

Profiling Energy Usage for Efficient Consumption

by Rajesh Chheda, Dan Shookowsky, Steve Stefanovich, and Joe Toscano

Summary

When measuring the effectiveness of application architecture, there are several key criteria by which it can be measured. First and foremost, is the architecture's ability to fulfill the requirements. These requirements often include specific, measurable items relating to cost, scalability, testability, and so forth. In the future, two new measurable criteria will become critical to evaluating application architectures: energy consumption/cost and environmental impact. But as we will see, the future is now.

The location: your corporate server room.

The time: present day.

The event: an unexpected visit from the "Software Architect of the Future."

There you sit, monitoring the performance of your latest line-of-business application freshly rolled out just 24 hours ago, when a strange visitor strolls in. She begins asking you about your enterprise architecture. Instead of calling security, you decide this is a chance to show off how cutting edge your organization is. You hit all the buzzwords: C#, SQL Server, WPF, WCF, and SilverLight. It scales up and out, is accessible from mobile devices, it slices and it dices. "Excellent," the Architect replies, "but what does your EUP look like?"

"My EUP? What's that?" The Architect explains that in the future, an Energy Usage Profile (EUP) is a fundamental part of every application design — through the use of a set of Energy Usage Profiles (EUPs), organizations of the future measure the ongoing energy costs and environmental impact of specific applications within the enterprise. "Sit down," she intones, "and I'll tell you all about it."

The Math

Before diving into specifics, a basic discussion of energy usage is in order. When measuring the electrical usage of any electronic component, simple math can be employed to assign a cost to that component. Let's begin with an example: the venerable 60 watt (W) incandescent lightbulb (common in your time but widely regarded as legacy hardware in the future). The electric company measures energy consumption in kilowatt-hours (kWh). Using our lightbulb as an example, if we were to leave it on for one hour, we would have consumed 60 watt-hours (or .06 kWh) of power ($60/1000 = .06$). Now say, for example, that the electric company charges \$0.0897 per kilowatt-hour (excluding delivery charges, capacity charges, taxes, and all other fees), that means that the lightbulb costs roughly \$0.0054 per hour to run ($0.06 * 0.0897 = 0.0054$). That may not seem like much but, if we were to leave that lightbulb on 24 hours a day, seven days a week, it would cost us \$47.15 to run annually ($0.0054 * 24 * 365 = 47.15$). Now consider that each kilowatt-hour of electricity used also contributes approximately 2.3 pounds of CO₂ to the atmosphere. So, using a few simple calculations we can state an annual cost (\$47.15) as well as an annual environmental impact (1,208 pounds of CO₂) for our little lightbulb. Going forward, we will be employing these formulae to calculate energy usage costs and environmental impact:

$kWh = Watts / 1000$

$Annual\ Energy\ Cost = kWh * 24 * 365 * cost\ per\ kWh$

$Annual\ CO_2\ Emissions = kWh * 24 * 365 * pounds\ of\ CO_2\ per\ kWh$

For the purposes of this discussion, we will use the values of \$0.0897 for cost per kWh and 2.3 for pounds of CO₂ per kWh. These values may vary depending on where you live and how your electricity is generated.

Developing an Energy Usage Profile for the Hardware

Unfortunately we're not talking about lightbulbs, we're talking about servers. So how does the energy usage of the average server compare to that of a 60 W lightbulb? Well, unless you're running your company on laptops, it's much worse. According to research published by Dell and Principled Technologies, a single Dell PowerEdge M600 blade server consumes an average of 383.75 W when idle and 454.39 W under stress (see Table 1).

Table 1: Sample Hardware Energy Usage Profile (in watts)

Component	Idle Power Usage	Average Power Usage	Maximum Power Usage
Server 1	383.75	454.39	600 (Estimated)
CPU	40.8		130
HDD	14.35		17
DIMM1	3		3
DIMM2	3		3
Video	18.3		25.6
Network Card	4.95		4.95
CD/DVD	2		18

Keep in mind that this figure can go up or down depending on workload. Using that baseline to recalculate our numbers, that one server costs \$301.54 per year to run. It also produces 7,731.8 pounds of CO₂. The picture gets even bleaker when you consider how many servers the typical application uses. For example, if you run SharePoint, are you running it on a farm? Maybe a farm with two Web front-end servers, an index server, and a database server? Suddenly the annual electricity cost increases to \$1,206.16 and the CO₂ produced increases to over 30,927 pounds. The cost and impact calculations should be directly included in any calculations of return on investment for an application.

