International Conference on Parallel Processing (ICPP 2002), Vancouver, Canada, August 2002.

Honey, I Shrunk the Beowulf!*

W. Feng†, M. Warren‡, E. Weigle†
{feng, msw, ehw}@lanl.gov

† Computer & Computational Sciences Division
‡ Theoretical Physics Division

Los Alamos National Laboratory
Los Alamos, NM 87545

Abstract

In this paper, we present a novel twist on the Beowulf cluster — the Bladed Beowulf. Designed by RLX Technologies and integrated and configured at Los Alamos National Laboratory, our Bladed Beowulf consists of compute nodes made from commodity off-the-shelf parts mounted on motherboard blades measuring 14.7″× 4.7″× 0.58″. Each motherboard blade (node) contains a 633-MHz Transmeta TM5600™CPU, 256-MB memory, 10-GB hard disk, and three 100-Mb/s Fast Ethernet network interfaces. Using a chassis provided by RLX, twenty-four such nodes mount side-by-side in a vertical orientation to fit in a rack-mountable 3U space, i.e., 19″ in width and 5.25″ in height.

A Bladed Beowulf can reduce the total cost of ownership (TCO) of a traditional Beowulf by a factor of three while providing Beowulf-like performance. Accordingly, rather than use the traditional definition of price-performance ratio where price is the cost of acquisition, we introduce a new metric called ToPPeR: Total Price-Performance Ratio, where total price encompasses TCO. We also propose two related (but more concrete) metrics: performance-space ratio and performance-power ratio.

Keywords: Beowulf, cluster, blade server, RLX, Transmeta, code morphing, VLIW, NAS benchmarks, price-performance ratio, ToPPeR, performance-space ratio, performance-power ratio, n-body code, treecode.

1 Introduction

In a relatively short time, Beowulf clusters [9, 12] have revolutionized the way that scientists approach high-performance computing. In contrast to tightly-coupled supercomputers, Beowulfs primarily use commodity off-the-shelf (COTS) technologies to deliver computational cycles at the lowest price, where price is defined as the cost of acquisition. However, when price is defined as the total cost of ownership (TCO), the advantages of Beowulfs, while still apparent, are not as compelling due to the added costs of system integration, administration, and maintenance (although many software tools have become available to reduce the impact of these added costs).

In this paper, we present our novel "Bladed Beowulf" cluster. Designed by RLX Technologies and integrated and configured at Los Alamos National Laboratory, our Bladed Beowulf cluster consists of compute nodes made from COTS parts mounted on motherboard blades called RLX ServerBlades™ (see Figure 1). Each motherboard blade (node) contains a 633-MHz Transmeta TM5600™ CPU [5], 256-MB memory, 10-GB hard disk, and three 100-Mb/s Fast Ethernet network interfaces. Twenty-four such ServerBlades mount into a chassis, shown in Figure 2, to form a "Bladed Beowulf" called the RLX System 324™ that fits in a rack-mountable 3U space, i.e., 19″ in width and 5.25″ in height.[1]

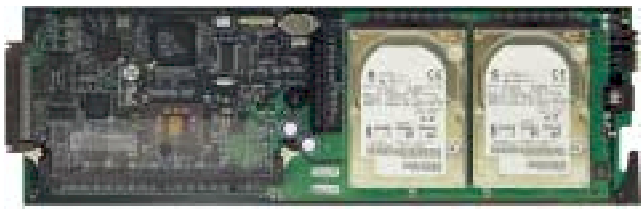

Figure 1. The RLX ServerBlade

*This work was supported by the U.S. Dept. of Energy's *Los Alamos Computer Science Institute* and *Information Architecture - Linux* Programs through Los Alamos National Laboratory contract W-7405-ENG-36. Also available as Los Alamos Unclassified Report 02-1210, March 2002.

[1] While the blade-to-chassis interface is RLX proprietary, the remainder of the cluster is COTS. However, a recent announcement (Feb. 5, 2002) by HP provides for an open enhancement of the CompactPCI (cPCI) specification to standardize blade servers across manufacturers.

