

Power Aware Computing Survey

Hesham Hassan
Computer Science Department
Faculty of Computer and Information
Cairo University, Egypt

Ahmed Shawky Moussa
Computer Science Department
Faculty of Computer and Information
Cairo University, Egypt

ABSTRACT

Power-aware computing has caught the interest of researchers and users of all computing systems. In embedded systems and small devices, better management of energy translates into longer lasting and smaller batteries, which in turn implies smaller and lighter devices. In cloud, distributed, and high performance computing systems, better management of power translates into saving a significant amount of money and natural resources. This paper surveys the different power-aware computing approaches and techniques, focusing mostly on software approaches. It also introduces power-aware computing and why it is very important these days. The paper discusses the ways and challenges of measuring the energy consumption of systems and devices.

Keywords

Power-aware computing, power-aware algorithms, energy consumption, power-aware compilers, power-aware cloud computing, power-aware high performance computing, measurement tools.

1. INTRODUCTION

Energy consumption has become the limiting factor in keeping up with developing faster, smaller, and larger capacity computers; thus it is of major concern. With the widespread use of handheld devices, smart phones and notebook computers, and with the increasing size of super computers and data centres, energy consumption has come to the attention of researchers. Users of battery powered devices are concerned with energy consumption because they want a longer device uptime. However, enhancing the battery itself is generally in conflict with the battery size and weight, which more recently have become a problem for small sized computing devices. Users of non-battery powered systems like cloud, distributed, and high performance computing systems are concerned with energy consumption because powering and cooling these systems are very expensive, and consume a lot of resources. Therefore, power and energy have become key considerations across designing devices and software algorithms. Everyone is looking for an answer to the question of how to save energy and keep up with developing more powerful computing devices at the same time. Everyone is searching for efficient usage of resources, ways to reduce power consumption, and energy-efficient hardware and software.

In the next sections studies of different techniques and approaches to power-aware computing will be examined.

2. IMPORTANT DEFINITIONS

Before proceeding, it is important to know the definition of Energy and Power, and the difference between them.

2.1 Energy, and Power Definitions

Energy is the ability to do work, or make a change. Since energy is defined in terms of work, it has the same work measurement unit – joule. The term joule is used in honour of the English scientist James Prescott Joule who generalized the notions of mechanical energy that followed from Newton's Laws, and discovered that heat is a type of energy. A joule is the amount of energy expended as work if a force of one newton over a distance of one meter is exerted [1].

Power is the rate at which energy is expended, used, transferred, or transformed. People are usually more interested in power than energy, as it shows the rate of energy usage, and how much energy their appliances draw. When speaking about electrical power the watt is the unit of measurement, which is joules per second.

$$Power = Energy / time$$

$$Watt = joule / second$$

For example, imagine a 1000 watt electrical oven converting 1000 joule of electrical energy into 1000 joule of heat energy every second; if it is left open for 10 minutes (600 seconds), the total energy consumption would be:

$$Energy = Power \times Time = 1000 \times 600 = 600,000 \text{ joules}$$

In practical situations, it becomes very hard to use joule as a measurement for consumed energy because numbers may become very large. Therefore, another unit, called kilowatt hour, could be used. This represents the energy delivered by 1000 watts of power over one hour. So, using the same 1000 watt oven for one hour (3600 seconds) would consume 1 kilowatt hour, which is equivalent to 3,600,000 joules [1].

2.2 Power vs. Energy

Osman S. Unsal, in his dissertation about system-level power-aware computing in complex real-time and multimedia systems [2], explains that there are big differences between power efficiency and energy efficiency, power-aware and low power, and power-constrained and energy-constrained systems. Below we discuss some of the points mentioned in his paper.

2.2.1 Power efficiency vs. energy efficiency

Power efficiency and energy efficiency are two separate goals. Reducing power does not necessarily imply reduced energy consumption. For example, a power efficiency design that aims to reduce the processor's power by decreasing the clock rate, might oppose energy efficiency as it could lead to a longer execution time which will lower the energy efficiency by increasing the total energy consumption. In other words,

low power devices operating for a long time could consume more energy than higher power devices operating for a shorter time. Now the question is how to meet a performance target despite power and energy constraints, and how to meet power and energy constraints without affecting performance.

