The Search for Energy-Efficient Building Blocks for the Data Center

Laura Keys[1], Suzanne Rivoire[2] and John D. Davis[3]

[1] University of California, Berkeley
[2] Sonoma State University
[3] Microsoft Research – Silicon Valley
laurak@eecs.berkeley.edu, suzanne.rivoire@sonoma.edu,
john.d@microsoft.com

Abstract. This paper conducts a survey of several small clusters of machines in search of the most energy-efficient data center building block targeting data-intensive computing. We first evaluate the performance and power of single machines from the embedded, mobile, desktop, and server spaces. From this group, we narrow our choices to three system types. We build five-node homogeneous clusters of each type and run Dryad, a distributed execution engine, with a collection of data-intensive workloads to measure the energy consumption per task on each cluster. For this collection of data-intensive workloads, our high-end mobile-class system was, on average, 80% more energy-efficient than a cluster with embedded processors and at least 300% more energy-efficient than a cluster with low-power server processors.

1 Introduction

Power consumption is a first-order design constraint in the data center (DC). Although still small in absolute terms, DC power consumption is growing rapidly, doubling between 2000 and 2005 [1]. The energy usage of the computational building blocks of the DC is critical to the overall power consumption, since it affects the design and operation of the cooling and power distribution infrastructure as well as the computational infrastructure [2, 3, 4].

Traditionally, the computational nodes in DCs operate with low system utilization but require high availability and fast response time. Researchers have therefore advocated the design of hardware whose power consumption is proportional to the system load [5]. However, there is a new class of DC benchmarks that use as many resources as are available. Many of these applications are I/O- and network-bound but exhibit phases of high CPU utilization. Dryad, Hadoop, MapReduce, and Condor are frameworks for this type of application [6, 7, 8, 9].

In the past, research on these data-intensive workloads has assumed that the applications would be bottlenecked by low I/O bandwidth and high latency. However, the introduction of NAND flash-based solid-state drives (SSDs) virtually eliminates the disk seek bottleneck, enabling much higher I/O bandwidth and very low latency. Although SSDs do not yet provide the capacity of magnetic disk drives, SSDs can be very low-power devices and have the ability to consolidate the storage system by providing far more IOPS, better feeding the processor with data [10].

In this paper, we characterize clusters across a variety of system types in order to find energy-efficient DC building blocks, with a focus on emerging data-intensive applications.

We initially characterize a variety of embedded, mobile, desktop, and server systems using single-machine performance, power, and energy efficiency. Using these benchmarks as a guide to prune the system space, we build homogeneous clusters of the top three systems. We execute DryadLINQ applications on these clusters in order to understand their energy efficiency for different application types.

This paper makes the following contributions:

- We characterize a wide range of real systems from embedded and mobile to desktop and server processors, focusing on single-thread and/or single-system performance.
- We characterize homogeneous compute clusters composed of embedded, mobile, and server processors in the context of data-intensive applications to find the most energy-efficient computing infrastructure over a wide range of workloads.
- We compare the energy efficiency of system classes that have not been compared in previous work, and we make this comparison across workloads with varying computational and I/O demands.

The rest of this paper is organized as follows. Section 2 is an overview of related work in this area. Section 3 describes our experimental infrastructure and the hardware and software evaluated. Section 4 presents our experimental results. We further discuss these results in Section 5 and conclude with Section 6.

2 Related Work

A growing body of research proposes energy-efficient building blocks for cluster and DC computing, but this work has typically investigated only a limited subset of system types and/or applications.

One major trend is to propose building blocks for data-intensive computing that combine embedded processors, such as the Intel Atom, with solid-state disks. However, many of these proposed systems have been evaluated for only a single workload or against a limited set of alternative hardware.

