12-1-2003

Building Energy-Efficient Web Servers Using Low Power Devices: Opportunities and Challenges

Le Cai

BUILDING ENERGY-EFFICIENT WEB SERVERS USING LOW POWER DEVICES: OPPORTUNITIES AND CHALLENGES

LE CAI

TR-ECE 03-15
DECEMBER 2003

Building Energy-Efficient Web Servers Using Low Power Devices: Opportunities and Challenges

Le Cai[*]

School of Electrical and Computer Engineering

465 Northwestern Ave.

Purdue University

West Lafayette, IN 47907-2035

ii

Contents

List of Tables

List of Figures

Abstract

Power consumption has become one major challenge in designing high-performance servers. Recently, some researchers postulate that better energy efficiency can be achieved by using low power devices originally designed for mobile embedded systems. In this report, we validate this hypothesis by measuring the power and the performance of web servers on system level. We consider the power and the energy consumption of processors, memory, and hard disk drives. Our data indicate that, compared with a Pentium4-based web server, a XScale-based computer can achieve more than 1.7 times higher energy efficiency (web pages per Joule) while its throughput (web pages per second) is less than 20%. Our comparison includes both static and dynamic web pages with different sizes. We also indicate some challenges to design an energy-efficient high-performance web server by using this approach.

x

1   Introduction

Internet servers require high performance, reliability, scalability, low operational cost, easy deployment and maintenance. Unfortunately, today's Internet servers are too expensive, often hard to maintain, and consume too much power— contributing to power shortage and generator pollution [3]. For example, the Google data centers use thousands of computers and each center consumes nearly one third of a million Watts for computation (excluding cooling) [9]; a high percentage of cost is associated with energy and heat, including electrical bills, cooling, and heat-caused malfunction and repair. It is imperative to reduce the energy consumption of Internet servers.

Web servers are one type of Internet servers. Most web servers consist of high-performance processors and IO devices. These components consume most of power of web servers. Today's high-performance processors aggressively exploit instruction-level parallelism by performing out-of-order and speculative execution. As a result, a significant portion of power is consumed for checking data and controlling dependences, maintaining precise exceptions, or sometimes even executing unnecessary instructions. Most of today's processors are built with CMOS technology. The dynamic power consumption of a CMOS transistor can be formulated as $p = c \cdot v^2 \cdot f \cdot \alpha$, here c is the load capacitance, v is the bias voltage, f is the clock speed, and α is the switching probability [21]. Also, the gate delay is proportional to the reciprocal of the bias voltage (first-order approximation). Thus, the maximum clock speed f is proportional to the voltage v. Reducing the voltage and the frequency can achieve cubic reduction in power consumption, while the performance degrades only linearly with the frequency. Intel XScale [5] processor is an example of low-power / low-performance processors. These processors are usually used in portable systems, such as personal digital assistants (PDAs) and cellular phones. Even though their performance is lower than desktop or server processors, they have better energy efficiency (instructions per Joule). On the other hand, many high-performance IO devices in web servers, such as SCSI hard disk drives, consume considerable power because of their high speed rotation and high bandwidth. In contrast, some embedded IO devices have lower speed but their power consumption is much less. For example, IBM microdrive consumes less than 1 mW power for reading or writing one MB data [10].

Several projects are investigating how to build energy-efficient web servers using low-power / low-performance components. Even though this approach seems straightforward, there have been insufficient evidences to demonstrate the advantages and challenges. In this report, we compare the performance and the energy efficiency of two web servers: a Pentium4-based computer with an IDE hard disk drive and an XScale PDA-like computer with a microdrive. We consider the power consumed by processors, memory, and IO devices. The performance is divided based on the source of web pages: static or dynamic, in memory or on disk. Our experimental results indicate that the XScale-based web server has only 10% performance of the Pentium4-based server (web pages per second). However, the XScale server consumes only 6% of power, compared with the Pentium4 server. This makes the XScale server nearly twice more energy-efficient (web pages per Joule). The contributions of this report are three-fold. First, we present system-level

measurement results to compare the energy efficiency of the two different approaches in building web servers. Second, our results indicate how the comparison scales in different scenarios. Third, we indicate the challenges in building energy-efficient web servers using low-power / low-performance components.

This report is organized as follows. Section 2 surveys related work on reducing energy or power consumption and building energy-efficient servers. Section 3 describes our experimental environment and presents the experimental results. Section 4 discusses these experimental results and demonstrates challenges. Finally, Section 5 concludes this report.

2   Related Work

Figure 1: Matchbox web server and Linux watch.

