

COMPUTER POWER CONSUMPTION BENCHMARKING
FOR GREEN COMPUTING

A Thesis Presented to the Faculty of
the Department of Computing Sciences
Villanova University

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science

in

Computer Science

by

Mujtaba Talebi

April 2008

Under the Direction of
Dr. Thomas P. Way

ACKNOWLEDGMENTS

This work could not have been performed without the guidance of my research advisor, Dr. Thomas Way, who provided valuable direction, feedback and enthusiastic support, and was truly both a mentor and a friend. I would like to thank my friend and colleague Christopher Continanza for our past collaborations on power consumption and Green Computing which led me to get started on this thesis. I would like to thank Dr. Robert Beck, Dr. Lillian Cassel, and the rest of the Villanova University Department of Computing Sciences which provided me not just research assistant funding but the opportunity to be challenged and to accomplish more than I thought I could. Most of all, sincere thanks are due to my family and friends. Without their encouragement and help this work would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
ABSTRACT.....	vii
1. Introduction.....	1
2. Green Computing Background.....	4
2.1 Types of Computing Energy Use.....	5
2.2 Interpreting Computer Population Data.....	6
3. Techniques for Managing Power Consumption.....	9
3.1 Energy Saving Primer.....	9
3.2 Screen Savers.....	12
3.3 Monitor Sleep Mode.....	12
3.4 Hard Disk Sleep Mode.....	13
3.5 System Stand By.....	13
3.6 Hibernate.....	14
3.7 Phantom Loads.....	15
4. Bottom Up Electrical Efficiency Improvement.....	17
4.1 The Tiered Approach.....	17
4.2 Lower Tier – Component Level Efficiency.....	18
4.3 Middle Tier – Desktop, Notebook, and Server Level Efficiency.....	19
4.4 Upper Tier – Network Level Efficiency.....	19
4.5 Implementing Bottom Up Electrical Efficiency Improvements.....	21
5. Computer Power Use Benchmarking.....	23
5.1 General Computer Benchmarking.....	23
5.2 Measuring Power Consumption.....	24
5.2.1 Techniques.....	24
5.2.2 Kill A Watt Power Consumption Meter.....	26
5.2.3 watts up? Power Consumption Meter.....	30
5.3 Creating Practical and Repeatable Benchmarks.....	31
6. Evaluation of Power Benchmarks.....	33
6.1 Implementation of Benchmarks.....	33
6.1.1 Internet Web Surfing Benchmark.....	34
6.1.2 3D Gaming Benchmark.....	36
6.1.3 System Power Mode Benchmarks.....	38
6.1.4 Distributed Application Benchmark.....	39
6.2 Experimental Methodology.....	40
6.3 Results.....	43
6.3.1 Internet Web Surfing Benchmark – Villanova AutoSurf Benchmark.....	43
6.3.2 3D Gaming Benchmark – Team Fortress 2.....	45
6.3.3 System Power Mode Benchmarks – Off, Stand by, Hibernate and Idle.....	48
6.3.4 Distributed Application Benchmark – Folding@home.....	50
6.4 Analysis.....	51
7. Conclusion.....	54
Appendix A. Villanova AutoSurf Benchmark.....	62
Appendix B. 3D Gaming Benchmark.....	67
Appendix C. Folding@home for Windows.....	68
Installation and Configuration.....	68
Appendix D. Folding@home for Linux.....	69
Installation and Configuration.....	69

LIST OF TABLES

Table 1. A Table of PC Stock Projections in California for 2005-2012.....	7
Table 2. Comparison of Electricity Usage Costs of A Desktop and Laptop Configuration With Different Use Patterns.....	11
Table 3. Villanova AutoSurf Benchmark Test Results (Watts used in 5 minutes).....	44
Table 4. Villanova AutoSurf Benchmark Test Results (Appr. Watts used in 1 Hour).	44
Table 5. Team Fortress 2 Benchmark Test Configurations.....	46
Table 6. Team Fortress 2 Benchmark Test Results.....	47
Table 7. System Idle Benchmark Test Results (Watts Used in 3 Minutes).....	49
Table 8. System Idle Benchmark Test Results (Appr. Watts Used in 1 Hour).....	50
Table 9. System Off, Stand by, and Hibernate Benchmark Test Results (Appr. Watts Used in 1 Hour).....	50
Table 10. Folding@home Benchmark Results (Appr. Watts Used in 1 Hour).....	51

LIST OF FIGURES

Figure 1. Power Options in Windows XP.....	15
Figure 2. P3 INTERNATIONAL KILL A WATT.....	27
Figure 3. P3 INTERNATIONAL KILL A WATT Specifications.....	29
Figure 4. Photo of a “watts up?” Power Consumption Meter.....	30
Figure 5. Example of a benchmark test.....	41

ABSTRACT

As population has increased, energy use has also increased. The widespread use of technology, particularly computers, means that computer power consumption is becoming a more important topic as the cost of energy rises and pollution, together with its side-effects, is recognized as an urgent challenge. Within computer science, benchmarks that measure various aspects of performance are common and widely used in the literature, yet benchmarking in order to measure power consumption of computers has not received the same attention.

In this thesis, techniques for accurately measuring the impact of techniques that can be used to reduce the power consumption of computers are explored. Pertinent background information is presented to put power benchmarking within the proper context of Green Computing, and common techniques for reducing power use are surveyed. Power benchmarking tools and techniques are introduced, a number of new benchmarks are presented and evaluated. An analysis of these techniques and their impact is provided, demonstrating the importance of benchmarking as a key metric in the area of Green Computing.

1. INTRODUCTION

Computer power consumption is becoming a more important topic as electricity prices climb and as pollution is becoming a bigger problem in the world. It is common knowledge that most of the world's power plants emit pollution as they generate electricity. As computers become more powerful and plentiful, their electrical demand increases which creates a need for more pollution generating power plants. The overriding goal of this paper is to try to make some contribution to the computing sciences which will raise awareness and ideally reduce computing consumption and pollution caused by powering computers.

One area of the computing sciences that is starting to become more important is computer power consumption benchmarking. *Benchmarking* is a general and widely known approach where a standard approach, or *benchmark*, is used to measure some desired aspect, behavior or other characteristic of the thing being observed. In computing, benchmarks are typically computer programs that are run on a system, enabling accurate and repeatable measurement of some specific characteristic of interest.

Recently, various computer hardware review websites like Tomshardware.com and Anandtech.com have begun including power consumption results with their other computing results. This extra effort to include power consumption benchmarks shows that readers are becoming more interested in how much power our computing equipment can consume. While the early computing power consumption benchmarks show a lot of promise, there is little standardization going on among the various computer hardware review websites. Each hardware review site has their own methods for measuring power

consumption. This makes cross site comparisons difficult for readers and researchers, alike.

One of the primary goals of this thesis is to develop a computer power consumption benchmarking toolkit. This toolkit is free for anyone to use and offers a standard approach to computer power consumption benchmarking. Our computer power consumption benchmarking toolkit contains benchmarks for different computing use patterns including web browsing, gaming, distributed computing, and various computer power states including Idle, Stand by, Hibernate, and Off modes. Computer users make use of computers in different ways so it should not be expected that a one-size-fits-all benchmark would accurately represent how much power every user would use.

Another goal of our benchmarking toolkit is that it can be used to raise awareness of computer power consumption. If people become aware that computer power consumption is a growing problem, they will be more able to deal with the problem. Unless a problem is known and understood, it is much more difficult to deal with it and devise solutions for it. Our benchmarking toolkit makes the task of understanding computer power consumption more straightforward.

This thesis is organized in several chapters, each of which builds on its predecessor. First, we provide a green computing background which provides convincing evidence that the green computing movement is just and worthwhile. Next, we review matters of practical computer power consumption reduction. Benchmarking on its own cannot reduce power consumption. This chapter provides an overview of how individuals can reduce power consumption of personal computers both at home and at places of

business. The fourth chapter discusses bottom-up efficiency improvement. While the bottom-up model has been applied to other areas of the computing sciences, we believe our approach is unique to green computing and provides meaningful insight into the measurement and subsequent reduction of power consumption. The fifth chapter is dedicated to a discussion of our benchmarking toolkit and how it can be used to measure power consumption. The sixth chapter evaluates the toolkit benchmarks, presenting results and analysis of a number of benchmarking experiments. Conclusions are presented in the final chapter which demonstrate the efficacy of our benchmarking approach to the field of green computing and illustrate the benefits of the resulting power reduction.

2. GREEN COMPUTING BACKGROUND

Green computing is a general term describing a facet of computing that is interested in improving energy efficiency and reducing waste in the full life cycle of computing equipment [1]. The computing life cycle includes the energy consumed to create computing equipment, get the computing equipment to a consumer, used to run and maintain the computing equipment and discarding/recycling of computing equipment at the end of their life cycle [1]. Computing equipment can range from desktop personal computers, laptops, servers, networking equipment, cabling, and more. Green computing is an important realm of the computing sciences because of the significant demand computing requires of our resources.

The computing life cycle includes pollution in the form of carbon dioxide, lead and other toxic materials [2]. The carbon dioxide pollution occurs at power plants where electricity is generated to power computers [2]. These same power plants also emit mercury emissions and pollute our land and our water [3]. The more inefficient our computers run, the more electricity and pollution is generated by these power plants. At the end of a computer's life cycle it is either recycled or dumped into a landfill. If computers are recycled improperly (as has been the case in poor third world countries, [4]) or if the computer is simply dumped into a landfill, lead and other toxic substances are leached into the ground and water tables. It is imperative that we carefully examine ways of reducing the environmental burden of computing to prevent or reduce the pollution caused by computers.

2.1 TYPES OF COMPUTING ENERGY USE

The University of California Energy Institute described direct energy use, indirect energy use and their relation to greenhouse gas emissions in a paper called “An Analysis of Measures to Reduce the Life-Cycle Energy Consumption and Greenhouse Gas Emissions of California's Personal Computers” [1]. In the University of California paper direct energy use and greenhouse gas emissions (GHG) are defined as energy use and greenhouse gas emissions attributable to the operational electricity consumption of California's PC stock [1]. This definition is also applicable to any state, country, or the entire world in general. Essentially direct energy use and greenhouse gas emissions come from the electricity requirements required to power a computer. The greenhouse gas emissions come from power plants that emit varying levels of greenhouse gas emissions as they convert natural resources to electricity[5].

The amount of greenhouse gas emissions created by computer direct energy use depends on what source that computer obtains its electricity. The amount of greenhouse gas emissions and pollution emitted by a power plant depends on its fuel source. An example that may not be obvious is that coal power plants generate radioactive pollution than nuclear power plants [6].

“As of 2005 it is estimated that 19 million PCs were installed in California homes and businesses, and this number is expected to grow significantly through 2012 [1].”

This many PCs require significant amounts of electricity to operate. The previous quote only estimates the amount of PCs in California. If we look at the world as a whole there are nearly 1 billion computers out there [7]. The environmental impact of computer

power consumption becomes clearer as we try to understand how much electricity computers consume as a collective.

The University of California Energy Institute describes indirect energy use and greenhouse gas emissions as energy use and greenhouse gas emissions attributable to activities that support ongoing PC stock maintenance, namely, the manufacture of new PCs and PC components as needed, and the end-of-life treatment (i.e., waste disposal and recycling) of obsolete PCs and PC components [1]. In lay man's terms indirect energy use and GHG emissions are incurred when computers are built, repaired, maintained, shipped, sold, recycled, and disposed of. Indirect energy use and GHG emissions are typically not thought of when a person imagines the energy costs of using a computer. People tend to only look at the direct energy cost from their local utility company. A computer's indirect energy use and GHG emissions are a significant part of its complete energy life cycle. Indirect energy use and GHG emissions need to be looked at in order to have a more complete understanding of what kind of impact computers make in the world.

2.2 INTERPRETING COMPUTER POPULATION DATA

The real focus of the University of California Energy Institute paper is to characterize the effectiveness of their data collection measures to inform PC-related policies for near-term energy efficiency improvements and GHG emissions reduction in California [1]. They gathered data (Table 1) about California's residential PC stock from information from the U.S. Census Bureau as well as other surveys [1]. When taking data from different sources and comparing them to each other there can be some variation. In

order to try to provide an accurate picture of what the PC stock projections look like in California, the authors tabulate the data into low and high scenarios [1]. These low and high scenarios provide boundaries which show the reader that they can be confident that the real PC stock lies somewhere in between these boundaries.