Using the previously mentioned formula for Annual Energy Cost, we can build an Energy Cost Profile for the hardware, as shown in Table 2. Finally, we can also build an Energy Impact Profile for the hardware, as shown in Table 3.

Table 2: Sample Hardware Energy Cost Profile (in USD)

Component	Idle Annual Cost	Average Annual Cost	Peak Annual Cost
Server 1	\$301.54	\$357.41	\$471.95
CPU	\$32.06		\$102.15
HDD	\$11.28		\$13.36
DIMM1	\$2.36		\$2.36
DIMM2	\$2.36		\$2.36
Video	\$14.38		\$20.12
Network Card	\$3.89		\$3.89
CD/DVD	\$1.57		\$14.14

Table 3: Sample Hardware Energy Impact Profile (in pounds of CO₂)

Component	Idle Annual Cost	Average Annual Cost	Peak Annual Cost
Server 1	7,731.8	9,164.46	12,101.22
CPU	822.04		2,619.24

HDD	289.12		342.52
DIMM1	60.44		60.44
DIMM2	60.44		60.44
Video	368.70		515.79
Network Card	99.73		99.73
CD/DVD	40.30		362.66

The numbers for specific components are not easily measured but can be obtained from manufacturers' Web sites and Web sites that perform independent hardware research. The server's energy consumption is much easier to measure. Several low-cost devices are available to monitor energy consumption. The important thing is to build an overall picture of how much energy the server uses at idle and under stress. The energy usage of individual components does not need to be exact, but is important because it provides a ranking system for energy consumers within the server. In the example above, the CPU is clearly the largest consumer of energy.

Using the EUP for the hardware, we can draw some general conclusions. Clearly, CPU usage is the most expensive resource in terms of actual cost and environmental impact. Memory usage has a minimal cost at best. Hard disk usage has minimal cost. The gist is that if we are going to attempt to optimize our infrastructure and application to minimize energy usage, the CPU should be the primary target.

Developing an Energy Usage Profile for the Application

Dealing in averages may be fine if you're analyzing a large data center but what about the energy usage of a specific application's architecture? Average power consumption is not good enough. We want to know how much power a single application uses and, while we cannot get to the exact number, we can get pretty close. The first step is building an energy usage profile for the hardware.

As mentioned previously, the manufacturers' Web sites are a good place to start. While the manufacturer's figures can give you a good approximation for best and worst case scenarios for the hardware, it cannot tell you how much energy your particular application uses. In this case, we need to measure. Luckily, inexpensive products are available that will help measure the energy consumption of your application. These products, used with a good set of application load tests, can make the energy consumption picture much clearer. The key is to build a set of load tests that reflect both the average and peak loads for your application.

The idle consumption can be used to calculate a baseline cost for the application. When doing nothing, a three-server application has an idle energy consumption of 354 W (.345 kWh). The hourly cost is \$0.025875 and the annual cost is \$226.67. Through our EUP, we can thus calculate the baseline annual cost of the application. The measurements taken during average and peak application loads can then be used to calculate costs and impacts when the application is running at those levels (see Tables 4-6).

Table 4: Sample Application Energy Usage Profile (in watts)

Server	Idle Power Usage	Power Usage at Average Load	Power Usage at Peak Load
Server 1 – Application	304	425	525
Server 2 – Database	304	425	525
Server 3 – Web	304	425	525
Total Power Usage	912	1,275	1,575

Table 5: Sample Application Energy Usage Cost Profile (in USD)

Server	Idle Power Usage	Power Usage at Average Load	Power Usage at Peak Load
Server 1 – Application	\$238.87	\$333.95	\$412.53
Server 2 – Database	\$238.87	\$333.95	\$412.53
Server 3 – Web	\$238.87	\$333.95	\$412.53
Total Cost	\$716.61	\$1,001.85	\$1,237.59

Table 6: Sample Application Energy Usage Cost Profile (in pounds CO₂)

Server	Idle Power Usage	Power Usage at Average Load	Power Usage at Peak Load
Server 1 – Application	6,125	8,562.9	10,577.7
Server 2 – Database	6,125	8,562.9	10,577.7
Server 3 – Web	6,125	8,562.9	10,577.7
Total Cost	18,375	25,688.7	31,733.1

Once an application-level EUP has been created, specific areas of the application can be targeted for improvement.