For example, Szalay et al. propose "Amdahl blades," consisting of Intel Atom processors with SSDs, and present a scaling study comparing these blades to traditional high-end cluster nodes using data from a synthetic disk-stressing benchmark [11]. The Gordon system, designed by Caulfield et al., also combines Atom processors with flash memory. It was evaluated against a Core 2 (single-core)-based server over a variety of MapReduce workloads using simulation and modeling rather than physical measurements [12].

The FAWN cluster, proposed by Andersen et al., consists of ultra-low-end embedded processors and high-end solid-state disks [13]. A version using the Intel Atom was evaluated across a wide range of workloads [14]. This evaluation showed FAWN breaking the energy-efficient sorting record set by Beckmann in 2010 with similar hardware [15]. The overall conclusion of the evaluation was that the FAWN hardware was superior to desktop- and server-class hardware for I/O-bound workloads and for memory-bound workloads with either poor locality or small working sets. However, high-end mobile processors were not evaluated in the FAWN study. Reddi et al. use embedded processors for web search and note both their promise and their limitations; in this context, embedded processors jeopardize quality of service because they lack the ability to absorb spikes in the workload [16].

Several studies have proposed high-end laptop hardware for energy-efficient DC computing. Rivoire et al. used a laptop processor and laptop disks to set an energy-efficient sorting record in 2007 [17], while Lim et al. proposed a laptop-processor-based building block for Web 2.0 workloads in 2008 [18]. However, these systems preceded the movement toward embedded processors and SSDs, and their conclusions must be revisited in light of these recent developments.

Finally, the CEMS servers proposed by Hamilton use a variety of desktop processors and a single enterprise-class magnetic disk [19]. These servers are evaluated using a CPU-bound webserver workload designed to exercise the CPU at varying utilizations up to 60%. Unlike much of the previous work, this study found that for this workload, the systems with the lowest power consumption were not the most energy-efficient systems.

3 System Overview

In this section, we describe the hardware platforms we examine, the benchmarks we use to evaluate them, and the infrastructure used to measure power.

3.1 Hardware

We consider a variety of systems based on embedded, mobile, desktop, and server processors. Table 1 provides a list of the important features of the systems under test (SUTs). All systems are running 64-bit Windows Server 2008 with support for Dryad and DryadLINQ jobs. We tried to provision the systems with 4 GB of DRAM per core when possible, but two of the embedded systems were only able to address a fraction of this memory. The industry-standard server system used 10,000 RPM enterprise hard disks, and the other systems each contained a single Micron RealSSD. This difference affected the server's average power by less than 10% and had a negligible effect on the system's overall energy efficiency.

3.2 Benchmark Details

We ran an assortment of benchmarks, some CPU-intensive, others utilizing disk and network, in order to find the most energy-efficient cluster building block and see how robust this choice is across different types of workloads. A few of these benchmarks are used to evaluate single-machine performance, and the rest are DryadLINQ jobs

Table 1. Systems evaluated in this paper. Costs are approximate and given in US dollars at the time of purchase. Costs are not given for systems that were donated as samples. In the memory column, the star denotes the maximum amount of addressable memory.

System Under Test	CPU	Memory	Disk(s)	System Information	Approx. cost
1A (embedded)	Intel Atom N230, 1-core, 1.6 GHz, 4W TDP	4 GB DDR2-800	1 SSD	Acer AspireRevo	$600
1B (embedded)	Intel Atom N330, 2-core, 1.6 GHz, 8W TDP	4 GB DDR2-800	1 SSD	Zotac IONITX-A-U	$600
1C (embedded)	Via Nano U2250, 1-core, 1.6 GHz	2.37 GB DDR2-800*	1 SSD	Via VX855	sample
1D (embedded)	Via Nano L2200, 1-core, 1.6 GHz	2.86 GB DDR2-800*	1 SSD	Via CN896/VT8237S	sample
2 (mobile)	Intel Core2 Duo, 2-core, 2.26 GHz, 25W TDP	4 GB DDR3-1066	1 SSD	Mac Mini	$1200
3 (desktop)	AMD Athlon, 2-core, 2.2 GHz, 65W TDP	8 GB DDR2-800	1 SSD	MSI AA-780E	sample
4 (server)	AMD Opteron, 4-core, 2.0 GHz, 50W TDP	32 GB DDR2-800	2 10K rpm	Supermicro AS-1021M-T2+B	$1900

dispatched to five-node clusters. We ran a single instance of each application at a time.

