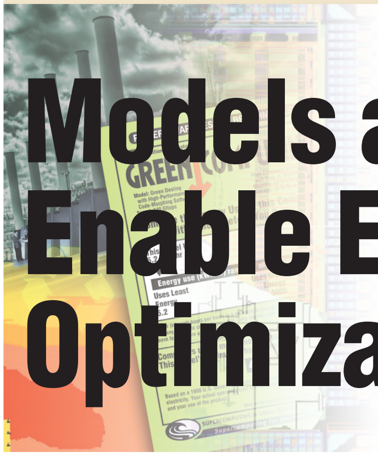
COVER FEATURE



Models and Metrics to Enable Energy-Efficiency Optimizations

Suzanne Rivoire, Stanford University
Mehul A. Shah and Parthasarathy Ranganathan, Hewlett-Packard Laboratories
Christos Kozyrakis, Stanford University
Justin Meza, University of California, Los Angeles

Power consumption and energy efficiency are important factors in the initial design and day-to-day management of computer systems. Researchers and system designers need benchmarks that characterize energy efficiency to evaluate systems and identify promising new technologies. To predict the effects of new designs and configurations, they also need accurate methods of modeling power consumption.

In recent years, the power consumption of servers and data centers has become a major concern. According to the US Environmental Protection Agency, enterprise power consumption in the US doubled between 2000 and 2006 (www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf), and will double again in the next five years. Server power consumption not only directly affects a data center's electricity costs, but also necessitates the purchase and operation of cooling equipment, which can consume from one-half to one watt for every watt of server power consumption.

All of these power-related costs can potentially exceed the cost of purchasing hardware. Moreover, the environmental impact of data center power consumption is receiving increasing attention, as is the effect of escalating power densities on the ability to pack machines into a data center.[1]

The two major and complementary ways to approach this problem involve building energy efficiency into the initial design of components and systems, and adaptively managing the power consumption of systems or groups of systems in response to changing conditions in the workload or environment. Examples of the former approach include

- circuit techniques such as disabling the clock signal to a processor's unused parts;
- architectural techniques such as replacing complex uniprocessors with multiple simple cores; and
- support for multiple low-power states in processors, memory, and disks.

At the system level, the latter approach requires policies to intelligently exploit these low-power states for energy savings. Across multiple systems in a cluster or data center, these policies can involve dynamically adapting workload placement or power provisioning to meet specific energy or thermal goals.[1]

To facilitate these optimizations, we need metrics to define energy efficiency, which will help designers compare designs and identify promising energy-efficient technologies. We also need models to predict the effects of dynamic power management policies, particularly over many systems. Unlike the significant body of work on power management and optimization, there has been relatively little focus on metrics and models.

We address the challenges in defining metrics for energy efficiency with a specific case study on JouleSort, which provides a complete, full-system benchmark for energy efficiency across a variety of system classes.[2] The "Approaches to Power Modeling in Computer Systems"

0018-9162/07/$25.00 © 2007 IEEE

Approaches to Power Modeling in Computer Systems

Power models are fundamental to energy-efficiency research, whether the goal is to improve the components' and systems' design or to efficiently use existing hardware. Developers use these models offline to evaluate proposed designs, and online in policies to exploit component power modes within a system, or to efficiently distribute work across several systems.

An ideal power model has several properties. First, it must be accurate. It should also be portable across a variety of existing and future hardware designs, and applicable under a variety of workloads. Finally, it should be cost-effective in its hardware and software requirements and execute swiftly.

Power models used in simulators of proposed hardware trade speed and portability for increased accuracy, relying on detailed knowledge of component architecture and circuit technology. Wattch[1] is a widely used CPU power model that estimates the power costs of accessing different parts of the processor and combines this information with activity counts from a performance simulator to yield power estimates. Similar models have been proposed for other components, including memory, disks, and networking, as well as complete systems.[2,3] These simulators are highly accurate, but also closely tied to specific systems and simulation infrastructures that are much slower than actual hardware.

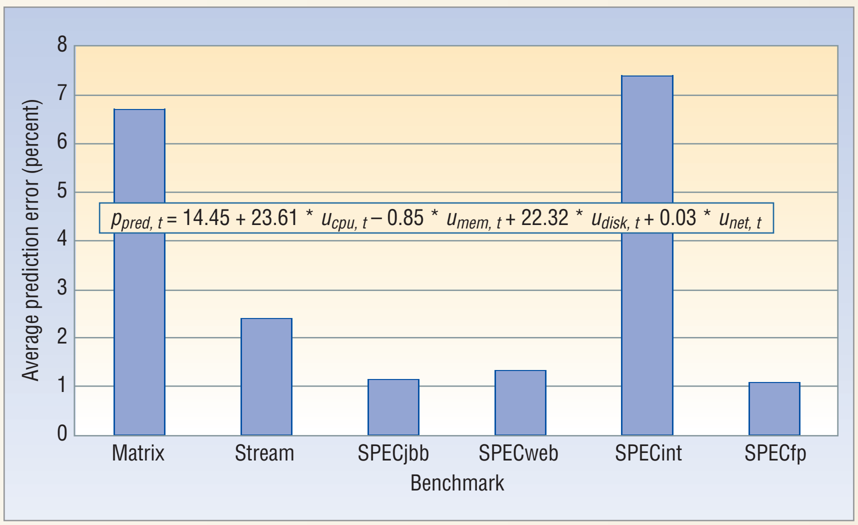


Figure A. Accuracy of the model created by Mantis[9] for a low-power blade. The equation shows the power predicted at a given time t ***as a function of CPU utilization, number of memory and disk accesses, and number of network accesses sampled at that time. Each utilization input*** u ***is given as a percentage of its maximum value. The average error of this coarse-grained linear model is less than 10 percent for every benchmark—sufficient for most scheduling optimizations.***

