IEEE Workshop on High-Performance, Power-Aware Computing (HP-PAC).
in conjunction with the IEEE Parallel & Distributed Processing Symposium
Denver, Colorado, April 2005, LA-UR 05-0936.

Towards Efficient Supercomputing: A Quest for the Right Metric *

Chung-Hsing Hsu, Wu-chun Feng, Jeremy S. Archuleta
Los Alamos National Laboratory
{chunghsu, feng, jsarch}@lanl.gov

Abstract

Over the past decade, we have been building less and less efficient supercomputers, resulting in the construction of substantially larger machine rooms and even new buildings. In addition, because of the thermal power envelope of these supercomputers, a small fortune must be spent to cool them. These infrastructure costs coupled with the additional costs of administering and maintaining such (unreliable) supercomputers dramatically increases their total cost of ownership. As a result, there has been substantial interest in recent years to produce more reliable and more efficient supercomputers that are easy to maintain and use. But how does one quantify efficient supercomputing? That is, what metric should be used to evaluate how efficiently a supercomputer delivers answers?

We argue that existing efficiency metrics such as the performance-power ratio are insufficient and motivate the need for a new type of efficiency metric, one that incorporates notions of reliability, availability, productivity, and total cost of ownership (TCO), for instance. In doing so, however, this paper raises more questions than it answers with respect to efficiency. And in the end, we still return to the performance-power ratio as an efficiency metric with respect to power and use it to evaluate a menagerie of processor platforms in order to provide a set of reference data points for the high-performance computing community.

1. Motivation

In [6], Patterson noted that the computer science research community has focused on performance, performance, and more performance for the past few decades. This observation is even more evident in the high-performance computing (HPC) community, where the Top500 Supercomputer List (http://www.top500.org) and the Gordon Bell Awards for *Performance* and *Price/Performance* take center stage. Unfortunately, this focus on performance (and to a lesser degree, price/performance, where price only captures the acquisition cost) has led the HPC community to build less and less efficient supercomputers with lower reliability, reduced availability, lower productivity, and significantly higher total cost of ownership (TCO).

For example, while the performance of our n-body code has increased by 2000-fold since the Cray C90 of the early 1990s, the performance per watt has only increased 300-fold and the performance per square foot by only 65-fold. This trend in inefficiency has resulted in the construction of substantially larger and more complex machine rooms and even new buildings. Further, because of the thermal envelope of such supercomputers, a small fortune must be spent to simply cool them, e.g., $6M/year at Lawrence Livermore National Laboratory (LLNL) [10]. The primary reason for this less efficient use of space (as well as power) has been the exponentially increasing power requirements of compute nodes, i.e., Moore's Law for Power Consumption, as shown in Figure 1. When nodes consume more power, they must be spaced out and aggressively cooled, e.g., LLNL re-

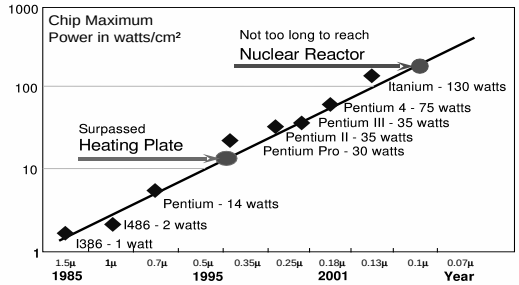

Figure 1. Moore's Law for Power Consumption.

* This work was supported by the DOE ASC Program through Los Alamos National Laboratory contract W-7405-ENG-36.

quires an additional 0.7 watts of cooling for every 1.0 watt of power consumed by their supercomputers [10].

