

A Survey on Energy Efficiency from Programming and Algorithmic Perspective

Deepthi T¹, Birunda Antoinette Mary J.²

PG Scholar, Department of Computer Science, Stella Maris College, Chennai, India¹

Assistant Professor, Department of Computer Science, Stella Maris College, Chennai, India²

ABSTRACT: All Handheld and IoT devices rely on batteries for power sources. But a battery can store only limited energy. Therefore, there is a necessity for continuous loop of evaluation of hardware and software that runs on these devices from energy utilization point of view. This paper studies the impact of programming languages and algorithms on power consumption. Also, this study includes the tools to measure the power consumption at operating system process level.

KEYWORDS: Algorithms, programming languages, programming language features, internet of things, energy efficiency, power measurement tools, green computing.

I. INTRODUCTION

The primary goal of energy efficient computing is to produce products and services which reduce the amount of energy required. (i.e) using small amount of energy to provide the same service [1]. For example, a smart phone detects user inactivity and switches off display unit to conserve energy.

Need for energy efficiency: Energy is considered as work which we do and is generated from various resources like biomass, coal, gas, geothermal, nuclear, solar, water and wind. The usage of electricity in the home is produced by burning the coal, nuclear reaction or hydro-electrical. The energy which is generated by means of sources is known as energy sources. The energy source can be further divided into two category renewable energy which can be restored easily and non-renewable energy which can't be restored. With the help of energy source we can produce useful energy called heat, which is a primary energy and electricity which is the secondary energy.

Burning of fossil fuel release carbon dioxide, a powerful greenhouse gas that has been stored in the earth for millions of years contributing to global warming. Mining of coal leads to irreversible damage to the environment.

The need for storing generated energy had to lead to the invention of batteries. Also to avoid the emission of hazardous gas, usage of batteries has increased in many countries. The battery converts chemical energy into electrical energy and is filled with electrolyte (sulfuric acid). The battery can be non-rechargeable (primary) and rechargeable (secondary). Major difference between these two is, non rechargeable battery can be used for a limited time and the drained batteries are disposed directly to the environment, in most cases. Whereas rechargeable battery will work for a limited time and can be charged even after the battery is drained. Whenever batteries are thrown directly to ground without proper precautions; there is a chance for chemical leakage from the battery will lead to various environmental pollutions and again will increase in the emission of greenhouse gas. The main idea behind saving energy in the high-tech world is to use energy efficient devices, but researchers say 70% of the energy a typical handheld device will consume during the manufacturing. So, more energy would be conserved by reducing power used in the manufacturing of devices, rather than only reducing the quantity of energy required to perform operation. Reuse has the highest impact to reduce carbon emissions more than recycling. Reducing energy use reduces energy costs and may result in a financial cost saving to consumers. Cost for implementing an energy efficient technology is expensive, but it is one time investment for efficient future.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

Energy efficiency and its impact on the internet of things (IoT): Before that we must define the meaning for “things”. It could be a very elementary object or multiplex object. [2] The things don't need to be connected directly to public internet, but connected via network. The IOT is considered as elementary objects that contain embedded equipment to interact with the outside environment.

The IoT devices comprises of hardware, embedded software, connectivity/communication services, sensors and information services for updating and retrieving live data. All elementary objects in an IoT system require energy to keep running. Most IoT devices run 24 x 7. An example may be an IoT device that monitors patients in an ICU and send alert messages to doctor. Therefore, IoT devices have to be programmed ideally to consume less energy. This requires efficient computing algorithms and datastructures.

In the next section, this paper studies the various models, principles, computational style for developing energy efficient programs and outlining the possibilities and potentials to save energy.

II. BACKGROUND STUDY

1.1 Landauer's principle:

Landauer's principle states that the observer couldn't able to extract work from that system, if an observer loses information about a physical system. This principle states that a sure of one bit of information requires $kT \ln 2$ energy. This energy E_L known as landauer limit and is given as

$$E_L = kT \ln 2$$

Where,

k - Boltzmann's constant (approximately $1.38 \times 10^{-23} \text{J/K}$), T - Temperature of the heat sink in kelvins,

$\ln 2$ - Natural logarithm of 2 (approximately 0.69315).[14]

Importance of landauer's limit:

Computational styles in today's context use irreversible logic, wherein an input cannot be identified from the result the system produces. This logic usually destroys information every clock cycle. Thereby leading to more energy consumption much greater than the Landauer's limit.

