We can trace the evolution from Crays, to clusters, to supercomputing centers. But where does it go from here?

What's Next in High-Performance Computing?

GORDON BELL AND JIM GRAY

AFTER 50 YEARS of building high-performance scientific computers, two major architectures exist: clusters of Cray-style vector supercomputers; and clusters of scalar uni- and multiprocessors. Clusters are in transition from massively parallel computers and clusters running proprietary software to proprietary clusters running standard software, and to do-it-yourself Beowulf clusters built from commodity hardware and software. In 2001, only five years after its introduction, Beowulf mobilized a community around a standard architecture and tools. Beowulf's economics and sociology are poised to kill off the other architectural lines—and will likely affect traditional supercomputer centers as well.

Peer-to-peer and Grid communities are beginning to provide significant advantages for addressing parallel problems and sharing vast numbers of files. The Computational Grid can federate systems into supercomputers far beyond the power of any current computing center. The centers will become super-data and super-application centers. While these trends make high-performance computing much less expensive and much more accessible, there is a dark side. Clusters perform poorly on applications that require large shared memory.

Although there is vibrant computer architecture activity on microprocessors and on high-end cellular architectures, we appear to be entering an era of supercomputing monoculture. Investing in next generation software and hardware supercomputer architecture is essential to improve the efficiency and efficacy of systems.

High performance comes from parallelism, fast-dense circuitry, and packaging technology. In the 1960s, Seymour Cray introduced parallel instruction execution using parallel and pipelined (7600) function units (CDC 6600, 7600), and by 1975 a vector register processor architecture (Cray 1). These were the first production supercomputers. By 1982, Cray Research had synthesized the multiprocessor (XMP) structure and vector processor to establish the modern supercomputer architecture. That architecture worked extremely well with Fortran because the innermost loops could be carried out by a few pipelined vector instructions, and multiple processors could execute the outermost loops in parallel. Several manufacturers adopted this architecture for large machines (for example, Fujitsu, Hitachi, IBM, and NEC), while others built and delivered mini-supercomputers aka "Crayettes" (Alliant, Ardent, and Convex) in the

early 1980s. In 2001, Cray-style supercomputers remain a significant part (10%) of the market and are vital for applications with fine-grain parallelism on a shared memory (for example, legacy climate modeling and crash codes.) Single node vector supers have a maximum performance. To go beyond that limit, they must be clustered.

It has been clear since the early 1980s that clusters of CMOS-based killer micros would eventually challenge the performance of the vector supers with much better price performance and an ability to scale to thousands of processors and memory banks. By 1985, companies such as Encore and Sequent began building shared memory multiple-microprocessors with a single shared bus that allowed any processor to access all connected memories. Combining a cache with the microprocessor reduced memory traffic by confining memory accesses locally and by providing a mechanism to observe all memory transactions. By *snooping* the bus transactions, a single coherent memory image could be preserved. Bell predicted that all future computers or computer nodes would be *multis* [2]. A flurry of new multidesigns emerged to challenge custom bipolar and ECL minicomputers and mainframes.

A cluster is a single system comprised of interconnected computers that communicate with one another either via a message passing; or by direct, internode memory access using a single address space. In a cluster, internode communication is 10–1000 times slower than intranode memory access. Clusters with over 1000 processors were called massively parallel processors or MPPs. A *constellation* connotes clusters made up of nodes with more than 16 processor multis. However, parallel software rarely exploits the shared memory aspect of nodes, especially if it is to be portable across clusters.

Tandem introduced its 16-node, uniprocesssor cluster architecture in 1975, followed in 1983 by Digital VAXClusters and the Teradata's 1,024 node database machine. This was followed by the IBM Sysplex and SP2 in the early 1990s. By the late 1990s most manufacturers had evolved their micro-based products to be clusters or multicomputers [3]—-the only known way to build an arbitrarily large, scalable, computer system. In the late 1990s, SGI pioneered large, non-uniform memory access (NUMA) shared memory clusters.

In 1983 ARPA embarked on the Strategic Computing Initiative (SCI) to research, design, build, and buy exotic new, scalable, computer architectures. About 20 research efforts and 40 companies were funded by ARPA to research and build scalable computers to exploit the new technologies. By the mid-1990s, nearly all of these efforts had failed. The main benefit was increased effort in scalability and parallelism that helped shift the market to coarse-grain parallelism required by a cluster.

Several other forces aided the transition to the cluster architecture. They were helped by exorbitant tariffs and by policies that prevented U.S. government agencies from purchasing Japanese supercomputers. Low cost clusters empowered users to find an alternative to hard-to-use, proprietary, and expensive architectures.