Without exotic cooling facilities, traditional (inefficient) supercomputers would be so unreliable that they would oftentimes be unavailable for use by the application scientist. For instance, the empirical data from our Little Blue Penguin cluster (2000-2001) as well as unpublished empirical data from a leading vendor corroborates that the failure rate of a compute node approximately doubles with every 10°C (18°F) increase in temperature, as per Arrhenius' equation when applied to microelectronics. Given that temperature is directly proportional to power consumption, keeping a compute node cool is obviously critical. Unfortunately, building such exotic cooling facilities can cost as much as the supercomputer itself, e.g., the building for the ASCI Q computer cost nearly $100M. Operating and maintaining such facilities costs even more and is still no guarantee that the supercomputer will not suffer failures, as shown in Table 1 [9]. In sum, all of the above contributes to an astronomical total cost of ownership (TCO).

To address the above problems, we took a *systems design and integration* approach towards building an efficient, reliable, and relatively high-performance supercomputer dubbed Green Destiny in April 2002.[1] Green Destiny is a 240-processor supercomputer that fits in a telephone booth (i.e., a footprint of five square feet) and sips only 3.2 kW of power (when booted diskless) while sitting in an 85-90°F warehouse at 7,400 feet above sea level. Despite its harsh operating environment, the reliability of Green Destiny is arguably second to none: no unscheduled failures in its 24 months of existence.

Since the debut of Green Destiny, and most notably after a New York Times article in June 2002 that compared and contrasted the ASCI Q and Green Destiny supercomputers [4], we have observed a dramatic shift away from the "Moore's Law for Power Consumption" forecast made at IEEE/ACM MICRO-32 in 1999 [8]. Specifically, AMD, and more recently Intel, have temporarily veered off of "Moore s Law for Power Consumption" and have stopped explicitly labeling microprocessors based on clock speed, which is directly proportional to power consumption. They are instead focusing on more efficient microprocessor design. More directly, IBM used the issues of space and cooling to help create a renaissance for its Blue Gene project in late 2002. In sum, what we are really observing is a quest for more efficient horsepower.

Despite the (over-exposed) success of Green Destiny, diehard HPC system researchers reacted with incredulity. Why would anyone sacrifice so much performance (i.e., 1.5x to 2x worse) to achieve better reliability and availability (i.e., "infinitely" better) as well as lower TCO (i.e., 2x better)? On the other hand, HPC application researchers from biological, pharmaceutical, and application software companies have embraced the idea enough that a start-up company based on Green Destiny was recently launched — Orion Multisystems (http://www.orionmulti.com). Thus, the question that begs to be asked in the quest for more efficient horsepower is as follows:

What metric should be used to evaluate how efficiently a given system delivers answers?

This paper presents an initial attempt at answering the above question — finding the right metric for efficient supercomputing. For now, we only focus on efficiency in terms of system performance and power consumption as well as their subsequent effects. System performance and power consumption are easily quantifiable measures; they are also first-class design constraints for supercomputers. We start by examining the use of existing metrics for efficient supercomputing, in particular, the performance-power ratio commonly seen in the field of low-power circuit design. We then argue that the use of such an existing metric is insufficient to address several key issues in efficient supercomputing. In other words, we need a new type of efficiency metric to address the issues of reliability, availability, productivity, and total cost of ownership (TCO).

2. Background

One place that we can borrow an existing metric for efficient supercomputing is the low-power circuit design community where researchers face a similar dilemma. These researchers must tackle the challenging problem of designing a circuit that simultaneously optimizes performance and power. Naively focusing on performance ultimately leads to a design that dissipates too much power whereas optimizing merely for the most power-efficient design rarely achieves the needed performance. As a consequence, the community has proposed metrics that combine both performance and power measures into a single index.

2.1. The Use of An Existing Metric: ED^n

The most commonly used metric in the low-power circuit design community is in the form of ED^n [7] where E is the energy, D is the circuit delay, and n is a nonnegative integer. The power-delay product (PDP), the energy-delay product (EDP) [3], and the energy-delay-squared product (ED^{2}P) [5] are all special cases of ED^n with n = 0, 1, and 2, respectively. Intuitively, ED^n captures the "energy usage per operation." A lower ED^n value indicates that power is more efficiently translated into the speed of operation. The

1 Complementary approaches include the recently unveiled IBM Blue Gene/L, which focuses on hardware design & manufacturing to deliver "five 9s" reliability (i.e., 99.999%) and Google's server farm, which focuses on delivering reliability at the software level while assuming hardware unreliability underneath.