Figure 2. The RLX System 324

The rest of the paper is organized as follows. In Section 2, we discuss the architecture and technology behind our Bladed Beowulf. Next, Section 3 presents the performance evaluation of our Bladed Beowulf via a gravitational microkernel benchmark, an N-body parallel simulation, NAS parallel benchmarks, and a treecode benchmark. With these performance numbers in hand, we then propose a new performance metric for the high-performance computing community — Total Price-Performance Ratio (ToPPeR), where Total Price encompasses the total cost of ownership — and discuss two related metrics, namely performance-space ratio and performance-power ratio.

2 Architecture of a Bladed Beowulf

The Crusoe family of processors takes a radically different approach to microprocessor design. In contrast to the traditional transistor-laden, and hence, power-hungry CPUs from AMD and Intel, the Transmeta Crusoe TM5600 CPU is a software-hardware hybrid. It consists of a 128-bit VLIW hardware engine surrounded by a software layer called code morphing. This code morphing software (CMS) presents an x86 interface to the BIOS, operating system (OS), and applications.

2.1 VLIW Engine

Having CMS handle x86 compatibility frees hardware designers to create a very simple, high-performance VLIW engine with two integer units, a floating-point unit, a memory (load/store) unit, and a branch unit. Each of the integer units is a 7-stage pipeline, and the floating-point unit is a 10-stage pipeline.

In Transmeta's terminology, the Crusoe processor's VLIW is called a *molecule*. Each molecule can be 64 bits or 128 bits long and can contain up to four RISC-like instructions called *atoms*, which are executed in parallel. The format of the molecule directly determines how atoms get routed to functional units, thus greatly simplifying the decode and dispatch hardware. And unlike superscalar architectures, molecules are expected in order, eliminating the need for complex out-of-order hardware which currently accounts for approximately 20% of the transistor count in a superscalar architecture.

This last issue has resulted in the current crop of *complex* RISC chips. For instance, the MIPS R10000 and HP PA-8000 are arguably much more complex than today's standard CISC architecture — the Pentium II. Furthermore, because modern CPUs are more complex, have more transistors, and perform more functions than their early RISC predecessors, the hardware requires *lots* of power, and the more power a CPU draws, the hotter it gets. The hotter that a CPU gets, the more likely it will fail, and perhaps, cause other components to fail (which is what happens in our traditional Beowulf clusters). In fact, unpublished (but reliable) empirical data from two leading vendors indicates that the failure rate of a component doubles for every 10°-C increase in temperature.

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 4) require a large number of transistors that increase power consumption significantly. At load, the Transmeta TM5600 and Pentium 4 CPUs generate approximately 6 and 75 watts, respectively, while an Intel IA-64 generates over 130 watts![2] Because of this substantial difference, the TM5600 requires no active cooling whereas a Pentium 4 (and most definitely, an Intel IA-64) 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 tremendous savings in the total cost of ownership with respect to reliability, electrical usage, cooling requirements, and space usage.

The current generation of Crusoe processors *eliminates* roughly 75% of the transistors traditionally found in all-hardware CPU designs to dramatically reduce power requirements and die size. CMS then "replaces" the functionality that the eliminated transistors would have provided. And because CMS typically resides in standard flash ROMs on the motherboard, improved versions can be downloaded into already-deployed CPUs. This ability to change CMS provides two huge advantages over traditional microprocessor fabrication. First, optimizing and fixing bugs amounts

[2] At the end of 2001, the fastest Crusoe CPU (i.e., TM5800) at load dissipated less than 1 watt (on average) with a 366-MHz TM5800 and approximately 2.5 watts (on average) with an 800-MHz TM5800 [2].

to replacing CMS in Transmetas whereas it may result in a costly hardware re-design and/or re-fabrication in Intels and AMDs. Second, changing to a different instruction set, e.g., from x86 to SPARC, simply involves a change in CMS rather than a complete change from one hardware microprocessor to another.

2.2 Code Morphing Software (CMS)

While the VLIW's native instruction set bears no resemblance to the x86 set, the CMS layer gives x86 programs the illusion that they are running on x86 hardware. That is, CMS dynamically "morphs" x86 instructions into VLIW instructions.