The shift from vectors to micro-based clusters can be quantified by comparing the Top500 machines in 1993 with 2001.[1] Clusters and constellations from Compaq, Cray, HP, IBM, SGI, and Sun comprise 90% of the Top500. IBM supplied 42% of the 500, including the fastest (12.3Tflops peak with 8192 processors) and slowest (96Gflops peak with 64 processors). Vector supercomputers, including clustered supers from Fujitsu, Hitachi, and NEC comprise only 10%. NEC's 128-processor clustered vector supercomputer operates at a peak of 1.28Tflops. Based on the ratio of their peak speeds, one vector processor is equal to 6–8 microprocessors. Although supers' peak advertised performance (PAP) is very expensive, their real applications performance (RAP) can be competitive or better than clusters on some applications. Shared memory computers deliver RAP of 30–50% of the PAP; clusters typically deliver 5–15% [1].

High-performance computing has evolved into a small, stable, high-priced market for vector supers and constellations. This allows suppliers to lock customers into a unique hardware-software environment, for example, PowerPC/Linux or SPARC/Solaris. Proprietary environments allow vendors to price systems at up to $30K per microprocessor versus $3K per slice for commodity microprocessors, and to maintain the margins needed to fund high-end, diseconomies of scale.

Enter Beowulf

The 1993 Beowulf Project goal was to satisfy NASA's requirement for a 1Gflops workstation costing less than $50,000. The idea was to use commercial off-the-shelf (COTS) hardware and software configured as a cluster of machines. In 1994, a 16-node $40,000 cluster built from Intel 486 computers achieved that goal. In 1997, a Beowulf cluster won the Gordon Bell performance/price Prize. By 2000, several thousand-node Beowulf computers

[1]The Top500 is a worldwide roster of the most powerful computers as measured by Linpack; see www.Top500.org.

were operating. In June 2001, 28 Beowulfs were in the Top500 and the Beowulf population is estimated to be several thousand. High schools can now buy and assemble a Beowulf using the recipe "How to Build a Beowulf" [6].

Beowulf is mostly about software. The success of Beowulf clusters stems from the unification of public domain parallel tools and applications for the scientific software community. It builds on decades of parallel processing research and on many attempts to apply loosely coupled computers to a variety of applications. Some of the components include:

- Message passing interface (MPI) programming model;
- Parallel virtual machine (PVM) programming, execution, and debugging model;
- Parallel file system;
- Tools to configure, schedule, manage and tune parallel applications (for example, Condor, the Maui scheduler, PBS); and
- Higher-level libraries, for example Linpack, BLAS.

Beowulf enabled do-it-yourself cluster computing using commodity microprocessors—the Linux/GNU or Windows 2000 operating system, plus tools that have evolved from the research community. This standard software platform allows applications to run on many computer types—and thereby fosters competition (and avoids lock-in). Most importantly, Beowulf is a convergent architecture that will run over multiple computer generations, and hence protects application investment. Beowulf fosters a community of users with common language, skills, and tools, but with diverse hardware. Beowulf is the alternative to vector supercomputers and proprietary clusters normally found in centers.

Comparison of computer types in the Top500 between 1993 and 2001.

Type	1993		2001			
	Number	Vendors	Number	Vendors	New	Defunct*
Scalar	133	9	450	6	3	6
Vector	332	4	50	3	0	1
SIMD**	35	1	0	0	0	1

*Either computer or the company producing it has ceased to exist.
**Single Instruction stream, Multiple Data operations. An architecture with 16–64 thousand units to exploit VLSI that was abandoned as microprocessors overtook it.

Centers: Haven't We Seen this Movie?

We have seen computation and data migrate over time from central facilities when no low-cost facilities were available, to distributed VAX minicomputers in the early 1980s, then back to a few large NSF and state-supported centers with PCs for access in the mid-1980s, to fewer, large centers in the late 1990. Now, we are back to build-it-yourself clusters.

Beowulf's economics have important socioeconomic-political effects. Now individuals and laboratories believe they can assemble and incrementally grow any size supercomputer anywhere in the world. The decision of where and how to compute is a combination of cost, performance, availability (for example, resource allocation, application program, ease of access, and service), the applications focus and dataset support, and the need or desire for individual control.

Economics is a key Beowulf advantage. The hardware and software is much less expensive. Centers add a cost factor of 2 to 5. Indeed, a center's costs are explicit: space (equals air conditioning, power, and raised floors for wiring and chilled air ducts), networking, and personnel for administration, system maintenance, consulting, and so on. A center's explicit costs are implicit when users build and operate their own centers because homegrown centers ride free on their organizational overhead that includes space, networks, and especially personnel.

