

# ;login:

THE MAGAZINE OF USENIX & SAGE

August 2001 • Volume 26 • Number 5

inside:

CLUSTERS

BOWWOLF CLUSTER COMPUTING AT THE  
THIRD PLATEAU

by Dr. Thomas Sterling

Special Focus  
Issue: Clustering

Guest Editor: Joseph L. Kaiser

**USENIX & SAGE**

The Advanced Computing Systems Association &  
The System Administrators Guild

# beowulf cluster computing at the third plateau

## Introduction

It was only seven or eight years ago that the early NOW (Network Of Workstations) and Beowulf projects launched their exploration of the opportunity to harness clusters of low-cost workstations and PCs, respectively, to achieve significant speedups on real-world applications at superior price-performance. Prior to that, and even to this day, throughput (also referred to as “capacity”) computing was pursued by means of workstation and PC farms: desktop and server computers on shared local area networks whose first obligation was to serve dedicated users but whose available idle cycles might be applied in concert to some additional workload, albeit largely decoupled.

While such capacity processing leverages existing resources, increases efficiency of system utilization, and yields advantage in cost of computation, workstation clusters and Beowulf-class systems employ parallel processing to increase the size and speed of individual applications. Even the inchoate Beowulf systems of the mid-1990s were competitive in performance with respect to their MPP counterparts (of equal numbers of processors) while exhibiting price-performance advantage of an order of magnitude or more for some (but not all) technical and scientific problems.

By 1996, Beowulf systems were delivering supercomputer performance at high-end scientific workstation prices, earning the 1997 Gordon Bell Prize for price-performance and being dubbed “do-it-yourself supercomputing” by *Science* magazine. Beowulf cluster computing had reached the first plateau. Beowulf was useful, distinct, and attracted many practitioners. As a subdiscipline of parallel computing, it was self-sustaining.

In the intervening years, Beowulfs have experienced an explosive growth in the scale of capability and capacity as well as their installed base and range of application domains. On a per processor basis, clock rate has increased by a factor of more than an order of magnitude and peak floating-point performance by more than two orders of magnitude. Network bandwidth has increased as well by two orders of magnitude while latency has dropped by more than a factor of 10. Storage capacity of both main memory and hard disks has been expanded by more than an order of magnitude on a per node basis. Overall, price-performance has improved by better than 200 and total system performance for the largest Beowulf-class systems, taking into consideration system scale (number of processors) as well as per node performance, has exploded by 10,000 – truly a revolutionary capability.

Today, many of the Top 500 computing systems are commodity clusters that include Beowulf-class systems. In the near future, such clusters are likely to be at the top of the list with important new systems under development by NSF and DOE. But the dramatic evolution and impact of Beowulf-class clusters have been enabled as much by software as hardware, and their future is as dependent on next generation software technology as their hardware technologies. Today, commodity clusters are at the second plateau, defined as much by their supporting software as their assembly of hardware. Slowly and incrementally, elements making up the software environment have matured and been enhanced in scope to enrich the tools with which clusters are managed and applied.

## by Dr. Thomas Sterling

Thomas Sterling holds a joint appointment at the NASA Jet Propulsion Laboratory's High Performance Computing Group, where he is a principal scientist, and the California Institute of Technology's Center for Advanced Computing Research, where he is a faculty associate.



For the past 20 years, Sterling has carried out research on parallel-processing hardware and software systems for high-performance computing. Since 1994, he has been a leader in the national petaflops initiative. He heads the hybrid technology multithreaded architecture research project.

[tron@cacr.caltech.edu](mailto:tron@cacr.caltech.edu)

## Beowulf clusters are at a pivotal stage in their evolution

But now Beowulf clusters are at a pivotal stage in their evolution. They are poised to dominate the realm of high-performance computing, but only if they can provide the level of services and robustness demanded of early generations of vector supercomputers, MPPs, DSMs, and SMPs. The promise of commodity clusters is their potential ability to ratchet up the performance by creating ensembles of these and other classes of computing elements including simple PCs. But they will fail if they prove too difficult to use. The question is: What are the key attributes that must be achieved to bring Beowulf clusters to the level needed to dominate high-end parallel systems, to reach the third plateau?

### The First Plateau

Though there was more than one choice, it was Linux combined with MPI that made Beowulf-class systems both possible and practical. Initially, PVM was the message-passing library employed for the first Beowulf systems. With the emergence of the community-wide standard across platforms and the potential for true portability, the use of Beowulfs took off, but these were primitive environments at best. Linux provided the virtual memory multi-tasking node environment needed to build a distributed capability. And MPI provided the logical inter-process data transfer and coordination mechanisms necessary for cooperative computing.

