SC2002: High-Performance Networking and Computing Conference (SC2002),
Baltimore, Maryland, November 2002. LA-UR 02-5194

High-Density Computing: A 240-Processor Beowulf in One Cubic Meter

Michael S. Warren[†] msw@lanl.gov
Eric H. Weigle[‡] ehw@lanl.gov
Wu-Chun Feng[‡] feng@lanl.gov

Los Alamos National Laboratory
Los Alamos, NM 87545

Abstract

We present results from computations on Green Destiny, a 240-processor Beowulf cluster which is contained entirely within a single 19-inch wide 42U rack. The cluster consists of 240 Transmeta TM5600 667-MHz CPUs mounted on RLX Technologies motherboard blades. The blades are mounted side-by-side in an RLX 3U rack-mount chassis, which holds 24 blades. The overall cluster contains 10 chassis and associated Fast and Gigabit Ethernet switches. The system has a footprint of 0.5 meter^2 (6 square feet), a volume of 0.85 meter^3 (30 cubic feet) and a measured power dissipation under load of 5200 watts (including network switches). We have measured the performance of the cluster using a gravitational treecode N-body simulation of galaxy formation using 200 million particles, which sustained an average of 38.9 Gflops on 212 nodes of the system. We also present results from a three-dimensional hydrodynamic simulation of a core-collapse supernova.

Keywords: Beowulf, cluster, blade server, RLX, Transmeta, code morphing, VLIW, performance-per-square-foot, MIPS-per-watt

1 Introduction

In 1991 a Cray C90 vector supercomputer occupied about 600 square feet and required 500 kilowatts of power. Over the past decade, machines have become much faster, but they have also grown much larger. The ASCI Q machine at Los Alamos will require 17,000 square feet of floor space and 3 megawatts of power. Performance has gone up a factor of 2000 since the C90, but performance per square foot has only grown a factor of 65. When we restrict this comparison to CMOS logic in distributed memory parallel supercomputers, it is even more striking. In 1991 the 512 processor Intel Delta could sustain performance close to that of the C90, but dissipated only 53 kilowatts of power and required 200 square feet of floor space, so performance per sq. foot among parallel computers has only increased a factor of 20 in a decade. This growth has led several institutions to design whole new buildings just to support these ever-larger machines. However, this trend must end soon. Space, power and performance constraints all place limits on the size to which supercomputers can grow.

Our cluster is named Green Destiny after the mythical sword in the movie *Crouching Tiger, Hidden Dragon*. As shown in Section 5, our computational density of 6500 Mflops per square foot is over 10 times better than the current generation of supercomputers, and also exceeds the computational density of blade servers based on the Intel Tualatin processor (such as the Nexcom HiServer) or a conventional Beowulf cluster using 1U cases by about a factor of 1.5 to 2. A bladed Beowulf such as Green Destiny can reduce the total cost of ownership by a significant factor due to reduced costs for space, power and cooling, as well as simplifying the maintenance of the cluster. Further analysis of the total cost of ownership of Green Destiny is available in [1].

[†]Theoretical Astrophysics, Mail Stop B210
[‡]Advanced Computing Laboratory, Mail Stop D451

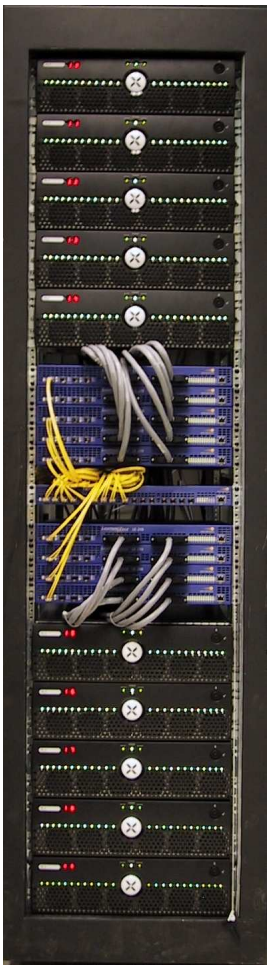

Figure 1: The Green Destiny cluster

2 Green Destiny: A Bladed Beowulf

In a relatively short time, Beowulf clusters [2] have filled a wide niche in high-performance computing. The Beowulf architecture first garnered attention in the supercomputing community at SC ’96 with the Loki and Hyglac Pentium Pro clusters, which won a Gordon Bell price/performance prize in 1997 [3], and again in 1998 with the Avalon cluster using the Alpha microprocessor [4]. The primary advantage of Beowulf clusters is often thought to be cost, but even more important is their convergent architecture that supports a standard software environment, allowing applications to run on many processor types over multiple generations of machines [5]. With the project described here, we again demonstrate that a well-designed portable message-passing code can take advantage of new technologies within the framework of the Beowulf architecture, GNU development tools, and Linux operating system.