Researches in domain have lead to a computational style that is reversible and does not require destroying information. Reversible logic can be a breakthrough in the field of energy efficient computing thereby leading to fundamental computational changes which are in demand to meet the toll on increasing energy requirements.

1.2 Reversible computing and Irreversible computing:

Reversible computing, in a given program, the program must be executed in such a way that its forward execution could be paused at any moment to obtain the previous steps exactly by reversing the execution. The backward execution can also be paused at any moment and the forward execution of the program can be retraced. This ability to pause the forward or backward execution at any point and the ability to switch the direction of execution is called reversible computing. Whereas, irreversible computing follows only forward execution we cannot trace the previous step by pausing the execution, in other words switching the direction of execution is not possible.[15]

1.3 Instruction level energy model:

This model relates energy consumption directly to software code, which help the developers to understand about the energy cost. An energy model needs to consider the following:

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

- The energy consumption of all machine instructions has to be included in the model.
- The energy consumed due to these state changes can be minimized by optimizing the instruction schedule.
- The compiler has to decide on the location of memory objects.
- Bit toggling on buses.
- Parameter of the energy model for electrical measurements and these measurements are simple to perform without requiring knowledge about the internal structure.
- Reusability of the model and should be able to describe more than one specific processor. It is necessary to consider all these properties to optimize and sustain efficiency by using this model.[3]

1.4 Language used for development of IOT application:

- **Assembly:** it is a low level programming language and is abbreviated as **asm**.
- **B#:** This language can be used in multiple platforms using an *EVM* (Embedded Virtual Machine). It is the best language for basic small IOT applications.
- **C:** is the one of the most important programming language for embedded devices. C has been used with IOT boards.
- **C++:** It has the processing power of an object oriented preprocessor whereas c doesn't have this feature. It is used as a preprocessor for C to enable it to run high level languages and help developers to extend their programming code for IOT and embedded code. Further, C++ helps you use other languages including C#, Java, and Python.
- **Go:** This language shares many features of C. Go is developed by google and it is stronger than C. It makes the devices to send and receive data in many channels simultaneously. But there is a high possibility of data loss or errors.
- **Java:** it is probably the most trendy language and choice among IOT developers because when it comes to end user IOT we can write it once, and run it anywhere and can be executed in different places like Smart phones and Servers.
- **JavaScript:** it is a programming language used in web browser and html. Although it takes a section from other programming languages like C and Python but it shares libraries of java language.
- **Parasail:** Parasail is a good choice if your application needs parallel processing. The developer must be able to differentiate between simultaneous processors and parallel processors. Parasail has similar syntax of languages like c#, java and python.
- **Php:** Website prototypes and blogs are things which come to the mind when we think about PHP, not in the field of Internet of things. In many applications developers have involved a php code.
- **Python:** Python is a scripting language. It is used to combine code together. It is one of the primary languages used by a many developers. Small device has restriction in power and memory, so developers had to end up with Python. As of now, most of the popular micro controllers are also using python.
- **Rust:** It is an open source language developed by Mozilla and considered to be same as *Go*, but it can do certain things that are not possible in *Go* language. Rust is capable of sharing information among different channels spontaneously. Limitation is, the processor must be enabled to maintain concurrent processing.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

- **Swift:** Swift is the language for developing iPhone operating system applications, that can communicate through Apple phones and Apple Pads with your home hub.
- Currently Java is the programming language that is used in all Internet of Things (IoT) devices.

1.5 Tool for measuring energy efficiency:

- **Perfmon:** Performance Monitor is a simple and powerful monitoring tool for analyzing performance data, both in real time and from log files. Performance data is represented by means of graph, histogram, or report. This tool can be accessible in Windows 7, Windows Server 2008 R2, Windows Server 2012, Windows Vista.[17]
- **Powertop:** Unix/Linux environments it is a powerful tool to detect energy efficiency in applications. It monitors while a specific program is running, it gathers details about statistics of CPU usage, CPU time, memory usage, caching information and statistics about P- and C-States as well as the different frequencies the CPU was clocked at during measurement. This tool must be within the basic toolbox of every developer who is concerned about energy efficiency in Linux/Unix environments.[13]
- **SoftWatt:** It is an energy profiling tool and it runs on SIMOS. This tool is especially useful for detecting what parts of the software consume the most energy. SoftWatt will provide details like number of cycles used; kind of hardware is used in those cycles and during those cycles the power-state of hardware is to come with estimation and it is expressed in watts.[20]
- **Joulemeter:** It is a tool where alpha version has been released for free download. Technically it is same as SoftWatt. This tool will be used for energy profiling and less for estimation energy consumption. Joulemeter has three main components: a workload manager, event logger and energy profiler.[20]

