Green Supercomputing in a Desktop Box*

Wu-chun Feng*, Avery Ching†, and Chung-Hsing Hsu‡

*Virginia Tech
Dept. of Computer Science
Blacksburg, VA 24061 USA
feng@cs.vt.edu

†Northwestern University
Dept. of EECS
Evanston, IL 60208 USA
aching@ece.northwestern.edu

‡Los Alamos National Laboratory
Advanced Computing Lab
Los Alamos, NM 87545 USA
chunghsu@lanl.gov

Abstract

The advent of the Beowulf cluster in 1994 provided dedicated compute cycles, i.e., supercomputing for the masses, as a cost-effective alternative to large supercomputers, i.e., supercomputing for the few. However, as the cluster movement matured, these clusters became like their large-scale supercomputing brethren — a shared (and power-hungry) datacenter resource that must reside in a actively-cooled machine room in order to operate properly. The above observation, coupled with the increasing performance gap between the PC and supercomputer, provides the motivation for a "green supercomputer" in a desktop box. Thus, this paper presents and evaluates such an architectural solution: a 12-node personal desktop supercomputer that offers an interactive environment for developing parallel codes and achieves 14 Gflops on Linpack but sips only 185 watts of power at load — all this in the approximate form factor of a Sun SPARCstation 1 pizza box.

1 Introduction

Sun Microsystems introduced the first workstation to the scientific computing community in 1982. By the late 1980s, Sun had become the undisputed leader of the workstation market when they introduced the Sun SPARCstation 1, rated at 12.5 MIPS and 1.4 Mflops while running at 20 MHz. The workstation's features were so tightly integrated that they fit in a 16" x 16" x 3" enclosure — the first "pizza box" workstation. By 1992, Sun introduced the first multiprocessing desktop workstation, the Sun SPARCstation 10 with dual 60-MHz SuperSPARC processors; but rather than continue to scale-up the number of processors in the SPARCstation 10, Sun instead delivered the 64-bit UltraSPARC in the mid-1990s even though its price-performance ratio was much worse than PCs. This arguably led to the demise of the computer workstation, when coupled with the emergence of PCs as cost-effective alternatives to workstations.

Concurrent to the emergence of the PC was the open-source Linux operating system (OS). This confluence of technologies ultimately led to the Beowulf commodity-clustering movement [1], a movement that dramatically lowered the entry costs into high-performance computing for computational scientists and provided dedicated "supercomputing for the rest of us." However, as this movement matured through the late 1990s and early 2000s, commodity clusters became the very thing that they were purported to be an alternative to, i.e., an expensive, expansive, and power-hungry resource that resides in a specially-cooled datacenter whose shared use is arbitrated by a batch scheduler such as LSF or PBS.

With the notion of "supercomputing for the rest of us" now effectively obsolete, how does an application scientist develop a parallel code *on the desktop*? A dual-processor SMP platform like the Dell PowerEdge 2650 may neither be enough to debug a parallel code nor to test its scalability.

*This paper is also available as Los Alamos Technical Report LA-UR-07-0360.

1-4244-0910-1/07/$20.00

On the other hand, using a shared datacenter HPC resource like a large-scale cluster can result in scheduling conflicts or long queues, resulting in longer turnaround times for the program developer. This makes debugging a parallel application more of a batch process than an interactive one. Thus, to address the above, this paper presents the origin, architecture, and performance evaluation of a green supercomputer in a *desktop* box, i.e., a 12-processor desktop cluster in an oversized pizza box and with a peak power envelope of only 185 watts when running Linpack.[1]

The remainder of the paper is organized as follows. Section 2 briefly explains the origin of the aforementioned green supercomputer in a desktop box, specifically the Orion Multisystems DT-12. Section 3 provides an architectural overview of the low-power DT-12. Section 4 presents an initial performance evaluation of the DT-12 (circa 2004) versus a typical SMP "desktop" server (also circa 2004), specifically, a Dell PowerEdge 2650 server. Finally, Section 5 concludes our work.