However, these simple systems were usually modest in scale, rarely more than 64 processors and often only 16 processors or less. They usually ran only one parallel application at a time, often administered by a single individual or small group where scheduling was coordinated by word of mouth. Frequently, a simple set of tools for monitoring the low-level status of the cluster nodes was developed in-house. In some cases, not even MPI was used, the programmers preferring to optimize their application communications using the sockets layer protocol. Communication performance could be improved by a factor of two to three when custom crafted compared to early implementations of MPI.

Although basic, these simple systems were responsible for a strong grassroots community effort to establish PC clusters as a viable low-cost alternative to expensive more tightly coupled vendor-specified parallel computers. More than cost, these early ensembles had other attributes that a portion of the user community found desirable. One was flexibility of configuration and easy upgrades. Many aspects of the system structure could be determined by the end user without permission from some vendor. Another was the low vulnerability to vendor market and product decisions. If one vendor stopped producing the elemental components of a Beowulf, it was easy to acquire comparable units from any one of several other distributors. This sense of confidence and empowerment led to the view that Beowulf-class systems represented a convergent architecture – one for which application software could be guaranteed many generations of compatible hardware.

### The Second Plateau

By 1997 it was clear that commodity clusters were having an impact on high-end computing and would become a major force in parallel processing. Vendors began to support commodity clusters and, ultimately, even Linux-based Beowulf systems. Both hardware and software products were developed to directly support a burgeoning commodity clusters market.

At the first plateau, Beowulf systems leveraged existing pieces of hardware and software developed for other purposes with only small additions such as network drivers contributed by the community where essential. But by 1999, if not before, Beowulf-class

cluster computing had reached the second plateau, where hardware and software were being developed and, in some cases, marketed explicitly for clusters. At the second plateau, node packaging and system area networks were implemented to facilitate commodity clusters. Rack-mounted 1U packages are now available that permit 40 or more nodes to be assembled in a single rack where less than half that many could be contained in the same footpad using desk-side towers.

SCI and Myrinet were devised initially for workstation clusters, but as their costs were reduced per node and the scale of topologies they could implement was increased including the degree per switch, the target system was increasingly large PC clusters. The virtual interface architecture (VIA) was devised by a consortium of industrial partners for the express purpose of reducing the latency of communication between nodes within a cluster, with example implementations including Servernet II and cLAN. Second-plateau cluster scale grew from moderate-sized systems to those comprising more than a thousand processors.

However, the most significant advance marking the transition to the second plateau is the improvement in software environments and tools. Linux, once a hobbyist's plaything, emerged as the foremost UNIX-like operating system, providing serious competition to NT in certain markets. Linux now has myriad distributions, some of which are sophisticated environments including some support for Beowulf clusters.

The Extreme Linux consortium reworked the Linux kernel internals to eliminate bottlenecks to scalability and to reduce inefficient mechanisms. Equally important was the development of distributed resource-management software compatible with Linux. Several schedulers were developed that have wide distribution, including PBS and the Maui schedulers. PVFS from Clemson University provides one example of a parallel file system developed explicitly for commodity clusters. A number of tools exist for monitoring the status, operation, and behavior of the system nodes. Etnus provides a parallel debugger. Oscar, a consortium led by Oak Ridge National Laboratory, is one of several efforts to provide an inter-operable collection of the basic cluster software derived from a number of existing tools. Today, commodity clusters of the second plateau are challenging all other forms of high-performance computing for preeminence.

### The Third Plateau

Commodity clusters may remain, as they are, ensembles of conveniently packaged nodes, interconnected by means of quasi-independent networks and running a collection of separate but mutually friendly software packages that provide various services to allow users to get by. In such a scenario, clusters will continue to contribute to the technical computing arena and aspects of the transaction-processing commercial domain. However, for commodity clusters to rise above their limited condition and forge a new path, then significant advances in both hardware and software will be required. A new generation of Beowulf clusters will be required, ones that achieve the third plateau.

Hardware for Beowulf clusters at the third plateau is being pursued aggressively by industry independently and collectively. The transition to 64-bit architecture for high-end PCs, now currently dominated by Compaq's Alpha family, will be accelerated with the market emergence of the long-awaited Intel IA-64 processor. Meanwhile, IBM describes future plans of incorporating multiple processors on a single chip, thus ensuring continued increase in performance per chip through at least the end of the decade made possible by such new techniques as EUV lithography.

