity: in this young science, David and Goliath (or Beowulf and Grendel!) still compete on a level playing field.

The Beowulf trend has accelerated since we built the Stone SouperComputer. New clusters with exotic names—Grendel, Naegling, Megalon, Brahma, Avalon, Medusa and theHive, to mention just a few—have steadily raised the performance curve by delivering higher speeds at lower costs. As of last November, 28 clusters of PCs, workstations or servers were on the list of the world's 500 fastest computers. The LosLobos cluster at the University of New Mexico has 512 Intel Pentium III processors and is the 80th-fastest system in the world, with a performance of 237 gigaflops. The Cplant cluster at Sandia National Laboratories has 580 Compaq Alpha processors and is ranked 84th. The National Science Foundation and the U.S. Department of Energy are planning to build even more advanced clusters that could operate in the teraflops range (one trillion floating-point operations per second), rivaling the speed of the fastest supercomputers on the planet.

Beowulf systems are also muscling their way into the corporate world. Major computer vendors are now selling clusters to businesses with large computational needs. IBM, for instance, is building a cluster of 1,250 servers for NuTec Sciences, a biotechnology firm that plans to use the system to identify disease-causing genes. An equally important trend is the development of networks of PCs that contribute their processing power to a collective task. An example is SETI@home, a project launched by researchers at the University of California at Berkeley who are analyzing deep-space radio signals for signs of intelligent life. SETI@home sends chunks of data over the Internet to more than three million PCs, which process the radio-signal data in their idle time. Some experts in the computer industry predict that researchers will eventually be able to tap into a "computational grid" that will work like a power grid: users will be able to obtain processing power just as easily as they now get electricity.

Above all, the Beowulf concept is an empowering force. It wrests high-level computing away from the privileged few and makes low-cost parallel-processing systems available to those with modest resources. Research groups, high schools, colleges or small businesses can build or buy their own Beowulf clusters, realizing the promise of a supercomputer in every basement. Should you decide to join the parallel-processing proletariat, please contact us through our Web site (http://extremelinux.esd.ornl.gov/) and tell us about your Beowulf-building experiences. We have found the Stone Soup to be hearty indeed. SA