The single-machine benchmarks are as follows:

- *SPECpower_ssj 2008.* This benchmark uses a CPU- and memory-intensive Java webserver workload to probe the power usage of a SUT's CPU at various utilizations. Since the performance of this benchmark can vary drastically depending on the JRE used, we use the Oracle JRockit JRE tuned with platform-specific parameters based on similar reported benchmark runs.
- *SPEC CPUint 2006.* This benchmark suite runs a variety of CPU and memory-intensive jobs and then provides a score based on the aggregate performance of these individual benchmarks. We do not make any architecture-specific optimizations for this workload.
- *CPUEater.* This benchmark fully utilizes a single system's CPU resources in order to determine the highest power reading attributable to the CPU. We use these measurements to corroborate the findings from SPECpower.

The multi-machine DryadLINQ benchmarks are:

- *Sort.* Sorts 4 GB of data with 100-byte records. The data is separated into 5 or 20 partitions which are distributed randomly across a cluster of machines. As all the data to be sorted must first be read from disk and ultimately transferred back to disk on a single machine, this workload has high disk and network utilization.
- *StaticRank.* This benchmark runs a graph-based page ranking algorithm over the ClueWeb09 dataset [20], a corpus consisting of around 1 billion web pages, spread over 80 partitions on a cluster. It is a 3-step job in which output partitions from one step are fed into the next step as input partitions. Thus, StaticRank has high network utilization.
- *Prime*. This benchmark is computationally intensive, checking for primeness of each of approximately 1,000,000 numbers on each of 5 partitions in a cluster. It produces little network traffic.

- *WordCount*. This benchmark reads through 50 MB text files on each of 5 partitions in a cluster and tallies the occurrences of each word that appears. It produces little network traffic.

3.3 Measurement Infrastructure

The measurement infrastructure consists of a hardware component to physically measure both the total system power and power factor and a software component to collect both the power measurements and application-level Event Tracing for Windows (ETW) metrics.

We use WattsUp? Pro USB digital power meters to capture the wall power and power factor once per second for each machine or group of machines. We use the API provided by the power meter manufacturer to incorporate measurements from the power meter into the ETW framework.

4 Evaluation

In this section, we first examine the single-machine performance of a range of machines. We use these results to identify the three most promising candidate systems for the cluster-level benchmarks. The results from both the single-machine and multi-machine benchmarks show that the mobile-class system consistently provides high energy efficiency on a wide range of tasks, while the other classes of systems are suitable for a more limited set of workloads.

4.1 Single-Machine Benchmarks

To pare down our list of systems, we used three single-machine benchmarks to characterize the systems' single-thread performance and power consumption. Based on this characterization, we can eliminate any systems that are Pareto-dominated in performance and power before proceeding to the cluster benchmarks.

Performance. We use SPEC CPU2006 integer benchmarks to compare the single-threaded SPEC-rate performance across all the platforms in Table 1. This benchmark, because it is CPU-intensive, should favor the processors with more complex cores. In addition to the dual-socket quad-core AMD Opteron server in Table 1 (SUT 4), we included two more Opteron servers: a dual-socket single-core server (2x1) with 8 GB of RAM and a dual-socket dual-core server (2x2) with 16 GB of RAM. These systems were included to quantify single-core performance improvements over time, as well as the benefits of additional cores. Figure 1 shows the per-core results, which are normalized to the Intel Atom single-core-based system (SUT 1A).