The RLX System 324 comes in three sets of easy-to-integrate pieces: the 3U system chassis (Figure 2), 24 ServerBlades (Figure 4), and bundled cables for communication and power. The system chassis fits in an industry-standard 19-inch rack cabinet and measures 5.25 high, 17.25 inches wide, and 25.2 inches deep. It features two hot-pluggable 450-watt power supplies that provide power load-balancing and auto-sensing capability for added reliability. Its system midplane integrates the system power, management, and network signals across all RLX ServerBlades. The ServerBlade connectors on the midplane completely eliminate the need for internal system cables and enable efficient hot-pluggable ServerBlade support.

The chassis also includes two sets of cards: the Management Hub card and the Network Connect cards. The former provides connectivity from the management network interface of each RLX ServerBlade to the external world. Consolidating 24 ServerBlade management networks in the hub card to one “RJ45 out” enables system management of the entire chassis through a single standard Ethernet cable. The latter provides connectivity to the public and private network interfaces of each RLX ServerBlade.

Figure 2: The RLX System 324

Figure 3: At the Supercomputing in Small Spaces project launch. Back row from left to right: Mark Gardner, Michael Warren, Gordon Bell, Eric Weigle, Chris Hipp, Bill Feiereisen, Robert Bedichek. Front row: Wu Feng, Linus Torvalds

2.1 Hardware

The Green Destiny cluster consists of 240 compute nodes. 216 of the nodes contain a 667-MHz Transmeta TM5600 CPU (100% x86 compatible), 128-MB DDR SDRAM, 512-MB SDRAM, 20-GB hard disk, and 100-Mb/s network interface, and a tenth chassis with 24 633-MHz processors. (The tenth chassis was purchased six months earlier than the rest of the system). We connect each compute node to a 100-Mb/s Fast Ethernet switch, which are in turn connected to a top-level gigabit Ethernet switch, resulting in a cluster with a 1-level tree topology. There is an additional ethernet port available on each blade, so channel bonding could be used to double the available bandwidth. The server blades themselves can support up to 1152 Mbytes of memory and 160 Gbytes of disk. The total price of the cluster was $335k, or $1400 per node.

The cluster (see Figure 1) is currently mounted in a standard rack with dimensions of 84x24x36 inches (42 cubic feet, 6 square feet, 1.19 cubic meters, 0.558 square meters). The cluster would fit in a rack with less height and depth, with measurements 72x24x30 inches (30 cubic feet, 5 square feet, 0.85 cubic meters, 0.465 square meters). We use the more conservative 6 square feet in the computational density calculations below. The power measurements, which include all 240 processors and all of the network switches, were taken from the APC Smart-UPS systems that the cluster was attached to using Linux `apcupsd` tool which read the serial data stream from the UPS units while the treecode was running.

It is important to note that the cluster is fairly bal-

anced in terms of computation, memory, communication and disk storage. One could improve computational density and power consumption significantly by running a diskless system or using special-purpose processors, but this would greatly restrict the functionality of the system. Also, advertised power consumption figures often do not take into account the power required for network switches, which becomes quite significant when processor power dissipation is small. As it stands, the Green Destiny cluster presents a very familiar and completely x86 compatible general purpose parallel computing environment. Green Destiny is part of the Supercomputing in Small Spaces project at Los Alamos. The initial project launch was attended by a number of notable computing personalities (Figure 3).

2.2 The Transmeta Crusoe TM5600

The Crusoe family of processors has emerged from a different approach to microprocessor design. In contrast to the traditional transistor-laden, and hence, power-hungry CPUs from AMD and Intel, the Crusoe CPU is fundamentally software-based with a small hardware core. The Transmeta Crusoe TM5600 CPU consists of a VLIW hardware engine surrounded by a software layer called Code Morphing. This Code Morphing software presents an x86 interface to the BIOS, operating system (OS), and applications.