Table 1. A Table of PC Stock Projections In California for 2005-2012 [1].

Year	California Population	Low Scenario				High Scenario			
		Total PCs	Notebook PCs	Desktop PCs		Total PCs	Notebook PCs	Desktop PCs	
				w/ CRT	w/ FPD			w/ CRT	w/ FPD
2005	37,005,000	12,102,422	2,987,441	5,651,521	3,463,460	12,398,645	3,060,563	5,789,849	3,548,233
2006	37,461,449	12,251,703	3,263,952	4,521,725	4,466,026	12,626,549	3,363,814	4,660,069	4,602,666
2007	37,917,899	12,400,984	3,585,559	3,563,154	5,252,271	12,856,280	3,717,201	3,693,973	5,445,106
2008	38,374,348	12,550,265	3,948,239	2,790,313	5,811,714	13,087,838	4,117,356	2,909,832	6,060,650
2009	38,830,798	12,699,546	4,348,121	2,166,738	6,184,688	13,321,222	4,560,973	2,272,805	6,487,444
2010	39,246,767	12,835,588	4,776,365	1,648,788	6,410,436	13,542,466	5,039,408	1,739,589	6,763,469
2011	39,703,216	12,984,869	5,238,323	1,227,347	6,519,199	13,779,424	5,558,860	1,302,449	6,918,114
2012	40,159,666	13,134,150	5,725,155	880,400	6,528,596	14,018,208	6,110,514	939,659	6,968,035

When we look at the California PC stock projections, there are several important trends to see. California's population is predicted to rise by hundreds of thousands of people each year [1]. As children get older and students graduate there will be a need for families, schools, and businesses to purchase more computers to accommodate them. If we look at the “Total PCs” columns we can see that PCs purchases are projected to increase by hundreds of thousands each year [1]. As new PC purchases are made electricity demand will increase.

Fortunately the rate of notebook purchases in California is higher than the rate of PC purchases [1]. Notebooks tend to use significantly less electricity to operate than a PC and monitor [see Results section 6.3 comparing desktop and notebook power consumption]. Another positive trend we see is that CRT purchases are expected to drop each year while while flat panel monitor purchases are increasing [1]. Flat panel LCD

monitors use less electricity [8] than CRT monitors, and are much smaller [8] which means they require less resources to build. In addition to the resource savings flat panel LCD monitors contain less mercury than CRT monitors do [9].

Once manufacturers begin to replace compact fluorescent back lights in flat panel monitors with LEDs, we will get one step closer to having mercury free monitors. It is important to mention that it is common knowledge that mercury is a toxic substance to humans and animals. When monitors are thrown away in the regular trash or recycled improperly, the mercury can find its way into our atmosphere, land and waterways [10].

3. TECHNIQUES FOR MANAGING POWER CONSUMPTION

Understanding the ways in which power consumption impacts the greenness of any technology, and specifically computing technology, is essential in order to find ways to reduce that consumption. There are a variety of techniques available to manage power consumption that are standard on most computers on the market today. These techniques involve changing the settings that control the behavior, and therefore power consumption, of various software and hardware components.

3.1 ENERGY SAVING PRIMER

One of the primary goals of this research is to create practical and repeatable benchmarks for computer power consumption. The reason for this goal is to ultimately assist people in choosing more efficient computing equipment. That being said, the efficiency of currently running machines can be significantly improved without affecting usability by following some of the following guidelines. The University of Colorado in Boulder released an article for their faculty, staff, and students which is aimed at improving awareness of green computing, computing power efficiency, and reducing computing waste [11].

The University of Colorado has around 18,000 computers on campus [11]. These computers cost the university up to \$550,000 in annual computing energy costs [11]. The \$550,000 estimate only accounts for the direct electricity cost to power the computers [11]. The estimate does not include the costs of cooling university buildings which is required due to the heat generated by the computers [11]. The estimate increases to around \$700,000 when cooling is taken into account [11]. The financial concerns with

computing power usage are clear but there is also the environmental concern. CO₂ gas and other pollutants are released into the atmosphere by power plants that generate electricity[12]. CO₂ is believed to be a leading cause for humanity's contribution to global warming [2]. Pollutants such as sulfur, mercury, and other pollutants contaminate our environment from these same power plants[12]. Reducing the electrical demand by adhering to the following guidelines will reduce the amount of harmful pollution generated by power plants.

According to the University of Boulder article, a typical desktop system is comprised of a computer, monitor and a printer [11]. The computer may require approximately 100 watts of electrical power. A 15-17" CRT monitor may require 50-150 watts of electrical power [11]. A conventional laser printer can use up to 100 watts or more during printing and can suck down several watts while it is idle [11]. An ink jet printer can use around 12 watts while printing and 5 watts during idling [11]. There are several scenarios which can occur on a college campus. A laptop computer may use around 35 watts of power during idling, but does not require a separate monitor because it has a built in screen. A desktop computer may use 100 watts of power idling, but may also use more power if the monitor is left on.

Table 2 illustrates some approximate and typical computing energy costs. The cost of electricity in the southeast Pennsylvania suburbs is around 15 cents per Kilowatt Hour. A Kilowatt Hour is a unit of measurement used to measure electricity use for 1000 watts in an hour. The cost of electricity varies between areas. The cost of electricity per kilowatt hour can be replaced below with your local rates.

Table 2. Comparison of Electricity Usage Costs of A Desktop and Laptop Configuration With Different Use Patterns.

Scenario	Desktop PC 100 Watts, 19" LCD 50 Watts	Laptop PC 35 Watts
Worst Case 24 hours per day usage, 365 days per year	150Watts/Hour * 8760Hours * 1Kilowatt / 1000 Watts * \$0.15/ Kilowatt = \$197.10 total cost for 1 year of use	35Watts/Hour * 8760Hours * 1Kilowatt / 1000 Watts * \$0.15/ Kilowatt = \$45.99 total cost for 1 year of use
24 hours per day usage, turned off during weekends	150Watts/Hour * 6264Hours * 1Kilowatt / 1000 Watts * \$0.15/ Kilowatt = \$140.94 total cost for 1 year of use	35Watts/Hour * 6264Hours * 1Kilowatt / 1000 Watts * \$0.15/ Kilowatt = \$32.89 total cost for 1 year of use
8 hours per day usage, turned off when not in use	150Watts/Hour * 2080Hours * 1Kilowatt / 1000 Watts * \$0.15/ Kilowatt = \$46.80 total cost for 1 year of use	35Watts/Hour * 2080Hours * 1Kilowatt / 1000 Watts * \$0.15/ Kilowatt = \$10.92 total cost for 1 year of use

Several points are demonstrated by this data. We can tell immediately that by doing something as simple as shutting off your PC when it is not in use, significant energy and financial savings can be made. Over the course of a year around \$50 is saved when turning off the computer/monitor only on the weekend, and around \$150 is saved when turning off the computer/monitor on weekends and on weekdays when the computer is not in use.

Even more substantial savings can be seen while following some of the power savings guidelines discussed later (Stand by, Hibernate) in addition to turning off the computer outside of an eight hour work day. When comparing a task completed on a desktop PC to a standalone laptop PC, there is a huge difference in power usage. For example even if you leave your computer on 24 hours a day 365 days a year, you can save around \$150 per year by switching to a laptop and following the same usage pattern.

The laptop in our study generally used about six times less electricity than our desktop PC and monitor use.

3.2 SCREEN SAVERS

Screen savers are a bit of a misnomer in green computing. A general misconception is that screen savers use less electricity. This is actually far from correct. Screen savers will use up just as much or more power as if your computer was idle [13]. Sometimes a complex screen saver or a 3D screen saver may use more power than your computer would normally use when idling. Screen savers can prevent the monitor and CPU from going to sleep which can waste a significant amount of power [13].

Screen savers do not put computers in an energy saving state. One common thought about screen savers are that they were designed to keep monitors from suffering from “burn-in” [13]. Burn-in occurs in monitors when the same image is displayed on the screen for a long period of time [13]. Burn-in has little affect on modern LCD monitors [13]. Plasma television screens can suffer from “burn-in” but they are generally not used with computers. As a rule of thumb, screen savers should be disabled. It is better to allow the monitor to fall asleep than to have a screen saver turn on after a period of idling.

3.3 MONITOR SLEEP MODE

Allowing the monitor to fall asleep after idling for some time period is a good start in improving energy efficiency[14]. A monitor falling asleep or entering Stand by generally means the same thing. The monitor screen will go blank and no light will emit from the screen. There is usually a green light on the monitor that shows the user that the

monitor is turned on. When the monitor is in sleep mode, the green light usually becomes amber.

Experiments performed for this research showed that a Dell 20" widescreen LCD uses around 55 watts of power when it is turned on. When the LCD goes into a sleep state, the power usage drops to around 3 watts resulting in significant energy savings. Using the monitor sleep mode can result in significant carbon dioxide reductions [14]. This mode can be changed in an operating systems power options control panel. Major operating systems such as Microsoft Windows [14], Linux, FreeBSD, and Mac OS X have power options which allow users to modify the monitor sleep feature as well as other energy saving features we will discuss in the upcoming sections.

3.4 HARD DISK SLEEP MODE

Similar to the monitor sleep mode, a computer can place its hard disk drives in a sleep mode when they are not in use [15]. Hard disk drives on desktop computers can use 10 watts or more power when in use [16]. Hard disk drives on laptop computers tend to use less than desktop computers but the energy savings on a laptop maybe more valuable, especially when the laptop is running on battery power. Some desktops, workstations, and servers have multiple hard disk drives. Hard disk drives that are not in use can be placed into Stand by mode while hard disk drives that are in use can be left turned on. The operating system manages this automatically for the user.

3.5 SYSTEM STAND BY

System Stand by is one of the most useful power saving features computers have to offer. After a preset time period of idling, a computer will shutdown most of its

components [17] giving us significant energy savings. The memory will remain active so that whatever the user was working on will still be there when the computer wakes up from Stand by mode [17]. The desktop computer that we tested used more than 100 watts idling and used as little as 5 watts when in Stand by mode. That is more than 20 times less electricity used than when idling. Another advantage of the system Stand by mode is that the computer can wakeup within a few seconds. This is much faster than shutting down a computer completely and booting it back up. Since the computer's state is saved in the memory which still consumes power in Stand by mode, the computer will wakeup with everything the user was working on prior to going into Stand by mode [17]. Another useful method of using Stand by is to configure the computer's power button to send the computer into Stand by mode instead of shutting it down. This feature will allow a user to send a computer into Stand by immediately, rather than waiting some time period idling before it is automatically put into Stand by mode. Configuring a computer's power button to act as a Stand by button can be accomplished by changing the computer's power options in the control panel.

3.6 HIBERNATE

The Hibernate feature is similar to the system Stand by mode. The Hibernate feature can be enabled via the interface shown in Figure 1. Hibernate goes one step further than Stand by and completely turns off the computer [17]. Prior to turning the computer off, the Hibernate feature will save the memory state onto the hard disk drive [17]. When the computer comes out of Hibernate mode, it will access the memory file which it stored on disk before hibernating and copy it back into memory. The desktop we

tested used around 3 watts of power in Hibernate mode vs the 5 watts of power the Stand by feature uses. Why does the computer use 3 watts of power if Hibernate turns the computer completely off? The cause is due to phantom loads, which will be described in the next section. A disadvantage of the Hibernate feature is that it takes slightly longer to Hibernate or wake up than it would to enter Stand by mode or wake up from Stand by [17]. This is because it takes several seconds to save or load the memory's state onto the hard disk drive.