Models used in online power-management policies, for which speed is a first-class constraint, cannot rely on such detailed simulation. Using real-time system events instead of simulated activity counts addresses this drawback. Frank Bellosa proposed using processor performance counter registers to provide on-the-fly power characterization of real systems.[4] His simple and portable model used the counts of instructions executed and memory accesses to drive the selection of a processor's frequency states. More detailed and processor-specific performance-counter-based models have been developed to model both power[5] and thermal[6] properties. Finally, since researchers developed performance counter options with application profiling rather than power estimation in mind,

sidebar describes different approaches to modeling power consumption in components, systems, and data centers.

ENERGY-EFFICIENCY METRICS

An ideal benchmark for energy efficiency would consist of a universally relevant workload, a metric that balances power and performance in a universally appropriate way, and rules that provide impossible-to-circumvent, fair comparisons. Since this benchmark is impossible, several different approaches have addressed pieces of the energy-efficiency evaluation problem, from chips to data centers. Table 1 summarizes these approaches.

Each metric in Table 1 addresses a particular energy-related problem, from minimizing power consumption in embedded processors (EnergyBench) to evaluating the efficiency of data center cooling and power provisioning (Green Grid metrics). However, only JouleSort specifies a workload, a metric to compare two systems, and rules for running the benchmark.

At the processor level, Ricardo Gonzalez and Mark Horowitz argued in 1996[3] that the energy-delay product provides the appropriate metric for comparing two designs. They observed that a chip's performance and power consumption are both related to the clock frequency, with performance directly proportional to, and power consumption increasing quadratically with, clock frequency. Comparing processors based on energy, which is the product of execution time and

Ismail Kadayif and colleagues proposed an interface based on "energy counters" that would virtualize the existing performance counters.[7]

Processor performance counters can be used to estimate processor and memory power consumption, but do not take other parts of the system, such as I/O, into account. Some optimizations, such as data-center-level optimizations that turn off unused machines, must consider the full-system power. In this case, OS utilization metrics can be used to model the base system components quickly, portably, and with reasonable accuracy. Taliver Heath and colleagues[8] and Dimitris Economou and colleagues[9] build linear models based on OS-reported utilization of each component. Both approaches require an initial calibration phase, in which developers connect the system to a power meter and run microbenchmarks to stress each component. They then fit the utilization data to the power measurements to construct a model.

Figure A shows an example of one such model.[9] Parthasarathy Ranganathan and Phil Leech used a similar approach to predict both power and performance by constructing lookup tables based on utilization.[10] Finally, researchers from Google found that an even simpler model, based solely on OS-reported processor utilization, proved sufficiently accurate to enable optimizations over a large group of homogeneous machines.[11]

Optimizations for energy efficiency rely on accurate, fast, cost-effective, and portable power models. While many models have been developed to address individual needs, creating systematic methods of generating widely portable and highly accurate models remains an open problem. Such methods could facilitate further innovations in energy-efficient system design and management.

References

1. D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proc. Int'l Symp. Computer Architecture* (ISCA), ACM Press, 2000, pp. 83-94.
2. W. Ye et al., "The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool," *Proc. Design Automation Conf.* (DAC), IEEE CS Press, 2000, pp. 340-345.
3. S. Gurumurthi et al., "Using Complete Machine Simulation for Software Power Estimation: The SoftWatt Approach," *Proc. High-Performance Computer Architecture Symp.* (HPCA), IEEE CS Press, 2002, pp. 141-150.
4. F. Bellosa, "The Benefits of Event-Driven Energy Accounting in Power-Sensitive Systems," *Proc. SIGOPS European Workshop*, ACM Press, 2000, pp. 37-42.
5. G. Contreras and M. Martonosi, "Power Prediction for Intel XScale Processors Using Performance Monitoring Unit Events," *Proc. Int'l Symp. Low-Power Electronics and Design* (ISLPED), ACM Press, 2005, pp. 221-226.
6. K. Skadron et al., "Temperature-Aware Microarchitecture: Modeling and Implementation," *ACM Trans. Architecture and Code Optimization*, vol. 1, no. 1, 2004, pp. 94-125.
7. I. Kadayif et al., "vEC: Virtual Energy Counters," *Proc. PASTE,* ACM Press, 2001, pp. 28-31.
8. T. Heath et al., "Energy Conservation in Heterogeneous Server Clusters," *Proc. ACM SIGPLAN Symp. Principles and Practice of Parallel Programming* (PpoPP), ACM Press, 2005, pp. 186-195.
9. D. Economou et al., "Full-System Power Analysis and Modeling for Server Environments"; http://csl.stanford.edu/%7Echristos/publications/2006.mantis.mobs.pdf.
10. P. Ranganathan and P. Leech, "Simulating Complex Enterprise Workloads Using Utilization Traces"; www.hpl.hp.com/personal/Partha_Ranganathan/papers/2007/2007_caecw_bladesim.pdf.
11. X. Fan, W-D. Weber, and L.A. Barroso, "Power Provisioning for a Warehouse-Sized Computer," *Proc. Int'l Symp. Computer Architecture* (ISCA), ACM Press, 2007, pp. 13-23.

power, would therefore motivate processor designers to focus solely on lowering clock frequency at the expense of performance. On the other hand, the energy-delay product, which weighs power against the square of execution time, would show the underlying design's energy efficiency rather than merely reflecting the clock frequency.