Due to the complexity of the x86 instruction set, the decode and dispatch hardware in superscalar out-of-order x86 processors (such as the Pentium III) require a large number of power-hungry transistors that increase the die size significantly. The large difference in the number of transistors also corresponds to a large difference in heat dissipation. At idle, a Transmeta TM5600 CPU in our Green Destiny cluster generates 7 watts of thermal power. At load, the Transmeta TM5600 generates approximately 15 watts. Using our measurements on the entire integrated 240-processor cluster, including network switches, each blade accounts for 22 watts of power dissipation while running code. Equivalent measurements we have made on an 18-processor Nexcom HiServer using 1133 Mhz Intel Pentium III processors with a single 3com Gigabit Ethernet switch, shows that each of the Nexcom blades dissipates 70 watts (see Table 4). Because of this substantial difference, the TM5600 requires no active cooling. Although no detailed statistics have been published, informal reports indicate that the failure rate for CMOS CPUs doubles for every 10 degree Celsius increase in temperature.

At the end of 2001, the fastest Crusoe CPU (i.e., TM5800) at load dissipated less than 1.6 watts with a 366-MHz TM5800 and less whereas a Pentium III (and most definitely, a Pentium 4) processor can heat to the point of failure if it is not aggressively cooled. Consequently, as in our Bladed Beowulf (24 CPUs in a 3U), Transmetas can be packed closely together with no active cooling, thus resulting in a savings in the total cost of ownership with respect to reliability, electrical usage, cooling requirements, and space usage.

By demonstrating that "equivalent" microprocessors can be implemented as software-hardware hybrids, Transmeta's Code Morphing technology dramatically expands the microprocessor design space. For example, upgrades to the software portion of the chip can now be rolled out independently from the chip. More importantly, decoupling the hardware design of the chip from the system and application software that use the chip frees hardware designers to evolve and replace their designs without perturbing legacy software.

3 N-body methods

N-body methods are widely used in a variety of computational physics algorithms where long-range interactions are important. Several methods have been introduced which allow N-body simulations to be performed on arbitrary collections of bodies in time much less than $O(N^2)$, without imposition of a lattice [6, 7]. They have in common the use of a truncated expansion to approximate the contribution of many bodies with a single interaction. The resulting complexity is usually determined to be $O(N)$ or $O(N \log N)$, which allows computations using orders of magnitude more particles. These methods represent a system of N bodies in a hierarchical manner by the use of a spatial tree data structure. Aggregations of bodies at various levels of detail form the internal nodes of the tree (cells). These methods obtain greatly

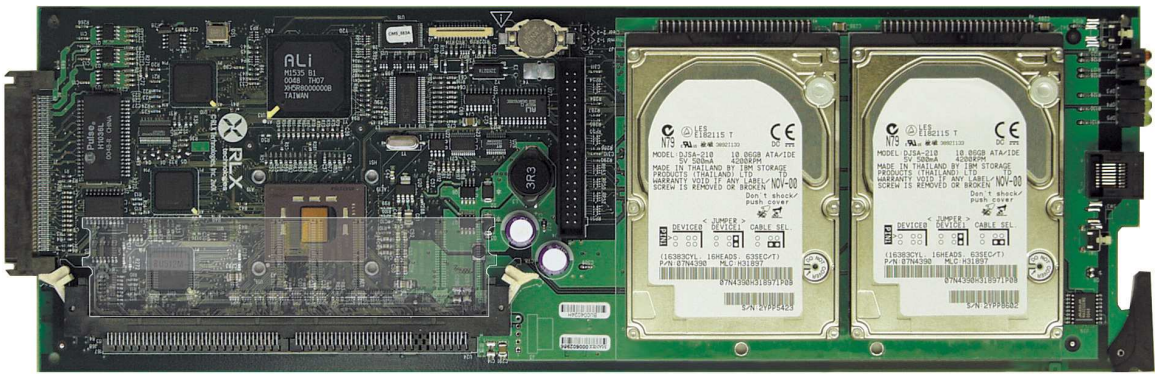

Figure 4: The RLX ServerBlade

increased efficiency by approximating the forces on particles. Properly used, these methods do not contribute significantly to the total solution error. This is because the force errors are exceeded by or are comparable to the time integration error and discretization error.