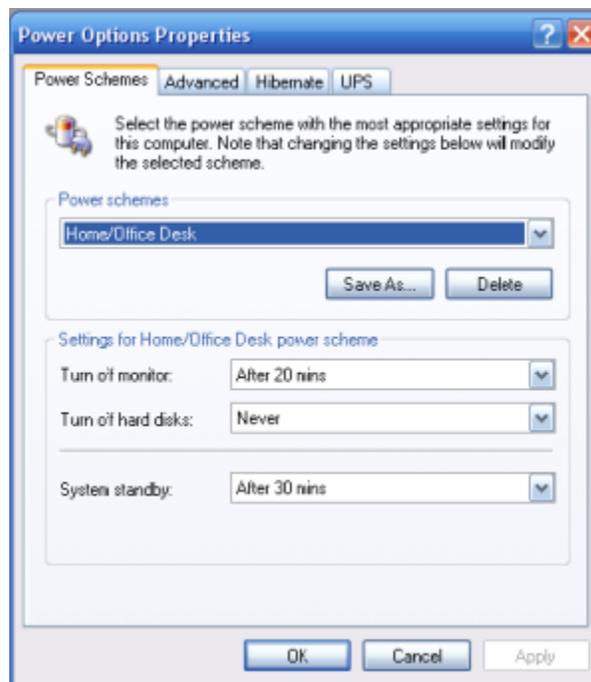


Figure 1. Power Options in Windows XP

3.7 PHANTOM LOADS

Phantom loads occur when electrical devices are turned off but still drain electricity from a wall outlet [18]. Many electrical devices that do not have a physical shut off switch that breaks the electrical connection to a wall socket will continue using

power when connected to an outlet [18]. Computers for example will use a little bit of power, typically around 1-3 watts, because of wake up on LAN functionality, constant power drawn from AC/DC adapters, and more. Wake up on LAN allows a completely shut off computer to be turned on remotely from a machine on its network. The computer that is shut off uses a little bit of power so that it can sense a wake up signal to its network card.

Phantom loads can cost a lot of money if left unchecked. If a user has 10 electrical devices that use as little as 3 watts when turned off, approximately the following amount of power is used over a year:

$$(10 \text{ devices} * 3\text{W} * 24 \text{ Hours} * 365 \text{ Days} * 1 \text{ KW} / 1000 \text{ Watts}) = 263 \text{ KW Hours.}$$

At the rate of \$0.15 per KW/H, this ends up costing almost \$40 over the year. This results in significant electrical waste for devices that are not in use. A solution to the phantom load is to pull the power plug when the electrical device is not in use [18]. If this is too much of a hassle, the devices that create phantom loads can be connected to a power strip. When these devices are not in use, the power strip can be turned off with the flick of a switch.

4. BOTTOM UP ELECTRICAL EFFICIENCY IMPROVEMENT

The bottom up electrical efficiency improvement works by improving the efficiency of smaller components and then moving up to improving larger components. This approach is similar to the bottom up approach to programming. The bottom up approach to programming aims to divide a program into small pieces [19]. Each part of the program must be large enough to do something substantial but small enough to stand-alone [19]. One of the primary benefits of the bottom up programming approach is that it is easier to work with smaller pieces. Similarly, the bottom up electrical efficiency improvement tackles small pieces at a time. Each component alone may bring in some electrical efficiency improvement but when each piece is looked at as a whole the improvements become more significant. We have devised a bottom up efficiency model which can be followed and modified if needed for unique situations.

4.1 THE TIERED APPROACH

Our bottom up model consists of a three tiered approach to improving overall computing efficiency. The lower tier represents the individual components in a computer. The middle tier represents the computer as a whole. The upper tier represents computers in groups such as in local area networks, wide area networks such as the Internet, distributed networks, grids, and more. All tiers may not be applicable to every computing environment. For example a home computer may not be connected to the Internet and may not benefit from improvements in the upper tier. Conversely a computer that is a member of a distributed network will benefit from improvements made in all tiers.

4.2 LOWER TIER – COMPONENT LEVEL EFFICIENCY

The lower tier concentrates on improving electrical efficiency at the individual component level. Examples of components are processors, memory, hard disk drives and other components that make up desktops, laptops, and servers. Improving the efficiency at the component level can be accomplished by replacing less efficient hardware with more efficient hardware.

Computers are comprised of many different components and each component does not consume an equal amount of power. Computer components include the central processing unit (CPU) [20], motherboard, memory, hard disk drives, optical drives, input devices, video cards, fans, power supply unit (PSU) [21][22], light emitting diodes (LED), and accessories. These devices have varying degrees of power usage. CPUs are generally the most power consuming component of computers. The type of load they use differs from that of devices like a hard disk drive. CPU power use tends to fluctuate greatly depending on the load that is placed upon it. Hard disk drives on the other hand tend to have a lesser differences in their various power states. This is due to a couple of reasons. Hard disk drives use much less power so the minimum and maximum power consumption of a hard disk drive are closer to each other than a CPUs minimum and maximum power consumption. A typical hard disk drive might use around 0 watts when it is shut off and 10 watts when it has a heavy load placed upon it. A CPU may use around 20 watts when it is idling or 90 watts when it has a heavy load placed upon it. Improving the most power hungry components such as the CPU [20] first will lead to better electrical efficiency. Of course the cost of replacing each component will be different so the cost to improvement ratio should also be taken into account when

upgrading components. It may be more worthwhile to complete inexpensive upgrades that result in greater power savings per dollar than expensive upgrades that result in less power savings per dollar.

4.3 MIDDLE TIER – DESKTOP, NOTEBOOK, AND SERVER LEVEL EFFICIENCY

The middle tier concentrates on improving electrical efficiency at the desktop, notebook, and server level. The primary way to improve the electrical efficiency of a complete system is by using the power saving features of the systems operating system. As was discussed in chapter 3, there are many ways to save electricity with operating system power management features. Turning a computer and monitor off when the system is not in use will prevent the waste of a lot of electricity. Pulling the computer's plug from the wall socket or using a power strip to turn off power to the computer will cut electricity use by 100%, thereby eliminating the machine's phantom loads. Using the Hibernate and Stand by feature can eliminate most of a computer's power draw when a computer is not in use. Hibernate and Stand by will also allow the machine to turn on quicker than a cold boot. Enabling the hard disk drives and monitor to go to sleep when the system is idle will also bring additional efficiency improvements.

4.4 UPPER TIER – NETWORK LEVEL EFFICIENCY

Network level efficiency can improve computing electrical efficiency at the group level. Groups of computers that are networked are usually used by a common organization or used for a common purpose. Many times these computers are left on all day long and never shut down. This leads to a lot of electrical waste and pollution. The

following are several examples for improving electrical efficiency at the upper tier or network level.

For a local area network, a network administrator can automatically turn off computers using the operating system management features or computer BIOS settings. When the computers need to come back up, a wake up on LAN signal can be sent to the machines in order to turn them back on. This makes sense not only from an electrical efficiency perspective but also from a management perspective. It is much easier for a network administrator to have his network's machines automatically shutdown and then be brought back up with the push of a button rather than going to each machine and individually turning them off and on.

A distributed network or a grid could benefit from having unused machines automatically power down or enter Stand by when the distributed network or grid is not making use of them. As soon as the load level increases, the sleeping machines could be woken up by LAN by a load balancing server. Internet level electrical efficiency can be improved for situations where a user leaves their machine turned on all night or all day to complete a download while they are away. A better method would be to have the web browser or download manager configured so that the machine turns off automatically after the download is completed. More research needs to be completed to flesh out all of the different scenarios where electricity usage can be optimized.

4.5 IMPLEMENTING BOTTOM UP ELECTRICAL EFFICIENCY IMPROVEMENTS

Each tier in the bottom up efficiency model builds upon the other. Using the middle tier electrical efficiency improvements builds on the energy savings gained on the lower tier component level, and the upper tier builds upon the techniques used in the lower and middle tiers. Although it would sound logical to start at the lower tier efficiency improvements and move to the higher tiers, this is not the most practical way to improve computing electrical efficiency. One reason not to start at the lower tier and move up is because all of the lower tier energy efficiency improvements involve buying new hardware components to replace old components. This does not provide a good cost to efficiency ratio and it would send a lot of hazardous waste to the dump.

The bottom up electrical efficiency model is aimed at giving a clear vision of how to improve efficiency at different levels of computing. Though it would make less sense intuitively, rearranging the bottom up approach so that the lower tier includes power management savings and the upper tier includes replacing hardware would be more practical. Power management features save electricity, money, and reduce pollution immediately with less investment and less waste. After these improvements are made computing equipment should be used for their full life cycle and then upgraded. Sometimes following an inherent order does not make real world sense, as is the case with our bottom up electrical efficiency model.

After the computing equipment has been upgraded as much as is practical and useful, then it is time to begin thinking about replacing entire machines. Consumers gain

experience from using power management features and upgrading to more efficient components and extending the life cycle of current equipment. With this experience consumers ideally will have a much better understanding of what kind energy efficient computer to purchase.

5. COMPUTER POWER USE BENCHMARKING

Understanding how to benchmark computers and measure power consumption is important in order to benchmark computer power consumption and improve computer electrical efficiency. Our goal is to create a reliable and practical benchmarking toolkit with low cost power consumption meters. There are a variety of tools and metrics available to measure computer power consumption and we evaluate them next in order to gain an understanding for the choices made when creating our benchmarking toolkit.

5.1 GENERAL COMPUTER BENCHMARKING

General computer benchmarking can usually be completed by using software tools or scripts. This is advantageous because everyone with a computer can download a free benchmarking program and run it on their machine to see how well their computer performs on that particular benchmark. There are benchmarks available to test a computer's performance in many areas including CPU throughput, memory bandwidth, video frames per second and more. General computer benchmarking has evolved over the past several decades. Products such as 3DMark2006 [23] can be downloaded for free and used to benchmark a computer's graphics system. The benchmark's results can then be compared to online databases which contain the results of thousands of other machines. Unfortunately benchmarking for green computing does not come close to general computer benchmarking in terms of advancement, simplicity and freely available material. Ideally efforts like the creation of our computer power consumption benchmarking toolkit will push the envelope on computer power consumption benchmarking.

5.2 MEASURING POWER CONSUMPTION

There are a variety of computer power consumption benchmarking techniques in practice today. Despite there being many available techniques, there is no universal one size fits all technique that is appropriate for every benchmarking situation. Different methods of benchmarking must be completed for different usage patterns. Additionally, measuring computer power consumption is different than general computer benchmarking because a tool is usually required to measure how much electricity is being consumed by a running machine.

General computer benchmarks such as CPU or video benchmarks generally do not require any special tools. Since power consumption benchmarks require electricity measuring devices, it makes it harder for people to participate in the power consumption benchmarking area. The requirement and cost of a tool adds a burden to those who wish to run their own computer power consumption benchmarks. We discuss two electricity consumption meters in section 5.2.2, the affordable **KILL A WATT** meter and the more expensive **watts up? PRO es** meter.

5.2.1 TECHNIQUES

There are two fundamental ways of measuring power consumption: measuring power consumption at one moment in time and measuring power consumption over time. Each method has its pros and cons. Measuring power consumption over one moment in time is useful when measuring a device that is using a constant amount of power. Less time is needed to take a sample when measuring a device one moment in time rather than when measuring a device over some period of time. A disadvantage of measuring over

one moment in time is that if the device being measured has power fluctuations, you may not get an accurate average of how much power the device uses.

For example a refrigerator may use a lot of power when the compressor is activated but it may use very little power when the compressor is turned off. It may use a little bit more power when the refrigerator door is opened and the light turns on. The power consumption of a refrigerator fluctuates over the day. In order to get a better idea of how much power a refrigerator consumes, it is important to measure the load over some time period that represents typical refrigerator use patterns. In the case of a refrigerator it may be necessary to measure a weeks worth of use in order to get a clear picture of how much power the refrigerator consumes on average. Similarly computers fluctuate in their power consumption as they carry out various tasks.

When comparing a computer's overall power usage, you must compare power usage over time [24]. Simply measuring a computer's power consumption when it is idling will not factor in how much power maybe consumed while it has a load placed on it. A good way of comparing two systems is to compare real tasks on each system identically from start until finish[24]. This reveals actual power requirements for tasks. It also exposes differences between systems. Here is an example:

“Lets say you are comparing system A that uses 45 watts of power under load and system B that uses 55 watts of power under load. If system B finishes the same task as system A in half as much time (because it is a better performing CPU), system B is actually a more power efficient system for the benchmarked task than system A even though it uses more

power under load. Since it takes a lot less time to finish, less power is spent [24].”