CMS consists of two main modules that work in tandem to create the illusion of running on an x86 processor: (1) the interpreter and (2) the translator. The interpreter module interprets x86 instructions one at a time, filters infrequently executed code from being needlessly optimized, and collects run-time statistical information about the x86 instruction stream to decide if optimizations are necessary.

When CMS detects critical and frequently used x86 instruction sequences, CMS invokes the translator module to re-compile the x86 instructions into optimized VLIW instructions called *translations*. These native translations reduce the number of instructions executed by packing atoms into VLIW molecules, thus resulting in better performance.

Caching the translations in a *translation cache* allows CMS to re-use translations. When a previously translated x86 instruction sequence is encountered, CMS skips the translation process and executes the cached translation directly out of the translation cache. Thus, caching and reusing translations exploits the locality of instruction streams such that the initial cost of the translation is amortized over repeated executions.

2.3 The RLX System 324: Bladed Beowulf

The RLX System 324 comes in three sets of easy-to-integrate pieces: the 3U system chassis, 24 ServerBlades, and bundled cables for communication and power.

The system chassis fits in the industry-standard 19-inch rack cabinet and measures $5.25''$ high, $17.25''$ wide, and $25.2''$ deep. It features two hot-pluggable 450-watt power supplies that provide power load-balancing and autosensing 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.

3 Experimental Study

We evaluate our Bladed Beowulf (internally dubbed *MetaBlade*, or short for Trans*meta*-based *blade*s) in four contexts. First, we use a gravitational microkernel benchmark based on an N-body simulation to evaluate the performance of instruction-level parallelism in commodity off-the-shelf processors — two of which are comparably clocked to the 633-MHz Transmeta TM5600 (i.e., 500-MHz Intel Pentium III and 533-MHz Compaq Alpha EV56) and two others which are not (i.e., 375-MHz IBM Power3 and 1200-MHz AMD Athlon MP). Second, we run a full-scale N-body simulation to obtain a Gflop rating for our *MetaBlade* Bladed Beowulf and take a brief look at the scalability of the simulation code on *MetaBlade*. Third, we use the NAS Parallel Benchmarks (NPB) 2.3 [1] to evaluate the task-level parallelism of the above processors. And lastly, we run a treecode simulation to compare the performance of *MetaBlade* to past and current clusters and supercomputers.

3.1 Experimental Set-Up

Our *MetaBlade* Beowulf cluster consists of twenty-four compute nodes with each node containing a 633-MHz Transmeta TM5600 CPU (100% x86 compatible), 256-MB SDRAM, 10-GB hard disk, and 100-Mb/s network interface. We connect each compute node to a 100-Mb/s Fast Ethernet switch, resulting in a cluster with a star topology.

3.2 Gravitational Microkernel Benchmark

The most time-consuming part of an N-body simulation is computing components of the accelerations of particles. For example, the x-component of the acceleration for particle j under the gravitational influence of particle k is given by

$$\frac{Gm_k(x_j - x_k)}{r^3} \tag{1}$$

where G is the gravitational constant, m_k is the mass of particle k, and r is the separation between the particles, i.e.,

$$r = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \tag{2}$$

Evaluating $r^{-3/2}$ is the slowest part of computing the acceleration, particularly when the square root must be performed in software.

Because of the importance of the above calculation to our N-body codes at Los Alamos National Laboratory, we evaluate the instruction-level parallelism of the Transmeta TM5600 using two different implementations of a reciprocal square root function. The first implementation uses the $sqrt$ function from a math library while the second implementation uses Karp's algorithm [4]: table lookup, Chebychev polynomial interpolation, and Newton-Raphson iteration. To simulate Eq. (1) in the context of an N-body simulation (and coincidentally, enhance the confidence interval of our floating-point evaluation), our microkernel benchmark loops 500 times over the reciprocal square-root calculation.

Table 1 shows the Mflops ratings for five commodity processors over the two different implementations of the gravitational microkernel benchmark. Considering that the Transmeta TM5600 is a software-hardware hybrid and the other CPUs are all-hardware designs, the Transmeta performs quite well. In the "Math $sqrt$" benchmark, the Transmeta performs as well as (if not better than) the Intel and Alpha, relative to clock speed. The performance of the Transmeta suffers a bit with the "Karp $sqrt$" benchmark, primarily because the other processor implementations of the code have been optimized to their respective architectures whereas the Transmeta was not due to the lack of knowledge on the internal details of the Transmeta TM5600.