Using a generic design, we have implemented a variety of modules to solve problems in galactic dynamics [8] and cosmology [9] as well as fluid-dynamical problems using smoothed particle hydrodynamics [10], a vortex particle method [11] and boundary integral methods [12].

3.1 The Hashed Oct-Tree Library

Our parallel N-body code has been evolving for over a decade on many platforms. We began with an Intel ipsc/860, Ncube machines, and the Caltech/JPL Mark III [13, 8]. This original version of the code was abandoned after it won a Gordon Bell Performance Prize in 1992 [14], due to various flaws inherent in the code, which was ported from a serial version. A new version of the code was initially described in [15].

The basic algorithm may be divided into several stages. Our discussion here is necessarily brief. First, particles are domain decomposed into spatial groups. Second, a distributed tree data structure is constructed. In the main stage of the algorithm, this tree is traversed independently in each processor, with requests for non-local data being generated as needed. In our implementation, we assign a `Key` to each particle, which is based on Morton ordering. This maps the points in 3-dimensional space to a 1-dimensional list, which maintaining as much spatial locality as possible. The domain decomposition is obtained by splitting this list into N_p (number of processors) pieces. The implementation of the domain decomposition is practically identical to a parallel sorting algorithm, with the modification that the amount of data that ends up in each processor is weighted by the work associated with each item.

The Morton ordered key labeling scheme implicitly defines the topology of the tree, and makes it possible to easily compute the key of a parent, daughter, or boundary cell for a given key. A hash table is used in order to translate the key into a pointer to the location where the cell data are stored. This level of indirection through a hash table can also be used to catch accesses to non-local data, and allows us to request and receive data from other processors using the global key name space. An efficient mechanism for latency hiding in the tree traversal phase of the algorithm is critical. To avoid stalls during non-local data access, we effectively do explicit "context switching" using a software queue to keep track of which computations have been put aside waiting for messages to arrive. In order to manage the complexities of the required asynchronous message traffic, we have developed a paradigm called "asynchronous batched messages (ABM)" built from primitive send/recv functions whose interface is modeled after that of active messages.

All of this data structure manipulation is to support the fundamental approximation employed by

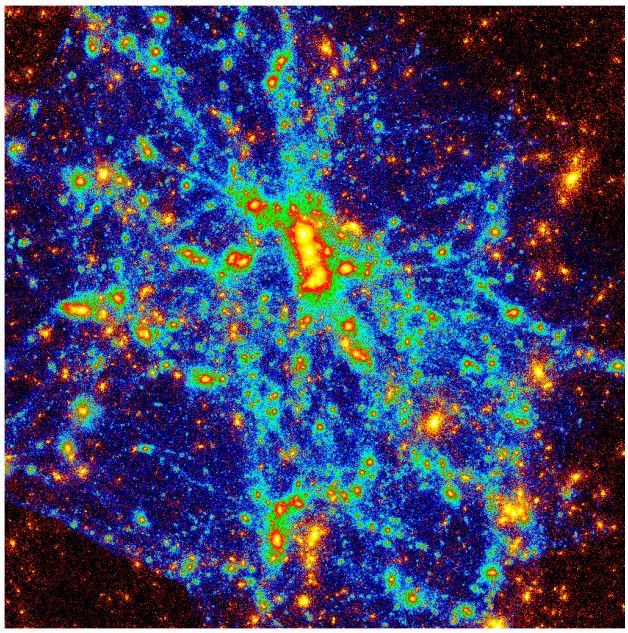

Figure 5: An intermediate Stage of a Gravitational N-body Simulation with 9.7 Million Particles, performed on the Green Destiny cluster. The overall simulation of 1000 timesteps with over 10^{15} floating point operations was completed in less than a day. The region shown is about 150 million light years across.