2 Background

The roots of the Orion Multisystems DT-12 can be traced back to the energy-efficient Green Destiny cluster [5, 14, 3], which leveraged Beowulf cluster technology (such as commodity hardware, Linux, and MPI) while being energy conscious (i.e., power awareness at design time) in order to improve the reliability and availability of compute cycles. The basic idea behind the DT-12 was to deliver the above advantages of Green Destiny but in the form factor of a "pizza box," thus filling the widening performance gap between supercomputers and traditional PC workstations.

Back in 2001, we observed that supercomputers were becoming less efficient with respect to both power and space consumption. For example, though the performance on our n-body code that simulates galaxy formation increased 2000-fold from the early 1990s to the early 2000s, the performance-per-watt and performance-per-square-foot only improved by 300-fold and 60-fold, respectively. This has resulted in the construction of massive datacenters with exotic cooling facilities (and even, entirely new buildings) to house these supercomputers, thus leading to an extraordinarily high total cost of ownership.

The main reason for this inefficiency has been the exponentially increasing power requirements of compute nodes, i.e., Moore's Law for Power Consumption [5, 3, 4, 7]. When nodes draw more power, they must be spaced out and aggressively cooled.[2] Our own empirical data as well as unpublished empirical data from a leading vendor indicates that the failure rate of a compute node *doubles* with every $10°$C ($18°$F) increase in temperature above $75°$F, and temperature is proportional to power density. Thus, the supercomputers in the TOP500 List require exotic cooling facilities; otherwise, they would be so unreliable (due to overheating-induced failures) that they would be unavailable for use by the application scientist.

To address the above, we started the Supercomputing in Small Spaces (http://sss.lanl.gov/) project in late 2001 and identified low-power building blocks to construct our energy-efficient Green Destiny [5, 14, 3], a 240-processor Beowulf cluster that fit in a telephone booth (i.e., a footprint of five square feet) and sipped only 3.2 kilowatts when running diskless, i.e., two hairdryers, but with performance slightly better than a 172-processor Cray T3E 900 (circa 11/2001 TOP500 List) when running Linpack. Green Destiny provided reliable compute cycles with no unscheduled downtime from April 2002 to April 2004, all while sitting in an $85°$F dusty warehouse at 7,400 feet above sea level — thus illustrating its ability to be moved out of the datacenter and into an "office space" that did not have any special cooling facilities. This transformation from datacenter cluster to office cluster ultimately led to a subsequent transformation into a *desktop* cluster, as embodied by the Orion Multisystems DT-12.

3 Architectural Overview

The Orion Multisystems DT-12, as shown in Figure 1, is a personal *desktop* cluster workstation that contains 12 individual x86 compute nodes in a 24" x 18" x 4" (or one cubic foot) pizza-box enclosure. Collectively, these nodes provide the horsepower of a small supercomputer, the administrative ease of a single-processor computer,[3] and the low noise, heat, and power draw of a conventional desktop.

Figure 1. The DT-12 Personal Desktop Cluster Workstation.

Each compute node contains a Transmeta Efficeon processor running the Linux operating system and Transmeta's

[1] For reference, a Dell PowerEdge 2650 desktop server with dual 2.2-GHz Intel Xeons consumes nearly 220 watts when running Linpack.

[2] Perhaps a more insidious problem to the above inefficiency is that the reliability of these systems continues to decrease as traditional supercomputers continue to aggregate more processors together.

[3] Booting the DT-12 amounts to depressing a single power switch, and in just over a minute, the entire single-system image is then up and ready to run a parallel job.

power-aware LongRun2 software, its own memory and Gigabit Ethernet interface, and optionally, its own hard disk drive. The nodes share a power supply, cooling system, and external 10-Gigabit Ethernet network connection. (The head node provides an interface to the end user.)

In short, the DT-12 arguably exports the utmost in simplicity with only one power plug, one power switch, one monitor connection, one keyboard, and one mouse. At load, it achieves 14 Gflops on Linpack while drawing only 185 watts of power, i.e., less than two 100-watt light bulbs.

3.1 Hardware

The DT-12 is a cluster of 12 x86-compatible nodes linked by a switched Gigabit Ethernet fabric.[4] The cluster operates as a single computer with a single power switch and a single-system image rapid-boot sequence, which allows the entire system to come on-line in just over a minute.