Processor	Math $sqrt$	Karp $sqrt$
500-MHz Intel Pentium III	87.6	137.5
533-MHz Compaq Alpha EV56	76.2	178.5
633-MHz Transmeta TM5600	115.0	144.6
375-MHz IBM Power3	298.5	379.1
1200-MHz AMD Athlon MP	350.7	452.5

Table 1. Mflop Ratings on an Gravitational Microkernel Benchmark

3.3 Gravitational N-body Simulation

Raw Performance Benchmark: In November 2001, we ran a simulation with $9,753,824$ particles on the 24 processors of our Bladed Beowulf (i.e., *MetaBlade*) for about 1000 timesteps. The latter half of the simulation was performed on the showroom floor of the SC 2001 conference. Figure 3 shows an image of this simulation. Overall, the simulation completed about 1.3×10^{15} floating-point operations sustaining a rate of 2.1 Gflops during the entire simulation.[3] With a peak rating of 15.2 Gflops, this real application code running on our Bladed Beowulf achieves 2.1 / 15.2 = 14% of peak.

[3]We achieved a 3.3-Gflop rating when running the simulation on *MetaBlade2*, a 24-processor chassis with 800-MHz Transmetas and a newer version of CMS, i.e., 4.3.x., courtesy of RLX Technologies.

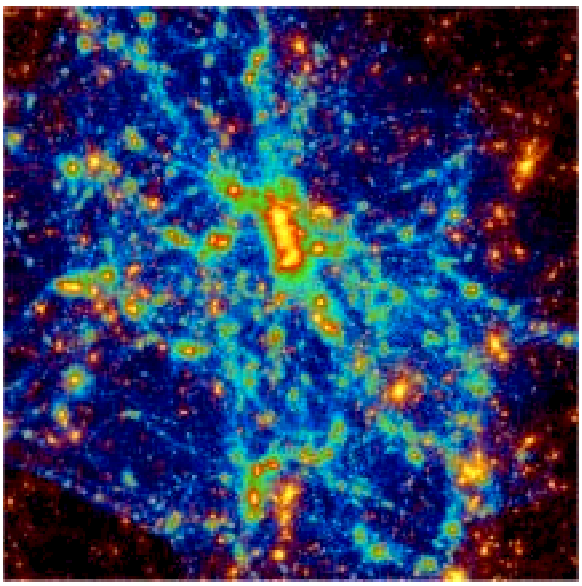

Figure 3. Intermediate Stage of a Gravitational N-body Simulation with 9.7 Million Particles.
The region shown is about 150 million light years across.

Scalability Benchmark: Here we run our N-body simulation code on different numbers of processors to evaluate the scalability of the simulation code over our *MetaBlade* Bladed Beowulf. Table 2 shows the results of these runs.

The scalability results for our Bladed Beowulf are in line with those for traditional clusters. And although the N-body code is highly parallel, the communication overhead is enough to cause the drop in efficiency.

# CPUs	Time (sec)	Speed-Up
1	1367.22	1.00
2	713.60	1.92
4	368.50	3.71
8	210.45	6.50
16	112.71	12.13
24	78.91	17.33

Table 2. Scalability of an N-body Simulation on the *MetaBlade* Bladed Beowulf

3.4 NAS Parallel Benchmarks

The results shown in Table 3 use the NAS Parallel Benchmarks, Version 2.3 [1]. These benchmarks, based on Fortran 77 and the MPI standard, approximate the performance that a typical user can expect for a portable parallel program on a distributed memory computer.

Briefly, the benchmarks are

- BT: simulated computational-fluid dynamics (CFD) application that solves block-tridiagonal systems of 5x5 blocks.
- SP: simulated CFD application that solves scalar pentadiagonal systems.
- LU: simulated CFD application that solves a block lower triangular-block upper triangular system of equations.
- MG: multigrid method to compute the solution of the three-dimensional scalar Poisson equation.
- EP: embarrassingly parallel benchmark to generate random numbers.
- IS: parallel sort over small integers.