Although Idle benchmarks can be compared between different reviewers and magazines for identical systems, the load benchmarks depend on the specific tests conducted by the reviewers. If two reviewers have identical systems but have different benchmarking and measuring schemes, the load benchmarks cannot be compared. Luckily Idle should be comparable because in order to reach an idle state, the system has to be left doing effectively nothing or very little. The systems being compared should come installed with the same hardware, operating system, service packs, patches, software, and background processes. This would ensure that comparisons could be made between benchmarks performed on two different systems.

An interesting future project would be to create a bootable CD such as a Linux live CD and include a test suite for measuring power efficiency. With a Linux live CD, many users could download an ISO image and run the same exact set of benchmarks on their machines while measuring power usage. This could help improve repeatability of benchmarks and allow users to be assured that, except for drivers, they have the same exact software and processes running as other machines that boot off of the same CD.

5.2.2 KILL A WATT POWER CONSUMPTION METER

For this research, two power meters were used. The first is called KILL A WATT (Figure 2) and is a simple meter made by P3 INTERNATIONAL[25].



Figure 2. P3 INTERNATIONAL KILL A WATT [25].

In order to use KILL A WATT, you must plug it into the wall and then plug a device into KILL A WATT. KILL A WATT can be connected directly to the wall outlet or can be connected to an extension cable or power strip which is then connected to the wall outlet. While conducting this research, we found it easier to work with KILL A WATT with an AC extension cable so that it could be positioned closer to the experimenter for easier viewing.

KILL A WATT has several modes including Volt (voltage), Amp (amperes), Watt (wattage), VA (volt-amperes), Hz (hertz), PF (power factor), KWH (kilowatt hours), and Hour (the length of time the device has been connected to a power source) [25]. The voltage tends to approximately 120 volts for most devices and appliances.

Some heavy appliances like electric stoves and electric dryers use 240 volts of alternating current (VAC). This research deals with the standard American wall plug voltage of 120 VAC. KILL A WATT is not able to be used with 240 VAC wall outlets.

The Amp mode records amperes, a unit of electric current or amount of electric charge per second [26]. The VA mode measures volt-amperes, the SI unit of apparent power measurement [27]. The Watt mode displays how many watts are being consumed through the device at the moment of time the wattage value is recorded. This value tends to fluctuate with most devices as the electrical loads vary. The wattage is what we were mostly concerned with in this research. The Hz mode measures hertz or how many times the alternating current cycles per second (usually 60 times) [25]. The PF mode displays Power Factor [25]. The KWH mode displays the amount of kilowatt hours consumed since the KILL A WATT meter was connected to a power source [25]. This mode is very useful in determining how much power a device or appliance uses over some time period. The last mode is the “Hour” mode, which displays the amount of time the KILL A WATT meter has been connected to a power source [25].

Measuring power use over time is preferred because this type of measurement will capture all the fluctuations of power usage as the electrical load varies. Measuring power consumption by looking at the amounts of watts used at any given moment in time is not representative of true power consumption of devices whose loads vary. A computer is a good example of a device whose load varies. It can be idling one moment, using some processing power another moment as the user browses the web, and then its processing could be maxed out due to participation in a distributed application like [Folding@home](#) [28]. For the purposes of this research, the “KWH” mode was not

sensitive enough to record electricity consumption over time for our benchmarks. Our goal was to create benchmarks which would realistically represent how much power a computer would use in a variety of situations. Without the ability to measure power used over time, we would not have been able to achieve our goal.

One of our primary goals was to create a benchmark which would not take too long to complete, to both encourage its use and to allow the completion of this research in a reasonable time frame. In order to measure the amount of power that an electrical device would use over a 10 minute benchmark, we would need a mode that would measure the amount of watt hours consumed by the electrical load. Most devices would take hours or even days to use one kilowatt of power. The functionality (Figure 3) and reasonable cost (approximately \$25 US at the time of this writing) of the KILL A WATT meter makes it a good choice for entry level power benchmarking.

Specifications:	
Model:	P4400
Operating Voltage:	115 VAC
Max Voltage:	125 VAC
Max Current:	15 A
Max Power:	1875 VA
Weight:	5 oz.
Dimensions:	5 1/8"H x 2 3/8 "W" x 1 5/8 "D

Specifications subject to change without notice.

Figure 3 - P3 INTERNATIONAL KILL A WATT Specifications [25]

5.2.3 WATTS UP? POWER CONSUMPTION METER

The watts up? power consumption meter (Figure 4) features the same functionality as the KILL A WATT meter and more. In addition to the KILL A WATT features, the watts up? meter can record power measurements over time in watt hours sensitively enough to use for computer power consumption measurements. In addition to recording the amount of power consumed over time, the meter comes with memory which saves the data that is recorded during measurements. The data can then be accessed by loading software from the watts up? website [29] and connecting the meter to a Windows computer with an included USB cable.

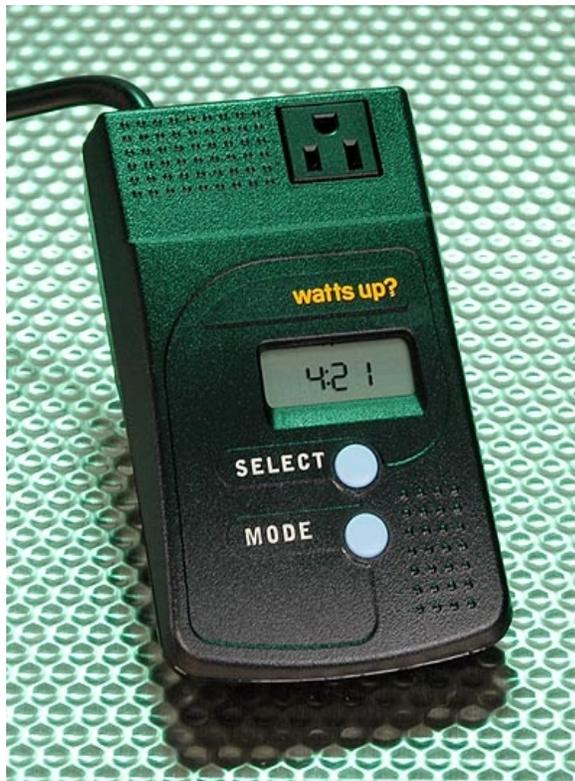


Figure 4. Photo of a “watts up?” Power Consumption Meter
(<http://www.smarthome.com/9056.html>).

There are several watts up? meter models available. The model that we tested was the watts up? PRO es model which included extra memory for saving measurements that are taken over long periods of time. The watts up? meter has several advantages over the KILL A WATT meter. It has more complex features than the KILL A WATT meter such as more sensitive measurements and memory for data storage. It can connect to a Windows machine with a USB cable and transfer data which can then be graphed and used for analysis. It has an extension cable that gives the user a lot of leeway to position the meter in a viewable spot. The watts up? power consumption meter's functionality made it a great tool for serious power consumption benchmarking.

There are also several important disadvantages to note about the watts up? meter. The watts up? meters cost several times more money than the KILL A WATT meter, around \$90-\$220 as of the time of this writing. The watts up? meters are more complicated and might be harder for some people to learn how to use. Despite these drawbacks the watts up? meter was the better bet for our research in order to carry out serious computer power consumption benchmarks. Without a watt hour measuring option on a power meter, it is not possible to accurately measure how much power a computer uses when its power fluctuates unless the benchmarks last for hours or days.

5.3 CREATING PRACTICAL AND REPEATABLE BENCHMARKS

Practicality and repeatability are important considerations when creating a power consumption benchmark. General computer benchmarks have already done well with repeatability. Programs like 3DMark06 [23] provide repeatability for computer video benchmarks. Each 3DMark06 benchmark runs exactly the same on each run.

Unfortunately, 3DMark06 benchmarks are not practical in the sense that 3DMark06 is not a real game. 3DMark06 is a benchmark that simulates gaming by running several gaming like graphics modes that are not used in any practical context outside of benchmarking.

Synthetic benchmarks do not give any inherently practical information but are somewhat useful for comparing systems against each other. If after running 3DMark06 on two systems and one system gets the score “5000 3DMarks” and the second system gets the score “5500 3DMarks”, these results do not say much except that one system performed better than the other in an instance of a 3DMark06 benchmark. Benchmarking a game like Team Fortress 2 [30] gives meaningful results in that Team Fortress 2 is actually played by people. This is why one of our main requirements for our benchmarking toolkit was to create practical benchmarks.

Repeatability was another main requirement for our benchmarking toolkit. Repeatability is critical in order to compare one benchmark run against another. If our benchmarks are not repeatable than the results of two runs on the same system with the same configuration might be different. A benchmark like this is not useful. General computing benchmarks like 3DMark06 do a great job at being practical and we try to follow their example in making our benchmarking toolkit repeatable.

6 EVALUATION OF POWER BENCHMARKS

Evaluating the benchmarking toolkit we created is equally important to the research and time spent creating the toolkit. The best way to prove that these benchmarks are relevant in the real world is to use them in real world systems. First a discussion of the benchmarking requirements and design are necessary to gain an understanding into why and how the benchmarking toolkit was created. Second a set of real system benchmark results and analysis is provided to (hopefully) show that the benchmarks are easy to use and consistent with the goals of this thesis.

6.1 IMPLEMENTATION OF BENCHMARKS

The power consumption benchmarks for green computing that we implemented must pass the following criteria:

- 1) The benchmark should be straightforward to use. This means that there should not have to be any complicated configuration files to configure and each iteration of the benchmark should be easy to run. Benchmarking computers requires at least a few iterations to be run using the same test state (i.e., repeating the test on the Idle power state) in order to obtain averages. These averages help assure the tester as well as the reader that the benchmark was run enough times to raise confidence that the results are consistent. In addition to running multiple iterations of the same test state, we need to run multiple iterations on several different test states (i.e., Idle, gaming, office, full load). This could mean that more than dozens of benchmarks might need to be completed in order to thoroughly benchmark a particular system. In order to support multiple iterations of benchmarks on the various test states, it is important for testers to be able to run tests without having to repeat configuration or typing in a lot of commands. Using shortcuts and simple commands are an ideal way of kicking off benchmarks.
- 2) The benchmark should be reasonably inexpensive to obtain. None of the benchmarks described in this paper will have any direct cost associated with them. Unfortunately there are indirect costs that are out of our control that may make it difficult to run some of the benchmarks. For example, the tester needs to have possession of a Microsoft Windows based computer to run some of our tests. Usually most computers come with Microsoft Windows so this may not be a problem. Another indirect cost would be the cost of the software which the particular benchmark runs from. Our gaming benchmark runs off of a proprietary game called Team Fortress 2. Team Fortress 2 cost around \$30 at the time of this writing. Although \$30 may seem like a lot of money to run a benchmark, this benchmark will be unique in that the results closely mirror results that would be recorded if the actual game would be played. Therefore we believe that the expense is worth the realism of the results.
- 3) The benchmark should be reasonably practical. There are synthetic benchmarks that can be downloaded on the Internet for free which give no inherently useful results such as 3DMark06

[23] (discussed in section 5.3.3). The reason we argue that synthetic benchmarks give no inherently useful results is because synthetic benchmarks are not programs which are used in the real world. Synthetic benchmarks attempt to perform tasks similar to real world program but their results cannot be interpreted clearly on their own. Synthetic benchmarks are somewhat useful when comparing one system to another since at least you can see if one system can complete the benchmark better than the other. Unfortunately a difference in performance of a synthetic benchmark may not actually result in any perceivable difference when using a real world system with a real world program.

We have attempted to create a benchmark toolkit which can test a broad range of computer uses. Computers are used differently by different people so it is important to understand the usage pattern of a particular population when creating a benchmark. A person who surfs the Internet and plays 3D games will benefit from seeing the results of a web browsing benchmark and a 3D gaming benchmark but will probably not benefit from seeing the results of a distributed application benchmark. We describe each of our benchmarks in the upcoming sections.

6.1.1 INTERNET WEB SURFING BENCHMARK

The Internet web surfing benchmark attempts to simulate the power consumption of a machine whose user is surfing the Internet through a web browser. The Villanova AutoSurf benchmark is an HTML and JavaScript based benchmark. Using HTML and JavaScript is ideal because they allow the benchmark to be browser neutral. All major web browsers support HTML and JavaScript. Since all major graphical operating systems have web browsers, this benchmark is also operating system neutral.