In the DT-12, one node functions as the head node, providing an interface to the end user and controlling jobs throughout the cluster. The other nodes, referred to as compute nodes, are available to the head node for parallel computing. When idle, the head node can also act as a compute node.

The DT-12 plugs directly into a standard 15-A office outlet with no special cooling or power requirements. The included I/O board provides video, keyboard and mouse, serial port, USB, and fan control. The DT-12 also provides a DVD/CD-RW and one 3.5" hard drive on the head node. The board can accommodate one 2.5" hard-disk drive per each of the other nodes although disk drives on the compute nodes are optional.

3.2 Software

With the single-system image rapid-boot sequence from Orion Multisystems, the DT-12 appears as monolithic as possible to the end user, e.g., logging into individual compute nodes is typically unnecessary. The system software accomplishes this by providing the same Linux kernel on all nodes via a single OS installation on the head node that is shared among all compute nodes. Each compute node then locally runs commonly used cluster tools such as rsh, Sun Grid Engine, and MPICH2.

4 Experimental Results

In this section, we present an initial performance evaluation of the DT-12 from 2004 and compare it to a high-end development platform in use from 2004, specifically the Dell PowerEdge 2650 SMP. We chose this comparison for several reasons. First, with Beowulf clusters having become shared datacenter resources, we argue that application programmers have turned to high-end *desktop* server platforms like the Dell PowerEdge 2650 in order to develop their parallel codes. Such a desktop SMP machine can fit on one's desk without the need for special wall outlets or cooling facilities, just like the similarly sized Orion Multisystems DT-12. Second, in addition to the DT-12 having similar dimensions to the Dell PowerEdge 2650, is also uses a similar amount of power. Third, the price of each system was four digits, i.e., approximately $9,000 for the DT-12 and $5,000 for the PowerEdge 2650.

Platform	DT-12	PowerEdge 2650
SPECint2000	526	792
SPECfp2000	358	726

Table 2. SPEC CPU2000 Results

Our Dell PowerEdge 2650 is a dual-processor 2U chassis machine, configured with dual 2.2-GHz Xeon processors, 1.5-GB memory, and a Fujitsu 18.4-GB 15000 RPM U160 SCSI drive. Each Xeon processor used hyper-threading for a total of 4 virtual processors. A quick comparison of the DT-12 and the PowerEdge 2650 in Table 1 illustrates some of the key hardware differences between the two developmental platforms. The dimensions of the two platforms are nearly identical, and the power draws at load (i.e., Linpack) are comparable.

We conducted many experiments to show that a personal desktop cluster workstation that is energy efficient, such as the Orion Multisystems DT-12, can provide a more suitable development platform for parallel codes versus the typical SMP workstation. More importantly, we demonstrate that the DT-12 can deliver green supercomputing in a *desktop* box. Our benchmarks included SPEC CPU2000, HPL, NAS MPI, STREAMS MPI, ttcp, bonnie, and mpi-io-test.

4.1 Processor Performance

As an initial performance characterization of the Efficeon processor used in the DT-12, we began our experiments with the SPEC CPU2000 benchmarks [11] using an Intel compiler (version 8.1) and with the baseline optimization options `-O3 -xW -ipo`. Table 2 shows our baseline results. The DT-12 achieved 526 for SPECint2000 and 358 for SPECfp2000 while the numbers for the PowerEdge 2650 were 792 and 726, respectively. If we had run the same customized (but proprietary) high-performance code-morphing software on the DT-12 as we did on Green Destiny, the floating-point performance would have improved by about 50% to the neighborhood of 535 for SPECfp2000.

[4] Linking DT-12 systems together can then be achieved via the external 10-Gigabit Ethernet interface.