Code	Athlon MP	Pentium 3	TM5600	Power3
BT	191.9	71.9	65.9	180.5
SP	167.6	52.7	43.6	155.6
LU	206.3	78.1	80.2	387.3
MG	180.1	41.9	61.6	249.3
EP	4.7	1.4	1.4	3.9
IS	36.4	6.6	12.4	11.0

Table 3. Single Processor Performance (Mops) for Class W NPB 2.3 Benchmarks.

Based on these results, we see that the 633-MHz Transmeta Crusoe TM5600 performs as well as the 500-MHz Intel Pentium III and about one-third as well as the Athlon and Power3 processors.

3.5 Treecode Benchmark

In this section, we run a treecode benchmark on our *MetaBlade* and *MetaBlade2* Bladed Beowulf clusters and compare it to the historical performance of the benchmark running on other Beowulf clusters and supercomputers.

3.5.1 Background on the Treecode Library

N-body methods are widely used in a variety of computational physics algorithms where long-range interactions are important. Several proposed methods allow N-body simulations to be performed on arbitrary collections of bodies in $O(N)$ or $O(N \log N)$ time. These methods represent a system of N bodies in a hierarchical manner by the use of a spatial tree data structure, hence the "treecode" connotation.

Isolating the elements of data management and parallel computation in a treecode library dramatically reduces the amount of programming required to implement a particular physical simulation [10]. For instance, only 2000 lines of code external to the library are required to implement a gravitational N-body simulation. The vortex particle method [7] requires only 2500 lines interfaced to the same treecode library. Smoothed particle hydrodynamics [11] takes 3000 lines. As a point of comparison, the treecode library itself runs nearly 20,000 lines of code.

3.5.2 Treecode Benchmark Results

Table 4 shows the relative placing of the *MetaBlade* (633-MHz Transmetas with CMS 4.2.x) and *MetaBlade2* (800-MHz Transmetas with CMS 4.3.x) Bladed Beowulfs with respect to Mflops/processor. The latter only places behind the SGI Origin 2000 supercomputer. So, although the RLX System 324 was designed for web-server farms, it demonstrates prowess as a supercomputing cluster. Per processor, the performance of the Transmeta Crusoe TM5600 is about twice that of the Intel Pentium Pro 200 which was used in the Loki Beowulf cluster that won the Gordon Bell price/performance prize in 1997 [12] and performs about the same as the 533-MHz Compaq Alpha processors used in the Avalon cluster.

Machine	CPU	Gflop	Mflop/proc
LANL SGI Origin 2000	64	13.10	205.0
SC'01 MetaBlade2	24	3.30	138.0
LANL Avalon	128	16.16	126.0
LANL MetaBlade	24	2.10	87.5
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

Table 4. Historical Performance of Treecode on Clusters and Supercomputers

4 Performance Metrics

Although Hennessy and Patterson [3] have shown the pitfalls of using processor clock speed, instructions per second (ips), and floating-point operations per second (flops) as performance metrics, scientists still tend to evaluate the performance of computing platforms based on floating-point operations per second (and even worse, some scientists compare processor clock speeds across different families of processors) despite the introduction of benchmark suites such as NAS [1] and SPEC [6]. In fact, since June

1993, the most prominent benchmarking list in the high-performance computing community has been the Top500 list at http://www.top500.org. This list is based on the "flop" rating of a single benchmark, i.e., Linpack, which solves a dense system of linear equations.

4.1 The ToPPeR Metric

The use of "flops" remains and will continue. Even at SC, the world's premier supercomputing conference, the Gordon Bell Awards are based on performance (where performance is measured in "flops") and price-performance ratio (where price is the cost of acquisition and performance is in "flops"). In contrast, we propose a new (but related) performance metric: total price-performance ratio (ToPPeR) where total price is the total cost of ownership.

Our *MetaBlade* Bladed Beowulf turns out to be approximately twice as expensive as a similarly performing traditional Beowulf cluster. So, based solely on price-performance ratio (where price encompasses only the cost of acquisition), there exists no reason to use a Bladed Beowulf other than for its novelty. However, we argue that there is more to price than just the cost of acquisition, and hence, propose the notion of Total Price-Performance Ratio (ToPPeR) where total price encompasses the total cost of ownership. We will demonstrate that the ToPPeR metric for Bladed Beowulf clusters is a factor of three times better than traditional Beowulf clusters.