In the embedded domain, the Embedded Microprocessor Benchmark Consortium (EEMBC) has proposed the EnergyBench[4] processor benchmarks. EnergyBench provides a standardized data acquisition infrastructure for measuring processor power when running one of EEMBC's existing performance benchmarks. Benchmark scores are then reported as "netmarks per Joule" for networking benchmarks and "telemarks per Joule" for telecommunications benchmarks.

The single-system level is the target of several recent metrics and benchmarking efforts. Performance per watt became a popular metric for servers once power became an important design consideration. Performance is typically specified with either MIPS or the rating from peak performance benchmarks like SPECint or TPC-C. Sun Microsystems has proposed the SWaP (space, watts, and performance) metric to include data center space efficiency as well as power consumption.[5]

Two evolving standards in system-level energy efficiency are the US government's Energy Star certification guidelines for computers and the SPEC Power and Performance

Table 1. Summary of energy efficiency benchmarks and metrics.

Benchmark	Metric	Level	Domain	Workload	Comment
Analysis tool	Performance^N per watt	Any	Any	Unspecified	Different balances of performance and power are important in different contexts. $N = 0$ represents power alone, and $N = 2$ corresponds to the energy-delay product.
EnergyBench	Throughput per Joule	Processor	Embedded	EEMBC benchmarks	
SWaP	Performance/(space × watts)	System(s)	Enterprise	Unspecified	Addresses both space and power concerns
Energy Star certification: workstations	Certify if "typical" power is less than 35 percent of "maximum" power	System	Enterprise	Sleep, idle, and standby power (typical); Linpack and SPECviewperf (maximum)	
Energy Star certification: other systems	Certify if each mode is below a predefined threshold for that system class	System	Mobile, desktop, small server	Sleep, idle, and standby modes	
SPEC Power and Performance	Not yet released	System	Enterprise	Server-side Java under varying loads	Expected late 2007
JouleSort	**Records sorted per Joule**	**System**	**Mobile, desktop, enterprise**	**External sort**	**Has three benchmark classes with different workload size**
Green Grid DCE	Percent of facility power that reaches IT equipment	Data center	Enterprise	n/a	
Green Grid DCPE	Work done/total facility power (W)	Data center	Enterprise	Not yet determined	

committee's upcoming benchmark. Energy Star is a designation given by the US government to highly energy-efficient household products, and has recently been expanded to include computers.[6] For most system classes, systems with idle, sleep, and standby power consumptions below a certain threshold will receive the Energy Star rating. For workstations, however, the Energy Star rating requires that the "typical" power—a weighted function of the idle, sleep, and standby power consumptions—not exceed 35 percent of the "maximum power" (the power consumed during the Linpack and SPECviewperf benchmarks, plus a factor based on the number of installed hard disks). Energy Star certification also requires that a system's power supply efficiency exceed 80 percent.

The SPEC power and performance benchmark remains under development.[7] The workload will be server-side Java-based, and designed to exercise the system at a variety of usage levels, since servers tend to be underutilized in data center environments. The committee expects to release the specific workload and metric of comparison in late 2007.

At the data center level, metrics have been proposed to guide holistic optimizations. To optimize data center cooling, Chandrakant Patel and others have advocated a metric based on weighing performance against the *exergy* destroyed. Roughly speaking, exergy is the energy available for doing useful work.[8]

Finally, the Green Grid, an industrial consortium that includes most major hardware vendors, recently introduced the *data center efficiency metric*.[9] The Green Grid proposal defines DCE as the percentage of total facility power that goes to the "IT equipment"—primarily compute, storage, and network. In the long term, rather than using IT equipment power as a proxy for performance, the Green Grid advocates *data center performance efficiency*, or the useful work divided by the total facility power.

Each of these metrics is useful in evaluating energy efficiency in a particular context, from embedded processors to underutilized servers to entire data centers. However, researchers have not methodically addressed energy-efficiency metrics for many important computing domains. For example, there are no full-system benchmarks that specify a workload, a metric to compare two systems, and rules for running the benchmark. The recently proposed JouleSort benchmark addresses this space.

JOULESORT BENCHMARK

We designed the JouleSort benchmark with several goals in mind. First, the benchmark should evaluate the

power-performance tradeoff—that is, the benchmark score should not reward high performance or low power alone. Two reasonable metrics for the benchmark are thus energy (the product of average power consumption and execution time) and the energy-delay product, which places more emphasis on performance.

We chose energy for two reasons. First, plenty of performance benchmarks already exist, so we wanted to be sure our benchmark emphasized power. Second, the tradeoff between performance and power at the system level does not display the straightforward quadratic relationship seen at the processor level, which motivated use of the energy-delay metric.