2.2.2 Power aware vs. low power systems

A low power system is a system whose main goal is to decrease the power. A power-aware system is a system where power and energy constraints are the main system design considerations, which could change the system's behaviour based on the energy consumption rate, or based on the amount of remaining energy.

2.2.3 Power constrained & energy constrained systems

There are two types of systems, power constrained systems and energy constrained systems. Battery operated devices are energy constrained systems because they have a finite energy resource, so meeting energy constraints is a major design consideration. Power constrained devices are those that work with infinite energy resources, although even in an infinite energy resource system energy constraints could be valid. However, the main goal would be to reduce power as much as possible.

3. ALGORITHM ENHANCEMENT APPROACH

One of the main approaches to saving energy is to enhance the algorithms. It has been discovered that different algorithms for the same problem could consume different amounts of energy, regardless of their processing time.

A research paper by Hannah Bayer and Markus E. Nebel about evaluating algorithms according to their energy consumption [3] stated that although it was a common belief that faster algorithms consume less energy than slower ones, this is not universally valid. Their study showed that slower algorithms could consume less energy than faster algorithms at some input sizes. Bayer and Nebel proved this by introducing a new energy model, which determines the energy consumption of algorithms with regard to the input size. Their main point was that different instructions are performed by different components of the processor, such as the ALU, or the multiplier, or by a combination of components. Those components consuming different amounts of energy result in different energy consumption for a variety of instructions. Since not every component of the processor is used for every instruction, components can be switched off when they are not used; however, energy is also consumed when the components are switched on again. All these aspects, when taken into consideration in the energy model, show that with different input sizes, slower algorithms could consume less energy than faster ones.

Another research paper by Hagen Höpfner and Christian Bünse about energy Aware DBMS [4], states that energy consumption grows with the inbuilt flash memory size. They assumed that the energy consumption related to the execution of a specific algorithm mostly depends on its memory requirements, and that the algorithm's complexity class (i.e. number of execution cycles) plays only a minor role. This highlights the fact that memory intensive implementations consume more energy than CPU intensive ones, and that energy consumption is not solely correlated to the complexity class (performance) of an algorithm.

4. POWER AWARE COMPILERS

Power-aware compilers are another approach to enhancing energy and power consumption.

Chung-Hsing Hsu, Chingren Lee, Jenq Kuen Lee, and TingTing Hwang [5] [6] stated that the energy, E , consumed by a program is given by $E = P \times T$, where T is the number of execution cycles of the program and P the average power.

The average power P is given by $P = \frac{1}{2} \cdot C \cdot V_{dd}^2 \cdot f \cdot D$,

where C is the load capacitance, V_{dd} the supply voltage, f the clock frequency and D the transition density. In the compiler optimizations, if the program is optimized for performance, T will be reduced and so will energy consumption. If the compiler performs software refinements to reduce P , without a software performance penalty, it will also reduce the energy consumption. Therefore, it is preferable that any power minimization technique should incur no performance penalty.

There are many ways and techniques to enhance energy consumption via power-aware compilers. For example,

- Reducing switching activities of the instruction bus: In CMOS circuits, power is dissipated in a gate when the gate output changes from 0 to 1 or from 1 to 0. Minimization of power dissipation can be arranged at instruction-level by scheduling instructions to reduce the power consumption on the instruction bus [6].
- Dynamic Voltage Scaling: This is mainly applicable for embedded systems. Its main idea is to change the voltage of the power supply unit in some places (called power management points, PMPs) during program execution in such a way that energy consumption decreases while the needed performance is still maintained. This is possible because energy consumption has a quadratic behaviour depending on voltage, and the processor frequency (and hence the performance) depends on voltage only linearly [7].
- Bit-switching minimization: It is known that switching bits on control and data buses of the processor accounts for a significant amount of power consumption. This could be minimized by predicting the bit encoding of the assembler instruction(s), and then using a scheduling heuristic to control the weight of the instructions based on the critical path priority and the number of bit-switching [7].
- Tuning of memory optimizations: The memory subsystem is considered one of the most energy-consuming parts. Hence, minimization of memory accesses and optimization of cache behaviour is an important way to lower energy consumption [7].
- Power-aware loop optimization is also a promising technique within the power-aware compilers area.