Total cost of ownership (TCO) refers to all the expenses related to buying, owning, and maintaining a computer system within an organization. We break TCO into two components: acquisition cost (AC) and operating cost (OC), i.e., TCO = AC + OC.

The AC simply consists of hardware costs (HWC) and software costs (SWC), i.e., AC = HWC + SWC. This cost is generally a *fixed*, *one-time* cost at the time of purchase. The OC, however, is much more difficult to quantify as it tends to be highly variable and recurring; this cost includes, but is not necessarily limited to, system-administration costs (SAC) such as installation, configuration, maintenance, upgrading, and support, power-consumption costs (PCC), space-consumption costs (SCC), and downtime costs (DTC).[4] The system administration costs (SAC) of a Beowulf cluster can be particularly onerous as they involve the recurring costs of labor and materials.

In sum, using the notation defined above, we propose the following equations as steps towards defining the total cost of ownership in high-performance computing.

$$TCO = AC + OC$$

where

$$\begin{aligned} AC &= HWC + SWC \\ OC &= SAC + PCC + SCC + DTC \end{aligned}$$

and

$$SAC = \sum \text{labor costs} + \sum \text{recurring material costs}$$

Table 5 presents a summary of the total cost of ownership (TCO) on five comparably-equipped, 24-node clusters based on AMD Athlons, Compaq/DEC Alphas, Intel Pentium IIIs (PIIIs) and Pentium 4s (P4s), and Transmeta Crusoe TM5600s, respectively, where each compute node has a 500 to 650-MHz CPU, 256-MB memory, and 10-GB hard disk. The exception is the Pentium 4 CPU which can only be found at 1.3 GHz and above.

For the purposes of our TCO calculation, we assume that the operational lifetime of each cluster to be four years. Based on our own empirical data from our Bladed Beowulf and four traditional Beowulf clusters that support small application-code teams, the system administration cost (SAC) of a traditional Beowulf runs about \$15K/year or \$60K over four years when operating in typical office environment where the ambient temperature hovers around $75°$-F. In contrast, our Bladed Beowulf (in a dusty $80°$-F environment) has been highly reliable with zero hardware failures and zero software failures in nine months; this translates to zero additional labor and zero additional hardware costs. And if there were a failure, we would leverage the bundled management software to diagnose a hardware problem immediately. Our only system administration cost incurred thus far was the initial 2.5-hour assembly, installation, and configuration of our Bladed Beowulf; at \$100/hour, that amounts to \$250 in the first year. Although there have been no failures thus far, we will assume that one major failure will occur per year, e.g., a compute node fails. The cost of the replacement hardware and the labor to install it amounts to \$1200/year. Thus, over a four-year period, SAC runs \$5050.

We estimate the power drawing and cooling costs of the clusters based on the power dissipation of each node. For example, a complete Intel P4 node (with memory, disk, and network interface) generates about 85 watts under load, which translates to 2.04 kW for 24 nodes. Assuming a typical utility rate of \$0.10/kWh over 8760 hours per year (or 35,040 hours over four years), the cost runs \$7,148. In addition, the traditional Beowulfs require power to cool the nodes from overheating, which typically amounts to half a watt per every watt dissipated, thus pushing the total

[4] Other OC components that may be seen more in an enterprise environment rather than a high-performance computing (HPC) environment include centralization, standardization, evaluation for re-investment, training, and auditing. In our calculation for TCO, we only use the OC components relevant to HPC but note that the calculation can be extended for other environments.

Cost Parameter	Alpha	Athlon	PIII	P4	TM5600
Acquisition	$17K	$15K	$16K	$17K	$26K
System Admin	$60K	$60K	$60K	$60K	$5K
Power & Cooling	$11K	$6K	$6K	$11K	$2K
Space	$8K	$8K	$8K	$8K	$2K
Downtime	$12K	$12K	$12K	$12K	$0K
TCO	$108K	$101K	$102K	$108K	$35K

Table 5. Total Cost of Ownership for a 24-node Cluster Over a Four-Year Period

power cost 50% higher to $10,722. In contrast, our 24-node *MetaBlade* Bladed Beowulf based on the Transmeta TM5600 dissipates 0.4 kW at load and requires no fans or active cooling, which results in a total power cost of $2,102 over four years.