Further, the benchmark should evaluate a system's peak energy efficiency, which for today's systems occurs at peak utilization. While peak utilization offers a realistic scenario in some domains, data center servers in particular are notoriously underutilized. However, benchmarking at peak utilization is justified for several reasons. First, peak utilization is simpler to define and measure, and it makes the benchmark more difficult to circumvent. Additionally, knowing the upper bound on energy efficiency for a particular system is useful. In enterprise environments, for example, this upper bound provides a target for server consolidation.

Next, the benchmark should be balanced. It should stress all core system components, and the metric should incorporate the energy that all components use. It should also be representative of important workloads and simple to implement and administer.

Finally, the benchmark should be inclusive, encompassing as many past, current, and future systems as possible. For inclusiveness, the benchmark must be meaningful and measurable on as many system classes as possible. The workload and metric should apply to a wide range of technologies.

Benchmark workload

For our benchmark's workload, we chose to use the external sort from the sort benchmarks' specification (http://research.microsoft.com/research/barc/SortBenchmark/default.htm). External sort has been a benchmark of interest in the database community since 1985, and researchers have used it to understand the system-level effectiveness of algorithmic and component improvements and identify promising technology trends. Previous sort benchmark winners have foreshadowed the transition from supercomputers to commodity clusters, and recently showed the promise of general-purpose computation on graphics processing units (GPUs).[10]

The sort benchmarks currently have three active categories, as summarized in Table 2. PennySort is a price-performance benchmark that measures the number of records a system can sort for one penny, assuming a three-year depreciation. MinuteSort and TerabyteSort measure a system's pure performance in sorting for a fixed time of one minute and a fixed data set of one Tbyte, respectively. JouleSort, to measure the power-performance tradeoff, is thus a logical addition to the sort benchmark repertoire. The original Datamation sort benchmark compared the amount of time systems took to sort 1 million records; it is now deprecated since this task is trivial on modern systems.

Table 2. Summary of sort benchmarks.[10]

Benchmark	Description	Status
PennySort	Sort as many records as possible for one cent, assuming a 3-year depreciation.	Active
MinuteSort	Sort as many records as possible in less than a minute.	Active
TerabyteSort	Sort a Tbyte of data (10 billion records) as quickly as possible.	Active
Datamation	Sort 1 million records as quickly as possible.	Deprecated
JouleSort	Sort a fixed number of records (approx. 10 Gbytes, 100 Gbytes, 1 Tbyte) using as little energy as possible.	Proposed

The workload can be summarized as follows: Sort a file consisting of randomly permuted 100-byte records with 10-byte keys. The input file must be read from—and the output file written to—nonvolatile storage. The output file must be newly created rather than overwriting the input file, and all intermediate files that the sort program uses must be deleted.

This workload meets our benchmark goals satisfactorily. It is balanced, stressing I/O, memory, the CPU, the OS, and the file system. It is representative and inclusive; it resembles sequential, I/O-intensive data-management workloads that are found on most platforms, from cell phones processing multimedia data to clusters performing large-scale parallel data analysis. The Sort Benchmark's longevity testifies to its enduring applicability as technology changes.

Benchmark metric

Designing a metric that allows fair comparisons across systems and avoids loopholes that obviate the benchmark presents a major challenge in benchmark development. For JouleSort, we seek to evaluate the power-performance balance of different systems, giving power and performance equal weight. We could have defined the JouleSort benchmark score in three different ways:

- Set a fixed energy budget for the sort, and compare systems based on the number of records sorted within that budget.

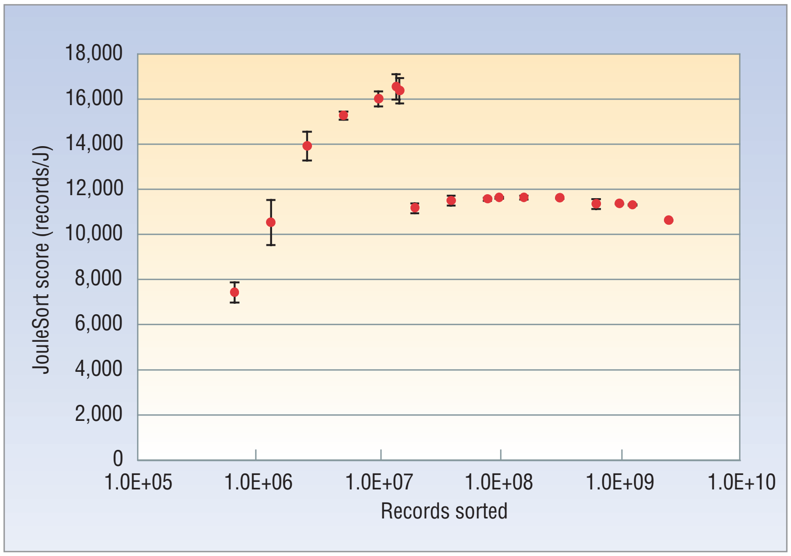


Figure 1. Problems with using a fixed time budget and a metric of records sorted per Joule. The dramatic drop in efficiency at the transition from one-pass to two-pass sorts (here, at 15 million records) creates an incentive to sleep for some, or even most, of the time budget. The (N lg N) complexity of sort causes the slow drop-off in efficiency for large data sets at the rightmost part of the graph and creates a similar problem.

- Set a fixed time budget for the sort, and compare systems based on the number of records sorted and the amount of energy consumed, expressed as records sorted per Joule.
- Set a fixed workload size for the sort, and compare systems based on the amount of energy consumed.