Platform	Orion Multisystems DT-12	Dell PowerEdge 2650
Dimensions	24(W) x 18(D) x 4(H) (in) 1 cubic foot	19(W) x 26(D) x 3.5(H) (in) $\sim$ 1.02 cubic feet
Power at Load Processors	185.3 watts Twelve 1.2-GHz Efficeon CPUs	217.0 watts Two 2.2-GHz Xeon CPUs
Memory	1 GB/Processor	1.5 GB
Network Interfaces	Gigabit Ethernet NIC/Processor	Dual Gigabit Ethernet NIC
Storage	80-GB 5400 RPM IDE (Head) 20-GB 4200 RPM IDE (Other)	18.4-GB 15000 RPM SCSI drive

Table 1. Experimental Hardware Configurations

According to the SPEC measurements, the integer performance of a 1.2-GHz Efficeon processor is roughly equivalent to a 1.5-GHz Pentium 4 (i.e., between 515 and 534). However, the floating-point performance of the Efficeon processor does not keep up with the 1.5-GHz Pentium 4 (i.e., between 543 and 549, compared to the Efficeon's 358) although it likely would have if Transmeta had used the prototypical high-performance code-morphing software that was originally developed for Green Destiny and tailored towards iterative scientific codes.

Our next test, HPL [6], is a freely available software package that implements the Linpack benchmark. It solves a (random) dense linear system in double precision (64-bits) arithmetic on distributed-memory computers. Using HPL requires an MPI 1.1 compliant implementation and either the Basic Linear Algebra Subprograms (BLAS) or the Vector Signal Image Processing Library (VSIPL). Generally, HPL is considered scalable with respect to the number of processors used during testing since its overall performance is mostly attributed to the system's CPU.

For HPL, the DT-12 delivers 14.17 Gflops while the Dell PowerEdge 2560 achieves 5.1 Gflops, or roughly three times slower than the DT-12. Since the DT-12 has twelve processors compared to the two processors in the PowerEdge 2650, we infer that a 2.2-GHz Xeon processor performs slightly more than twice as fast as a 1.2-GHz Efficeon in this application.

In our final CPU test, we ran experiments using the NAS Parallel Benchmarks (NPB) [9]. Collectively, they mimic the computation and data-movement characteristics of applications in computational fluid dynamics (CFD). NPB is based on Fortran and MPI. These implementations, which are intended to be run with little or no tuning, approximate the performance that a typical user can expect from a portable parallel program in a distributed-memory computing system.

Table 3 and Table 4 show the results from the class A and class B workload, respectively. Due to process restrictions in some tests, we only obtained results for certain numbers of processes. For example, SP and BT require that the number of processors be a square of an integer, therefore we have results from 1, 4, and 9 processors. In Table 4, FT was not able to complete for the PowerEdge 2650 (even after waiting numerous minutes). We attribute this to insufficient memory and heavy disk swapping.

In general, as the number of processes increases, the DT-12 scales nearly linearly in the LU, BT, EP, and CG tests. The FT, MG, SP, and IS tests do scale to some degree on the DT-12, although not linearly. However, because some of these tests are dependent on system components besides the processor, they are not expected to scale linearly. In contrast, the PowerEdge 2650 does not achieve linear scalability on any of the 8 NPB tests and would provide poor feedback on demonstrating whether a parallel code is achieving an expected linear speedup. Running more processes than the number of actual processors on the dual-processor PowerEdge 2650 does not allow for any useful scalability testing.

4.2 Memory Performance

The STREAM benchmark [12] measures the sustainable memory bandwidth and the corresponding computation rate for simple vector kernels. The STREAM benchmark has four different components which are summarized in Table 5.

We used the MPI version of STREAM to test the scalability of the memory subsystem. Each test was compiled with MPICH2 [8] and run three times with the best run used in our results. We chose to use the best run for this benchmark since memory testing is particularly sensitive to any OS interrupts. Figure 2a shows that the DT-12 achieves linear scalability while the PowerEdge 2650 in Figure 2b struggles to achieve any speedup. While it is a two-way SMP and each processor has hyper-threading (thereby having four virtual processors), we hardly see any speedup between 1 to 4 processors on the PowerEdge 2650.