Remember that what is high priority in terms of performance improvement may not always be high priority for energy usage reduction. For example, an operation that does a long write operation to a physical disk might be seen as a high priority performance issue. In terms of energy usage, however, this operation might be low priority because the physical disk consumes less energy than the CPU. An operation that utilizes a large amount of CPU time might be a higher priority. This is not to say that CPU-intensive operations should be replaced at the cost of application performance. It simply gives us a ranking system for determining which components of the application can be optimized to realize the largest cost savings. The hardware EUP will tell you which physical components of the hardware carry the most cost and impact. Using hardware EUP, we know that CPU utilization carries the largest impact in terms of power usage, and should therefore be our first target.

Tools such as Visual Studio 2008 Profiler can be used to determine precise CPU usage for specific components of the application. These tracking statistics can then be used to attack the high cost portions of the application. In other words, find the most expensive operations and optimize them to reduce CPU usage. The ultimate goal is to lower resource usage to the point where multiple applications can be hosted on the same set of servers. Server sharing can reduce the energy footprint of all the applications involved.

Developing an Energy Usage Profile for the Operating System

Application-based optimizations are not the only way to save energy. Optimizing the operating system is just as important. The base hardware EUP also provides us with an idle operating system EUP. Looking at the data collected for physical servers when idle and at peak loads, we can see that a significant amount of energy is wasted on the system idle process. This wasted energy can be reclaimed through operating system virtualization. Virtualization allows the host machine to run at approximately 80 percent of peak processor utilization with fractionally increased power requirements. Table 7 translates these potential benefits into numbers.

Table 7: Potential Benefits

	Annual Cost
Average Cost of Server:	\$6,500

Average Power Cost:	\$301
CO ₂ Produced	7,731.8 pounds annually

Each physical server replaced with a virtual equivalent represents a significant savings, both in upfront costs as well as with respect to the environment. To put the carbon footprint in perspective, the average car produces 20 pounds of CO₂ per gallon of gas consumed. To maximize the number of virtual guest machines on a single host, optimization is a key requirement. This optimization includes minimizing disk and network utilization, as well as reducing the memory footprint.

The first step in tuning the virtual guest is to disable or deactivate unneeded services. Depending on the type of guest operating system and its purpose, there may be a number of unnecessary services running by default. Table 8 lists services which may be safely shutdown depending on your circumstances.

In addition to the services listed above, you should eliminate screen savers and evaluate your event logging requirements for each guest operating system. This will avoid wasted processor cycles and disk activity. Minimizing disk activity is essential to both performance and energy efficiency. For that reason, you should use fixed size virtual hard disks to avoid the need to resize the Virtual Hard Disk (VHD) file. If you don't require extensive logging, you can further reduce disk activity at the expense of data necessary for later analysis.

If you have the necessary RAM, you can use a RAM drive to cache frequently accessed data. A RAM drive trades increased memory utilization for improved performance. The advantage is that RAM drives don't have to hit the physical disk, providing energy efficiency in addition to the aforementioned performance.

Look closely at the network utilization of your applications with a network monitoring tool to locate and eliminate chatty protocols and unnecessary network communication. Wireshark is an effective and freely available tool for network analysis.

On the host server side, usage of the Windows Server 2008 Server Core installation will minimize the files and services used by the host operating system.

These optimizations reduce the amount of work done by the guest operating system as well as by the host. Less work translates into lower energy consumption.