There are two surprising results. First, the mobile Intel Core 2 Duo (SUT 2) has per-core performance that matches or exceeds that all of the other processors, including the server processors. Second, and more surprising, is the fact that the Atom processor performs so well on the libquantum benchmark. Overall, these results

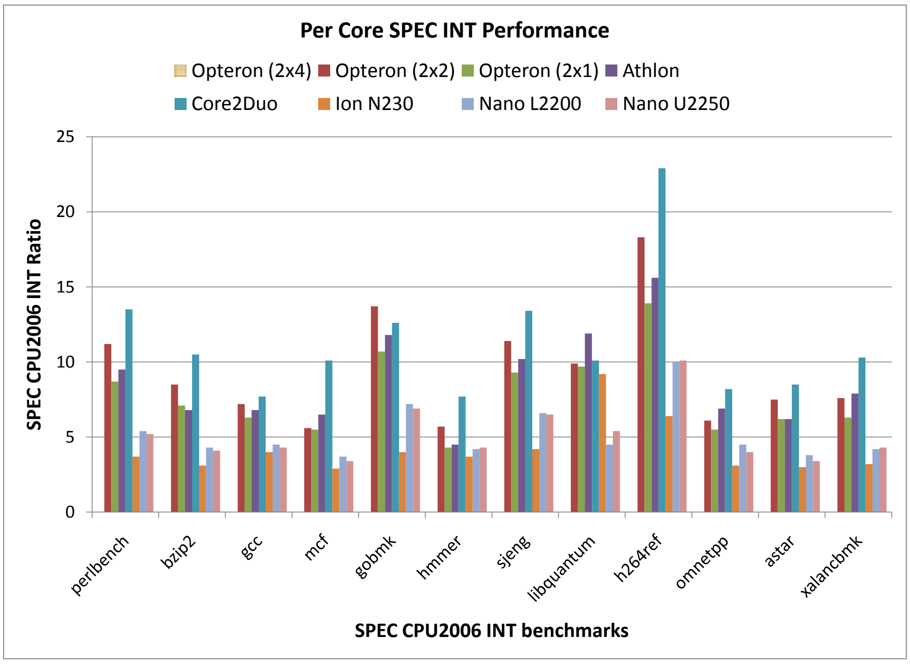

Fig. 1. Per-core SPEC CPU2006 integer performance normalized to the Atom N230 for the systems (embedded, mobile, desktop, and server processors) from Table 1 plus two legacy Opteron servers. The legend should be read from *left* to *right* and *top* to *bottom*. It lists the bars in order from *left* to *right.*

demonstrate that SUT 2 (Intel Core 2 Duo) and SUT4 (AMD Opteron 2x4) provide the highest single-thread performance.

Power Consumption. Single-thread performance is not the only factor to consider when selecting the appropriate energy-efficient building blocks for the DC. Before diving into benchmarks that provide data on work done per Watt or Joule, we measure system power at idle and when running CPUEater at 100% utilization. Figure 2 shows power consumption at these two utilization points for all of the systems from Figure 1, ordered by the maximum system power under full CPU load. Surprisingly, the four embedded-class systems do not have significantly lower idle power than the other systems; in fact, the mobile-class system with a 25 W TDP processor has the second-lowest idle power. However, the 100% utilized systems result in a different ordering. The mobile-class system now has significantly higher power than the embedded systems, which use processors with 4-16 W TDPs.

Balancing Performance and Power. To confirm our conclusions based on examining performance and power separately, we used SPECpower_ssj to characterize the amount of work or operations done per watt. As Figure 3 shows, the Intel Core 2 Duo system (SUT 2) and the Opteron (2x4) system (SUT 4) yield the