Sociology is an equally important Beowulf advantage. Its standards-setting and community nature, though not usually part of the decision, eliminates a barrier because users have access to both generic and profession-specific programs and talent that centers try to provide. Furthermore, a standard platform enables a market for programs and enhanced technical recognition.

The situation is similar to the late 1970s when VAX was introduced and Cray users concluded it was more productive and cost effective to own and operate their own, smaller, focused centers. Scientists left centers because they were unable to get sufficient computing power compared to a single-user VAX. Although the performance gap between the VAX and a center's Cray was a factor of 5–10 and could be 100; the performance per price was usually the reverse.

By the mid-1980s, government studies bemoaned the lack of supercomputer centers and supercomputer access for university scientists. These researchers were often competing to make breakthroughs with their counterparts in extremely well funded Department of Energy (DOE) labs. The various DOE labs had been given the mandate with the Advanced Strategic Computing Initiative (ASCI) to reach 10Tflops and

petaflops (10^{12} and 10^{15} floating-point operations per second, respectively) levels in 2001 and 2010 in order to fulfill their role as the nation's nuclear stockpile steward.

In response, the NSF established five centers in 1985. Keeping all of the expensive supercomputer centers at the leading edge was neither affordable nor justified, especially in view of the relatively small number of users. To be competitive, a center must house one of the world's largest computers (about two orders of magnitude larger than what a single researcher can afford).

In response to these realities, NSF reduced the number of supercomputing centers to two in 1999. This concentrated enough funding to achieve several teraflops at each center. The plan was that each year or so, one of the two centers would leapfrog the other with new technology to keep centers at the forefront and provide services that no single user could afford. In 2001, NSF seemed to have forgotten all this[2] and created a third center—or at least funded the CPU and memory with what turned out to be Compaq's last, Alpha cluster and inherently an orphan. Storage was unaffordable! The next act is predictable: The NSF will underfund all three centers and eventually discontinue one of them. The viability of individual centers decreases as more centers dilute funding.

Some centers claim a role with constellations built from large shared memory multiprocessor nodes. Each of these nodes is more powerful than a Beowulf cluster of commodity PC uni- or dual processors.

The centers idea may already be obsolete in light of Beowulfs, computational Grids, and peer-to-peer computing. Departmental Beowulfs are attractive for a small laboratory because they give low-overhead dedicated access to nearly the same capability a large center provides. A center typically allocates between 64 and 128 nodes to a job,[3] comparable to the Beowulf that most researchers can build in their labs (like their VAXen two decades earlier). To be competitive, a supercomputer center needs to have at least 1,000 new (less than two years old) nodes, large data storage for each user community, and some asset beyond the scope of a small laboratory.

We believe that supercomputer centers may end up being fully distributed computation brokers—either collocated with instrumentation sites as in the case of the astronomy community, or centers to support peer-to-peer computing (for example, www.seti.org averaging 10Tflops from 1.6 million participants who donate their computer time, or www.Entropia.com that brokers fully distributed problems to Internet PCs).

We foresee two possible future scenarios for supercomputer centers:

Exotic. An application-centric vector or cellular supercomputer (www.research.ibm.com/BlueGene) for an area like weather genomics to run applications that users have been unable to use to a Beowulf architecture or Japan's Earth Observation Research Center Simulator; www.eorc.nasda.go.jp.

Data Center. A concentration of peta-scale datasets (and their applications) in one place so that users can get efficient and convenient access to the data. The various NASA Data Access Archives and Science Data Centers fit this model. The Data Center becomes increasingly feasible with an Internet II delivering 1–10Gbits per second.

Both these models cast the supercomputer center as the steward of a unique resource for specific application domains.

Paths to Petaflops Computing

The dark side to Beowulf commodity clusters is they perform poorly on applications that require large shared memory. We are concerned that traditional supercomputer architecture is dead and that we are entering a supercomputer monoculture. At a minimum we recommend increased investment in research on ultra-high-performance hardware-software architectures including new programming paradigms, user interfaces, and especially peta-scale distributed databases.

In 1995 a group of eminent architects outlined approaches that would achieve a petaops by 2010 [7]. They recommended three interconnected machines: a 200Tflops multithreaded shared memory architecture; a 10,000 node cluster of 0.1Tflops nodes; and 1 million, 1Gflops processor-in-memory nodes. Until recently, Sterling had been pursuing data-flow architectures with radical packaging and circuit technology. IBM's BlueGene is following the third path (a million gigaflops chips) to build a petaflops machine by 2005 geared to protein folding and other parallel tasks with limited memory needs (it has mips:megabyte ratio of 20:1 versus 1:1). IBM is also considering a better balanced machine codenamed Blue Light. Only a small number of unconventional experimental architectures, for example, Berkeley's processor-in-memory are being pursued.