The fixed-energy budget and fixed workload both have the drawback that a single fixed budget will not be applicable to all classes of systems, necessitating multiple benchmark classes and updates to the class definitions as technology changes. The fixed-energy budget has the further drawback of being difficult to benchmark. Since energy is the product of power and time, it is affected by variations in both quantities. Measurement error from power meters only compounds this problem.

By contrast, using a reasonably low fixed-time budget and a metric of records sorted per Joule would avoid this problem; however, two more serious issues eliminate it from consideration. Figure 1 illustrates these, showing the records sorted per Joule for our best-performing system while running different workload sizes. From the left, the smallest data set sizes take only a few seconds and thus poorly amortize the startup overhead.

As data sets grow larger, this overhead amortizes better, while efficiency increases, up to 15 million records. This is the largest data set that fits completely in memory. For larger sizes, the system must temporarily write data to disk, doubling the amount of I/O and decreasing performance dramatically. After this transition, energy efficiency stays relatively constant, with a slow trend downward.

The first problem, then, is the disincentive to continue sorting beyond the largest one-pass sort. With a budget of one minute, this particular machine would achieve its best records-sorted-per-Joule rating if it sorted 15 million records, which takes 10 seconds, and went into a low-power sleep mode for the remaining 50 seconds. In the extreme case, a system optimized for this benchmark could spend most of the benchmark's duration in sleep mode—thus voiding the goal of measuring a utilized system's efficiency.

The second problem is the (N lg N) algorithmic complexity of sort, which causes the downward trend in efficiency for large data sets. While constant factors initially obscure this complexity, once the sort becomes CPU-bound, the number of records sorted per Joule begins to decrease because the execution time now increases superlinearly with the number of records. In light of these problems with a fixed time budget and fixed energy budget, we settled on using a fixed input size. This decision necessitates multiple benchmark classes, similar to the TPC-H benchmark, since different workload sizes are appropriate to different system classes. The JouleSort classes are 100 million records (about 10 Gbytes), 1 billion records (about 100 Gbytes), and 10 billion records (about 1 Tbyte). The metric of comparison then becomes the minimum energy or records sorted per Joule, which are equivalent for a fixed workload size.

We prefer the latter metric because it highlights efficiency more clearly and allows rough comparisons across different benchmark classes, with the caveats we have described. We do anticipate that the benchmark classes will change as systems become more capable. However, since sort performance is improving more slowly than Moore's law, we expect the current classes to be relevant for at least five years. Therefore, given our criteria, the fixed input size offers the most reasonable option.

Energy measurement

While we can borrow many of the benchmark rules from the existing sort benchmarks, energy measurement requires additional guidelines. The most important areas to consider are the boundaries of the system to be measured, constraints on the ambient environment, and acceptable methods of measuring power consumption.

The energy consumed to power the physical system executing the sort is measured from the wall outlet. This approach accounts for power supply inefficiencies in converting from AC to DC power, which can be significant.[1] If a component remains unused in the sort and cannot be physically removed from the system, we include its power consumption in the measurement.

The benchmark accounts for the energy consumed by elements of the cooling infrastructure, such as fans, that physically connect to the hardware. While air conditioners, blowers, and other cooling devices consume significant amounts of energy in data centers, it would be unreasonable to include them for all but the largest sorting systems. We do specify that the ambient temperature at the system's inlets be maintained at between 20° to 25° C—typical for data center environments.

Finally, energy consumption should be measured as the product of the wall clock time used for the sort and the average power over the sort's execution. The execution time will be measured as the existing sort benchmarks specify. The easiest way to measure the power is to plug the system into a digital power meter, which then plugs into the wall; the SPEC Power committee and the Energy Star guidelines have jointly proposed minimum power meter requirements,[6,7] which we adopt for JouleSort as well. Finally, we define two benchmark categories: Daytona, for commercially supported hardware and software, and Indy, which is unconstrained. Table 3 summarizes the final benchmark definition.

Table 3. Summary of JouleSort benchmark definitions.

Workload	**External sort**
Benchmark classes	10^8 records (10 Gbytes), 10^9 records (100 Gbytes), 10^{10} records (1 Tbyte)
Benchmark categories	Daytona = commercially supported hardware and software Indy = "no holds barred" implementations
Metric	Energy to sort a fixed number of records (records sorted per Joule)
Energy measurement	Measure power at the wall, subject to EPA power meter guidelines
Environment	Maintain ambient temperature of 20°-25° C

JOULESORT BENCHMARK RESULTS

Using this benchmark, we evaluated energy efficiency for a variety of computer systems. We first estimated the energy efficiency of previous Sort Benchmark winners and then experimentally evaluated different systems with the JouleSort benchmark.

Energy efficiency of previous sort benchmark winners

First, to understand historical trends in energy efficiency, we retrospectively applied our benchmark to previous sort benchmark winners over the past decade, computing their scores in records sorted per Joule. Since there are no power measurements for these systems, we estimated the power consumption based on the benchmark winners' posted reports on the Sort Benchmark Web site, which include both hardware configuration information and performance data. The estimation methodology relies on the fact that these historical winners have used desktop- and server-class components that should be running at or near peak power for most of the sort. Therefore, we can approximate component power consumption as constant over the sort's length.