System	#CPUs	Reliability
ASCI Q	8,192	MTBI: 6.5 hours. 114 unplanned outages/month. HW outage sources: sotrage, CPU , memory.
ASCI White	8,192	MTBF: 5 hours (2001) and 40 hours (2003). HW outage sources: storage, CPU, 3rd-party HW.
NERSC Seaborg	6,566	MTBI: 14 days. MTTR: 3.3 hours. SW is the main outage source.

Table 1. Reliability and Availability of Large-Scale HPC Systems [9].
(MTBI: Mean Time Between Interrupts, MTBF: Mean Time Between Failures, MTTR: Mean Time To Restore)

parameter n implies that a 1% reduction in circuit delay is worth paying an n% increase in energy usage; thus, different n values represent varying degrees of emphasis on deliverable performance over power consumption.

Sometimes, a variant of ED^n — the number of operations per second per watt — is used. In the circuit design community, the use of *performance^n/power* with a higher n value is encouraged for performance-optimized, high-end products. For example, using SPEC^3/W to evaluate server-class processors is considered the fairest when compared to the use of SPEC/W or SPEC^2/W [1]. It seems natural to adopt this metric to define the efficiency of a supercomputer and set $n \geq 2$ to put more weight on the performance side. However, we argue that there is a danger in using this metric for efficient supercomputing. Specifically, the metric biases itself toward massively parallel supercomputing systems.

2.2. The Biased Effect of ED^n

Using ED^n with $n \geq 2$ as an efficiency metric to compare two supercomputer designs has a biased effect towards a massively parallel HPC architecture. A large n value not only emphasizes the performance aspect of a HPC system, but it also exaggerates the performance gained from the massive parallelism. More specifically, the ED^n metric with $n \geq 2$ increases exponentially with respect to the number of processors in a supercomputer.

Let us consider a few current and future supercomputers, as listed in Figure 2. Here, for the sake of discussion, we set the efficiency metric as FLOPS^n/W. In order to better present the efficiency comparisons, we normalize FLOPS^n/W with respect to Blue Gene/L because Blue Gene/L has the best FLOPS^n/W efficiency for all n's and for all the listed supercomputers.

Based solely on the FLOPS^2/W metric in Figure 2, Blue Gene/L possesses a far superior design with respect to efficiency. The second most-efficient supercomputer is ASCI Purple, but it lags behind by several orders of magnitude. Blue Gene/L and ASCI Purple have totally different system architectures. Blue Gene/L uses many (131,072) embedded low-performance (0.7 GHz) processors to achieve performance-power efficiency; in contrast, ASCI Purple uses significantly fewer (12,288) but more powerful (2.0 GHz) processors. Thus, Blue Gene/L takes advantage of FLOPS^n/W by aggregating a gaggle of low-power profile embedded processors to achieve significantly better power efficiency. The performance loss that is sustained by using these embedded processors is more than offset by the additional order-of-magnitude increase in the number of processors.

Interestingly, Green Destiny, another supercomputer that is based on an "efficient design" philosophy like Blue Gene/L, falls behind all the listed supercomputers by several orders of magnitude in terms of FLOPS^n/W, where $n \geq 2$. The reason for the gargantuan difference in FLOPS^n/W is due to the metric's bias towards supercomputers with larger numbers of processors. Because Green Destiny has a miniscule processor count when compared to all the other listed supercomputers, particularly Blue Gene/L, the merit of its system architecture with respect to efficiency is marginalized. Therefore, we conclude that using FLOPS^n/W, where $n \geq 2$, as an efficiency metric is highly misleading given that two systems with similar "efficient design" philosophies have two completely different efficiency measures.