1.6 Tools for Operating System Optimizations:

- **Wattch:** It is a framework which is used to analyse and optimize power allocation of microprocessor at architecture level. It has the benefit of low level validation at the industry circuits. [18]
- **Power API:** It estimates the energy consumption of ongoing process by means of operating system. It gathers raw information from hardware devices to estimate the energy consumption. It is a modular architecture which provides measurement unit for specific hardware components. It does not require any external devices to evaluate the energy consumption. It is primarily specified as a C API because C is the preferred language for low-level software on HPC systems and it is also universally supported. However, alternative language for bindings is Python. [18][19]

II. LITERATURE REVIEW

In order to maintain energy efficiency in all handled devices and IoT devices, we need to concentrate from hardware level to application level. Here this paper provides us a comparative study on various techniques, tools and methods to maintain energy efficiency.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

ReferenceNo	Title	Author	Year
3	An Accurate and Fine Grain Instruction-Level Energy Model Supporting Software Optimizations	StefanSteinke, Markus Knauer,LarsWehmeyer,PeterMarwedel.	2001
4	The impact of source code transformations on software power and energy consumption.	Carlo brandolese,Williamfornaciari,Fabiosalice,Donatellasciuto.	2002
5	Principles of a Reversible Programming Language	Tetsuo Yokoyama,Holger Bock Axelsen, Robert Glück	2008
6	How to Measure Energy-Efficiency of Software: Metrics and Measurement Results	TimoJohann, MarkusDick, StefanNaumann,Eva Kern	2012
7	Green Software and Green Software Engineering –Definitions, Measurements, and Quality Aspect	Eva Kern, Markus Dick,StefanNaumann, AchimGuldner, Timo Johann	2013
8	A Novel Human Computer Interaction Aware Algorithm to Minimize Energy Consumption	P. K. Gupta , G. Singh	2014
9	Energy Efficient Software Development Life Cycle	Sunil Kumar Sharma ,P.K. Gupta, Reza Malekian	2015
10	Measuring energy footprint of software features	Syed Islam, Adel Noureddine, and RabihBashroush	2016
11	Jolinar: Analysing the Energy Footprint of Software Applications (Demo)	AdelNoureddine, Syed Islam,RabihBashroush	2016
12	Energy-Efficient Algorithms	ErikD.Demaine, JaysonLynch, Geronimo J. Mirano, NirvanTyagi	2016
13	Energy Efficient Programming	Merlin Steuer	2016

Table 1: Papers reviewed for energy efficiency

[3] Steinke et al proposed that, energy consumption in CPU is measured with help of haming distance. Haming distance is measured by sequence of two different data or instruction. They proposed instruction level model to optimize the software. Linear measuring method is used for measuring the efficiency and it doesn't require deep knowledge about the processor. The proposed model is established for ARM7TDMI processor.

[4] Brandolese et al proposed that, reducing the energy consumption of software by means of compiler technique. The technique used in this paper is, source code transformation. By using this technique some new transformation like: (i) Loop transformations, (ii) Data Structure Transformations, (iii) Inter-Procedural Transformations and (iv) Operators and Control Structure Transformations has been identified for the purpose of reducing the powerconsumption.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

[5] Yokoyama et al done implementation on reversible Turing machine interpreter and a fast Fourier transform algorithm to demonstrate strength and possibility of the language. Janus is a tool used to clarify and illustrate the principles of reversible programming. The tool is used to clarify other related concepts in future work, like reversible input/output models and recursive structured data. Further review must be on time and space complexity for reversible computing.

[6] Johann et al proposed metric and measurement method. Software Metrics are used to measure the characteristic of the project. Measurement is considered as quantitative indication size of attribute of some process or amount. This method enables software developers to continuously measure and monitor the energy consumption of the software during the development process.