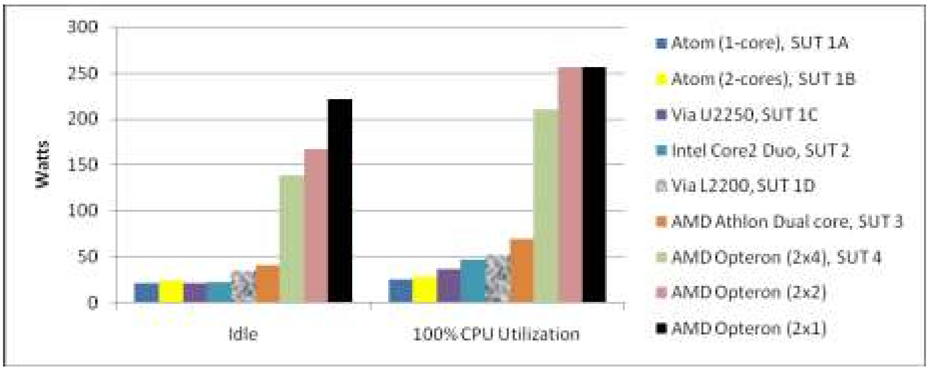

Fig. 2. Power consumption at idle and at 100% CPU utilization for all the systems in Figure 1. The systems are shown in order from lowest to highest power consumption at 100% utilization. The legend should be read from *top* to *bottom*. It lists the bars in order from *left* to *right*.

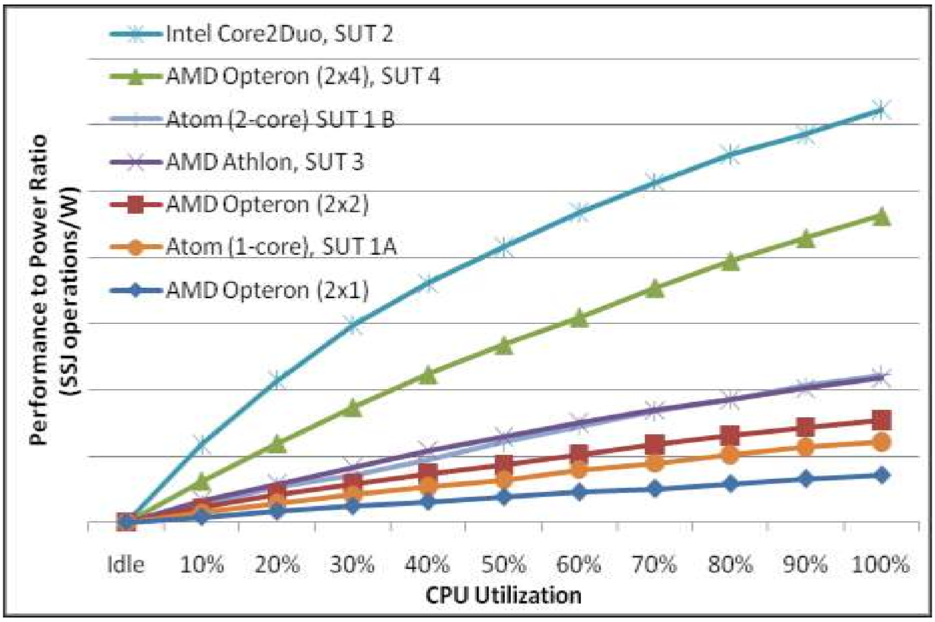

Fig. 3. SPECpower_ssj results for four of the systems from Table 1 plus the two previous generations of Opteron servers.

best power/performance, followed by the Atom system (SUT 1B). These results reinforce our conclusions from looking at power and performance separately. Furthermore, this benchmark goes beyond the single-core performance measured by SPEC CPU 2006.

4.2 Multi-machine Dryad Benchmarks

Based on the characterization from Section 4.1, we set up 5-node clusters of the three most promising systems (1B, 2, and 4) and ran the four DryadLINQ benchmarks: Sort, Primes, StaticRank, and WordCount.

Figure 4 shows the average energy usage for these benchmarks, normalized to the mobile system (SUT 2). It shows two versions of Sort that only differ by the number of data partitions, 5 or 20; the 20-partition version has better load balance.