Some studies have proposed different approaches to reduce the energy consumption of servers. In [19], Pinheiro at el. presented a shut-down policy for idle machines based on the traffic of incoming requests. An et al. compared how query types affected energy consumption [1]. Chase et al. proposed an economic model to charge processes that used excessive energy [4]. Another approach is to treat energy as a resource like CPU time or memory allocation [16, 20]. Meanwhile, some researchers suggested that high energy-efficiency could be achieved by using many slower processors, instead of few faster processors. In [6], Pedram compared the energy efficiency of Pentium III, Crusoe, and StrongARM and found that StrongARM was ten times more energy-efficient (MIPS per Watt) than Pentium III. However, their study did not include system-level comparison nor the comparison of different types of workloads. Researchers in IBM Austin Lab were building "dense servers" using multiple low-power / low-performance blades [13]. Similar efforts were taken by Intel [11] using low bias voltages.

Energy efficiency is important for portable systems. For example, XScale is a microprocessor often used in cellular phones or PDAs; it can execute up to 4×10^9 instructions per Joule [5]. A new PowerPC processor can achieve 500 DMIPS at 0.5 W [17]. For IO devices, a one-inch microdrive consumes only 3.4% of the power compared with a 3.5" SCSI disk [9]; the microdrive is three times more energy-efficient than the SCSI disk (5.25 MB/Joule vs. 1.72 MB/Joule). These data suggest that it is possible to build energy-efficient web servers using the components originally designed for battery-powered portable systems. Two recent prototypes demonstrated the feasibility of constructing servers using these components. The Matchbox PC was a web server in the size of a typical matchbox consuming less than two Watts [7]. In addition, a wrist watch could execute Linux and have networking capability [12, 15]. In [18], Phan et al. suggested adding PDAs to computing grids and utilizing the spare computation resources. Even though it seems intuitive to build energy-efficient servers using these components, few studies have been conducted in quantifying the energy savings at system level.

3 Measurement Results

	Servers		Client
	Pentium 4	XScale	Pentium 4
CPU (GHz)	1.8	0.4	2.26
Memory (MB)	256	64	256
Disk size(inch)	3.5	1	3.5
Disk capacity(GB)	60	1	40
Network (Mb/sec)	100	100	100
Linux kernel	2.4.18	2.4.18	2.4.18
Web program	Apache	Apache	Httperf

Table 1: Configurations of experimental systems.

Figure 2: Experimental setup.

3.1 Experimental Environment

Our experimental setup consists of three parts: (1) a Pentium4 PC and a XScale-based computer as web servers. The Pentium4 PC is a NetVista made by IBM and the XScale computer is an Integrated Development Platform made by Accelent. (2) a Pentium4 PC as a web client. (3) a Pentium4 PC to measure power consumption. Figure 2 and table 1 show the configuration of the setup. In the rest of this report, we use X and P to represent the XScale computer and the Pentium4 PC, respectively.

We assess the performance of the servers by their throughput: web pages per second. Energy efficiency is defined as pages per Joule: $\frac{\text{throughput (pages/sec)}}{\text{power (Joules/sec)}}$. We consider three different scenarios based on the types of web pages.

(1) static pages cached in memory

(2) static pages stored on hard disk

(3) dynamic pages produced by CGI (common gateway interface) programs

These scenarios represent three types of web requests. In the first scenario, a 10 KB web page stored in memory is served because typical web pages are about 10 KB [2]. We also study how the page size affects performance and power consumption. In the second scenario, the pages are retrieved from hard disks. In the third scenario, web pages are dynamically generated at run-time. We compare three C programs for generating the pages. Each program has two procedures: (a) integer operations[1] simulating computation for e-commerce or web searching and (b) a 2KB HTML page generated from the computation results. The three CGI programs have different numbers of integer operations. They have 0.2 million, one million, and two million multiplications respectively. The comparison under larger computation amount is also made by using CPU-intensive benchmarks.

In the web client, we use httperf [14] to measure the throughput and the response time of web servers. Httperf can create web requests at different rates and evaluate throughput and response time. We choose httperf instead of SPECWeb99 because the former provides more flexibility in adjusting the request rates. The measurement computer contains a data acquisition card from National Instrument. The card has 16-bit precision and 16 channels; the sampling rate can reach 200 thousand samples per second. In our experiments, we use 1000 samples per second as the sampling rate . We measure the total power consumption including the power of processor, motherboard, and hard disk; the power of the monitors is excluded.