We validated our estimation methodology on single-node desktop- and server-class systems, for which the estimates were accurate within 5 to 25 percent—sufficiently accurate to draw high-level conclusions.

The historical data, shown in Figure 2, supports a few observations.

First, the PennySort winners tend to be the most energy-efficient systems, for the simple reason that PennySort is the only benchmark to weigh performance against a resource constraint. While low cost and low power consumption do not always correlate, both metrics tend to encourage minimizing the number of

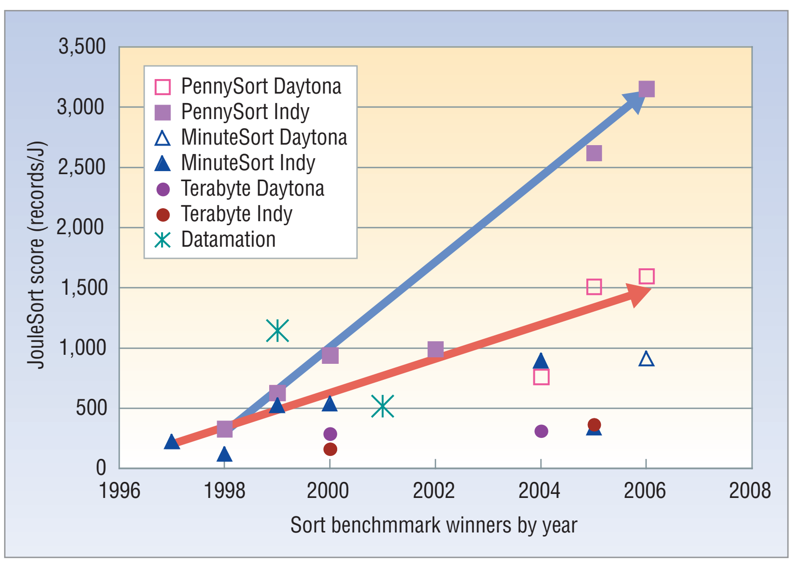


Figure 2. Estimated energy efficiency, in records sorted per Joule, of historical Sort Benchmark winners. The Daytona category is for commercially supported sorts, while the Indy category has no such restrictions. The pink arrow shows the energy efficiency trend for cost-efficient sorts, which is improving at a rate of 25 percent per year. The blue arrow shows the trend for performance-oriented sorts, whose energy efficiency is improving at 13 percent per year. Both of these rates fall well below the rates of improvement in performance and cost performance.

Table 4. Systems for which the JouleSort rating was experimentally measured.

Name	Description
Laptop	A modern laptop with an Intel Core 2 Duo processor and 3 Gbytes of RAM
Blade-wall	A single low-power blade plus the full wall power of its enclosure (designed for 16 blades)
Blade-amortized	A single low-power blade plus its proportionate share of the enclosure power
Standard server	A standard server with Intel Xeon processor, 2 Gbytes of RAM, and 2 hard disks
Fileserver	A fileserver with 2 disk trays containing 6 disks per tray
CoolSort	A desktop with a high-end mobile processor, 2 Gbytes of RAM, and 13 SATA laptop disks
Gumstix	An ultra-low-power system used in embedded devices
Soekris	A board typically used for networking applications
VIA-laptop	A VIA picoITX multimedia machine with laptop hard disks
VIA-flash	A VIA picoITX multimedia machine with flash drives

components and using lower-performance components within a class.

Second, comparing this graph to the published performance records shows that the energy-efficiency scores of sort benchmark winners have not improved at nearly the same rate as performance or price-performance scores. The PennySort winners have improved in both performance and cost efficiency at rates greater than 50 percent per year. Their energy efficiency, on the other hand, has improved by just 25 percent per year, most of which came in the past two years.

The winners of the performance sorts (MinuteSort, TerabyteSort, and Datamation) have improved their performance by 38 percent per year, but have improved energy efficiency by only 13 percent per year. It remains unclear whether these sort benchmark contest winners were the most energy-efficient systems of their time, which suggests the need for a benchmark to track energy efficiency trends.

Current-system and custom-configuration energy efficiency

We ran the JouleSort benchmark on a variety of systems, including off-the-shelf machines representing major system classes, as well as specialized sorting systems we created from commodity components. Table 4 summarizes these systems. Since we focus chiefly on comparing hardware configurations, we use Ordinal Technologies' NSort software for all our experiments.

The commodity machines span several classes of systems: a laptop, low-power blade, standard server, and fileserver. In all systems but the fileserver, the CPU is underutilized in the sort because I/O is the bottleneck; CPU and I/O utilizations balance in the file server. We give two measurements for the blade because the wall power measures the entire enclosure, which is designed to deliver power to 15 blades and is thus both overprovisioned and inefficient at this low load. We therefore include both the wall power of the enclosure—*blade-wall*—and a more realistic calculation of the power consumption for the blade itself, plus a proportionate share of the enclosure overhead, which we call *blade-amortized*.

The laptop and the fileserver proved to be the two most efficient "off-the-shelf" systems by far—both have energy efficiency similar to the most efficient historical system. The file server's high energy efficiency is not surprising because the CPU and I/O both operate at peak utilization, which corresponds to peak energy efficiency for today's equipment. The laptop, however, shows high energy efficiency even though its CPU is drastically underutilized. These results suggest that a benchmark-winning JouleSort machine could be constructed by creating a balanced sorting machine out of mobile-class components.