Dr. Thomas Way developed our Internet Web Surfing Benchmark and dubbed it the “Villanova AutoSurf Benchmark” [31]. In order to run a benchmark using the Villanova AutoSurf Benchmark, you must use an electricity consumption meter that can measure watt hours. The reason we need to measure watt hours is because when a user is browsing the Internet, the power consumption of the computer fluctuates. At one moment

the browser maybe loading a page which requires CPU, memory, and hard disk drive activity. At that moment there is a spike in power consumption. When the user is reading the content on a page, the computer maybe idle or using a small amount of processing power and memory access to scroll down a page. These fluctuations in power consumption need to be measured and averaged over time so that we can get an accurate picture of the computer's true power consumption.

Clearly a tester cannot measure the power used at any given moment and get a sense of how much power is consumed when a user is web browsing. An average must be taken by taking into account power consumption over time. This is why we need a power consumption meter that measures in watt hours. A watt hour is simply the amount of watts used in an hour of time. This does not mean our benchmark must be run for one hour. We can run the benchmark for 5 minutes and multiply the watt hours recorded by minutes in an hour (60) by the amount of time in minutes we spent running the benchmark (5 minutes) in order to eliminate the hours from the equation and end up with the average watts consumed.

The watts up? electricity consumption meter we used has a watt hour function that we used for the Villanova AutoSurf Benchmark tests. The KILL A WATT meter has a kilowatt hour function that was not really useful because computers needs to be on for hours or days in order to use even one kilowatt hour of electricity. This means that the KILL A WATT meter is not sensitive enough to be used for the Villanova AutoSurf benchmark. As long as watt hours or a similar sensitive function is used, the Villanova AutoSurf benchmark can be used for a few minutes and give results that can be

approximated to how much power a computer consumes when a user surfs the Internet.

The Villanova AutoSurf benchmark browses web pages from a list of hundreds of web sites. These web sites offer a variety of content including text, pictures, flash, movies, audio, web engine searches, and more. The Villanova AutoSurf Benchmark has an option which allows the user to configure the interval in which a new page is loaded. In order to run a power benchmark with the Villanova AutoSurf benchmark, follow these steps:

1. Have the computer and monitor, or laptop connected to a power measuring device which measures in watt hours or a similar power measurement over time.
2. Start the computer and load the Villanova AutoSurf page in a web browser.
3. Type in a time interval for page loads. The default is 5 seconds.
4. Start the benchmark by clicking the “Start Surfing” button.
5. Record measurements after the amount of time required for the benchmark is reached.

Once the watt hours are recorded, you can convert to average watts by multiplying the watt hours value by $[60 \text{ minutes (minutes in an hour)} / n \text{ minutes (amount of minutes spent benchmarking)}]$.

6.1.2 3D GAMING BENCHMARK

The 3D gaming benchmark is designed to give a benchmark user an accurate measurement of how much electricity is consumed when a computer user plays a game. The selected gaming benchmark is Team Fortress 2 [30], a game developed by Valve Corporation for the Microsoft Windows platform. Team Fortress 2 has not been released by Valve Corporation for other operating systems besides Windows but can be run using WINE (Wine Is Not an Emulator) [32] on other operating systems such as Linux. More information about running Team Fortress 2 under WINE can be found at the WINE application database [33].

The Team Fortress 2 benchmark was created by utilizing an in game feature called a time demo or simply a demo. Essentially a demo of actual game play is created and can then be played back for us to observe power consumption. In order to record a demo the console feature must be enabled in the keyboard advanced settings menu. The console can then be opened by pressing the tilde [~] key. After a game is started, a player can issue a “record [demoname]” command to the Team Fortress 2 console that will start recording a demo and save the file with the name specified in between the brackets. The demo records everything that goes on in the game including the game play from the player's point of view and the text.

At any point after the recording has started, the player can stop the recording by issuing a “stop” command in the console. There are two common ways of playing back the demo. More commonly users use the “timedemo [demoname]” command which runs the demo as fast as the player's computer can handle it. The timedemo will then output the average frames per second which their computer ran the timedemo. This has been a useful 3D gaming video graphics benchmark for over ten years as a way of seeing how fast a video card can render graphics. The other command is the “playdemo [demoname]” command. This is the command that is used for our 3D gaming power consumption benchmark. This command will run the demo at normal speed. Normal playback is what we require for our benchmark because we need the benchmark to run as closely to a real game as possible.

A demo is useful because it produces repeatable results. The demo will playback the same way every time. Furthermore anyone who has a copy of Team Fortress 2 can

copy the same demo we use and run the demo on their own machine. The demo is a good approximation of actual game play but there are important drawbacks. Team Fortress 2 is only a multi player game. While a demo of actual game play may look real, there are no networking streams going back and forth between the local machine and a server. There is also no keyboard and mouse usage while a demo is being played so less power is required to process input. Despite these drawbacks, we will see in the results section that running a demo is a very good approximation of actual game play.

In order to record power consumption from a Team Fortress 2 time demo, follow these steps:

1. Connect the computer and monitor, or laptop to a power measuring device which measures in watt hours or a similar power measurement over time.
2. Start the computer and copy the timedemo file to:
C:\Program Files\Steam\SteamApps\[USER PROFILE]\team fortress 2\tf\
Replace [USER PROFILE] with your user profile name. Modify this path if you have chosen a nonstandard installation location.
3. Open the console by pressing the tilde key [~] key on the keyboard.
4. Type playdemo [demoname]. Replace [demoname] with the name of the demo.
5. Press enter to start the time demo, record the start watt hours. Use a stopwatch or other timing device to measure how much time has passed.
6. When the amount of time required for the benchmark is reached, record the end watt hours.
7. Subtract the end watt hours from the start watt hours to get the watt hours for the time interval that was spent benchmarking.

Once the watt hours are recorded, you can convert to average watts by multiplying the watt hours value by $[60 \text{ minutes (minutes in an hour)} / n \text{ minutes (amount of minutes spent benchmarking)}]$.

6.1.3 SYSTEM POWER MODE BENCHMARKS

The System Power Mode Benchmarks are the easiest tests of our benchmark toolkit to run. The tester simply needs to turn off the computer, set it to Stand by, or Hibernate and then measure the wattage. In all of the computers that have been tested for

this research, the amount of power used by the system power saving modes is constant. This means that we can use the simpler KILL A WATT power consumption meter to measure wattage used at any moment as the system is in the power saving mode, and that value will represent how many watts will be consumed in one hour of operation in the respective power saving state.

In order to record power consumption for a system power saving mode benchmark, follow these steps:

1. Connect the computer and monitor, or laptop to a power consumption meter that displays wattage.
2. Turn on the computer and place the computer in the off, Stand by, or Hibernate power saving mode. For Windows machines this can usually be done by clicking on the start menu and clicking on “Turn Off Computer”.
3. The wattage consumed should not fluctuate. Record the value. This value represents how much wattage is consumed in one hour.

6.1.4 DISTRIBUTED APPLICATION BENCHMARK

Our distributed application benchmark makes use of a program called [Folding@home](#). [Folding@home](#) is a Stanford University distributed networking project which uses massive amounts of computing power to solve protein folding problems. The [Folding@home](#) client can be downloaded for free from the [Folding@home](#) website [34]. [Folding@home](#) has clients available for many major operating systems including Windows, Linux, Mac OS X, and more. This makes [Folding@home](#) a nice platform independent benchmark to use for our power consumption tests. [Folding@home](#) can be tested using the KILL A WATT power meter because when [Folding@home](#) is running, it tends to max out the computers processor and return a consistent wattage consumption value. In a situation where [Folding@home](#) is configured to run at less than 100% CPU

power or if [Folding@home](#) is not the only thing running in the background, it is best to use a watts up? meter or similar meter that can measure watts over time.

In order to record power consumption for a system power saving mode benchmark, follow these steps:

1. Connect the computer and monitor, or laptop to a power consumption meter that displays wattage.
2. Turn on the computer, and start the [Folding@home](#) application. A straightforward guide on configuring and using [Folding@home](#) can be found in Appendix C for Windows and Appendix D for Linux.
3. Record the wattage displayed by the meter. The wattage value represents the watts consumed in an hour. If the wattage displayed is not constant, you will need to repeat these steps and use a device that can measure watts over time.

Follow these steps if you measured using a watts over time consumption meter. Once the watt hours are recorded, you can convert to average watts by multiplying the watt hours value by $[60 \text{ minutes (minutes in an hour)} / n \text{ minutes (amount of minutes spent benchmarking)}]$.

6.2 EXPERIMENTAL METHODOLOGY

Our testing strategy made full use of the benchmarking suite that we have compiled and discussed in the previous sections. We ran each benchmark on two different test systems, a desktop and a laptop. We also tested Ubuntu Linux on the desktop test system. Each benchmark was run with either the KILL A WATT tool or the watts up? tool. Each tool was used for situations which took advantage of their design. The watts up? tool was used to measure benchmarks that caused fluctuating electricity consumption such as the Villanova AutoSurf benchmark and the Team Fortress 2 benchmark. When power consumption changes from one moment to the next, we need to use a power consumption meter that can record consumption over time. The watts up?

tool has the ability to log watt hours. By measuring the watt hours used over some time N , we can then determine the average wattage consumed by a device. The following example (Figure 5) illustrates this technique:

Configured with a 20 second load interval using Mozilla Firefox 2.0.0.12.
We tested for a duration of 5 minutes.
Watt hours consumed in 5 minute interval = 2 Watt hours
Once the watt hours are recorded, we can convert to average watts by multiplying the watt hours value by $[60 \text{ minutes (minutes in an hour)} / n \text{ minutes (amount of minutes spent benchmarking)}]$. After substituting n with 5, we get an average of 24 watts consumed or more accurately the laptop would use about 24 watts in an hour.

Figure 5. Example of a benchmarking test.

The KILL A WATT tool was used in situations where the wattage drawn by a test was constant such as when the computer was turned off or in Stand by mode. The KILL A WATT's simplicity was useful when the wattage was constant because the KILL A WATT does not seem to employ any averaging algorithms so the exact amount of watts currently being consumed was displayed. The watts up? tool on the other hand seemed to take a long time going from one wattage value to another such as in a situation where the computer was turned off and then turned on, leading us to believe it used an averaging algorithm.

We used the following test configurations for each of our benchmarks:

Test System #1 - Desktop

Brand – Custom Built with Cooler Master Case

Operating System – Windows XP Professional with Service Pack 2

Ubuntu 7.10 Desktop Edition 32Bit Gutsy Gibbon

Processor – AMD Athlon 5000+ Black Edition 2.6GHz operating overclocked to 3.1GHz

Memory – 2x 2GB Corsair XMS2 DDR2 800 Memory Sticks operating at 1.9V

Video – BFG 7900GS with 256MB of Memory

Sound – Integrated Realtek HDAudio

Network Device Used – RealTek RTL8168/8111 PCI-E Gigabit Ethernet NIC

Hard disk – Maxtor 100GB 7200RPM SATA (OS Drive) / Seagate 400GB 7200 IDE (Storage)

Optical drive – Samsung 20X DVDRW SATA with LightScribe

Power Supply – Antec Earthwatts 380W 80+ Certified Power Supply Unit

Floppy/Memory Card Reader – Internal Floppy Disk with multiple card reader

Fans – 2x 80MM Case Fans, 1x CPU Fan, 1x Video Fan, 1x Power Supply Fan

USB Devices – Keyboard, mouse

Speakers – 2 Antec Lansing externally powered speakers

Display – 20” Dell 2005FPW Widescreen LCD connected via DVI

Test System #2 - Notebook

Brand – Acer Travelmate 3200

Operating System – Windows XP Professional with Service Pack 2

Processor – Intel Pentium-M 1.5GHz

Memory – 2x 1GB Kingston DDR 400 Memory Sticks

Video – ATI Radeon 9700 with 64MB of Memory

Sound – Realtek AC'97 Audio

Network Device Used – Intel 2200BG 54Mbps Wireless Mini-PCI Card

Hard disk – Hitachi 80GB 7200 RPM 2.5” IDE Drive

Optical drive – MATSHITA UJDA755 DVD/CDRW

Power Supply – DELTA Electronics 19V 3.42 Amps

Floppy/Memory Card Reader – SD card reader

Fans – 1x Internal Fan

USB Devices – None used during testing

Speakers – 2 internally powered speakers

Display – 14” LCD Built-in

6.3 RESULTS

All of the benchmarks were run against both test systems. Test system #1 included benchmarks for Ubuntu Linux version 7.10. The benchmarks were run using the instructions that were laid out in the previous sections. Each benchmark is discussed and results and discussions are included for each test system in the upcoming sections.