More formally, consider a supercomputer that has s processors and each of the processors can deliver F flops at P watts. The FLOPS^n/W metric can then be written in terms of s, F, and P, as follows.

$$\text{FLOPS}^n/\text{W} = \frac{(s \cdot F)^n}{s \cdot P} = s^{n-1} \cdot \frac{F^n}{P} \qquad (1)$$

Equation (1) establishes a positive correlation between FLOPS^n/W and the processor count s. That is, the metric FLOPS^n/W is biased toward HPC systems that contain a larger number of processors (s) as n gets larger. This partly explains why Green Destiny has a very low FLOPS^n/W — all the other supercomputers have 10x-500x more processors than Green Destiny. As a result, Equa-

System Name	Processor Count	Theo. Perf. (TFlops)	Max. Power (MW)
Blue Gene/L (BG/L)	131,072	367.00	1.2
Red Storm (RS)	10,368	41.47	1.7
MACH 5 (M5)	3,132	25.06	0.4
ASCI Purple (Purple)	12,288	98.30	4.7
Columbia (Col)	10,240	61.44	2.0
Earth Simulator (ES)	5,120	40.96	8.0
MareNostrum (Mare)	4,536	39.92	0.6
ASCI Q (Q)	12,288	30.72	3.0
Green Destiny (GD)	240	0.22	0.0052

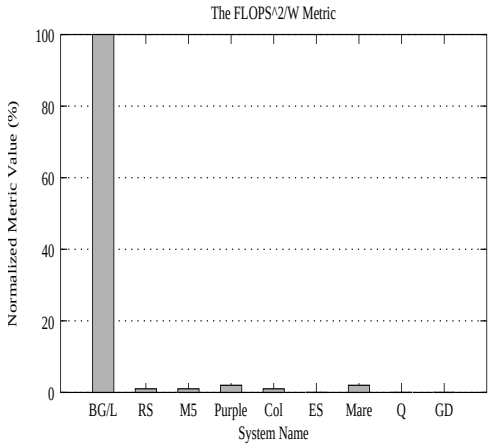

Figure 2. The Efficiency of Several Supercomputers in terms of FLOPS^2/W normalized with respect to Blue Gene/L (BG/L).

tion (1) suggests that setting n = 1 can eliminate such a bias.

In sum, using ED^n with a larger n value, so as to emphasize performance, produces a bias toward supercomputers with massively parallel HPC architectures. To eliminate this bias, n should be set to 1, i.e., the performance-power ratio is a better metric for efficient supercomputing, at least relative to power consumption. Yet efficiency is not just about how efficiently a supercomputer uses power. As discussed earlier, we need a new efficiency metric that addresses the issues of reliability, availability, productivity, and total cost of ownership (TCO). Though the aforementioned discussion influences these issues, it did not address any of them directly.

2.3. Reliability, Productivity, and TCO

The FLOPS^n/W metric is really a derivative of ED^n, which is borrowed from low-power circuit design and tailored for use in efficient supercomputing. Another metric that can be "borrowed" from the circuit-design community is *reliability*. Reliability refers to a system's ability to operate continuously without failure and to maintain data integrity; it is usually measured in terms of mean time to failure (MTTF). In addition, there exist new metrics that are specifically designed for HPC systems such as availability and serviceability. System availability refers to the availability of a system irrespective of its hardware and software reliability, e.g., though Google server farms assume unreliability in their hardware, Google software provides a reliability layer that ensures near 100% availability. Serviceability refers to the ease with which a system can be brought back to an operational state after a failure. Availability and serviceability contribute to the productivity of a system.

While a supercomputer may be capable of computing at petaflop speeds, it does not necessarily mean that it will be a productive machine. For example, in one unpublished instance, the expected boot time of a planned supercomputer exceeded the forecasted MTTF. In this case, the MTTF would have been so short that the productivity of the system would be zero. Even when the MTTF moderately exceeds the boot time, the question becomes how available and serviceable will the supercomputer be for productive use by an end user.