	X	P
In memory	3.9	0.7
On disk	25.4	12.9
From CGI1	14.6	1.5
From CGI2	66.4	5.4
From CGI3	131.1	10.4

Table 2: Response time (ms) of X and P.

3.2 Response Time

Response time is crucial for interactive web services. Table 2 shows the response time of X and P in three scenarios. Even though P has better response times, X's response time— tens of milliseconds— is acceptable because it is only a fraction of the delay of the Internet [8] and will be hardly noticed by a user. As a result, we focus on the comparison of throughput in the following text.

---

[1]XScale has no floating-point unit.

(a) Static pages in memory (b) Static pages on disk (c) Dynamic pages

Figure 3: Power, performance, and energy efficiency ($\frac{\text{pages}}{\text{Joule}}$) of X and P. We use different scales for X and P in comparing power and performance to make the curves clearer. In figure(a) and (c), the disk power is excluded.

3.3   Static Pages in Memory

Our first set of experiments compares the energy efficiency when web pages are cached in memory. To guarantee that the requested pages are always in memory, we use Apache's *mod_mmap_static* module to map web pages into memory during initialization. Figure 3(a) shows the throughput and the power consumption of X and P. The power consumption excludes the part consumed by hard disk drives because no pages are accessed from hard disk drives. In the figures, P achieves the peak throughput of 1086 pages/sec and this is 9 times better than that of X. However, P consumes 15.5 times more power than X. Consequently, X is 1.7 times more energy-efficient when P and X are running at peak throughput. Both P and X become more energy-efficient as their throughput increases. The energy efficiency of X grows faster than that of P.

The above large difference of performance and power is attributed to both the frequency difference and the architecture difference between Pentium4 and XScale. Pentium4's clock frequency is 4.5 times higher (1.8 GHz vs 0.4 GHz) while the performance of P is nearly 9 times better. On the other hand, Pentium4 consumes more power than the value calculated by purely scaling the frequency and the voltage. Both situations can be explained by the architectural difference.

3.4 Static Pages on Disk

Figure 3(b) and Figure 4 show the measurement results when the web pages are stored on the hard disks. Figure 3(b) indicates that P's peak throughput is reduced to only 5 times higher. Hence, X's energy efficiency becomes 2.3 times better. In Figure 4, X's microdrive can support 44 pages/sec at only 0.9 Watt; P's disk consumes 8.5 Watts to provide 220 pages/sec. In other words, the microdrive is 1.89 ($\frac{44}{0.9} \div \frac{8.5}{220} = 1.89$) times more energy efficient. Compared with the previous scenario, the throughput drops substantially for both machines (63% for X and 80% for P). The performance of P is more sensitive to the slow disk accesses since it has a faster processor. The power consumed by X increases 33% while the power consumption of P remains comparable. This can be explained by the aggressive pipelining used in P's processor to achieve the high performance in the first scenario. When the performance is bounded by IO, the total power consumption excluding hard disk drives is reduced substantially (32.5-8.5=24 Watts at 220 pages/sec compared with 31 Watts at 1086 pages/sec in the first scenario). Adding the power consumed by the disk makes the overall consumption similar to the power consumed in the first scenario. In contrast, the overall power of X increases significantly when the hard disk's power is included.

(a) Power

(b) Energy efficiency

Figure 4: Disks of P and X

3.5 Dynamic pages

Dynamic pages introduce further performance degradation on both machines: 90% for P and 94% for X. However, X is still more energy-efficient than P but the difference becomes smaller. Figure 3(c) shows the throughput, the power consumption, and the energy efficiency for the three CGI programs. X's power curve for CGI1 has a small drop near peak throughput. This can be understood as the power consumption fluctuation because of saturation. The curves for X's energy efficiency overlap because the energy efficiency

	CGI1		CGI2		CGI3	
	X	P	X	P	X	P
TP	6.6	98.0	5.0	85.6	3.0	59.3
Power	1.9	38.3	1.9	41.9	1.9	43.3
EE	3.5	2.6	2.6	2.0	1.6	1.4
X:P	1.36		1.28		1.15	

Table 3: Energy efficiency of X and P for dynamic pages. TP: throughput; EE: energy efficiency.

is nearly identical for all three CGI programs. From the figures, P only achieves a throughput of 59 pages/sec for CGI3 but still consumes 43 Watts. Compared with the first scenario, (peak throughput of 1086 pages/sec and 31 Watts power consumption), CGI3 increases P's power consumption by 40%, but with 90% performance degradation. CGI programs made the processor busier and consume more power. This power consumption increase does not happen in X because the processor consumes small part of X's total power. We summarize the energy efficiency for CGI programs in Table 3.