6.3.1 INTERNET WEB SURFING BENCHMARK – VILLANOVA AUTOSURF BENCHMARK

We ran the Villanova AutoSurf benchmark with the computer monitor turned on. Most people tend to browse the Internet with their monitor turned on so this is an appropriate benchmarking decision. If benchmarking needs to be done for a system that does not use a monitor, perhaps for a person with a visual disability, then the monitor can be turned off during benchmarking. We set the page load interval to 20 seconds which we arbitrarily chose. An idea for a future work would be to find the average amount of time a user spends on one page. We think 20 seconds is a good amount of time for a user to visit a page, read and look for something and then move on to another page. Each benchmark was run for 5 minutes. We used the Mozilla Firefox web browser version 2.0.0.12. Firefox is an ideal browser to use for power consumption benchmarking when comparing different operating systems. Firefox is available for many popular web browsers including but not limited to Windows, Linux, and Mac OS X. The watts up? meter was used for measuring power consumption. The results of these tests are shown in Table 3.

Table 3. Villanova AutoSurf Benchmark Test Results (Watts used in 5 minutes).

	Desktop (Windows XP)	Desktop (Ubuntu 7.10)	Laptop (Windows XP)
Run 1 Watts / 5 Minutes	13.8	12.5	2
Run 2 Watts / 5 Minutes	13.8	12.4	2
Run 3 Watts / 5 Minutes	13.9	12.4	2.1
Average Watts / 5 Minutes	13.8	12.4	2

The desktop system running Windows XP used 13.8 watts of power on average after running the Villanova AutoSurf benchmark for 5 minutes. Surprisingly the desktop system running Ubuntu Linux 7.10 used less power, 12.4 watts for 5 minutes. The laptop was the undisputed winner using an average of only 2 watts in 5 minutes. To put things in perspective, we have approximated the amount of power that would be used by each configuration if the Villanova AutoSurf benchmark were to be run for one hour (Table 4). Whenever you read a label on an device that says the device uses X number of watts, this value means the device uses X number of watts in one hour. This is why it is important to show the approximate amount of power that a device would use in one hour when showing power consumption results, so that the results can be compared to manufacturer labels and other data that display wattage used per hour.

Table 4. Villanova AutoSurf Benchmark Test Results (Appr. Watts used in 1 Hour).

watts up?	Desktop (Windows XP)	Desktop (Ubuntu 7.10)	Laptop (Windows XP)
Run 1 Appr. Watts / Hr	165.6	150.0	24
Run 2 Appr. Watts / Hr	165.6	148.8	24
Run 3 Appr. Watts / Hr	166.8	148.8	25.2
Average Appr. Watts / Hr	166	149.2	24.4

The desktop system with a 20" LCD monitor used approximately 166 watts of power, 15 watts more than the same system used running Ubuntu Linux 7.10. The laptop used approximately 24.4 watts on average. An energy savings of almost 7 times will be observed by using our laptop system for surfing the Internet vs our desktop system.

6.3.2 3D GAMING BENCHMARK – TEAM FORTRESS 2

We ran the Team Fortress 2 benchmark with the computer monitor and speakers turned on. Games are played with the monitor turned on and most of the time with a set of speakers or headphones. If benchmarking needs to be done for a system that does not use a monitor or speakers then the monitor and/or speakers can be turned off during benchmarking. Each benchmark was run for 5 minutes. We ran the Team Fortress 2 benchmark natively in Windows XP and through WINE [32]. The watts up? meter was used for measuring power consumption. Table 5 contains the video settings which were used for each configuration.

Table 5. Team Fortress 2 Benchmark Test Configurations.

	Desktop (Windows XP 1680x1050)	Desktop (Windows XP 1024x768)	Desktop (Ubuntu 7.10 1024x768)	Laptop (Windows XP 800x600)
Video Settings				
Resolution	1680x1050	1024x768	1024x768	800x600
Aspect Ratio	Widescreen 16:10	Normal (4:3)	Normal (4:3)	Normal (4:3)
Display Mode	Full screen	Full screen	Full screen	Full screen
Video Advanced				
Model detail	High	Medium	Medium	Medium
Texture detail	High	Medium	Medium	Medium
Shader detail	High	High	High	High
Water detail	Reflect all	Reflect all	Reflect all	Reflect all
Shadow detail	High	Medium	Medium	Medium
Color correction	Enabled	Enabled	Enabled	Enabled
Anti-aliasing mode	None	None	None	None
Filtering mode	Anisotropic 8x	Trilinear	Trilinear	Trilinear
Wait for vertical sync	Disabled	Disabled	Disabled	Disabled
High Dynamic Range	Full (if available)*	Full (if available)*	Full (if available)*	Full (if available)*
Motion Blur	Enabled	Disabled	Disabled	Disabled
Use 'bloom' effect when available	Unchecked (disabled)	Unchecked (disabled)	Unchecked (disabled)	Unchecked (disabled)
Hardware Direct X level:	DirectX v9.0+	DirectX v9.0+	DirectX v9.0	DirectX v9.0
Software Direct X level:	DirectX v9.0	DirectX v9.0	DirectX v9.0	DirectX v9.0

The video settings play an important part in power consumption for a 3D game because graphics chips can use so much power. We included results for the desktop power consumption as it was recording the initial demo so that a comparison could be made between recording and playing a demo. We also included a desktop resolution of 1024x768 which followed the laptop's lower video detail settings so we could compare the lower resolution 1024x768 settings to the desktop 1680x1050 resolution with high quality settings. This would give us an idea of how much more power a system uses for higher quality graphics.

Our Ubuntu configuration also followed the 1024x768 resolution with lower video quality settings because Team Fortress 2 performed slower under Ubuntu and WINE than under Windows. Team Fortress 2 was more playable at lower settings in Ubuntu and WINE. Finally the laptop ran at a resolution of 800x600 with lower video quality settings which made the game playable with its limited graphics capabilities. The results are shown in Table 6.

Table 6. Team Fortress 2 Benchmark Test Results.

watts up?	Desktop Recording Demo (Windows XP 1680x1050)	Desktop Playing Demo (Windows XP 1680x1050)	Desktop Playing Demo (Windows XP 1024x768)	Desktop Playing Demo (Ubuntu 7.10 1024x768)	Laptop Playing Demo (Windows XP 800x600)
Watts / 5 Minutes	18	18	16.6	15.7	4
Appr. Watts / Hr	216	216	199.2	188.4	48

The desktop system recording the demo and playing the demo had identical power consumption values. This is fortunate because it means playing a demo is an effective approximation of actual game play. The desktop used around 16 watts less power at

lower quality settings than it did at higher quality video settings. This shows that video cards have a significant role to play in power consumption when they have a load placed upon them. Comparing the Windows XP and Ubuntu power consumption again shows a trend of lower power consumption in Ubuntu. It is hard to pinpoint the reason why this is the case as both systems were running the same exact program at the same exact settings. Perhaps Ubuntu uses lighter weight processes, less background processes, and/or more efficient drivers or perhaps Ubuntu does not fully take advantage of some of the available hardware.

Fortunately there is no noticeable loss in functionality between Windows XP and Ubuntu during our tests except in Team Fortress 2 where Ubuntu performed slower than Windows. In the other tests there was no perceivable difference between the two operating systems. The laptop performs the best using approximately 48 watts of power. While the laptop uses more than 150 watts less power than the desktop, there are important trade offs to consider. It must be pointed out that playing a game on the desktop offers higher resolutions, better performance, and a larger screen, so a laptop may use less power but does not offer all the features a desktop has to offer.

6.3.3 SYSTEM POWER MODE BENCHMARKS – OFF, STAND BY, HIBERNATE AND IDLE

We ran each system power mode benchmark with the monitor left off except for the Idle benchmark. Certain modes such as the Off state would force the monitor into its own sleep state. The Dell 20" 2005FPW used 2 watts in its sleep state. Some people do not turn off their monitors while some people do, so both sides can be taken into account

during these benchmarks by simply turning the monitor off or on. The laptop screen automatically turns off when the laptop turns off so if the user closes the lid, we run our benchmark with the lid closed. Conversely if a user leaves the lid open we can benchmark with the lid open. The Idle benchmark test runs lasted 3 minutes each. We used the watts up? [29] meter to keep track of power consumption over the three minutes. The Off, Stand by, and Hibernate states have constant levels of power consumption so we used the simpler KILL A WATT [25] tool for that purpose. The results of these tests are shown in Table 7.

Table 7. System Idle Benchmark Test Results (Watts Used in 3 Minutes).

watts up?	Desktop (Windows XP)	Desktop (Ubuntu 7.10)	Laptop (Windows XP)
Run 1 Watts / 3 Minutes	8.1	7.2	.9
Run 2 Watts / 3 Minutes	8.1	7.2	.9
Run 3 Watts / 3 Minutes	8.1	7.2	1.0
Average Watts / 3 Minutes	8.1	7.2	.9

The desktop using Windows XP is the worst performer in this test. Ubuntu performs slightly better on the same hardware. The laptop is the best performer once again.

The gaps widen significantly when the hour long approximation is calculated (Table 8). The desktop using Windows XP uses around 18 more watts than the desktop using Ubuntu Linux 7.10 at idle. The laptop uses 19 watts which is less than the power consumed by a typical 60W incandescent light bulb. This Idle benchmark shows us that a lot of power can be wasted when an unused computer is left on for extended amounts of time.

Table 8. System Idle Benchmark Test Results (Appr. Watts Used in 1 Hour).

watts up?	Desktop (Windows XP)	Desktop (Ubuntu 7.10)	Laptop (Windows XP)
Run 1 Appr. Watts / Hr	162	144	18
Run 2 Appr. Watts / Hr	162	144	18
Run 3 Appr. Watts / Hr	162	144	20
Average Appr. Watts / Hr	162	144	19

The Off state and the Hibernate state used the exact same amount of power (Table 9). This is expected because Hibernate saves the computers memory state to the hard disk drive and then shuts off the computer. Stand by in all cases uses 1 more watt than the Off and Hibernate states. Windows XP and Ubuntu did equally well in these power savings modes while the laptop was outstanding, using around 0 watts when it was Off or in Hibernate and 1 watt during Stand by. Savings of 158 watts for the Windows desktop, 140 watts for the Ubuntu desktop, and 19 watts for the laptop can be observed simply by using the Hibernate feature instead of leaving the machine idling.

Table 9. System Off, Stand by, and Hibernate Benchmark Test Results (Appr. Watts Used in 1 Hour).

KILL A WATT	Desktop (Windows XP)	Desktop (Ubuntu 7.10)	Laptop (Windows XP)
Off	4	4	0
Stand by	5	5	1
Hibernate	4	4	0

6.3.4 DISTRIBUTED APPLICATION BENCHMARK – FOLDING@HOME

We ran each test configuration with the monitor turned on and off (Table 10). The laptop monitor was turned off by closing the lid which automatically powers down the

screen. The KILL A WATT meter was used to get the results because when [Folding@home](#) is run by itself, the power usage is constant. If a benchmarker finds a case where the power usage is not constant, a watts up? [29] meter may be used to record watts consumed over time. Here are the results of our tests:

Table 10. [Folding@home](#) Benchmark Results (Appr. Watts Used in 1 Hour).

KILL A WATT	Desktop (Windows XP)	Desktop (Ubuntu 7.10)	Laptop (Windows XP)
F@H With Monitor	186	187	38
F@H Without Monitor	133	134	34

[Folding@home](#) did not make use of both processor cores on the desktop system, so the CPU usage was pegged at 50%. This issue did not affect the laptop because the laptop only had a single processor core. If the [Folding@home](#) client supported symmetric multi processing [35] (using more than one CPU), the power consumption would have been higher.