Space costs are rarely considered in the TCO of a computer system. Given that Pittsburgh Supercomputing Center leased space from Westinghouse and spent $750,000 to renovate the facilities in order to house its new 6-Tflop Terascale Computing System [8], these costs ought to be included as part of the total cost of ownership. In our space-cost calculation, however, we make the more conservative assumption that space is being leased at a cost of $100 per square foot per year. For example, a 24-node Alpha cluster takes up 20 square feet, which translates to a four-year space cost of $8000, whereas the 24-node Bladed Beowulf takes up 6 square feet for a four-year cost of $2400.[5]

Based on how supercomputing centers charge for time on their clusters and supercomputers, we can estimate the cost of downtime based on the amount of lost revenue. We assume a conservative $5.00 charged per CPU hour (although a recent keynote speech at IEEE IPDPS 2001 indicates that the downtime cost per hour for a NYC stockbroker is $6,500,000). In the case of a 24-node cluster, these costs are relatively small even when we assume that a single failure causes the entire cluster to go down. Specifically, we experience a failure and subsequent four-hour outage (on average) every two months on traditional Beowulf clusters. Thus, the cost of the downtime is 96 hours over four years for the cluster; with 24 nodes, the total CPU downtime is 96 hours $\times$ 24 = 2304 hours. The total downtime cost is then $11,520. In contrast, our Bladed Beowulf has yet to fail after nine months of operation; so, the downtime cost has been $0 thus far. Assuming one failure will occur by the end of the year and is diagnosed in an hour using to the bundled management software, the annual downtime is one hour or four hours over four years for a total cost of $20.

For the five comparably-equipped and comparably-performing, 24-node CPUs, the TCO on our *MetaBlade* Bladed Beowulf is approximately three times better than the TCO on a traditional Beowulf. In a larger-scale supercomputing environment, the results are even more dramatic, e.g., for a 240-node cluster, the space costs differ by a factor of 33. However, the biggest problem with this metric is identifying the hidden costs in the operational costs; furthermore, the magnitude of most of these operational costs is institution-specific. To address this issue more definitively, we propose two related (but more concrete) metrics — performance/space ratio and performance/power ratio — in the next section.

Before we do that, however, we conclude that with the TCO of our 24-node Bladed Beowulf being three times smaller than a traditional cluster and its performance being 75% of a comparably-clocked traditional Beowulf cluster; the ToPPeR value for our Bladed Beowulf is less than half that of a traditional Beowulf. In other words, the total price-performance ratio for our Transmeta-based Bladed Beowulf is over twice as good as a traditional Beowulf.

4.2 Performance/Space

As we noted earlier, space costs money. Thus, it is important to simultaneously maximize performance and minimize space. This provides the motivation for the "performance/space" metric. With respect to this metric, Table 6 compares a traditional 128-node Beowulf called Avalon (which won the Gordon Bell price/performance award in 1998) with our 24-node *MetaBlade* (MB) Beowulf and a recently-ordered 240-node Bladed Beowulf (dubbed *Green Destiny* or GD) that would fit in the same footprint as *MetaBlade*, i.e., six square feet. Even without a rack full of RLX System 324s, our 24-node *MetaBlade* Beowulf beats the traditional Beowulf with respect to performance/space by a factor of two. With a fully-loaded rack of ten RLX System 324s and associated network gear, our *Green Destiny* Bladed Beowulf would result in an over twenty-fold improvement in the performance/space metric when compared to a traditional Beowulf.

[5] It is very important to note that if we were to scale up our Bladed Beowulf to 240 nodes, i.e., cluster in a rack, the cost per square foot over four years would *remain* at $2400 while the traditional Beowulfs' cost would increase ten-fold to $80,000, i.e., 33 times more expensive!