5. POWER AWARE CLOUD COMPUTING

"If the cloud were a country, it would have the fifth largest electricity demand in the world." That is what Gary Cook said in his paper titled "How Clean is Your Cloud?"; his statement is based on the fact that the combined electricity demand of the internet/cloud (data centres and telecommunications network) globally in 2007 was approximately 623bn kWh [8]. A cloud architecture enables services to be accessible from multiple end-use devices, which allows hardware and software sharing. Doing the same tasks in the cloud instead of

a local computer can reduce the total energy use because of the efficiency of using remote shared resources. However, in some cases, cloud computing can consume more energy than doing the same computation on a personal computer. These cases are when the cloud is accessed frequently [9]. For example, according to the Music Tank Report, streaming an album over the internet many times can cost more energy than the manufacturing and production of its CD equivalent [10]. Another example can be taken from Seetharam et al. study titled "Shipping to Streaming: Is this shift green?" [11]. According to Seetharam et al., watching a movie on a laptop consumes 2.788MJ regardless of its delivery method; the energy cost of delivering the movie using a DVD shipping method is 1.396MJ; and the energy cost of delivering the same movie using an energy optimized streaming method is 0.265MJ. This means that watching the movie once on the cloud will consume less energy than delivering it using a DVD shipping method. However, watching the same movie 6 times on the cloud will consume more energy than watching it through the DVD.

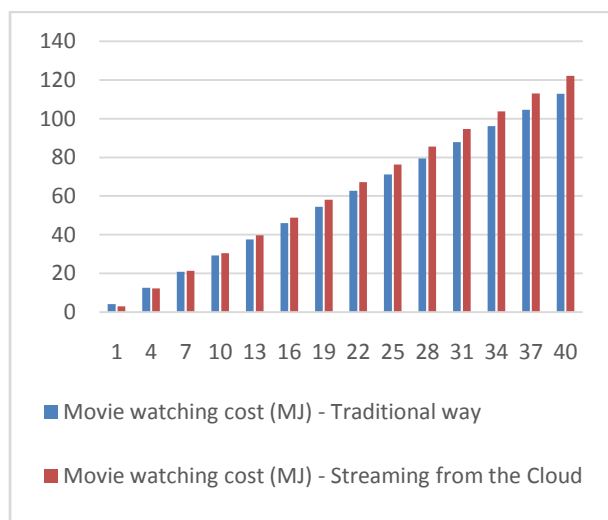


Fig 1: Energy consumption for watching a movie

Although a lot of research has been done on cloud computing, calculating the energy consumption of these systems is still an unresolved question because of the difficulty of calculating the energy consumed by a cloud system. In cloud systems the number of physical machines is reduced, but in order to keep the performance as it is, more powerful machines have to be used, which could consume more energy. Unlike private machines, these systems have to be continuously up and running. So it is very hard to decide whether it actually leads to energy savings or higher consumption.

James W. Smith in his paper "Green Cloud Literature Review" [12] explained that in order to reduce energy consumption, the hardware used in building datacentres could be improved to be more energy efficient, power efficient software could be used, or, setup, build, and operational related approaches could be applied. Below we discuss some of the points mentioned in his paper.

5.1 Setup & build approaches

In any hardware, as energy consumption rises, the emitted heat increases, which in turn increases the risk of hardware failure. This creates a situation where cooling systems have to be applied to the hardware, leading to the use of more and more energy. This problem becomes more apparent in large datacentres and cloud systems. Many ideas have been introduced to overcome this problem: one of them was to choose the right location for building the datacentre, taking into consideration the geographical location features. For example, choosing a low temperature location will reduce the energy that the infrastructure consumes for cooling.