Basically metric is defined as:

Energy efficiency = Productive work / Energy consumed

For measuring software energy consumption, the already exists approach is white box measurement to notify which part of software there is potential for energy conservation. To perform white box measurement the well known technique is source code instrumentation, which is mostly used in the circumstance of software profiling (the usage of specific instructions) e.g. for runtime analysis. ENERGY EFFICIENCY METRICS is applied to sorting application. The algorithms are: Bubble Sort the complexity is $O(n^2)$ and Heap Sort the complexity is $O(n \log(n))$. In that, the Heap Sort algorithm also sorted 200,000 within 5 seconds. The algorithm with the superior asymptotic run time seems to also have the greater energy efficiency.

[7] Kern et al proposed metric method, which is implemented to a model. This method can easily integrate into green soft model. The main aim of sustainability and Information and Communication Technology is Green IT and Green by IT. In Green IT, there are a lot of solutions to develop and use hardware in a more energy efficient way. Green by IT is basically, a software-based. The reason for inaugurate energy efficient software and systems is to attain a longer battery life and to reduce costs. But in this they are stepping forward for sustainability in ecological part. They also proposed a model called green soft model to measure the energy efficiency of software and help to arrange some aspects of sustainable software and its engineering. One limitation to all discussed approaches in this paper is, it rely on executable software artifacts to read or to estimate their energy consumption.

[8] Gupta et al proposed HCI method to understand the user interaction with computer system. In addition to this, they have also evaluated the memory performance using the proposed algorithm (ie) HCI algorithm. This algorithm estimates the percentage of total CPU usage over a given interval of time. The technique has been implemented on a Windows platform to evaluate the performance of proposed HCI algorithm using CPU to ensure there is no memory losses or breakdown in CPU speed. Various tools discussed in this paper are: pTop, SoftWatt, PowerSpy, SimWattch.

[9] Sharma et al proposed SDLC (Software Development Life Cycle) energy efficient model. It consists of six stages. Green requirement analysis, design, implementation, testing, maintenance and analysis are the stages of SDLC model. All other stages get connected to green analysis phase in a cyclic manner. When one stage is completed, then the analysis phase clarifies and validates to ensure the software is ecofriendly. Implementation is done using greedy capacity algorithm and using simple scalar tool accurate results are calculated. In future work is to simplifying the framework in order to minimize energy reduction and electronic wastage in SDLC.

[10] Islam et al currently focuses on measuring the energy consumption of applications at two main perspectives. The first considers the energy consumption of a particular function or a set of functions, while the second level of perspective is to look at the energy consumed by the entire software or process. Implementation is done on sort and uniq algorithm using feature slicing and energy management technique. Future work is to automate energy monitoring at feature level and expand energy models to additional hardware and platforms in the cloud. Various tools discussed in this paper are: pTop, Power Spy, PowerAPI, Joule Meter or Energy Checker.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

[11] Noureddine et al described about Jolinar, a tool that fill the gap between energy measurements, access to software engineers and end-users. Implementation is done on Chudnovsky's algorithm and Ramanujan's algorithm. They compare the algorithms using a single thread and a multithreaded environment. For calculating the digits of Pi the range is from 10 to 100 million digits using jolinar command line tool. Future work is to allow Jolinar to monitor already started applications. Handling this tool is quite difficult for non technical person.

[12] Demaine et al reviewed the energy complexity of algorithms and it works based on landauer's principle in physics. this principle states that the amount of energy a system will waste if it deletes any information. they also proposed three models like word RAM, circuit RAM and transdichotomous RAM for energy aware variations and primitives like control logic, memory allocation, and garbage collection on top this model. prime motive of this paper is to analyse and design the energy cost of algorithms by irreversibility and that idea leads to reversible computing of basic algorithms for positive results but there is a limitation with fully reversible algorithm .to overcome that semi reversible algorithm is applied for some basic algorithms. Implementation is done on algorithms like: counting sort, comparison sort, insertion sort, breadth-first search, Bellman-Ford, Floyd-Warshall, matrix all-pairs shortest paths, AVL trees, binary heaps, and dynamic arrays. General techniques used in constructing reversible algorithms are Complete Logging, Reversible Subroutine, Data Structure Rebuilding. Future work is about giving more important for some algorithms like: hashing, predecessor data structures, max-flow/min- cut, Fast Fourier Transforms, and dynamicprogramming.