Based on these insights, we identified two approaches to custom-assembled machines. The first builds a machine from mobile-class components and attempts to maximize performance. The second tries to minimize power while still designing a machine with reasonable performance. Both approaches lead to energy efficiencies more than 2.5 times greater than in previous systems.

The former approach led to the design of the CoolSort machine. CoolSort uses a high-end mobile CPU connected to 13 SATA laptop disks over two PCI-Express interfaces. The laptop disks use less than one-fifth of the power of server-class disks, while providing about one-half the bandwidth. At 13 disks, the CPU is fully utilized during the input pass of the sort, and the motherboard and disk controllers cannot provide any additional I/O bandwidth. In the 10-Gbyte and 100-Gbyte categories, CoolSort's scores of approximately 11,500 records sorted per Joule are more than three times better than those of any previously measured or estimated systems.

Since these notebook-class components have proven more energy efficient than their desktop- or server-class counterparts, it makes sense to ask whether a system with even lower-power components could be more energy efficient than CoolSort. We examined three embedded-class systems: a Gumstix device; a Soekris machine, typically found in routers and networking equipment; and a Via picoITX-based machine, typically used for embedded multimedia applications. Figure 3 shows the JouleSort results of all our measured systems.

Vendors use the smallest and lowest power of these systems, the Gumstix, in a variety of embedded devices.

Our version uses a 600-MHz ARM processor, 128 Mbytes of memory, and an 8-Gbyte CompactFlash card. Its power consumption is a mere 2 W, but the I/O bandwidth is low; a 100-Mbyte in-memory sort takes 137 seconds, sorting 3,650 records per Joule at a bandwidth of 730 Kbps. For a 2-pass, 1-Gbyte sort, the energy efficiency would probably drop to about 1,820 records per Joule.

Moving up in power and performance, the Soekris board we used, designed for networking equipment, contains a 266-MHz AMD Geode processor, 256 Mbytes of memory, and an 8-Gbyte CompactFlash card. The power used during sort is 6 W, three times that of the Gumstix, but the sort bandwidth is a much higher 3.7 Mbps, yielding 5,945 records sorted per Joule for a 1-Gbyte sort. We determined that the I/O interface caused the bottleneck, not the processor or I/O device itself.

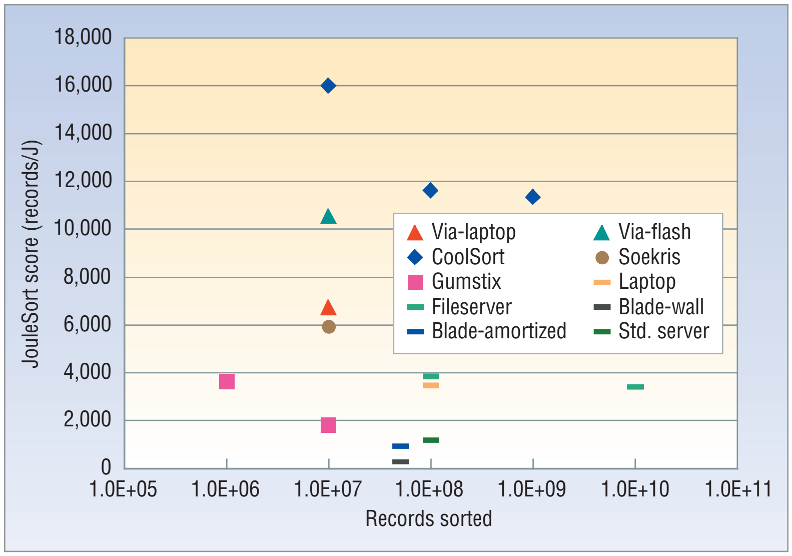


Figure 3. Measured JouleSort scores of commodity and custom machines. The commodity machines, marked with dashes, performed less well than the custom systems.

The final system, the Via picoITX, has a 1-GHz processor and 1 Gbyte of DDR2 memory. We tried both laptop disks and flash as I/O devices; the flash used less power and provided higher bandwidth. For a 1-Gbyte sort, the flash-based version consumed 15 W and sorted 10,548 records per Joule—a number close to that of CoolSort's two-pass sorts. Although laptop disks are theoretically faster, the limitations of the board allowed for fewer laptop disks than flash disks, and thus the flash configuration gave more total I/O bandwidth.

The CoolSort and VIA machines improve upon the previous year's efficiency by more than 250 percent, a marked departure from the 12 to 25 percent yearly improvement rates over the past decade. Thus, creating benchmarks helps to recognize and drive improvements in energy efficiency.

INSIGHTS AND FUTURE WORK

The highest-scoring JouleSort machines provide several insights into system design for energy efficiency. In the CoolSort machine, we chose components for their power-performance tradeoffs and connected them with high-performance interfaces. While CoolSort's mobile processor combined with 13 laptop disks offers an extreme example, it does highlight the promise of designing reasonably well-performing servers from mobile components. On the other hand, the lower-power machines suffered because of the limited performance of their I/O interfaces, rather than the flash devices or CPU. Integration of flash memory closer to the CPU could create more energy-efficient systems.