Productivity is oftentimes quantified as the ratio of the number of jobs that a system has run over a long period of time to the total amount of resources that went into buying and running the system. Basing decisions on this definition of productivity avoids the embarrassment of procuring "cheap" piles of hardware that never quite run applications properly and end up cluttering the data-center floor; it also avoids procuring fast, expensive systems solely for the purpose of securing Top500 bragging rights. In short, productivity shifts the focus towards producing meaningful computational output given fixed resources, i.e., getting the most out of what you have.

The power dissipation of a supercomputer has a significant impact on both reliability and productivity, and consequently TCO. Consider the following scenario. A scientist submits a parallel job that requires an entire day to complete on a large-scale supercomputing system. Partway through its execution, the application encounters an unscheduled failure, which causes the fault-management mechanism of the supercomputing system to shut the system down. Even if the scientist performed application-level checkpointing, (s)he still has to wait until the system is repaired and back

on-line before (s)he can re-submit the job to the system to complete. (In the case where no checkpointing was done, the scientist re-submits the job from the start and hopes that it does not fail the next time around.) The above scenario briefly illustrates why it is important for large-scale supercomputing systems to be highly available in delivering reliable and efficient computational cycles to scientists.

2.4. Where Art Thou Efficiency Metric?

In sum, we intuitively know that productivity is influenced by a system's reliability and that system reliability is sensitive to the amount of heat dissipation from the system. However, we are not yet aware of any efficiency metric that models this cascading effect. One of the primary obstacles in having such a model is the lack of substantive empirical data that supports a *quantitative* relationship between reliability and heat stress in a large-scale supercomputing system. In contrast, the relationship between reliability and heat stress has been well established at the circuit level. For example, Arrhenius' equation exponentially links the failure rate with an increase in temperature of a circuit.

While our quantitative understanding of the relationship between reliability and heat stress may be tenuous at best, we have some *qualitative* understanding that reducing the power draw of a supercomputer will improve both reliability and productivity. This is because the high temperature of a supercomputer will more likely induce overheating problems, and the increase in temperature is largely caused by the transformation of electrical energy into heat energy. In other words, if a supercomputer draws a reasonably low amount of power, the damages caused by heat stress to the hardware parts can be dramatically reduced, thus increasing the system's reliability, reducing downtime, and improving productivity.

3. Case Study: FLOPS/W Efficiency

The above discussion on identifying an efficiency metric raises more questions than it answers. What we are still left with is the imperfect FLOPS/W metric (and correspondingly, the imperfect LINPACK FLOPS metric). With that in mind, we resign ourselves to present FLOPS/W efficiency results with the LINPACK benchmark across a gamut of four-processor parallel-computing systems as well as a few larger-scale cluster systems whose "natural" sizes are larger than four processors, thus providing a set of reference data points for the high-performance, low-power/power-aware computing community. The microprocessors of the parallel-computing platforms that are evaluated span eight different architectures from the low-end 1.0-GHz VIA C3 (Nehemiah M10000), through the 1.133-GHz Intel PIII-M, 0.933-GHz Transmeta TM5800, 1.2-GHz Transmeta Efficeon, 2.0-GHz AMD Athlon64, 2.2-GHz Intel Xeon, and 1.0-GHz Intel Itanium2 (Deerfield), and to the high-end 2.0-GHz AMD Opteron.

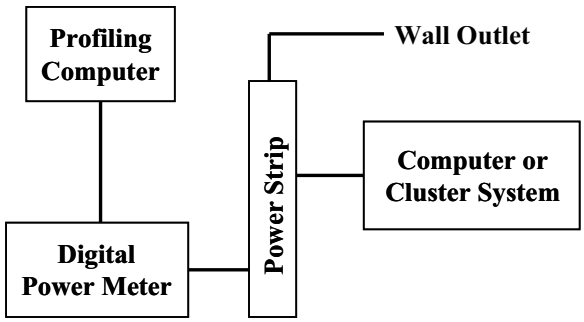

Figure 3. Experimental Set-Up for Benchmark Tests

Figure 3 shows the experimental set-up for all of our tests with the LINPACK benchmark (HPL). The computing system, shown in the bottom right of Figure 3 and denoted as "Computer or Cluster System," runs HPL over Linux 2.4.x and reports on its performance. In order to measure the system's power consumption, we use a Yokogawa digital power meter that is plugged into the same power strip as the system. The power meter continuously samples the instantaneous wattage at a rate of 50 kHz (i.e., every 20 s) and delivers the readings to the profiling computing, shown in the upper left of Figure 3.