[13] Steuer given an overview about the fundamentals and techniques of energy efficient programming. To reduce the energy loss they primarily focus on hardware layer, operating system and application layer. Bubble sort and quick sort algorithm where used for evaluation purpose using various techniques like: Time efficiency, data efficiency, logging, letting the CPU rest, choosing the programming language, libraries. Tool to measure energy efficiency for Unix is power top and perfmon for Windows

IV. CONCLUSION

This paper identifies the earlier researches that were done in the area of energy efficiency at programming and algorithmic levels. A detailed study had been done on different computational styles and energy model at instruction cycle level that can have an impact in minimizing energy consumption. Various tools that measure energy efficiency at operating system process level had been identified and discussed. All the more energy efficiency at software level relies to a greater extent on the underlying hardware. So future researches on energy efficiency should consider the entire hardware and software lifecycle i.e. the energy consumption involved in development, testing, deployment, production-run and maintenance.

REFERENCES

- [1] https://en.wikipedia.org/wiki/Efficient_energy_use
- [2] Jammes, F. (2016). Internet of Things in Energy Efficiency: The Internet of Things (Ubiquity symposium). *Ubiquity*, 2016(February),2.
- [3] Steinke, S., Knauer, M., Wehmeyer, L., &Marwedel, P. (2001, September). An accurate and fine grain instruction-level energy model supporting software optimizations. In *Proc. of PATMOS*.
- [4] Brandolese, C., Fornaciari, W., Salice, F., &Sciuto, D. (2002). The impact of source code transformations on software power and energy consumption. *Journal of Circuits, Systems, and Computers*, 11(05), 477-502.
- [5] Yokoyama, T., Axelsen, H. B., &Glück, R. (2008, May). Principles of a reversible programming language. In *Proceedings of the 5th conference on Computing frontiers* (pp. 43-54). ACM.
- [6] Johann, T., Dick, M., Naumann, S., & Kern, E. (2012, June). How to measure energy-efficiency of software: Metrics and measurement results. In *Proceedings of the First International Workshop on Green and Sustainable Software* (pp. 51-54). IEEE Press."
- [7] Kern, E., Dick, M., Naumann, S., Guldner, A., & Johann, T. (2013). Green software and green software engineering—definitions, measurements, and quality aspects. *Hilty et al.(2013)*, 87-94.
- [8] Gupta, P. K., & Singh, G. (2015). A Novel Human Computer Interaction Aware Algorithm to Minimize Energy Consumption. *Wireless Personal Communications*, 81(2), 661-683.
- [9] Sharma, S. K., Gupta, P. K., &Malekian, R. (2015, November). Energy efficient software development life cycle-An approach towards smart computing. In *Computer Graphics, Vision and Information Security (CGVIS), 2015 IEEE International Conference on* (pp. 1-5). IEEE.
- [10] Islam, S., Noureddine, A., &Bashroush, R. (2016, May). Measuring energy footprint of software features. In *Program Comprehension (ICPC), 2016 IEEE 24th International Conference on* (pp. 1-4). IEEE.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, September 2017

- [11] Noureddine, A., Islam, S., & Bashroush, R. (2016, July). Jolinar: analysing the energy footprint of software applications. In *The International Symposium on Software Testing and Analysis* (pp. Pages-445).
- [12] Demaine, E. D., Lynch, J., Mirano, G. J., & Tyagi, N. (2016, January). Energy-efficient algorithms. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science* (pp. 321-332). ACM.
- [13] Steuer, M. (2016). Energy Efficient Programming.
- [14] https://en.wikipedia.org/wiki/Landauer%27s_principle.
- [15] Perumalla, K. S. (2013). *Introduction to reversible computing*. CRC Press.
- [16] <https://iot.intersog.com/blog/12-popular-programming-languages-for-iot-development-in-2017/>
- [17] [https://technet.microsoft.com/enus/library/cc749115\(v=ws.11\).aspx](https://technet.microsoft.com/enus/library/cc749115(v=ws.11).aspx).
- [18] Goekkus, F., & Hilty, L. Energy Efficient Programming.
- [19] Grant, R. E., Levenhagen, M., Olivier, S. L., DeBonis, D., Pedretti, K. T., & Laros III, J. H. (2016). Standardizing power monitoring and control at exascale. *Computer*, 49(10), 38-46.
- [20] https://wiki.cs.vu.nl/green_software/index.php/Measuring_green_operation.