Class A Workload										
	DT-12					PowerEdge 2650				
	1	4	8	9	12	1	4	8	9	12
FT	143.29	287.84	543.37			286.14	361.20	347.42		
MG	97.42	359.49	1180.40			324.11	273.49	313.86		
SP	98.95	319.74		458.26		188.12	213.08		194.59	
LU	179.12	779.28	1521.10			325.47	419.88	419.46		
BT	268.56	946.81		1472.29		540.76	810.68		841.85	
IS	11.23	17.41	19.10			24.34	30.75	28.93		
EP	3.46	13.80	27.52	31.23	41.11	4.87	17.18	17.61	17.61	17.47
CG	70.27	260.61	526.59			283.42	249.97	221.15		

Table 3. NAS Parallel Benchmarks – Class A

Class B Workload								
	DT-12				PowerEdge 2650			
	4	8	9	12	4	8	9	12
FT	354.70	649.08			N/A	N/A	N/A	N/A
MG	605.56	1203.56			283.09	346.66		
SP	365.94		703.20		208.64		214.50	
LU	681.74	1466.66			374.46	417.98		
BT	1029.67		2018.45		814.64		858.07	
IS	21.15	32.97			29.45	28.33		
EP	14.72	29.52	33.26	44.29	17.22	17.28	17.53	17.78
CG	172.96	357.51			72.13	191.24		

Table 4. NAS Parallel Benchmarks – Class B

4.3 I/O Performance

In order to test sequential I/O performance, we used bonnie [2]. Bonnie performs a series of tests on a file of unknown size. It does sequential output using the putc() stdio macro, writes blocks using the write() command, and rewrites blocks of size 16 KB (reads the blocks into memory, dirties them, and writes them back). It does sequential input using getc() and the read() command. The test results can be significantly affected by the amount of system memory available since writes are buffered on most file systems.

Both of our test platforms used ext3, which certainly buffers write requests whenever memory is available. We also used the largest file size possible on the local file system (2047 MB) in our tests. Since this is a sequential I/O test, the DT-12 can only use a single processor and a single memory (1 GB) for file system buffering. The PowerEdge 2650 has both its processors available and 1.5 GB of memory for I/O buffering. Because the file size is 2047 MB, more of the file I/O can be buffered on the PowerEdge 2650, resulting in higher I/O performance (because it is mostly writing to memory and not to the actual hard disk). For the DT-12, we also tested I/O access to a node's local storage through ext3 and to remote storage on another node using NFS. On the PowerEdge 2650, testing NFS performance does not make sense as both processors have access to their local disks.

The results for sequential output and sequential input are shown in Figure 3. As expected, there are certain cases, for instance, the block write and the block read, that are much better for the PowerEdge 2650 due to the increased buffering capabilities from having more memory per processor. When bonnie rewrites data through NFS on the DT-12, we see that its performance really suffers due to fetching data over the network and rewriting it.

We took our parallel I/O test, mpi-io-test, from the PVFS2 test suite. mpi-io-test simply writes 16 MB per processor into a shared file and reads it back through the MPICH2 ROMIO [13] interface. We set-up the PVFS2 file system [10], a next-generation parallel file system for Linux clusters, on the nodes to attain the high possible bandwidth. For the DT-12 platform, we configured all 12 nodes as I/O servers, with one node doubling as the metadata server. For the PowerEdge 2650 platform, we configured two processes as I/O servers, with one doubling as the metadata server.

Test	Operation	Bytes/Iteration	Floating Point Operations/Iteration
COPY	a(i) = b(i)	16	0
SCALE	a(i) = q*b(i)	16	1
SUM	a(i) = a(i)+b(i)	24	1
TRIAD	a(i) = b(i)+q*c(i)	24	2