The energy usage per task of SUT 2, the mobile Core 2 Duo-based server, is always lower than that of SUT 4, the Opteron-based server, across all the benchmarks, using three to five times less energy overall for the different benchmarks.

The relative energy usage of SUT 1B, the Atom-based system, varies the most from benchmark to benchmark. It degrades significantly for Primes, which is the most CPU-intensive benchmark. For this benchmark, the traditional server system (SUT 4) is more energy-efficient than the Atom-based system. SUT 4 has a performance advantage with four times the number of cores, enabling it to finish parallel and computationally intense tasks more quickly but with a significantly higher power envelope than SUT 1B.

This advantage disappears, however, for StaticRank, which has a mix of CPU and I/O. SUT 4 can finish this job only slightly faster than SUT 2 or 1B, but it uses much more power. However, it should be noted that the partition size used for StaticRank is set by the memory capacity limitations of the mobile and embedded platforms. This biases the results in their favor, because at this workload size, SUT 4's execution is dominated by Dryad overhead.

More surprisingly, the Atom-based system is less energy-efficient for Sort than the mobile-CPU-based system. Previous work on platforms for sequential I/O-intensive workloads used Atom-based systems on the assumption that the I/O would be the bottleneck and the CPU would thus not be heavily utilized [11, 14, 15]. However, the SSDs in these systems mitigate this bottleneck for Sort, placing more stress on the CPU. In contrast, the Atom-based system is most energy-efficient for WordCount, which is the least CPU-intensive of the four benchmarks.

These energy measurements on cluster benchmarks complement the results on single-machine benchmarks: low-power mobile class platforms have an advantage over high-power, high-performing server-class platforms as energy-efficient DC building blocks that do not skimp on performance. Their performance and power also are more robust over a wider range of benchmarks than the embedded-class systems.

5 Discussion

The results demonstrate a clear class of systems that is well suited for data-intensive computing. This result is somewhat surprising due to the interface limitations of real mobile-class systems. We discuss this result in more detail, and we follow that discussion with some of the system improvements that would be necessary to build a

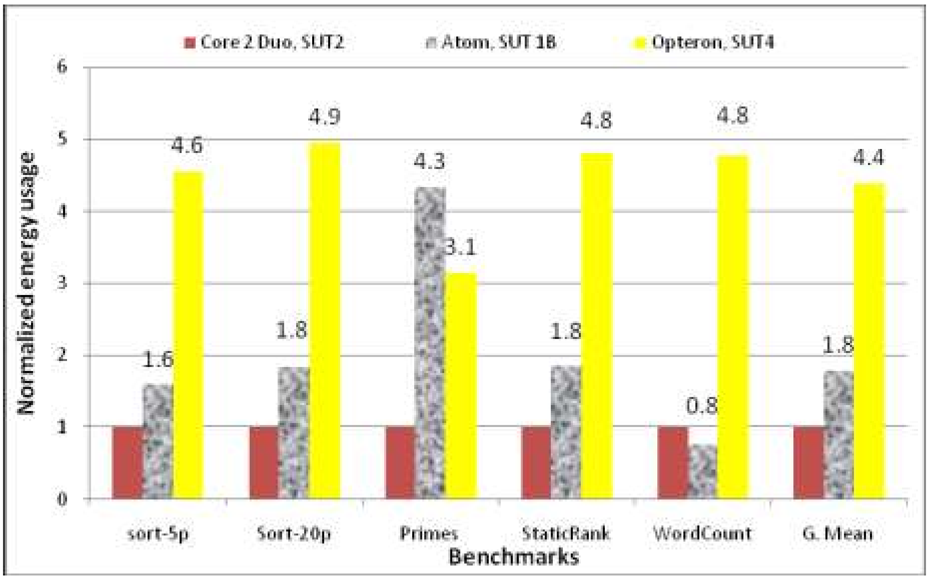

Fig. 4. Normalized average energy usage for SUT 2, SUT 1B, and SUT 4 for each benchmark on each system and the geometric mean.

more compelling energy-efficient system, requiring minor modifications to today's components.