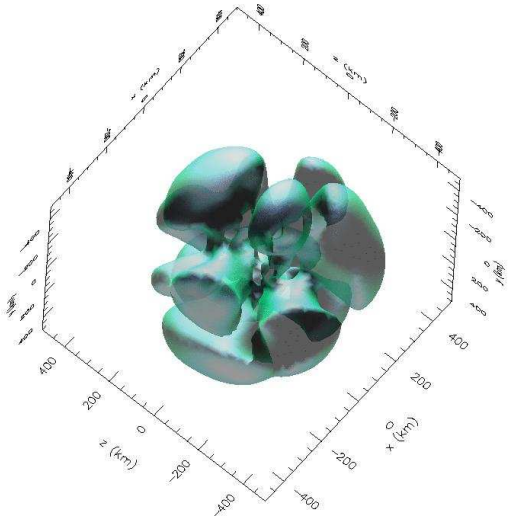

Figure 6: The isosurface of material with radial velocities of 1000km/s in a core collapse supernova. The isosurface outlines the outward moving convective bubbles. The open spaces mark the downflows. Note that the supernova figure was produced from simulation performed on the NERSC IBM SP, but further supernova simulations are underway on the Green Destiny Cluster.

treecodes:

$$\sum_j \frac{Gm_j\vec{d}_{ij}}{|d_{ij}|^3} \approx \frac{GM\vec{d}_{i,cm}}{d_{i,cm}^3} + \ldots, \qquad (1)$$

where $\vec{d}_{i,cm} = \vec{x}_i - \vec{x}_{cm}$ is the vector from $\vec{x}_i$ to the center-of-mass of the particles that appear under the summation on the left-hand side, and the ellipsis indicates quadrupole, octopole, and further terms in the multipole expansion. The monopole approximation, i.e., Eqn. 1 with only the first term on the right-hand side, was known to Newton, who realized that the gravitational effect of an extended body like the moon can be approximated by replacing the entire system by a point-mass located at the center of mass. Effectively managing the errors introduced by this approximation is the subject of an entire paper of ours [16].

4 Simulation Performance

In this section, we evaluate Green Destiny in several contexts. First, we use a gravitational microkernel benchmark based on the inner loop of our N-body code to evaluate raw performance of several processors. We also run a small-scale simulation to obtain a performance rating for Green Destiny which is directly comparable to a decade of similar measurements on most of the major supercomputer architectures. We then present some results from an N-body simulation of galaxy formation and a smoothed particle hydrodynamic simulation of supernova core-collapse.

The statistics quoted below are based on internal diagnostics compiled by our program. Essentially, we keep track of the number of interactions computed. We obtain optimal performance on the Transmeta processor by decomposing the reciprocal square

Processor	libm	Karp
533-MHz Alpha EV56	76.2	242.2
667-MHz Transmeta TM5600	128.7	297.5
933-MHz Transmeta TM5800	189.5	373.2
375-MHz IBM Power3	298.5	514.4
1133-MHz Intel Pentium III	292.2	594.9
1200-MHz AMD Athlon MP	350.7	614.0
2200-MHz Intel Pentium IV	668.0	655.5
1800-MHz AMD Athlon XP	609.9	951.9

Table 1: Mflops obtained on our gravitational microkernel benchmark. The first column uses the math library *sqrt*, the second column uses an optimization by Karp, which decomposes the reciprocal square root into a table lookup, Chebychev interpolation and Newton-Raphson iteration, which uses only adds and multiplies.

root function required for a gravitational interaction into a table lookup, Chebychev polynomial interpolation, and Newton-Raphson iteration, using the algorithm of Karp [17]. This algorithm uses only adds and multiplies, and requires 38 floating point operations per interaction. We *do not* use assembly language for any part of the code. The flop rates follow from the interaction counts and the elapsed wall-clock time. The flop counts are identical to the best available sequential algorithm. We *do not* count flops associated with decomposition or other parallel constructs. The reported times are for the entire application, including I/O, communication, program initialization, etc.

In Table 1 we present results from a benchmark based on the inner loop of our N-body code. These numbers provide an upper bound on the performance of the code on parallel machines using these processors. We typically sustain somewhere around 50% of the microkernel benchmark numbers for a production simulation on a parallel machine. The Transmeta performance is comparable to Intel processors at similar clock rates. Also note that we have measured the current generation TM5800 processor at 933 MHz, and obtain a performance increase. Overall, the Transmeta performance is about 1/3 to 1/2 that of the latest generation of Intel and AMD processors.

In Table 2 we show the performance of the cluster on a standard simulation problem which we have run on most of the major supercomputer architectures of the past decade. The problem is a spherical distribution of particles which represents the initial evolution of a cosmological N-body simulation. Overall, the performance of the full Green Destiny cluster is similar to that of a current-generation 128 processor IBM SP machine. We use these numbers below in the calculation of compute density per square foot.