Rack-mounted 1U packages are now available that permit 40 or more nodes to be assembled in a single rack

We can expect processor chips delivering as much as 10Gflops peak performance in the next few years.

We can expect processor chips delivering as much as 10Gflops peak performance in the next few years. The industry consortium developing the Infiniband network architecture will push bandwidths up beyond 10Gbps per channel and reduce network latency to near microsecond levels. With continued rapid increase in DRAM capacity and innovative structures for achieving greater effective memory bandwidth, hardware systems will be capable of petaflops-scale performance by the end of the decade. Market pressures for laptops, PDAs, and cellular phones will force power consumption down while significantly improving the size, weight, and power of micro-disks, further enhancing the practicality of large-scale Beowulf-class systems and their availability to a broad range of institutions and users. With these significant advances, commodity cluster hardware is well positioned to reach the third plateau in a few years.

The obstacle to this next quantum step is in enabling software. There are three main challenges to next generation commodity clusters that may limit their long-term impact: resource management, fault recovery, and programming methodology. Resource management involves all aspects of system initialization, maintenance, administration, and job allocation. Today, while some strides have been made in each of these areas for second plateau systems, they represent only partial and incomplete solutions. For example, system administration tools are not on a par with conventional uniprocessor systems. Also, launching new processes is often not nearly as efficient on remote cluster nodes as on the same local processor. Managing copies of software across a large number of nodes can be surprisingly difficult, and it is easy to have nodes within a cluster be inconsistent.

A major thrust in the commodity-cluster community is achieving what is called “single-system image,” or SSI. In any system, whether uniprocessor, multiprocessor, or cluster, there are multiple namespaces. These include the variable-address space, file-name space, process-id space, and others. On a uniprocessor, each of these namespaces is single: that is, there is only one such space for each class of name. But on a cluster, in the worst case, there can be as many separate namespaces for each class of name as there are nodes in the system. Commodity clusters at the third plateau will present a single-system image to the user and administrator for most if not all name classes. The exception may be the user-variable namespace. Even here, hardware solutions such as SCI and software solutions such as HPF provide ways to let the user think about a single variable namespace. The Scyld tool set manages all remote-process calls through a single master node providing a single process-id namespace. PVFS provides a single parallel-file namespace. It is possible that a synthesis of these methods may ultimately provide the SSI operation that is key to effective control of large clusters of the third plateau.

Efficiency of resource management is also critical to the successful deployment of clusters. Again, Scyld’s process migration mechanisms can achieve speedups of as much as an order of magnitude in starting a process compared to rsh. Another efficiency improvement may come from Linux BIOS being developed by Los Alamos National Laboratory which optimizes the onboard BIOS of each node for use with Linux. This can permit a new node to be brought up as part of a cluster system in less than a minute.

The challenge of reliability requires new fault-response techniques that will prevent an entire cluster system from crashing each time a failure occurs with a single node. For the largest scale clusters with as many as 10,000 processors, MTBF measured in hours or less is possible (numbers vary significantly depending on whether you include infant mortality as part of the life cycle). When a hard fault occurs, future management tools must allow the rest of the system to continue to operate. Ideally, the application running on

the failed node could be restarted at an earlier stage in its computation through automatic checkpointing, thus avoiding having to restart the application from the beginning. While progress in related areas has been made, no fully satisfactory solution exists, and one will be required.

Thus commodity clusters of the third plateau will present the user with a single-system image, manage its resources to varying demands of users and administrators, and provide high availability even in the presence of faults. It is unclear how third-plateau systems of the future will be programmed. New models are slow to achieve acceptance, and the heritage of legacy codes causes older languages to remain for many years. But in the long term, users are likely to be separated from the arduous task of directly managing the computing resources from within the application code and are more likely to rely on advanced operating system and runtime system software to allocate tasks to physical components.

## Conclusion

Beowulf-class systems and other forms of commodity clusters have evolved over less than a decade from primitive small piles of PCs to among the largest systems in the world. However, their future dominance and impact on both technical and commercial computing will depend on future advances that will allow them to attain the third plateau. This is a stage of robust, manageable, and effective computing systems that will permit their use in almost any domain of workload currently serviced by other system forms. The third plateau will represent the final stage in this evolution and will offer a stable, cost-effective convergent form of parallel architecture that will promise users dependable and scalable computing platforms for generations to come.

Beowulf-class systems and other forms of commodity clusters have evolved over less than a decade from primitive small piles of PCs to among the largest systems in the world