The next step would be to build power-aware systems that are able to identify the energy consumption of their components, and that are able to adapt themselves based on the energy consumption rate. However, this may require special monitoring hardware to be installed, which may add more cost and complexity to the system. These measures will allow researchers to easily identify improvement areas, leading to long term gains.

5.2 Powering off unused components

Powering off unused components or machines when they are not used is another well-known technique to reduce consumed energy. There are two approaches for applying this technique, and each one of them has its own features and drawbacks.

The first one is to power off some components of the system while keeping the other components on. This will allow a fast return back to the running state, and will of course reduce the consumed energy. However, this will still result in some energy being consumed by the other powered-on components whose only use is to allow fast return to the running state.

The second approach is to power off the whole system, and power it on again when needed. This approach is considered the best in terms of power saving, but the worst when it comes to the performance of returning to the running state.

Table 1. Comparison between different powering-off approaches

Powering off some components	Powering off the whole system
Less energy saving	Better energy saving
The powered on components will keep consuming energy	No powered on components, the whole system is off
Fast return back to running state	Slow return back to running state

Organizations could save a lot of energy and money by applying any of the above approaches to large cloud computing datacentres. However, in some systems the operational consumed energy of powering components on/off is much higher than the energy consumed by keeping the whole system up and running. In this case the above technique will not be useful.

5.3 Task-&resource-based techniques

In some cloud computing systems not all machines are completely utilized because they have to be always up and running to cover any task-requests peak at any given time. This ends up wasting a lot of resources and energy as well. Some techniques have been introduced to overcome this problem.

One of the techniques is to define the number of needed machines based on the number of tasks, and the amount of needed resources by each task (*Task Consolidation*). The approach is to assign the right combination of tasks to the right machine, and power off all the unneeded ones. This could be looked at as a knapsack problem of a set of items with different weights, and of the same value.

James W. Smith identified problems with this approach, as it relies upon being able to accurately predict the resources a task will consume in order to allocate it correctly. In some situations, such as when an application adjusts its performance according to the available resources, this approach may prove impossible as these applications will consume all resources given. Therefore, the new problem becomes a question of deciding how many resources should be allocated to each application.

Another introduced approach is to adjust the resources allocated to the task based on the deadline of delivering that task (*Resource Scaling*). So, if using fewer resources can perform the task within its time limit, then it could be possible to save some resources and reduce energy consumption. An example of Resource-Scaling is Speed Scaling, which adjusts the processor’s speed to consume less power.

However, the above approach does not cover the power consumed by other system components, such as cooling devices, RAM, H/D ...etc. By reducing processor power, the time to execute a task will increase, leading to a higher usage of other system components, which could lead to an increase in the total power consumed to finish the task.

5.4 Load balancing techniques

Traditional load balancing algorithms do not take the consumed energy into account. They spread the load equally according to the number of available resources in order to achieve the best performance. This could lead to a low level of resource utilization and unneeded energy consumption, as in some cases the same performance could be achieved by using fewer resources. Another solution would be the use of the “load-skewing” concept [13] which is exactly the opposite of load balancing. In load-skewing, new requests are routed to busy servers as long as the servers can handle them, which provides the ability to power off unneeded machines.

Table 2. Load balancing vs. Load Skewing

Load Balancing	Load Skewing
Spreads the load equally among the existing resources	Requests are routed to busy resources as long as they can handle them.
Low level of resource utilization	High level of resource utilization
Achieves better performance	May achieve less performance
All machines are used equally, which makes powering off machines as not possible.	Provides the ability to power off unneeded machines
Leads to more energy consumption	Gives the ability to have less energy consumption

6. POWER-AWARE HIGH PERFORMANCE COMPUTING

Although maximum performance is the main goal of High Performance Computing (HPC), energy and power efficiency must be taken into account. Using more energy leads to generating more heat. Handling this heat is considered one of the biggest difficulties in building more powerful supercomputers or larger high performance computing centres. In addition, powering and cooling these systems costs a significant amount of money. This is why energy consumption has become a primary concern in high performance computing. A lot of techniques can be applied to building a power-aware HPC system; some of these are listed below.