4.1 A 10 million body simulation of galaxy formation

In November 2001, we ran a simulation with $9,753,824$ particles on the first single chassis (24 x 633 MHz processors) of our Bladed Beowulf for about 1000 timesteps. The latter half of the simulation was performed on the showroom floor of the SC 2001 conference. Figure 5 shows an image of this simulation. The simulation was of a spherical region of space 100 Mpc (Megaparsec) in diameter, a region large enough to contain a few hundred thousand typical galaxies. The region inside a sphere of diameter 100 Mpc was calculated at high mass resolution, while a buffer region of 50 Mpc with a particle mass 8 times higher was used around the outside to provide boundary conditions. The initial conditions were extracted from a 134 million point initial dataset, calculated using a a 512^3 point 3-D FFT, from a Cold Dark Matter power spectrum of density fluctuations. Overall, the simulation completed about 1.3×10^{15} floating-point operations sustaining a rate of 2.1 Gflops during the entire simulation. We repeated this simulation on the full Green Destiny cluster in July 2002, which completed the entire simulation in a single run of just 24 hours, while saving 80 Gbytes of raw data.

4.2 A 3 million body hydrodynamics simulation

Warren and collaborators are currently performing the first ever full-physics three-dimensional simulations of supernova core-collapse [18] as part of the DOE SCIDAC Supernova Science Center `http://www.supersci.org`. The largest simulation using 3 million particles was finished recently, and required roughly one month of time on a 256 processor IBM SP (Figure 6). We have successfully run

Site	Machine	Procs	Gflops	Mflops/proc
NERSC	IBM SP-3(375/W)	256	57.70	225.0
LANL	SGI Origin 2000	64	13.10	205.0
LANL	Green Destiny	212	38.9	183.5
SC '01	RLX System 324	24	3.30	138.0
LANL	Avalon	128	16.16	126.0
LANL	Loki	16	1.28	80.0
NAS	IBM SP-2(66/W)	128	9.52	74.4
SC '96	Loki+Hyglac	32	2.19	68.4
Sandia	ASCI Red	6800	464.9	68.4
Caltech	Naegling	96	5.67	59.1
NRL	TMC CM-5E	256	11.57	45.2
Sandia	ASCI Red	4096	164.3	40.1
JPL	Cray T3D	256	7.94	31.0
LANL	TMC CM-5	512	14.06	27.5
Caltech	Intel Paragon	512	13.70	26.8
Caltech	Intel Delta	512	10.02	19.6
LANL	CM-5 no vu	256	2.62	5.1

Table 2: Historical Performance of Treecode on Clusters and Supercomputers

the supernova code on the Green Destiny cluster, and obtained performance per node about 1/5 that of the IBM SP. We have not spent any time optimizing the SPH code or tuning various performance parameters on the Green Destiny machine, and we expect to improve per-processor performance to the range of 1/3 to 1/2 that of the SP.

5 Performance Metrics

Table 3 contains the fundamental raw data upon which our claim of superior power density is based. The machines considered are Green Destiny, the ASCI Red, White and Q machines, and the Intel Delta and Avalon cluster for historical comparison. Performance is directly measured for the treecode solving the N-body problem, except for the ASCI machines where performance is extrapolated from measured performance on smaller machines with the same architecture. Our extrapolations are optimistic for the White and Q machines, and actual performance measurements would probably be somewhat smaller.

Power and space are actual measurements for Green Destiny and Avalon, and are based on personal communications from system administrators and figures on the Web for the remaining systems. The power figures do not include the additional power necessary for cooling, but that should be a constant factor of the power dissipation for all of these air-cooled machines.

We see that the computational density of 6480, measured in Mflops per square foot for Green Destiny, exceeds that of the fastest supercomputers by a factor of 10-25. Other striking figures are the DRAM density, where we are a factor of 35 denser than the nearest competitor, and the disk density (almost 1 Tbyte per square foot). A fully populated Green Destiny cluster would reach disk and memory densities exceeding the not-yet-functional ASCI Q by a factor of 70 each.