We first performed our benchmark tests on four-processor parallel-computing systems whose configurations ranged from four single-processor nodes to two dual-processor nodes to a single quad-processor node, as shown in Table 2. Each system had an aggregate memory of one or more gigabytes (GB) with each processor typically having an average of 0.5 to 1.0-GB memory. This set-up, however, put the Nexcom HiServer 318, 0.1 of Green Destiny (Figure 4), and the DT-12 (Figure 5) systems at a disadvantage, particularly the latter system. In all of these systems, the chassis for the compute nodes supported more than four processors: 18 for the HiServer 318, 24 for 0.1 of Green Destiny (i.e., one chassis), and 12 for the DT-12. Thus, even though only four processors were powered-on in each of these systems, the power consumption of the surrounding chassis was for 18, 24, and 12 processors, respectively. So, instead of the chassis' power consumption being amortized over 18, 24, and 12 processors, respectively, they were amortized only over four processors. In addition, the latter two systems contain embedded Ethernet backplane networks that must be powered-on, irrespective of the number of compute nodes that are powered-on in the chassis.

Figure 4. One 24-Node Chassis from Green Destiny

Figure 5. Orion Multisystems DT-12

Finally, while the HiServer 318 and 0.1 of Green Destiny can have any number of compute nodes on at any given time, the DT-12 must have all of its nodes on. So, measuring the power consumption of the DT-12 actually meant that we were measuring the power consumption of all twelve nodes, four of which were actively running HPL and eight of which were on but computationally idle. Therefore, to allow these systems to run HPL at their more natural sizes and allow power consumption to be amortized over an entire chassis, Table 3 presents the performance, power, and performance/power numbers for HPL over an 18-processor Nexcom HiServer 318, 24 processors (or one chassis) of Green Destiny, all 240 processors of Green Destiny, and a 12-processor Orion Multisystems DT-12.

Clearly, the most striking aspect of Tables 2 and 3 is that the Orion Multisystems DT-12 computing platform delivers a performance-power ratio that is at least 2.5 times better than any other system that was tested, i.e., 74.47 Mflops/W versus the next best number of 29.83 Mflops/W for the AMD Opteron-based system.

Future work entails an in-depth analysis of why the Orion Multisystems DT-12 platform was so much more efficient than the other platforms. We will also compare and contrast this work on low-power, relatively high-performance computing to our power-aware high-performance computing research with PowerNow!-enabled AMD Athlon XP processors [2] that was presented at the International Supercomputer Conference in June 2004, i.e., where the Top500 Supercomputer List is announced every June.

4. Summary

This paper sought to identify an appropriate metric for the efficiency of an HPC system. We started by examining the existing metrics used in the low-power circuit design community, mostly in the form of ED^n, and analyzed their appropriateness for use in efficient supercomputing. We found that $FLOPS^n/W$ with a larger n value was not an appropriate metric for comparing the efficiency of two supercomputers, unlike the circuit design community which encourages the use of a larger n for performance-optimized products. This is because the $FLOPS^n/W$ metric, where $n \geq 2$, is strongly influenced by the number of processors in a system, thus biasing the metric towards massively parallel-processing systems. Consequently, this left us back where we started — with the performance-power ratio, defined as FLOPS/W for the LINPACK benchmark. Despite the "elusiveness" of a new efficiency metric, we still advocated the need for one in supercomputing and discussed the issues that a new metric should consider: reliability, productivity, power efficiency (performance-power ratio), and sustained performance.