6.1 HPC centre location

Choosing the location of the HPC centre plays an important role in saving energy because, as mentioned in section 5.1(Power Aware Cloud Computing – Setup & build approaches), choosing a low temperature location will reduce the energy that the infrastructure consumes for cooling.

6.2 Task consolidation

A task consolidation technique could be used to define the number of needed nodes based on the number of tasks and the amount of required resources, so that the unneeded nodes could be powered off. This must not sacrifice performance, especially in an HPC centre. The time needed for bringing the powered off nodes up again must be taken into account.

6.3 Programming language support

Programming modules could be designed to give importance to power saving. For example, sequence matching tools in bioinformatics can be configured to use resources with power constraints [14].

6.4 Adaptive power allocation

An adaptive power allocation can be done by applying runtime power management to adapt the power use to the application demand. Adaptive power allocation is challenging because it is employed at component level and on application segments, so detailed power and energy profiles of every system component is required [15].

7. COMMONLY USED MEASUREMENT TOOLS

There are many ways to measure the energy consumption of a system. These vary in accuracy, cost, usability, and availability. For example, energy consumption can be measured by using a power meter. This is a device that is placed between the system and the power supply to measure the whole system’s power consumption. Power meters are common tools, easy to use, not expensive, and available everywhere. However, using power meters to measure energy consumption gives accurate results only if the goal is to measure the energy consumed by the whole system. It does not give the ability to measure the energy consumed by internal system components, which makes it hard to analyse which part of the system is drawing most of the energy.

Fitting special hardware tools to measure the energy consumption of specific system components is another example. These hardware tools give a very accurate result of the energy consumed by specific system components. They give the ability to do more analysis about system components and to determine which part of the system needs to be enhanced. Although these hardware tools give very accurate

results, they are expensive, hard to use, and require technical knowledge to set them up.

Software tools can also be used to measure the energy consumption. Using software tools is a lot easier than using hardware tools, and they can give an accurate energy consumption estimate at a very low cost; however, hardware tools give more accurate measurements.

Table 3. Examples of energy measurement tools

Power meter tools	Fitting special hardware tools	Software tools
Provide accurate energy consumption measurement	Provide a very accurate energy consumption measurement	Provide an estimation of energy consumption
Not expensive	Expensive	Not expensive, free most of the time
Easy to use	Hard to use	Easy to use
Available everywhere	Hard to find	Available online for download
Measure the whole system's energy consumption	Measure internal components' energy consumption	Estimate internal components' energy consumption
Hard to analyse which part of the system is drawing most of the energy	Very good for deep analysis	Good for deep analysis that does not require very accurate measurements

Listed below are a few of the common software tools, frameworks, and libraries that could be used in measuring and estimating energy consumption.

- **JouleMeter**— a JouleMeter is a software tool that estimates the energy consumption of virtual machines, servers, desktops, laptops, and even individual software applications running on a computer. It tracks computer resources, such as CPU utilization and screen brightness and estimates power usage. [16]
- **Intel® Power Gadget**— Intel® Power Gadget is a software-based power usage monitoring tool enabled for 2nd Generation Intel® Core™ processors or later. It uses the energy counter in the processor to monitor and estimate real-time processor package power information. [17]
- **Metertools**— Metertools is a shared library that can be integrated into applications to read measurement devices. It reads the power meter attached to the serial port locally or over the network. The library can easily be integrated into applications. It has the flexibility to either do continuous monitoring or a limited number of measurement readings. [18]
- **PMC (Performance Monitoring Counter tool)**— the PMC tool allows accessing the performance counter system of modern AMD processors using the pcounter shared library. It also provides tools to support a general performance log analysis. [18]
- **HotSpot**— HotSpot is an accurate and fast thermal model. It is based on an equivalent circuit of thermal resistances and capacitances which correspond to microarchitecture blocks and essential aspects of the thermal package. It has

a simple set of interfaces that allow integration with power performance simulators. [19]