Because custom system-on-a-chip experiments are

[2]Although the NSF is an independent agency directly funded by Congress, it is subject to varying political winds and climate that include Congressional people, conflicting centers, and directorate advisory committees, and occasionally its own changing leadership.

[3]At a center (with approximately 600 SP2 processors), one observed: 65% of the users ran on more than 16 processors; 24% on more than 32; 4% on more than 64; 4% on more than 128; and 1% on more than 256.

so complex and the tools so limited, we can only afford a few such experiments.

Next-generation Beowulfs represent the middle path. It has taken 25 years to evolve the crude clusters we have today. The number of processors has stayed below a maximum of 10,000 for at least five years, with very few apps able to utilize more than 100 processors. By 2010, the cluster is likely to be the principal computing structure. Therefore research programs that stimulate cluster understanding and training are a good investment for laboratories that depend on the highest performance machines. Sandia's computational plant program is a good example; www.cs.sandia.gov/cplant/.

Future Investments

Continued investments to assure that Moore's Law will continue to be valid underlies all of our assumptions about the future. Based on recent advances and predictions, progress is likely to continue for at least another decade. Assuming continued circuit progress, performance will come from a hierarchy of computers starting with multiprocessors on a chip. For example, several commodity chips with multiple processing units are being introduced that will operate at 20Gflops. As the performance of single, multiprocessor chips approaches 100Gflops, a petaflops machine will only need 10,000 units.

On the other hand, it is hardly reasonable to expect a revolutionary technology within this time period because we see no laboratory results for near-term revolution. Certainly petaflops performance will be achieved by special-purpose computers like IBM's Blue Gene project, but they stand alone.

SGI builds a shared memory system with up to 256 processors and then clusters these to form a constellation. But this architecture is low-volume and hence expensive. On the other hand, research into high-speed interconnections such as Infiniband, may make the SGI approach a commodity. It is entirely possible that huge cache-only memory architectures might emerge in the next decade. All these systems require good locality because on-chip latencies and bandwidth are so much better than off-chip. A processor-in-memory architecture or multisystem on a chip will no doubt be part of the high-performance equation.

In 2001, the world's Top500 computers consist of about 100,000 processors, each operating at about 1Gflops. Together they deliver slightly over 100Tflops.

SETI@home (www.seti.org) does not run Linpack, so does not qualify in the Top500. But SETI@home averages 13Tflops, making it more powerful than the top three of the Top500 machines combined. This suggests that GRID and peer-to-peer computing using the Internet II is likely to remain the world's most powerful supercomputer.

Beowulfs and Grid computing technologies will likely merge in the next decade. When multigigabit LANs and WANs become ubiquitous, and when message passing applications can tolerate high latency, the Grid becomes a Beowulf. So all the LAN-based PCs become Beowulfs—and together they form the Grid.

Progress has been great in parallelizing applications that had been challenging in the past (for example, n-body problems). It is important to continue on this course to parallelize applications heretofore deemed the province of shared memory multiprocessors. These include problems requiring random variable access and adaptive mesh refinement. For example, automotive and aerodynamic engineering, climate and ocean modeling, and applications involving heterogeneous space remain the province of vector multiprocessors. It is essential to have the list of challenges to log progress—unfortunately, the vector-super folks have not provided this list.

Although great progress has been made by computational scientists working with computer scientists, the effort to adopt, understand, and train computer scientists in cluster and constellation parallelism has been minimal. Few computer science departments are working with their counterparts in other scientific disciplines to explore the application of these new architectures to scientific problems. C

References

1. Bailey, D.H. and Buzbee, W. Private communication.
2. Bell, C.G. Multis: A new class of multiprocessor computers. *Science* 228. (Apr. 25, 1985), 452–457.
3. Bell, C.G. and Newell, A. *Computer Structures.* McGraw-Hill, New York 1971.
4. Bell, G. Ultracomputers: A teraflop before its time. *Commun. ACM 35*, 8 (Aug. 1992), 27–45.
5. Foster, I. and Kesselman, C., Eds. *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufman, San Francisco, 1999.
6. Sterling, T. *Beowulf PC Cluster Computing with Windows and Beowulf PC Cluster Computing with Linux*. MIT Press, Cambridge, MA, 2001.
7. Sterling, T., Messina, P., and Smith, P.H. *Enabling Technologies for Petaflops Computing.* July 1995. MIT Press, Cambridge, MA.

Gordon Bell(gbell@microsoft.com) is Senior Researcher at the Bay Area Research Center of Microsoft Research, San Francisco, CA.
Jim Gray(gray@microsoft.com) is Distinguished Engineer at the Bay Area Research Center of Microsoft Research, San Francisco, CA.