Cluster Name	CPU	Cluster Topology	Memory (GB)	HPL Perf. (GFlops)	$\text{Power}_{\text{HPL}}$ (W)	Perf/Power (MFlops/W)	Comment
Door Stop	1.0-GHz VIA C3 Nehemiah M10K	4×1P	1.92	1.01	217.3	4.65	Circa 2002.
HS-318	1.133-GHz Intel Pentium III-M	4×1P	1.00	3.15	351.6	8.96	Circa 2002. 18-node chassis on.
1/60 Green Destiny	0.933-GHz Transmeta TM5800	4×1P	2.56	2.32	132.8	17.47	Circa 2001-2002. 24-node chassis on.
DT-12	1.2-GHz Transmeta Efficeon	4×1P	4.00	4.99	159.1	31.36	Circa 2004. All 12 nodes on.
Agile	2.0-GHz AMD Athlon64	4×1P	4.00	8.44	612.9	13.77	Circa 2003.
Haste	2.2-GHz Intel Xeon	2×2P	3.00	10.10	434.5	23.25	Circa 2003.
Hot Shot	1.0-GHz Intel Itanium2	2×2P	8.00	10.51	601.0	17.49	Circa 2004.
CAFfeine	2.0-GHz AMD Opteron	1×4P	4.00	11.68	402.9	28.99	Circa 2004.
CAFfeine+	2.0-GHz AMD Opteron	1×4P	8.00	12.81	429.5	29.83	Circa 2004.

Table 2. Performance, Power, and Performance/Power for Four-Processor Parallel-Computing Systems

Cluster Name	CPU	Cluster Topology	Memory (GB)	HPL Perf. (GFlops)	$\text{Power}_{\text{HPL}}$ (W)	Perf/Power (MFlops/W)	Comment
HS-318	1.133-GHz Intel Pentium III-M	18×1P	4.5	13.23	1117.0	11.84	Circa 2002.
1/10 Green Destiny	0.933-GHz Transmeta TM5800	24×1P	15.4	12.60	512.6	24.58	Circa 2001-2002.
Green Destiny	0.933-GHz Transmeta TM5800	240×1P	153.6	101.01	5200.0	19.43	Circa 2001-2002.
DT-12	1.2-GHz Transmeta Efficeon	12×1P	12.00	13.80	185.3	74.47	Circa 2004.

Table 3. Performance, Power, and Performance/Power for Large-Scale Parallel-Computing Systems

References

[1] D. Brooks, P. Bose, S. Schuster, H. Jacobson, P. Kudva, A.Buyuktosunoglu, J.-D. Wellman, V. Zyuban, M. Gupta, and P. Cook. Power-aware microarchitecture: design and modeling challenges for the next generation microprocessors. *IEEE Micro*, 20(6):26–44, November/December 2000.

[2] W. Feng and C. Hsu. Green destiny and its evolving parts. *Innovative Supercomputer Architecture Award, The 19th International Supercomputer Conference*, Heidelberg, Germany, June 2004.

[3] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal of Solid-State Circuits*, 31(9):1277–1284, Sept. 1996.

[4] J. Markoff and S. Lohr. Intel's huge bet turns iffy. *New York Times*, Sept. 2002.

[5] A. Martin. Towards an energy complexity of computation. *Information Processing Letters*, 77:181–187, Feb. 2001.

[6] D. Patterson. New challenges for the postPC era. In *Invited Talk, IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Apr. 2001.

[7] P. Pénzes and A. Martin. Energy-delay efficiency of VLSI computations. In *The ACM Great Lakes Symposium on VLSI (GLSVLSI)*, Apr. 2002.

[8] F. Pollack. Keynote speech on new microarchitecture challenges in the coming generations of CMOS process technologies. In *International Symposium on Microarchitecture (MICRO)*, Nov. 1999.

[9] D. Reed. High-end computing: The challenge of scale. Director's Colloquium, Los Alamos National Laboratory, May 2004.

[10] M. Seager. Panelist on what are the future trends in high-performance interconnects for parallel computers? In *IEEE Symposium on High-Performance Interconnects HOTI*, Aug. 2004.