Table 5. STREAM Tests

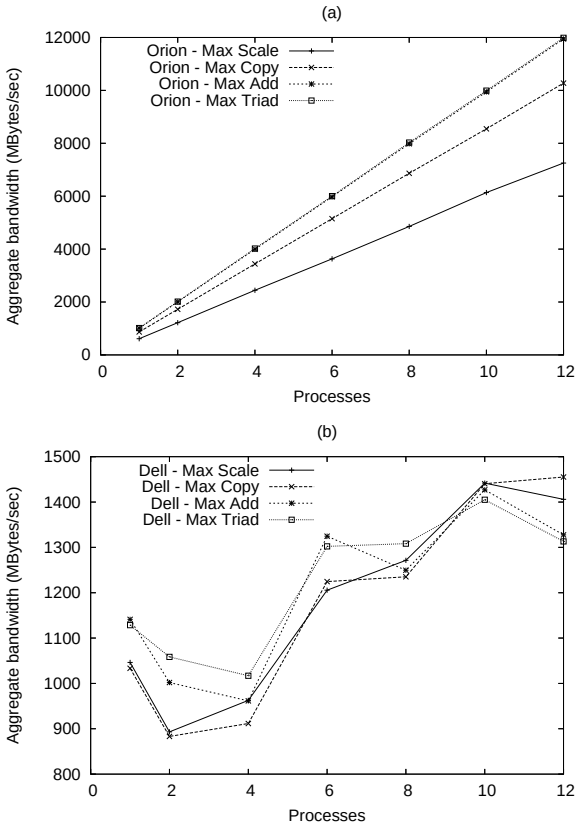

Figure 2. STREAM Results: (a) DT-12 and (b) PowerEdge 2650.

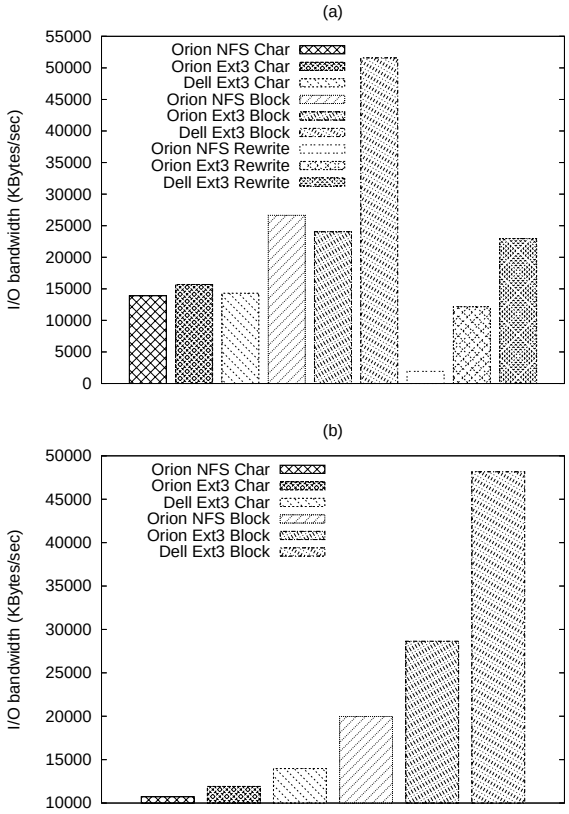

Figure 3. Results from bonnie: (a) Sequential Output and (b) Sequential Input.

Our results, shown in Figure 4, point to the scalability of the Orion DT-12. As we scaled up the number of processors, we saw nearly linear speedup in I/O bandwidth for the DT-12. In contrast, the PowerEdge 2650 I/O bandwidth is pretty constant between 1 and 12 processes. Thus, in this case, a personal desktop cluster workstation like the DT-12 is very capable for parallel I/O development (i.e., writing parallel codes that use MPI-IO).

Cluster Name	CPU	Cluster Topology	Memory (GB)	HPL Perf. (GFlops)	$\text{Power}_{\text{HPL}}$ (W)	Perf/Power (MFlops/W)
PowerEdge 2650	2.2-GHz Intel Xeon	1×2P	1.50	5.1	217.0	23.50
DT-12	1.2-GHz Transmeta Efficeon	12×1P	12.00	14.17	185.3	76.47

Table 6. Performance, Power, and Performance/Power

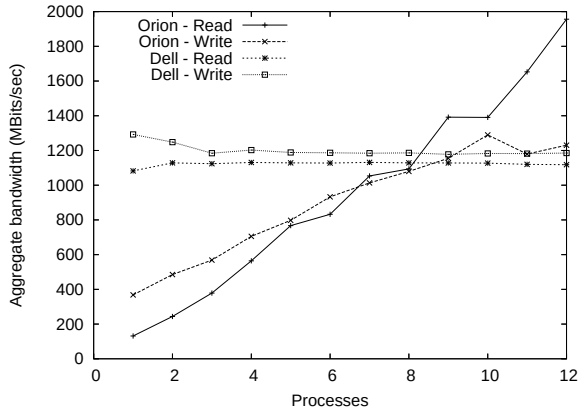

Figure 4. Results from mpi-io-test.