6.4 ANALYSIS

Some of our results were expected and some surprised us. In every single test the laptop used several times less power than the desktop computer. Even if you take away the power consumption of a desktop's monitor, the laptop still used significantly less power. Our comparisons of the Off, Stand by, Hibernate, and Idle power states showed us that significant savings can be had if computer users put forth a little bit of effort to avoid leaving a computer idling. Like a light bulb left on in an unoccupied room, an unattended computer left on wastes a lot of power. Turning a computer Off and using the Stand by or Hibernate features are great alternatives to leaving computers turned on.

One of the surprising results was that Ubuntu Linux 7.10 used significantly less power than the Windows XP operating system on the same machine. Sometimes up to as much as 18 less watts were expended while completing an identical benchmark on the Ubuntu Linux 7.10 operating system. Unfortunately the results only show us that Ubuntu uses less power than Windows XP and not why Ubuntu uses less power than Windows XP. As mentioned earlier there could be a plethora of reasons why Ubuntu uses less power than Windows XP in many of our tests. Reasons may include but not limited to lighter weight processes, less background processes, and/or drivers that may not take advantage of all of the system hardware. Finding out the cause of this difference is left to future work.

In our 3D gaming benchmark with Team Fortress 2 we learned that using lower quality video settings consumes less power than using higher quality settings on our desktop machine which uses a discrete graphics card (a graphics chipset that is on an add on board) [36]. This indicates that video processing can consume large amounts of power, especially when dedicated graphics cards with high clock speeds, large transistor counts, large amounts of memory, and large fast fans are used. The power efficiency of graphics cards must be researched so that ways of decreasing their power consumption can be found.

The results collectively indicate that the benchmarks we have developed are both repeatable as well as practical. They closely approximate how much power would be used during real world tasks. This is an important feat because we have a free toolkit to offer anyone which can help show a computers power consumption. This can lead to more efficient computing design by having regular people be able to see results of these

benchmarks and understand in simple terms how much power their devices use. In our capitalistic society consumer purchases have an impact on the designs computer manufacturers employ. If consumers demand energy efficient products, computer companies will have to employ energy efficient designs so that they do not lose business.

7 CONCLUSION

The results of our work answered some questions while raising some new ones. Our primary question was, “Can we create a set of benchmarks that are repeatable, practical, and free?”. After reviewing the results from the benchmark tests we ran using our green computing benchmarking toolkit, we think the answer is a resounding yes! Our green computing benchmarking toolkit includes benchmarks for many different computing use patterns such as web browsing, 3D gaming, distributed computing, and provides guidelines on how to benchmark Off, Stand by, Hibernate, and Idle power states.

While these benchmarks provide a variety of use patterns, there is still more work to be done. Benchmarks for programs such as word processing, spreadsheets, presentation software, email clients, calendar clients, photo editing software, multimedia software, rendering software, and more need to be completed. There are a plethora of areas which benchmarking tools would be beneficial for furthering the goals of green computing, improving energy efficiency, reducing waste, and encouraging reuse.

We have learned that laptops generally use less power than desktops in every benchmark we tried. There are definitely exceptionally energy efficient desktops that come close to laptops such as Everex's gPC [37], but even these desktops use more power than the average laptop especially when an external monitor's power consumption is taken into account. While laptops use a lot less power, their full life cycle energy cost should be taken into account. Laptops tend to be hard to upgrade [38], especially when the processors and video chipsets are soldered onto the motherboards. This reduces their

longterm expandability and may result in a shorter life cycle than an upgradeable desktop[38].

Desktop computers require external monitors which consume more raw materials due to their size. All of these matters and more must be taken into account when looking at the impact of purchasing a desktop or a laptop computer. Even if a person determines that a laptop is better for the environment, another important consideration is that laptops sometimes do not perform as well as desktops in certain tasks [38]. For mundane tasks such as web browsing and office work, laptops are usually more than adequate. However for 3D gaming, animation rendering, and other resource intensive programs a desktop may perform much better. Laptops have come a long way in the past decade with the advent of gaming laptops and solid state drives, but despite their advances sometimes laptops simply cannot compete with a performance desktop.

We have seen that turning off a machine or placing it into the Stand by or Hibernate mode uses several times less power than leaving a computer idling. This is one area which we are glad to report that has little trade offs for decreasing your computer's power consumption. A computer may take 5 seconds to wake from Stand by or 30 seconds to wake from Hibernate. Using these features takes a little longer to start a machine than coming back to an idling computer, but the amount of money and electricity not wasted by using these power saving features more than makes up for the lost seconds. A computer left idling 24 hours a day, 365 days a year can waste more than \$100 worth of electricity.

One of the more pleasant surprises of our testing was that Ubuntu Linux uses less power than Windows in most of the test cases. Not only is Ubuntu Linux free to use for anyone, it will also save you some money from electricity costs. We hypothesize that Ubuntu is a lighter weight operating system than Windows and uses less processes which in turn requires less resources. We also think that Ubuntu may do a better job of throttling processor speed than Windows does. Modern processors from Intel and AMD have the ability to run at lower clock speeds to conserve power when they are not doing intensive work. This may have played a role in Ubuntu's efficiency. Although Ubuntu excelled in power efficiency, it should be noted that there are trade offs to be made for the power efficiency gains. Ubuntu may not satisfy the requirements of all users. Ubuntu provides a wide variety of free software to accomplish many tasks but may not offer all of the features needed for jobs that can be done in Windows. For example, Ubuntu is not able to run games written for Windows without the use of separate programs like WINE [32] or Cedega [39]. WINE and Cedega can run many Windows programs under UNIX based operating systems but they tend to perform slower and may have glitches. More work needs to be done in the future to determine the exact reasons for Ubuntu's lower power consumption.

The overriding goal for this paper is to help people reduce electricity consumption which in turn reduces pollution. Saving money is a secondary benefit which is also important. Many people may not be willing to reduce consumption based on purely green reasons but maybe interested in learning how to save money. By reducing the amount of power computers consume there is a direct benefit in the form of lower electricity

consumption. A side benefit occurs in warmer climates where air conditioning is used. If computers generate less waste heat, the air conditioners used to cool rooms that contain computers will work less hard. This reduces electricity consumption further. By using our green computing benchmarking toolkit folks can have a better idea of how much power their computers use in different scenarios. In order to deal with the electrical inefficiencies of computers we must first learn where the inefficiencies lie. Only then can we understand that there is a problem and work to solve the problem. We believe that our green computing benchmarking toolkit can be used to show the world computer power consumption in simple terms most people should be able to understand. By getting knowledge about computer electrical consumption out in the open, we will have completed the first steps to reduce pollution and greenhouse gas emissions due to computing.

BIBLIOGRAPHY

- [1] Horvath, A, and Masanet, Eric, An Analysis of Measures to Reduce the Life-Cycle Energy Consumption and Greenhouse Gas Emissions of California's Personal Computers, University of California, 2007,
<http://repositories.cdlib.org/cgi/viewcontent.cgi?article=1036&context=upei>.
- [2] University of California, The Rise of CO₂ & Warming, University of California, 2002, http://earthguide.ucsd.edu/globalchange/global_warming/03.html.
- [3] Amar, Praveen, MERCURY EMISSIONS FROM COAL-FIRED POWER PLANTS, Northeast States for Coordinated Air Use Management, 2003,
<http://www.nescaum.org/documents/rpt031104mercury.pdf/>.
- [4] Markoff, John, Technology's Toxic Trash Is Sent to Poor Nations, NY Times, 2002,
<http://query.nytimes.com/gst/fullpage.html?res=9E00E5D71E3EF936A15751C0A9649C8B63>.
- [5] Sierra Club, Clean Air Dirty Coal Power, Sierra Club, 2006,
<http://www.sierraclub.org/cleanair/factsheets/power.asp>.
- [6] Hvistendahl, Mara, Coal Ash Is More Radioactive than Nuclear Waste, Scientific American, 2007, <http://www.sciam.com/article.cfm?id=coal-ash-is-more-radioactive-than-nuclear-waste>.
- [7] Chapman, Siobhan, PC numbers set to hit 1 billion, TechWorld, 2007,
<http://www.techworld.com/news/index.cfm?NewsID=9119>.
- [8] Beal, Vangie, All About Monitors CRT vs. LCD, Jupitermedia Corporation, , http://www.webopedia.com/DidYouKnow/Hardware_Software/2005/all_about_monitors.asp.
- [9] Planar Systems, Inc., Benefits of AMLCDs over CRTs as Related to the StereoMirror, Planar Systems, Inc., , <http://www.planar3d.com/3d-technology/lcd-vs-crt>.
- [10] Natural Resources Council of Maine, Electronic Waste and Other Solid Waste Issues, Natural Resources Council of Maine, ,
http://www.nrcm.org/issue_electronicwaste.asp.
- [11] UCSU Environmental Center, GREEN COMPUTING GUIDE, The University of Colorado - Boulder, 2004,
http://ecenter.colorado.edu/energy/projects/green_computing_guide.pdf.

- [12] MSNBC staff and news service reports, Deadly power plants? Study fuels debate, msnbc.com, 2004, <http://www.msnbc.msn.com/id/5174391>.
- [13] Gerdes, Justin, Screensavers: They Aren't Saving Your Screen, But They Are Sapping Your Savings, Efficiency Partnership, 2008, <http://fypower.org/news/?p=601>.
- [14] Microsoft Corporation, Help save energy and the environment by putting your monitor to sleep, Microsoft Corporation, 2006, <http://www.microsoft.com/windowsxp/using/setup/tips/sleep.mspx>.
- [15] Maintain, Sleep (Hard disks), Maintain, , <http://www.maintain.se/cocktail/help/tiger/disks/sleep.html>.
- [16] Digit-Life.com, HDD Diet: Power Consumption and Heat Dissipation, Digit-Life.com, 2005, <http://www.digit-life.com/articles2/storage/hddpower.html>.
- [17] Terra Novum, LLC, System Standby (S3) v. Hibernation In New Windows PCs, Terra Novum, LLC, , http://www.terranovum.com/projects/energystar/standby_v_hiber.html.
- [18] Kemp, William H., The Renewable Energy Handbook, Aztext Press, 2005, .
- [19] Paul Graham, Programming Bottom-Up, paulgraham.com, 1993, <http://www.paulgraham.com/progbot.html>.
- [20] Töpelt, Bert and Schuhmann, Töpelt, Energy Efficiency: AMD vs. Intel, Tom's Hardware, Bestofmedia Network, 2007, <http://www.tomshardware.com/2007/07/11/energy-efficiency-intel-left-out-in-the-cold/>.
- [21] Atwood, Jeff, When Hardware is Free, Power is Expensive, Jeff Atwood, 2007, <http://www.codinghorror.com/blog/archives/000868.html>.
- [22] Mammano, Bob, Improving Power Supply Efficiency –The Global Perspective, Texas Instruments, 2006, <http://focus.ti.com/download/trng/docs/seminar/Topic1BM.pdf>.
- [23] Futuremark, 3DMark06, Futuremark, 2007, <http://www.futuremark.com/products/3dmark06>.
- [24] Schmid, Patrick and Roos, Achim, The Truth About PC Power Consumption, Tom's Hardware, Bestofmedia Network, 2007, http://www.tomshardware.com/2007/10/19/the_truth_about_pc_power_consumption/index.html.