5.1 Energy Efficiency

Our results show that low-power embedded components are not necessarily ideal for energy efficiency, even for applications that are not normally considered CPU-intensive (e.g. Sort). With the increase in I/O capabilities provided by SSDs, our results indicate that embedded-class processors are not always sufficient to balance the I/O bandwidth. In fairness, one disadvantage that these systems had is that the chipsets and other components dominated the overall system power; in other words, Amdahl's Law limited the benefits of having an ultra-low-power processor. As the non-CPU components become more energy-efficient, this type of system will be more competitive.

Our results also confirm that standard servers are becoming more energy-efficient. We presented results from three consecutive generations of Opteron servers running SPEC benchmarks. Over time, these systems have maintained or improved single-thread performance, increased system throughput, and simultaneously reduced overall system power and energy. Until recently, embedded systems were the only systems that exhibited the same trends. This is a result of combining lower-power server processors with efficient power supplies and related components. However, there still is a long way to go.

5.2 The Missing Links

Research on energy-efficient DC building blocks has largely been limited to evaluations of existing hardware. While simulation provides the flexibility to derive any reasonable system imaginable, the runtimes for the applications used in this study make simulation of any type prohibitively expensive. For this data-intensive benchmark suite, the wall-clock runtime varied from just over 25 seconds (WordCount on SUT 4) to ~1.5 hours (StaticRank on SUT 1B). Therefore, this study was constrained to use existing hardware. However, there are several clear improvements that could be made to increase the energy efficiency of future datacenter hardware.

First, the embedded and mobile systems had very restrictive I/O subsystems, limited by the number of ports and overall bandwidth. Likewise, the network is also a limiting factor, which can be solved with more energy efficient designs and higher bandwidth, like 10 Gb solutions.

Finally, only configurations 3 and 4 supported ECC DRAM memory. Memory is the conduit to the processor, and memory errors are on the rise, especially for large systems [21, 22]. We view ECC as a requirement for any data-intensive computing system.

Our ideal system would couple a high-end mobile processor (like the Intel Core 2 Duo or AMD equivalent) with a low-power chipset that supported ECC for the DRAM, larger DRAM capacity, and more I/O ports with higher bandwidth.

6 Conclusions

Our results from small clusters demonstrate that systems built using high-end mobile processors and SSDs are the most energy-efficient systems for data-intensive cluster computing across all the applications we tested. We compared systems across the spectrum of available hardware, including systems advocated by other researchers proposing solutions to this problem [11]. A concern with ultra-low-power embedded systems is that the chipset and peripherals can dominate the overall power usage, making these systems less energy-efficient than their processors alone. Our results also show that the successive generations of server systems are becoming more energy-efficient, as we expected. We were able to use single-threaded and single system benchmarks to filter the systems down to a tractable set in order to run a variety of large-scale benchmarks. The initial benchmark results were consistent with the data-intensive benchmark results. Moving forward, we expect that embedded processor systems will be overpowered by their I/O subsystem requirements for data-intensive applications in the near future. Furthermore, by optimizing the chipset and peripherals, even more energy-efficient systems can be built for this application space. These systems will use less power, reducing overall power provisioning requirements and costs.

Finally, there is a large body of future work that we would like to pursue. First, we would like to use OS-level performance counters to facilitate per-application modeling for total system power and energy. Furthermore, we know of no standard methodology to build and validate these models. Likewise, developing standard metrics and benchmarks will make these comparisons easier in the future.