Second, JouleSort continues the Sort Benchmark's tradition of identifying promising new technologies. The VIA system demonstrates the energy efficiency advantages of flash storage over traditional disks. GPUTeraSort's success among the historical sort benchmark winners shows that the high performance of GPUs comes at a relatively small energy cost, although it is unclear whether this will continue to hold as GPUs grow ever more power hungry. Finally, because sort is a highly parallelizable algorithm, we speculate that multicores will be excellent processors in energy-efficient sorting systems.

Although JouleSort addresses a computer system's energy efficiency, energy is just one piece of the system's *total cost of ownership* (TCO). From the system purchaser's viewpoint, a TCO-Sort would be the most desirable sort benchmark; however, the TCO components vary widely from user to user. Combining JouleSort and PennySort to benchmark the costs of purchasing and powering a system is a possible first step. For the high-efficiency machines we studied, the VIA achieves a JouleSort score comparable to CoolSort, at a much lower price: $1,158 versus $3,032. This result highlights the potential of flash as a cost-effective technology for achieving high energy efficiency.

An emerging area of concern is the *scaledown efficiency* of components and systems—that is, their ability to reduce power consumption in response to low utilization.[11] Traditionally, components have consumed well over half their peak power, even when underutilized or idle. Manufacturers are starting to address this inefficiency. JouleSort captures scaledown efficiency to a small extent, since CPU and I/O will not be perfectly balanced during both sort phases, but it does not necessarily assess efficiency at low utilization.

Finally, JouleSort can be extended to include metrics of importance in the data center. The benchmark's current version does not account for losses in power delivery at the data center or rack level. An appropriate benchmark in that setting might be an Exergy JouleSort where the metric of interest is records sorted per Joule of exergy[8] expended.

As concerns about enterprise power consumption continue to increase, we need metrics that assess and improve energy efficiency. The JouleSort benchmark can help assess improvements in end-to-end, system-level energy efficiency. This addition to the family of sort benchmarks provides a simple and holistic way to chart trends and identify promising new technologies. The most energy-efficient sorting systems use a variety of emerging technologies, including low-power mobile components, and flash-based storage.

References

1. C.D. Patel and P. Ranganathan, "Enterprise Power and Cooling: A Chip-to-Datacenter Perspective," tutorial, *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems* (ASPLOS 2006), ACM Press, 2006, pp.1-101.
2. S. Rivoire et al., "JouleSort: A Balanced Energy-Efficiency Benchmark," *Proc. SIGMOD Conf.*, ACM Press, 2007, pp. 365-376.
3. R. Gonzalez and M. Horowitz, "Energy Dissipation in General-Purpose Microprocessors," *IEEE J. Solid-State Circuits*, Sept. 1996, pp. 1277-1284.
4. Embedded Microprocessor Benchmark Consortium (EEMBC), "EnergyBench v 1.0 Power/Energy Benchmarks"; www.eembc.org/benchmark/power_sl.asp.
5. Sun Microsystems, "SWaP (Space, Watts, and Performance) Metric"; www.sun.com/servers/coolthreads/swap.
6. US Environmental Protection Agency, "ENERGY STAR Program Requirements for Computers: Version 4.0"; www.energystar.gov/ia/partners/prod_development/revisions/downloads/computer/Computer_Spec_Final.pdf.
7. Standard Performance Evaluation Corporation (SPEC), "SPEC Power and Performance Committee"; www.spec.org/specpower.
8. C.D. Patel, "A Vision of Energy-Aware Computing from Chips to Data Centers," *Proc. ISMME*, Japan Soc. Mechanical Engineers, 2003, pp. 141-150.
9. The Green Grid, "Green Grid Metrics: Describing Datacenter Power Efficiency," 2007; www.thegreengrid.org/gg_content/Green_Grid_Metrics_WP.pdf.
10. N.K. Govindaraju et al., "GPUTeraSort: High Performance Graphics Coprocessor Sorting for Large Database Management," *Proc. SIGMOD Conf.*, ACM Press, 2006, pp. 325-336.
11. R. Mayo and P. Ranganathan, "Energy Consumption in Mobile Devices: Why Future Systems Need Requirements-Aware Energy Scale-Down," *Special Issue on Power Management*, LNCS, Springer, 2003, pp. 26-40.

Suzanne Rivoire *is a PhD candidate in the Department of Electrical Engineering, Stanford University. Her research interests include energy-efficient system design, data center power management, and data-parallel architectures. Rivoire received an MS in electrical engineering from Stanford University. Contact her at rivoire@stanford.edu.*

Mehul A. Shah *is a research scientist in the Storage Systems Department at Hewlett-Packard Laboratories. His research interests include energy efficiency of computer systems, database systems, distributed systems, and long-term digital preservation. Shah received a PhD in computer science from the University of California, Berkeley. Contact him at mehul.shah@hp.com.*

Parthasarathy Ranganathan *is a principal research scientist at HP Laboratories. His research interests include low-power design, system architecture, and parallel computing. Ranganathan received a PhD in electrical and computer engineering from Rice University. Contact him at partha.ranganathan@hp.com.*

Christos Kozyrakis *is an assistant professor of electrical engineering and computer science at Stanford University. His research interests include transactional memory, architectural support for security, and power management techniques. Kozyrakis received a PhD in computer science from the University of California, Berkeley. Contact him at christos@ee.stanford.edu.*

Justin Meza *is an undergraduate in computer science at the University of California, Los Angeles. His research interests include open source software, Web standards, electronics, and programming. Contact him at justin.meza@ucla.edu.*