- **CACTI**— CACTI is an integrated cache and memory access time, cycle time, area, leakage, and dynamic power model. It integrates all these models together to give the ability for more consistent trade-off decisions. [20]
- **McPAT**— The McPAT (Multicore Power, Area, and Timing) is an integrated power, area, and timing modelling framework for multicore and manycore architectures, which includes models for the components of a chip. It also has a flexible XML interface to facilitate its use with different performance simulators [21]
- **Multi2Sim**— Multi2Sim is a simulation framework to evaluate Multicore-Multithread processors. It provides a number of instrumentation capabilities that enable research in application characterization, code optimization, compiler optimization, and hardware architecture design. [22]

8. THE REAL ENERGY CONSUMPTION

Nowadays, no computer device works on its own without requiring connections and services from other systems. Hence, reducing the energy consumption of the direct device or system that we deal with, may not reduce the total energy that this device is actually consuming. For example, the introduction of smart phones has led to a major increase in data centres and cloud systems usage, leading to a major increase in energy consumption across the globe. The number of smartphones in use worldwide surpassed the 1 billion-unit mark in the third quarter of 2012 [23], and according to the latest forecasts from ABI Research the global installed base of smartphones will total 1.4 billion by the end of 2013 [24]. A report from Digital Power Group CEO Mark Mills titled “The Cloud Begins with Coal” stated that, although charging the handheld device itself may require a negligible amount of electricity, using one device to watch an hour of video weekly consumes annually more electricity in the remote networks than two new refrigerators use in a year [9]. This means that reducing the smartphone charging energy does not necessarily reduce the total energy that this device actually consumes, because using one of these smart devices to access the internet tends to use so many other systems. This leads to the fact that saving energy is not a single device problem anymore: it is now a problem for all Information Technology systems, and everyone needs to think differently by keeping energy-efficiency as one of the main design goals for designing hardware and software. In order to actually reduce the energy consumption of a device, all the systems that support and provide services to that device need to be taken into consideration. The first step toward solving the energy consumption problem is to measure and analyse the real amount of energy that this device actually consumes. This measurement can be done by calculating the amount of electricity used to run all the systems and components that deliver connections and services to that device. This means that a new method of calculating energy consumption is needed, a method that not only measures a single system's energy consumption, but also takes into account the energy consumed by all other interconnected systems.

9. SUMMARY AND CONCLUSION

A lot of research has been done on the power-aware computing area, resulting in different techniques and approaches to reducing energy consumption. This paper has discussed some of these approaches and techniques. It discussed the algorithm enhancement techniques that could be used to generate a power-aware algorithm. It talked about power-aware compilers, and explained that power-aware compilers could help in reducing energy consumption by applying dynamic voltage scaling, reducing switching activity, tuning memory, and performing loop optimization. The paper also discussed the power-aware cloud computing systems, and highlighted that in order to reduce energy consumption, the hardware used in building datacentres could be improved to be more energy efficient, power efficient software could be used, or setup, build, and operational related approaches could be applied. It also discussed power-aware HPC systems, and highlighted some power-aware techniques, such as choosing the HPC centre's location, task consolidation, programming language support and adaptive power allocation. Finally, this paper provided a brief description of commonly used energy measurement tools, showing that a new method of calculating energy consumption is needed, a method which not only measures a single system's energy consumption, but also takes into account the energy consumed by all other interconnected systems.