4.4 Power Efficiency

A personal desktop cluster workstation must not have special power requirements (e.g., it should be able to be plugged into a normal wall socket). We evaluated the power consumption of the DT-12 versus the PowerEdge 2650 using the HPL benchmark. In order to measure the system's power consumption, we used a Yokogawa digital power meter that was plugged into the same power strip as the system. The power meter continuously sampled the instantaneous wattage at a rate of 50 kHz and delivered its readings to a profiling computer.

Table 6 shows the results of HPL running with the power meter. With regard to power-consuming components, the DT-12 has 12 processors, 12 GB of memory, 12 hard disks, and an internal network compared to the PowerEdge's two processors and 1.5 GB of memory, a single disk, and no internal network. Surprisingly, despite having many more power-consuming components, the DT-12 actually consumed *less* power than the PowerEdge 2650 when running Linpack. Much of the energy savings can be attributed to the Transmeta Efficeon processors. In addition, the DT-12 delivered a performance/power ratio that was over three times better than the PowerEdge 2650.

5 Conclusion

In this paper, we presented a case for a green supercomputer in a *desktop* box. The particular incarnation that we chose to evaluate was the Orion Multisystems DT-12 (circa 2004), which was based on the energy-efficient Green Destiny cluster that featured in *The New York Times* in 2002. The DT-12, a 12-node personal desktop supercomputer, achieved 14 Gflops on Linpack while sipping only 185 watts at load and occupying a mere one cubic foot in space, i.e., an oversized pizza box *on the desktop*.

We also evaluated the suitability of the DT-12 as a scalable platform for parallel code development. Our experiments showed that this personal desktop cluster workstation was a more useful tool for developing parallel codes and running parallel applications than a high-end SMP workstation. Most of our experiments using the DT-12 demonstrated linear scalability with respect to processor, memory, and I/O. In contrast, the PowerEdge 2650, our reference "desktop" SMP platform, did not fare as well, achieving limited or no scalability as we increased the parallelism of our experiments.

References

[1] D. J. Becker, T. Sterling, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Packer. BEOWULF: A parallel workstation for scientific computation. In *Proc. of the 1995 International Conference on Parallel Processing (ICPP)*, pages 11–14, August 1995.

[2] `http://www.textuality.com/bonnie/`.

[3] W. Feng. Making a case for efficient supercomputing. *ACM Queue*, 1(7):54–64, October 2003.

[4] W. Feng. The importance of being low power in high-performance computing. *Cyberinfrastructure Technology Watch*, 1(3):12–20, August 2005.

[5] W. Feng, M. Warren, and E. Weigle. The bladed beowulf: A cost-effective alternative to traditional beowulfs. In *Proc. of the IEEE Int'l Conf. on Cluster Computing*, September 2002.

[6] `http://www.netlib.org/benchmark/hpl/`.

[7] C. Hsu and W. Feng. A power-aware run-time system for high-performance computing. In *Proc. of SC2005: The*

IEEE/ACM International Conference on High-Performance Computing, Networking, Storage, and Analytics, November 2005.

[8] `http://www-unix.mcs.anl.gov/mpi/mpich2/`.

[9] `http://www.nas.nasa.gov/Software/NPB/`.

[10] `http://www.pvfs.org/pvfs2/`.

[11] `http://www.spec.org/osg/cpu2000/`.

[12] `http://www.cs.virginia.edu/stream/ref.html/`.

[13] R. Thakur, W. Gropp, and E. Lusk. On implementing MPI-IO portably and with high performance. In *Proceedings of the Sixth Workshop on Input/Output in Parallel and Distributed Systems*, pages 23–32, 1999.

[14] M. Warren, E. Weigle, and W. Feng. High-density computing: A 240-processor beowulf in one cubic meter. In *Proc. of SC2002: High-Performance Networking & Computing Conf.*, November 2002.