Machine	Avalon	MB	GD
Performance (Gflop)	16.2	2.1	21.0
Area (ft^2)	120	6	6
Perf/Space ($Mflop/ft^2$)	135	350	3500

Table 6. Performance-Space Ratio of a Traditional Beowulf vs. Bladed Beowulfs

Machine	Avalon	MB	GD
Performance (Gflop)	17.6	2.1	21.4
Power (kW)	18.0	0.52	5.2
Perf/Power (Gflop/kW)	0.98	4.12	4.12

Table 7. Performance-Power Ratio for a Traditional Beowulf vs. Bladed Beowulfs

4.3 Performance/Power

Because the electricity needed to power (and cool) machines costs money, we also introduce the "performance/power" metric. Table 7 shows that the Bladed Beowulfs outperform the traditional Beowulf by a factor of four with respect to this metric.

5 Conclusion

In this paper, we presented our *MetaBlade* Bladed Beowulf cluster. Although the acquisition cost of this cluster is approximately 50%-75% more than a comparably-equipped but traditional Beowulf cluster, our experiences and calculations predict that the total cost of ownership of a Transmeta-based Bladed Beowulf will be three times cheaper than a traditional Beowulf cluster. This observation prompted us to propose a new metric called ToPPeR: Total Price-Performance Ratio, where total price encompasses TCO.

The disparity in power dissipation and space usage as well as for ToPPeR will increase in size as Intel pushes forward with its even more voracious IA-64 while Transmeta moves in the other direction, i.e., even lower power consumption but competitive performance. For instance, the 800-MHz Transmeta Crusoe TM5800 that we demonstrated at SC 2001 (http://www.sc2001.org) alongside the 633-MHz Transmeta Crusoe TM5600 produces a "flop" rating of 3.3 Gflops (about 50% better than the 633-MHz TM5600) while generating only 3.5 watts per CPU. The TM6000, expected in volume in the last half of 2002, is expected to improve "flop" performance over the TM5800 by another factor of two to three while reducing power requirements in half again.

Acknowledgements

The authors wish to thank the DOE Los Alamos Computer Science Institute and Information Architecture - Linux Programs 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 and Linus Torvalds for their encouragement on this endeavor.

This research is part of a larger project called *Supercomputing in Small Spaces*. For more information, go to http://sss.lanl.gov.

References

[1] D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow. The NAS Parallel Benchmarks 2.0. *The International Journal of Supercomputer Applications*, December 1995.

[2] D. Ditzel. The TM6000 Crusoe: 1-GHz x86 System on a Chip. 2001 Microprocessor Forum, October 2001.

[3] J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, 1995.

[4] A. Karp. Speeding Up N-body Calculations on Machines Lacking a Hardware Square Root. *Scientific Programming*, 1(2), 1992.

[5] A. Klaiber. The Technology Behind Crusoe Processors. *White Paper*, January 2000.

[6] SPEC Newsletter. SPEC Benchmark Suite Release, 1990.

[7] J. Salmon, M. Warren, and G. Winckelmans. Fast Parallel Treecodes for Gravitational and Fluid Dynamical N-body Problems. *Intl. J. Supercomputer Appl.*, 8:129–142, 1994.

[8] B. Spice. Wiring, Air-Cooling Systems Go In As Assembly of Terascale Approaches: Setting the Stage for the Supercomputer. *The Pittsburgh Post-Gazette*, April 2001.

[9] T. Sterling, D. Becker, D. Savarese, J. Dorband, U. Ranawake, and C. Packer. Beowulf: A Parallel Workstation for Scientific Computation. In *Proc. of the Int'l Conf. on Parallel Processing (ICPP)*, August 1995.

[10] M. Warren and J. Salmon. A Parallel Hashed Oct-Tree N-Body Algorithm. In *Supercomputing '93*, November 1993.

[11] M. Warren and J. Salmon. A Portable Parallel Particle Program. *Computer Physics Communications*, 87:266–290, 1995.

[12] M. Warren, J. Salmon, D. Becker, M. Goda, T. Sterling, and G. Winckelmans. Pentium Pro Inside: I. A Treecode at 430 Gigaflops on ASCI Red, II. Price/Performance of $50/Mflop on Loki and Hyglac. In *Proc. of SC 1997*, November 1997. Gordon Bell Awards for Both Performance and Price/Performance.