	gzip		gcc		gauss elimination		quicksort	
computer	X	P	X	P	X	P	X	P
execution time(s)	20.5	0.9	28.5	1.3	88.9	5.6	49.6	2.5
power(Watt)	2.4	45.3	2.3	45.5	2.1	49.76	2.1	50.6
EE (task/Joule)	0.02	0.025	0.015	0.017	0.0053	0.0036	0.0098	0.0078
ETR	23.2		21.4		15.9		19.8	
EER	0.83		0.92		1.47		1.26	

Table 4: Performance and energy efficiency for 4 benchmarks. ETR: ratio of the execution time. EER: ratio of the energy efficiency.

3.6 CPU-Intensive Benchmarks

In this section, we study the energy efficiency when X and P run CPU-intensive benchmarks. As web servers are used for more versatile applications (sometimes called application servers), it is important to understand the energy efficiency for computation-intensive programs. Because X has no floating point unit, we choose four integer benchmarks for the comparison. X has *lower* energy efficiency in two of the four benchmarks. Table 4 compares their performance and energy efficiency. For these benchmarks, P is more than 15 times faster. This large performance difference counteracts X's low-power feature and makes X and P have close energy efficiency.

(a) Throughput (b) Power (c) Energy efficiency

Figure 5: The effect of page sizes.

3.7 The Effect of Page Sizes

Previously, we used 10 KB for the sizes of the static pages. In this section, we show the effect of sizes for static web pages. Figure 5 compares the throughput, the power, and the energy efficiency of different page sizes. As expected, the throughput decreases for larger pages. However, X is still more energy efficient. The figure also shows that the throughput increases more rapidly in P when page sizes become smaller. This is because the bottleneck is the network. In the Figure 5(b), we notice that X's power drops with the decrease of file size. This can be explained by the decrease of output byte. For example, when file size changes from 5KB to 1KB, the output byte reduces 600KB ($160 \times 5KB - 200 \times 1KB$).

4 Challenges

	Memory	Disk	CGI
X(pages/Joule)	59.4	14.9	2.6
P(pages/Joule)	35.0	6.5	2.0
X:P	1.7	2.3	1.3

Table 5: Summary of energy efficiency of X and P for the three scenarios.

(a) Static pages in memory   (b) Static pages in disk   (c) Dynamic pages(CGI1)

Figure 6: Energy efficiency under different scalability.

4.1 High Performance

X is more energy-efficient than P in most scenarios. As shown in Table 5, X's energy efficiency is 77% higher on average. However, P has better performance, in both throughput and response time. In order to achieve the same throughput, we may use clusters of multiple X-based servers. To compare the energy efficiency under same performance, we *extrapolate* our measurement results for a cluster of multiple Xs by assuming different scalability. We define scalability as the percentage of performance improvement when the number of machines doubles. For example, the scalability of 80% means that the performance will improve 80% with twice the number of machines. If the number of machines quadruples, the performance improvement will be $1.8^2 = 3.24$. Scalability lower than 100% indicates diminishing returns after adding more machines. Figure 6(a) compares the energy efficiency of P and X with different scalability when the web pages are stored in memory. In this figure, X is more energy-efficient with 80% scalability. This comparison suggests that it is possible to build clusters of Xs with desirable performance and better energy efficiency. A similar comparison is used for the energy efficiency when serving pages from the hard disks. Figure 6(b) shows that X has better energy efficiency for the same performance if the scalability is higher than 60%.

4.2 Dynamic Pages

Our experimental results show that X has poorer performance when serving dynamic pages. Since many web sites provide dynamic pages, it has become one major challenge for X to improve its relatively small throughput. The large peak throughput difference between X and P requires more Xs in a cluster to explore more task-level parallelism. Based on Figure 6(c), to achieve same performance, a cluster composing of 15 Xs needs to have 100% scalability. If the cluster cannot achieve the required scalability, more Xs need to be added and they will consume more power. It is possible, however, to improve the performance in several ways. For example, the CGI programs may be kept in memory to reduce the time for loading them from the disk. Another approach is to use faster processors for dynamic pages and slower, but more energy-efficient processors for static pages.

4.3 CPU-Intensive Computation

P is more than 15 times faster for CPU-intensive computation. This performance difference implies the difficulty to achieve high performance using Xs for computation-intensive applications. Meanwhile, X's energy efficiency becomes close to P's. It may be possible to use multiple Xs and explore task-level parallelism. Their power states may be individually controlled to match the required performance. This approach is yet to be investigated.