10. REFERENCES

- [1] "How is Electrical Energy measured", http://www.nmsea.org/Curriculum/Primer/How_is_electrical_energy_measured.htm, Requested in April 2013.
- [2] Osman S. Unsal, 2008, "System-Level Power-Aware Computing In Complex Real-Time and Multimedia Systems", Doctor of Philosophy, Department of Electrical and Computer Engineering, University of Massachusetts Amherst.
- [3] Hannah Bayer and Markus E. Nebel, 2009, "Evaluating Algorithms according to their Energy Consumption", Mathematical Theory and Computational Practice.
- [4] Hagen Höpfner and Christian Bunse, 2009, "Towards an Energy Aware DBMS – Energy Consumptions of Sorting and Join Algorithms", International University in Germany, Campus 3; 76646 Bruchsal; Germany
- [5] Chung-Hsing Hsu, 2003, "Compiler-directed dynamic voltage and frequency scaling for CPU power and energy reduction", Doctor of Philosophy, Graduate Program in Computer Science, Rutgers, The State University of New Jersey.
- [6] Chingren Lee, Jenq Kuen Lee, and TingTing Hwang, 2000, "Compiler Optimization on Instruction Scheduling for Low Power", 13th International Symposium on System Synthesis (ISSS'00), Madrid, Spain, 20-22.
- [7] Dmitry Zhurikhin, Andrey Belevantsev, Arutyun Avetisyan, Kirill Batuzov, and Semun Lee, 2009, "Evaluating power-aware optimizations within GCC compiler", GCC Research Opportunities workshop, GROW.
- [8] Gary Cook, 2012, "How Clean is Your Cloud?" Catalysing an energy revolution.
- [9] Mark P. Mills CEO, Digital Power Group, 2013, "The Cloud Begins With Coal-An Overview of the Electricity Used By the Global Digital Ecosystem", Sponsored by: National Mining Association, American Coalition for Clean Coal Electricity.
- [10] Dagfinn Bach, 2012, "MusicTank Report- Dark Side Of The Tune - The Hidden Energy Cost Of Digital Music Consumption", Bach Technology AS.
- [11] Anand Seetharam, Manikandan Somasundaram, Don Towsley, Jim Kurose and Prashant Shenoy, 2010, "Shipping to Streaming: Is this shift green?" Proceedings of the first ACM SIGCOMM workshop on Green networking. ACM.
- [12] James W. Smith and Ian Sommerville, "Green Cloud Literature Review of Energy-Aware Computing", Dependable Systems Engineering Group School of Computer Science, University of St Andrews, UK.
- [13] G.Chen, W.He, J.Liu, S.Nath, L.Rigas, L.Xiao, and F.Zhao, 2008, "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services", NSDI (Vol. 8, pp. 337-350).
- [14] A. R. Surve, A. R. Khomane and S.D. Cheke, 2013, "Energy Awareness in HPC-A Survey, International Journal of Computer Science and Mobile Computing", Vol. 2, Issue. 3, pg.46 – 51.
- [15] Albert Y.Zomaya and Young Choon Lee, 2012, "Energy Efficient Distributed Computing Systems", Volume 88 of Wiley Series on Parallel and Distributed Computing.
- [16] "Joulemeter", 29 September 2011, <http://research.microsoft.com/en-us/downloads/fe9e10c5-5c5b-450c-a674-daf55565f794/>, Requested in January 2014.
- [17] "Intel® Power Gadget", 07 January 2014, <http://software.intel.com/en-us/articles/intel-power-gadget-20>, Requested in 23 January 2014.
- [18] "Power Aware Tools", http://fortknox.csc.ncsu.edu/wiki/pac:power_aware_tools, Requested in January 2014.
- [19] "HotSpot 5.0 Temperature Modeling Tool", 7 December 2011, <http://lava.cs.virginia.edu/HotSpot/>, Requested in January 2014.
- [20] "CACTI", <http://www.hpl.hp.com/research/cacti/>, Requested in January 2014.
- [21] Sheng Li, Jung Ho Ahn, Jay B. Brockman, and Norman P. Jouppi, 2009, "McPAT 1.0: An Integrated Power, Area, and Timing Modeling Framework for Multicore Architectures", Microarchitecture, MICRO-42. 42nd Annual IEEE/ACM International Symposium on (pp. 469-480). IEEE.
- [22] Ubal, R., Sahuquillo, J., Petit, S., & López, P, 2007, "Multi2Sim: A Simulation Framework to Evaluate Multicore-Multithread Processors", IEEE 19th International Symposium on Computer Architecture and High Performance computing, page (s) (pp. 62-68).
- [23] Scott Bicheno, 2012, "Global Smartphone Installed Base Forecast by Operating System for 88 Countries 2007 to 2017", Strategy Analytics.
- [24] ABI Research, 2013, "45 Million Windows Phone and 20 Million BlackBerry 10 Smartphones in Active Use at Year-end; Enough to Keep Developers Interested".