In terms of power efficiency, measured in Mflops per watt, the RLX cluster is 3-5 times more efficient than the supercomputers. One should also note that our power density is nearing 1 kW per square foot, which is the maximum supported in most (all?) data centers, and similar to that of the Cray vector machines of a decade ago. This is likely the limit of

Machine	Intel Delta	Avalon	ASCI Red	ASCI White	ASCI Q	**Green Destiny**
Year	1991	1996	1996	2000	2002	2002
Performance (Gflop/s)	10.0	17.6	600	2500	8000	38.9
Area (feet^2)	200	120	1600	9920	17000	6
Power (kilowatts)	53	18.0	1200	2000	3000	5.2
DRAM (Gbytes)	8	35.8	585	6200	12000	150
Disk (Tbytes)	-	0.4	2	160	600	4.8
DRAM density (Mbytes/foot^2)	40	300	365	625	705	**25000**
Disk density (Gbytes/foot^2)	-	3.3	1.25	16	35	**800**
Compute density (Mflop/s/foot^2)	50	150	375	252	470	**6480**
Power density (watts/foot^2)	265	147	750	200	176	920
Power efficiency (Mflop/s/watt)	0.19	1.0	0.5	1.25	2.66	7.5

Table 3: Space and Power statistics for Green Destiny compared to a number of other supercomputers.

power density that can be reached without special engineering or water cooling.

In comparison to other blade servers using the Pentium III, the differences in computational density and power dissipation are much smaller, but still significant. Table 4 shows that the Transmeta TM5600 solution is about 50% more power efficient per flop than the Nexcom server per blade. A rack of 10 Nexcom chassis would dissipate twice as much power as Green Destiny, reaching a power density of 2 kW per square foot, but would provide 50% more computing capability than Green Destiny. Given that the authors of this paper have only measured two blade architectures at the present time, it remains to be seen if blade offerings from other vendors can significantly improve compute density.

Although it is conceivable that one could package the 80-100 1.8-2.5 GHz Athlons or Pentium IVs required to match our performance in a single rack, they would dissipate 12-15 kW, which could not be supported without expensive additional power and cooling infrastructure. Given the limitation of 1 kW per square foot, we estimate we would have 2-3x the computational density of a cluster constructed from 1U cases using the latest AMD or Intel processors.

6 Conclusion

Green Destiny turns out to be approximately 2x as expensive as a similarly performing traditional Beowulf cluster. Although the fundamental components of the system are the same as those used in commodity mass-market PCs and laptops, the system design and integration creates an added cost. It is possible that if blade servers become more popular, economies of scale could reduce the price, but they are unlikely to reach the price of a typical Beowulf based on off-the-shelf parts. However, there is more to price than just the cost of acquisition, and our experience indicates that a cluster such as Green Destiny may in the end be cheaper than the initial cost of acquisition may imply, due to simplified hardware maintenance and less infrastructure cost.

The disparity in power dissipation and compute density will increase over time as the voracious IA-64, IBM Power4 and Compaq EV8 processors appear. Transmeta is moving to even lower power while maintaining competitive performance. With the currently available TM5800, we could build a cluster with 50% better performance and less power consumption today.

Processor	Athlon	Pentium IV	Pentium III	TM5600
Clock (MHz)	1800	2200	1133	667
Performance (Mflop/s)	952	656	595	298
Power (watts/proc)	160	130	70	22
Power efficiency (Mflop/s/watt)	5.95	5.05	8.5	13.5

Table 4: Power statistics for the Transmeta TM5600 processors used in Green Destiny compared to the Intel Tualatins used in a Nexcom HiServer 318, and single conventional Pentium IV and Athlon nodes. Performance is measured with the gravity micro-kernel. Power was measured with an APC Smart-UPS, and includes the per-port power required for an ethernet switch.

The largest supercomputers are the first to confront the power barrier, but the scale at which this problem occurs will become smaller as time goes on. Some reasonable extrapolations of single processor power dissipations approach 1 kW over the next ten years. It is certain that supercomputer design and low-power design will become much the same over the next decade. We hope this paper has set the stage for the much more exciting architectural developments to come.

Acknowledgments

We would like to thank the DOE Los Alamos Computer Science Institute for supporting this project, IEEE/ACM SC 2001 and Los Alamos National Laboratory for providing a venue to garner visibility for an earlier version of the project, and Gordon Bell for his encouragement on this endeavor. Support for MSW was provided by the LANL Information Architecture Linux project. This work was supported by the DOE through contract number W-7405-ENG-36.