Developing an External Energy Usage Profile

The impact of your application goes beyond CPU cycles and megabytes. The way end users interact with the application has a very real cost and impact. The actions taken within your application that are external to your design are measured in the External EUP. Think through the way end users will utilize a given application and try to anticipate the impact of those interactions. Following are examples of actions that should be tracked in an External EUP:

- **Printing:** Are users likely to print information? If so, how much? Are there ways to optimize screen layout to minimize the likelihood of printing?
- **Information Retrieval:** Are statements, invoices, or other materials mailed to users? Is there a way to present that information within the system?
- **Updates:** Do users need to leave their systems on to receive updates? If so, is there a way for the application to update itself while being used?
- **Emailing:** Purging bad email addresses eliminates energy wasted attempting to send messages.

While harder to quantify, these impacts are important because simple measures can be taken to mitigate them. For example, adding an on-screen print view might virtually eliminate the cost of paper, electrical power and toner usage for printers as well as the carbon production from both.

Table 8: Services that may be disabled in a virtual machine

Name	Description
Automatic Updates	This service is not necessary if you have disabled Windows Update in favor of manual patch management.
Background Intelligent Transfer Service	This service is not necessary if you have disabled Windows Update in favor of manual patch management.
Clipboard	This has nothing to do with the clipboard or with the ability to cut and paste.
Error Reporting Service	This service can be safely disabled on a server.
Help and Support	
Indexing Service	This service may be required depending on the server's purpose.
Messenger	Messenger services are unnecessary on servers running in lights-out mode.
Portable Media Serial number	Music players are not necessary on a server.
Print Spooler	Unnecessary if you aren't printing from the server.
Smart Card	Not needed if smart cards aren't being used.
Task Scheduler	Unnecessary on most servers.
Themes	Themes are unnecessary on a server machine.
Uninterruptible Power Supply	The host server can manage UPS. This service is not needed in a virtual machine.
Web Client	
Windows Audio	Audio is unnecessary on a server.
Wireless Configuration	This service is unnecessary on both physical and virtual servers.

Conclusions

It is not just the Architect of the Future who can use the methods described in this article. You can (and should) consider the energy usage when designing an application. Part of being an Architect is understanding how every aspect of your application functions. Energy usage is another key aspect and an energy usage profile is the tool for understanding it. Having a useful energy usage profile means:

- Understanding your hardware and its energy usage.
- Understanding your operating system and its energy usage.
- Understanding your application and its energy usage.
- Understanding the external impacts of your application.

Together, these areas of understanding combine to give you a total energy usage profile that you can act upon. More importantly, they give you a barometer for tracking the real cost of your application and the real gain of your optimization. The future of application architecture is looking greener.

Resources

Average Retail Price of Electricity to Ultimate Customers by End-Use Sector, by State, Energy Information Administration, October 3, 2008

http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html

Frequently Asked Global Change Questions, Carbon Dioxide Information Analysis Center
<http://cdiac.ornl.gov/pns/faq.html>

Performance and Power Consumption on Dell, IBM, and HP Blade Servers, Principled Technologies,
December 2007
http://www.dell.com/downloads/global/products/pedge/en/pe_blades_specjbb2005.pdf

About the Authors

Rajesh Chheda is a consultant at RDA with more than 10 years of experience designing and developing business solutions using Microsoft tools and technologies. He has worked on software solutions for real estate finance, lending, procurement. His current focus is on designing, customizing and deploying collaboration solutions for his clients using Microsoft SharePoint technology. He can be reached at chheda@rdacorp.com.

Dan Shookowsky is a senior software engineer with RDA. He has 15 years of experience in IT consulting with a significant focus on validated applications for the pharmaceutical industry running on Microsoft Windows platforms. He can be reached at shookowsky@rdacorp.com.

Steve Stefanovich has spent more than a decade with RDA as a developer and principal architect focusing on custom application development using Microsoft .Net and SharePoint technologies. Steve is a frequent contributor to RDA's Architecture and Collaboration blogs. He can be reached at stefanovich@rdacorp.com.

Joe Toscano currently is a SQL Server BI specialist with RDA and has worked with SQL Server since version 6.5 and with Sybase prior to that. Joe has both published materials and presented locally for the Philadelphia SQL Server User Group and numerous .NET Code Camps sessions, and internationally at The Professional Association for SQL Server. He can be reached at toscano@rdacorp.com.