References

1. United States Environmental Protection Agency Energy Star Program: Report on Server and Data Center Energy Efficiency. (2007)
2. Barroso, L.A., Hölzle, U.: The Datacenter as a Computer: an Introduction to the Design of Warehouse-Scale Machines. Morgan-Claypool, San Rafael (2009)
3. Koomey, J.G.: Estimating Total Power Consumption by Servers in the U.S. and the World. Analytics Press, Oakland (2007)
4. Poess, M., Nambiar, R.O.: Energy Cost, The Key Challenge of Today's Data Centers: a Power Consumption Analysis of TPC-C Results. Proceedings of the VLDB Endowment, vol. 1, no. 1, 1229--1240 (2008)
5. Barroso, L.A., Hölzle, U.: The Case for Energy-Proportional Computing. IEEE Computer, vol. 40, no. 12, 33--37 (2007)
6. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: 6th Symposium on Operating Systems Design and Implementation, pp. 137--150. USENIX, Berkeley (2004)
7. Hadoop Wiki, http://wiki.apache.org/hadoop/
8. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks. In: EuroSys Conference, pp. 59--72. ACM, New York (2007)
9. Thain, D., Tannenbaum, T., Livny, M.: Distributed Computing in Practice: The Condor Experience. Concurrency and Computation: Practice and Experience 17, 2--4 (2005)
10. Intel: Intel X18-M/X25-M SATA solid state drive product manual, http://download.intel.com/design/flash/nand/mainstream/mainstream-sata-ssd-datasheet.pdf
11. Szalay, A.S., Bell, G., Huang, H.H., Terzis, A., White, A.: Low-Power Amdahl-Balanced Blades for Data Intensive Computing. In: 2nd Workshop on Power Aware Computing and Systems (HotPower), online. ACM SIGOPS (2009)
12. Caulfield, A.M., Grupp, L.M., Swanson, S.: Gordon: Using Flash Memory to Build Fast, Power-Efficient Clusters for Data-Intensive Applications. In: 14th International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 217--228. ACM, New York (2009)
13. Andersen, D.G., Franklin, J., Kaminsky, M., Phanishayee, A., Tan, L., Vasudevan, V.: FAWN: a Fast Array of Wimpy Nodes. In: 22nd Symposium on Operating Systems Principles, online. ACM SIGOPS (2009)
14. Vasudevan, V., Andersen, D., Kaminsky, M., Tan, L., Franklin, J., Moraru, I.: Energy-Efficient Cluster Computing with FAWN: Workloads and Implications. In: 1st International Conference on Energy-Efficient Computing and Networking (e-Energy), pp. 195--204. ACM, New York (2010)
15. Beckmann, A., Meyer, U., Sanders, P., Singler, J.: Energy-Efficient Sorting Using Solid State Disks, http://sortbenchmark.org/ecosort_2010_Jan_01.pdf
16. Reddi, V.J., Lee, B.C., Chilimbi, T.M., Vaid, K.: Web Search Using Mobile Cores: Quantifying and Mitigating the Price of Efficiency. In: 37th International Symposium on Computer Architecture, pp. 314--325. ACM, New York (2010)
17. Rivoire, S., Shah, M.A., Ranganathan, P., Kozyrakis, C.: JouleSort: A Balanced Energy-Efficiency Benchmark. In: SIGMOD International Conference on Management of Data, pp. 365--376. ACM, New York (2007)

18. Lim, K.T., Ranganathan, P., Chang, J., Patel, C.D., Mudge, T.N., Reinhardt, S.K.: Understanding and Designing New Server Architectures for Emerging Warehouse-Computing Environments. In: 35th International Symposium on Computer Architecture, pp. 315--326. ACM, New York (2008)
19. Hamilton, J.: CEMS: Low-Cost, Low-Power Servers for Internet-Scale Services. In: 4th Biennial Conference on Innovative Data Systems Research, online (2009)
20. ClueWeb09 dataset, http://boston.lti.cs.cmu.edu/Data/clueweb09/
21. Schroeder, B., Pinheiro, E., Weber, W.-D.: DRAM Errors in the Wild: A Large-Scale Field Study. In: Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance), pp. 193--204. ACM, New York (2009)
22. Yelick, K.: How to Waste a Parallel Computer. Keynote address at 36th International Symposium on Computer Architecture (2008)