- [25] P3 INTERNATIONAL, P3 - Kill A Watt, P3 INTERNATIONAL, 2008, <http://www.p3international.com/products/special/P4400/P4400-CE.html>.
- [26] Wikipedia Contributors, Ampere, Wikipedia.org, 2008, <http://en.wikipedia.org/wiki/Ampere>.
- [27] Wikipedia Contributors, Volt-ampere, Wikipedia.org, 2008, <http://en.wikipedia.org/wiki/Volt-ampere>.
- [28] Vijay Pande and Stanford University, Folding@home, Stanford University, 2008, <http://folding.stanford.edu>.
- [29] Electronic Educational Devices, Watts Up? Products: Meters, Electronic Educational Devices, 2008, <https://www.wattsupmeters.com/secure/products.php>.
- [30] Valve Corporation, The Orange Box - Team Fortress 2, Valve Corporation, 2008, <http://www.whatistheorangebox.com/tf2.html>.
- [31] Thomas Way, Villanova AutoSurf Benchmark, Villanova University, 2008, <http://actlab.csc.villanova.edu/green/autosurf.html>.
- [32] WINE, WINE HQ, WINE, 2008, <http://www.winehq.org/>.
- [33] WINE, WINE APP Database for Team Fortress 2, WINE, 2008, <http://appdb.winehq.org/objectManager.php?sClass=version&iId=9207>.
- [34] Vijay Pande and Stanford University, Folding@home - Download the Folding@home software application, Stanford University, 2008, <http://folding.stanford.edu/English/Download>.
- [35] Salzman, Peter Jay and Pomerantz, Ori, Chapter 13. Symmetric Multi Processing, tldp.org, 2001, <http://tldp.org/LDP/lkmpg/2.4/html/c1294.htm>.
- [36] Wikipedia Contributor, Graphics Processing Unit, Wikipedia.org, 2008, http://en.wikipedia.org/wiki/Graphics_processing_unit#Dedicated_graphics_cards.
- [37] Everex, Everex - The Alternative PC, Everex, 2007, <http://www.everex.com/products/gpc/gpc.htm>.
- [38] OfZenAndComputing.com, Laptops versus Desktops: which is better for you?, OfZenAndComputing.com, 2006, <http://www.ofzenandcomputing.com/zanswers/208>.
- [39] TransGaming Inc., Cedega, TransGaming Inc., 2008, <http://www.transgaming.com>.

[40] Vijay Pande and Stanford University, Folding@home - FAQ, Stanford University, 2008, <http://folding.stanford.edu/English/FAQ>.

APPENDIX A. VILLANOVA AUTOSURF BENCHMARK

You can run the Villanova AutoSurf Benchmark by visiting the Villanova Applied Computing Technology Laboratory website at the following link:

<http://actlab.csc.villanova.edu/green/autosurf.html> [31].

Alternatively you can copy and paste the following source code into a blank HTML file.

Then open the HTML file with a web browser such as Mozilla Firefox.

```
<html>
<head>
<title>Villanova AutoSurf Benchmark</title>

<script language="javascript" src="sites.js"></script>

<script language="javascript">
<!-- Begin

var winName = "surfwin";
var surfTimerId = 0;
var stopwatchID = 0;
var startTime = 0;
var counter = 0;
var newWin = 0;
var readyToLoad = false;
var timeout = 60;
var waitedFor = 0;
var loadtime = new Date();

function surf()
{
  if (document.getElementById("surfButton").value == "Start Surfing")
  {
    document.getElementById("surfButton").value = "Stop Surfing";
    document.getElementById("duration").innerHTML = "0:00:00";
    document.getElementById("statusMessage").innerHTML =
"Starting...";
    startTime = new Date();
    setSurfTimerRegular();
    stopwatchID = setInterval("tick()", 1000);
    timeout = document.getElementById("delay").value;
  }
  else
  {
    document.getElementById("surfButton").value = "Start Surfing";
    clearInterval(surfTimerId);
    clearInterval(stopwatchID);
    stopwatchID = 0;
  }
}
```

```

    surfTimerId = 0;
    document.getElementById("statusMessage").innerHTML = "Ready";
  }
}

function setSurfTimerSecs(secs)
{
    clearInterval(surfTimerId);
    surfTimerId = setInterval ( "visitNextSite()", secs * 1000 );
}

function setSurfTimerRegular()
{
    setSurfTimerSecs(document.getElementById("delay").value);
}

function visitNextSite()
{
    // Would like to check status bar here to see if it is "Done"
    // and only load the next site when it says that.
    if (!readyToLoad)
    {
        setSurfTimerSecs(1);
        waitedFor++;
        if (waitedFor >= timeout)
        {
            readyToLoad = true;
        }
        return;
    }
    else
    {
        setSurfTimerRegular();
        waitedFor = 0;
    }
    //var url = randomsite;
    //var url = getNextURL();
    //var url = googleURL();
    var url = getNextListSite();
    counter++;
    document.getElementById("counter").innerHTML = counter;
    document.getElementById("index").innerHTML = currentSite + " of
" + siteCount; // from sites.js
    ms = (new Date()).getTime() - startTime.getTime();
    if (counter > 0)
    {
        secsPerPage = ms / counter / 1000;
        secsPerPage = secsPerPage.toFixed(1);
        document.getElementById("avgPerPage").innerHTML =
secsPerPage;
    }
    openWindow(url);
}

```

```

function tick()
{
    nowTime = new Date();
    ms = nowTime.getTime() - startTime.getTime();
    hours = Math.floor(ms / 3600000);
    ms = ms - (hours * 3600000);
    minutes = Math.floor(ms / 60000);
    ms = ms - (minutes * 60000);
    seconds = Math.floor(ms / 1000);
    ms = ms - (seconds * 1000);
    minutes = (minutes < 10?"0":"") + minutes;
    seconds = (seconds < 10?"0":"") + seconds ;
    document.getElementById("duration").innerHTML =
hours+":"+minutes+":"+seconds;
}

function openWindow(url)
{
    document.getElementById("statusMessage").innerHTML = "Loading: "+url;
    readyToLoad = false;
    loadtime = (new Date()).getTime();
    document.getElementById("surfview").src = url;
}

function setReady()
{
    readyToLoad = true;
    document.getElementById("statusMessage").innerHTML = "Ready";
    setLoadInfo();
}

function setLoadInfo()
{
    var url = document.getElementById("surfview").src;
    var time = ((new Date()).getTime() - loadtime) / 100;
    if (url == document.location)
    {
        url = "";
        time = "";
    }
    else
    {
        url = "<a href='"+url+"' target='_blank'>"+url+"</a>";
        time = (Math.round(time) / 10) + " secs";
    }
    document.getElementById("location").innerHTML = url;
    document.getElementById("loadtime").innerHTML = time;
}

//-----
// Random URL support
//-----

var alphabet = "abcdefghijklmnopqrstuvwxyz";

```



```

        web browsing activity. To use, enter desired number of seconds
between page
        loads and click the &quot;Start Surfing&quot; button. Adjust
your speaker
        volume as some of the links are very noisy.</font><p />
<b><font size="2">Example tests:</font></b><br />
<ul style="line-height: 100%; margin-top: 0; margin-bottom: 0">
    <li><font size="2">Compare power use of different
browsers.</font></li>
    <li><font size="2">Compare
        effect of power conservation settings on power use
while browsing.</font></li>
</ul>
</td></tr>
<tr><td width="155">Pages visited:</td><td width="295"><span
id="counter">0</span> (#<span id="index">0</span>)</td></tr>
<tr><td width="155">Status:</td><td width="295"><span
id="statusMessage" style="color: green;">Ready</span></td></tr>
<tr><td width="155">Location:</td><td width="295"><span
id="location"></span></td></tr>
<tr><td width="155">Load time:</td><td width="295"><span
id="loadtime"></span></td></tr>
<tr><td width="155">Total runtime:</td><td width="295"><span
id="duration"></span></td></tr>
<tr><td width="155">Avg secs/page:</td><td width="295"><span
id="avgPerPage"></span></td></tr>
</table>
<p align="center">
<iframe id="surfview" width="100%" height="500"
src="http://www.csc.villanova.edu/" onLoad="setReady()"></iframe>
<br clear="all" />

&copy; <font size="2">2008 Villanova Applied Computing Technology
Lab<br>
updated
<!--webbot bot="Timestamp" S-Type="EDITED" S-Format="%m/%d/%Y"
startspan -->02/10/2008<!--webbot bot="Timestamp" endspan i-
checksum="12504" --></font>

</p>

</body>
</html>

```

APPENDIX B. 3D GAMING BENCHMARK

How to Use the Team Fortress 2 Demo Feature

In order to record a demo the console feature must be enabled in the keyboard advanced settings menu. The console can then be opened by pressing the tilde [~] key. After a game is started, a player can issue a “**record [demoname]**” command to the Team Fortress 2 console that will start recording a demo and save the file with the name specified in between the brackets. The demo records everything that goes on in the game including the game play from the player's point of view and the text. At any point after the recording has started, the player can stop the recording by issuing a “**stop**” command in the console. The **stop** command will stop the demo from recording. There are two common ways of playing back the demo. More commonly users use the “**timedemo [demoname]**” command which runs the demo as fast as the player's computer can handle it. The timedemo will then output the average frames per second which their computer ran the timedemo. The other command is the “**playdemo [demoname]**” command. This is the command that is used for our 3D gaming power consumption benchmark. This command will run the demo at normal speed. Normal playback is what we require for our benchmark because we need the benchmark to run as closely to a real game as possible.

APPENDIX C. FOLDING@HOME FOR WINDOWS

INSTALLATION AND CONFIGURATION

Open <http://folding.stanford.edu/English/Download> [34] in a web browser. Find the Windows text-only console version of [Folding@home](#) [28] and download it to your computer (note you can follow most of these steps for other popular operating systems). Once the download completes browse to the download directory and double click on the executable.

The first time you run [Folding@home](#) [28], you are asked a series of configuration questions. We used the following configuration settings for our benchmark, based on information in the online FAQ [40]. Feel free to experiment with the settings for your application. The settings we went with are in bold.

```
User name [Anonymous]?
Team Number [0]?
Launch automatically at machine startup, installing this as a
  service (yes/no) [no]?
Ask before fetching/sending work (no/yes) [no]?
Use Internet Explorer Settings (no/yes) [no]?
Use proxy (yes/no) [no]?
Allow receipt of work assignments and return of work results greater
  than 5MB in size (such work units may have large memory demands)
  (no/yes) [no]? yes
Change advanced options (yes/no) [no]? yes
Core Priority (idle/low) [idle]?
CPU usage requested (5-100) [100]?
Disable highly optimized assembly code (no/yes) [no]?
Pause if battery power is being used (useful for laptops) (no/yes) [no]?
Interval, in minutes, between checkpoints (3-30) [15]?
Memory, in MB, to indicate (2047 available) [2047]?
Request work units without deadlines (no/yes) [no]?
Set -advmethods flag always, requesting new advanced
  scientific cores and/or work units if available (no/yes) [no]? yes
Ignore any deadline information (mainly useful if system clock
  frequently has errors) (no/yes) [no]?
Machine ID (1-8) [1]?
```

APPENDIX D. [FOLDING@HOME](#) FOR LINUX

INSTALLATION AND CONFIGURATION

Open <http://folding.stanford.edu/English/Download> [34] in a web browser. Find the Linux text-only console version of [Folding@home](#) [28] and download it to your computer (note you can follow most of these steps for other popular operating systems). Once the download completes open a terminal and go to the download directory and issue the following command to change the file's permissions to executable.

```
~$ chmod +x FAH504-Linux.exe
```

Next run the executable by issuing the following command:

```
~$ ./FAH504-Linux.exe
```

The first time you run [Folding@home](#) [28], you are asked a series of configuration questions. We used the following configuration settings for our benchmark, based on information in the online FAQ [40]. Feel free to experiment with the settings for your application. The settings we used are in bold.

```
User name [Anonymous]?  
Team Number [0]?  
Ask before fetching/sending work (no/yes) [no]?  
Use proxy (yes/no) [no]?  
Allow receipt of work assignments and return of work results greater  
than  
5MB in size (such workunits may have large memory demands) (no/yes)  
[no]? yes  
Change advanced options (yes/no) [no]? yes  
Core Priority (idle/low) [idle]?  
Disable highly optimized assembly code (no/yes) [no]?  
Interval, in minutes, between checkpoints (3-30) [15]?  
Memory, in MB, to indicate? 2047  
Request work units without deadlines (no/yes) [no]?  
Set -advmethods flag always, requesting new advanced  
scientific cores and/or work units if available (no/yes) [no]? yes  
Ignore any deadline information (mainly useful if  
system clock frequently has errors) (no/yes) [no]?  
Machine ID (1-8) [1]?
```



This work is licensed under a [Creative Commons Attribution 3.0 United States License](https://creativecommons.org/licenses/by/3.0/us/).