We thank Dean Gaudet of Transmeta for helping to optimize the N-body inner loop. Our thanks also go out to Chris Hipp of RLX Technologies and Randy Hinson of World Wide Packets. Chris Hipp, in particular, was instrumental in expediting the delivery of the necessary resources for us to conduct this research.

References

[1] W. Feng, M. S. Warren, and E. Weigle, "Honey, I shrunk the Beowulf!," in *Proceedings of ICPP 2002*, 2002.

[2] 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 *Proceedings of the 1995 International Conference on Parallel Processing (ICPP)*, pp. 11–14, 1995.

[3] M. S. Warren, J. K. Salmon, D. J. Becker, M. P. Goda, T. Sterling, and G. S. Winckelmans, "Pentium Pro inside: I. A treecode at 430 Gigaflops on ASCI Red, II. Price/performance of $50/Mflop on Loki and Hyglac," in *Supercomputing '97*, (Los Alamitos), IEEE Comp. Soc., 1997.

[4] M. S. Warren, T. C. Germann, P. S. Lomdahl, D. M. Beazley, and J. K. Salmon, "Avalon: An Alpha/Linux cluster achieves 10 Gflops for $150k," in *Supercomputing '98*, (Los Alamitos), IEEE Comp. Soc., 1998.

[5] G. Bell and J. Gray, "High performance computing: Crays, clusters and centers. what next?," Tech. Rep. MSR-TR-2001-76, Microsoft Research, 2001.

[6] J. Barnes and P. Hut, "A hierarchical O(NlogN) force-calculation algorithm," *Nature*, vol. 324, p. 446, 1986.

[7] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comp. Phys.*, vol. 73, pp. 325–348, 1987.

[8] M. S. Warren, P. J. Quinn, J. K. Salmon, and W. H. Zurek, "Dark halos formed via dissipationless collapse: I. Shapes and alignment of angular momentum," *Ap. J.*, vol. 399, pp. 405–425, 1992.

[9] W. H. Zurek, P. J. Quinn, J. K. Salmon, and M. S. Warren, "Large scale structure after COBE: Peculiar velocities and correlations of cold dark matter halos," *Ap. J.*, vol. 431, pp. 559–568, 1994.

[10] M. S. Warren and J. K. Salmon, "A portable parallel particle program," *Computer Physics Communications*, vol. 87, pp. 266–290, 1995.

[11] J. K. Salmon, M. S. Warren, and G. S. Winckelmans, "Fast parallel treecodes for gravitational and fluid dynamical N-body problems," *Intl. J. Supercomputer Appl.*, vol. 8, pp. 129–142, 1994.

[12] G. S. Winckelmans, J. K. Salmon, M. S. Warren, and A. Leonard, "The fast solution of three-dimensional fluid dynamical N-body problems using parallel tree codes: vortex element method and boundary element method," in *Seventh SIAM Conference on Parallel Processing for Scientific Computing*, (Philadelphia), pp. 301–306, SIAM, 1995.

[13] J. K. Salmon, P. J. Quinn, and M. S. Warren, "Using parallel computers for very large N-body simulations: Shell formation using 180k particles," in *Proceedings of 1989 Heidelberg Conference on Dynamics and Interactions of Galaxies* (A. Toomre and R. Wielen, eds.), New York: Springer-Verlag, 1990.

[14] M. S. Warren and J. K. Salmon, "Astrophysical N-body simulations using hierarchical tree data structures," in *Supercomputing '92*, (Los Alamitos), pp. 570–576, IEEE Comp. Soc., 1992.

[15] M. S. Warren and J. K. Salmon, "A parallel hashed oct-tree N-body algorithm," in *Supercomputing '93*, (Los Alamitos), pp. 12–21, IEEE Comp. Soc., 1993.

[16] J. K. Salmon and M. S. Warren, "Skeletons from the treecode closet," *J. Comp. Phys.*, vol. 111, pp. 136–155, 1994.

[17] A. H. Karp, "Speeding Up N-body Calculations on Machines without Hardware Square Root," *Scientific Programming*, vol. 1, pp. 133–140, 1993.

[18] C. L. Fryer and M. S. Warren, "Modeling core-collapse supernovae in three dimensions," *Ap. J. (Letters)*, vol. 574, p. L65, 2002.