5   Conclusion

In this report, we present system-level measurements to compare the performance and the energy efficiency of two types of web servers. Our results indicate that it is possible to build a more energy-efficient web server by using low-power / low-performance components, such as a processor and a hard disk used in PDAs. Such a server achieves 1.12 to 2.3 times better energy efficiency than a Pentium4 based server. Finally, we illustrate several challenges introduced by this approach.

References

[1] N. An, A. Sivasubramaniam, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, and S. Gurumurthi. Analyzing Energy Behavior of Spatial Access Methods for Memory Resident Data. In *International Conference on Very Large Data Bases*, pages 411–420, 2001.

[2] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *The Proceedings of the ACM SIGMETRICS'96 Conference*, pages 126–137, May 1996.

[3] California Energy Commission. News Release, Research Funds for Energy Efficient Web Server Farms. www.energy.ca.gov/releases/2002_releases/, March 1 2002.

[4] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing Energy and Server Resources in Hosting Centers. In *ACM Symposium on Operating Systems Principles*, pages 103–116, 2001.

[5] L. T. Clark, E. J. Hoffman, J. Miller, M. Biyani, Y. Liao, S. Strazdus, M. Morrow, K. E. Velarde, and M. A. Yarch. An Embedded 32-b Microprocessor Core for Low Power and High Performance Applications. *IEEE Journal of Solid-State Circuits*, 36(11):1599–1608, November 2001.

[6] DARPA / ITO. Power Aware Computing / Communication Principal Investigator Meeting. www.dyncorp-is.com/darpa/meetings/paccmay00/, May 2000.

[7] G. Defouw and V. Pratt. The Matchbox PC: A Small Wearable Platform. In *International Symposium on Wearable Computers*, pages 172–175, 1999.

[8] J.-R. Gruser, L. Raschid, V. Zadorozhny, and T. Zhan. Learning response time for websources using query feedback and application in query optimization. *The VLDB Journal – The International Journal on Very Large Data Bases*, 9(1):18–37, 2000.

[9] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 3 edition, 2003.

[10] IBM. Inc. IBM 1G Microdrive Specification. http://www.hgst.com/, 2002.

[11] Intel. Press Release, Low-Power Server Products. www.intel.com/pressroom/archive/releases /20011113comp.htm, November 2001.

[12] N. Kamijoh, T. Inoue, C. M. Olsen, M. Raghunath, and C. Narayanaswami. Energy Trade Offs in the IBM Wristwatch Computer. In *International Symposium on Wearable Computers*, pages 133–140, 2001.

[13] C. Lefurgy. Super-Dense Servers: An Energy-efficient Approach to Large-Scale Server Clusters. www.research.ibm.com/people/l/lefurgy/Publications/, November 2002. Texas A&M University.

[14] D. Mosberger and T. Jin. Httperf: A Tool for Measuring Web Server Performance . In *Workshop on Internet Server Performance* , pages 59–67, June 1998.

[15] C. Narayanaswami, T. I. Kamijoh, M. Raghunath, T. Inoue, T. Cipolla, J. Sanford, E. Schlig, S. Venkiteswaran, D. Guniguntala, V. Kulkarni, and K. Yamazaki. IBM's Linux Watch: The Challenge of Miniaturization. *IEEE Computer*, 35(1):33–41, January 2002.

[16] R. Neugebauer and D. McAuley. Energy is Just Another Resource: Energy Accounting and Energy Pricing in the Nemesis OS. In *Workshop on Hot Topics in Operating Systems*, pages 59–64, 2001.

[17] K. J. Nowka, G. D. Carpenter, E. W. MacDonald, H. C. Ngo, B. C. Brock, K. I. Ishii, T. Y. Nguyen, and J. L. Burns. A 32-bit PowerPC System-on-a-chip With Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling. *IEEE Journal of Solid-State Circuits*, 37(11):1441–1447, November 2002.

[18] T. Phan, L. Huang, and C. Dulan. Challenge: Integrating Mobile Wireless Devices into the Computational Grid. In *International Conference on Mobile Computing and Networking*, pages 271–278, 2002.

[19] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Dynamic Cluster Reconfiguration for Power and Performance. *Compilers and Operating Systems for Low Power*, Kluwer, 2003.

[20] A. Vahdat, A. Lebeck, and C. S. Ellis. Every Joule is Precious: the Case for Revisiting Operating System Design for Energy Efficiency. In *ACM SIGOPS European Workshop*, pages 31–36, 2000.

[21] N. H. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. 1993.