

## Energy consumption and architecture and computer technology subjects

**Ishaan Acharya**

Department of Computer Engineering, Vellore Institute of Technology, Tamil Nadu, India.

**How to cite this paper:** Ishaan Acharya, "Energy consumption and architecture and computer technology subjects", IJIRE-V3I04-106-112.

Copyright © 2022 by author(s) and 5<sup>th</sup> Dimension Research Publication.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).  
<http://creativecommons.org/licenses/by/4.0/>

**Abstract:** Program power consumption has become as important a performance measure as processing time, even though it is not usually included in program performance measures. For this reason, it is convenient to include in the subjects of the area of architecture and computer technology content related to the energy performance of programs and architectures and to have tools that allow characterizing the power and energy consumed according to the characteristics of the code to be executed. The need to evaluate programs according to their energy efficiency and execution time constitutes a multi-objective approach to performance evaluation that should be introduced in Computer Engineering subjects. In this article we also describe an Arduino-based system that allows obtaining measurements of power and energy consumed in practices and projects that address the generation of optimal codes for a given platform.

**Key Word:** Arduino, Energy Efficient Computing, Python, Energy Sensitive Planning.

### I. INTRODUCTION

The energy consumed by computers has always been an important measure from the point of view of their own viability. The energy consumption of the first electronic computers was so great that anecdotes circulate about the effect of their operation on the electrical network of the city or the campus where they were located. Thus, the ENIAC I, capable of performing 5000 additions per second, 357 multiplications per second, or 35 divisions per second, consumed a power of about 175KW. From ENIAC I, capable of 0.03operations/s per watt in 1946, to today's computers that can achieve  $10^{10}$ operations/s per watt, the energy efficiency of computers has grown by a factor of  $10^{12}$ , almost equal to the factor of increase in processing speed. Despite this improvement in energy performance, the widespread use of personal computing devices and the development of ever faster supercomputers has meant that the energy consumption of various ICT applications is an important part of global energy consumption and constitutes an essential barrier to the development of supercomputers and data processing centers. In addition, the energy consumption of the processors has been a factor to consider in the proposal of new microarchitectures and poses essential challenges in the development of ever faster supercomputers: currently the development of supercomputers that provide Exaflops.

Thus, the energy efficiency of programs has become a measure of performance as important as processing speed, although it is not usually considered in publications that include information on the efficiency of programs. On the one hand, there is the need to improve the autonomy of devices that use batteries, and on the other, the economic and environmental reasons that motivate the reduction in energy consumption of computing platforms. These platforms can include a considerable number of computing cores, and other elements, which give rise to high instantaneous power consumption values. In addition to the increase in temperature caused by the greater instantaneous power, with the consequent greater probability of component failure and the need for more complex ventilation systems, the cost of the energy consumed can be higher than that of the platform itself. [1]. In fact, the not inconsiderable contribution to emissions of CO<sub>2</sub> information and communication technologies has been observed for some time. For example, the environmental impact of emissions from CO<sub>2</sub> data processing centers in the United States was similar, in 2010, to that of economies in countries such as Argentina [2]. The interest in the energy efficiency of high performance computers is evident in the appearance of the Green500 list [3] that has shown improvements of a factor of 5 in the MFLOPS/W values between 2009 and 2013, and the feasibility Exaflop-scale computing by 2020 requires even higher rates of improvement if you want to have a system with a speed of the order of  $10^{18}$ flops that consumes a few 20MW at most [4].

Therefore, it is convenient to incorporate and intensify the contents related to energy consumption in undergraduate architecture and computer technology subjects, especially in Computer Engineering studies, highlighting the multi-objective perspective from which the optimization of energy must be approached. code and performance evaluation. For the teaching of these topics, among other aspects, it is necessary to have tools and strategies to measure the energy consumption of computers and conveniently implement the codes.

In Section 2 of this article, some of the circuit energy consumption models are described along with some techniques to improve the energy efficiency of computers and programs. In Section 3 we will describe the Arduino-based measurement system that we have developed, and in Section 4 we will propose content related to energy consumption that we believe could be included in some of the subjects related to architecture and computer technology in engineering degrees. computer science, and more specifically in the Computer Engineering degrees. Finally, Section 5 provides the conclusions of the article.

## II. STRATEGIES TO IMPROVE ENERGY CONSUMPTION

Power consumption has been an important factor in the design of microarchitectures. In the field of embedded processors, given the characteristics of the systems in which they are included, restrictions related to power consumption and heat dissipation have always been essential. On the other hand, the evolution towards multi-core general purpose architectures has been a consequence of the need to limit the power per unit area dissipated by the microprocessor, which could have reached unacceptably high values if the improvement in performance had been kept associated. capabilities to increasingly complex superscalar microarchitectures, in order to complete more instructions per cycle. Thus, at the beginning of the century, different alternatives were proposed to maintain the pace set by Moore's law. Among them, the general-purpose VLIW microarchitectures, in which the compiler was primarily responsible for extracting parallelism and avoiding structures that consume power and hardware resources, such as renaming or reordering buffers. Microarchitectures with several processing cores were also proposed, which have ultimately become the dominant trend in the evolution of microprocessors.

To understand the influence of technology on the dissipated power of a CMOS integrated circuit, expression (1) [4] can be used, in which the first term is due to the dynamic charge and discharge consumption of the output capacity,  $C$ , of a logic gate, in an integrated circuit with an activity coefficient,  $A$ , which represents the fraction of circuit gates that switch in each cycle, at a voltage,  $V$ , and a clock frequency,  $f$ . The second term of (1) is due to the current between power supply and ground,  $I_{cortoc}$ , "during an instant of time,  $t$ , in which the" gate. The third and last term is the power consumed due to the leakage current,  $I_{leak}$ , "independent of the state of the logic gate. It is possible to reduce the power consumed by lowering the frequency of the circuit or the supply voltage (with the consequent effect on the speed of the circuit).

$$Power = ACV^2f + tAVI_{cortoc} + VI_{leak}$$

However, it must be taken into account that the maximum frequency at which the circuit can work is related to the supply voltage, as shown in expression (2):

$$f_{max} \sim \frac{(V - V_{umbral})^2}{V}$$

Thus, to increase the processing speed by increasing the processor frequency, the threshold voltage would have to be reduced, which  $V_{umbral}$ , in (2). Nevertheless, this would mean increasing the loss current, which grows exponentially as the threshold voltage is reduced, as shown in (3):

$$I_{leak} \sim e^{\frac{-qV_{umbral}}{kT}}$$

Therefore, the increase in the maximum frequency of the microprocessors by reducing the supply voltage together with the threshold voltage is limited by the increase that originates in the power associated with the leakage current, and thus alternatives arise, such as the VLIW microarchitectures of general purpose or multicore microarchitectures, to which we have referred.

Along with power, there are other performance measures related to the power consumption of integrated circuits. These include power density, or power consumed per unit area; the peak power, or maximum power that a circuit can consume without suffering damage; and dynamic power, which refers to the sudden change in power consumption that can occur in the circuit without the noise associated with temporary current variations causing abnormal behavior. The energy required to perform a calculation is an important magnitude, for example, to assess the efficiency with which the energy stored in a battery is used. Thus, a processor may consume less power than another to execute a program, but if it takes longer, it may happen that in the end the energy consumed is greater. Related to the power required to perform a computation is the MIPS/W (millions of instructions per second per watt) figure of merit, which connects the instructions executed and the power consumed in executing them. The higher the MIPS/W of a processor, the better it is from the point of execution. The higher the MIPS/W of a processor, the better it is from the point that equations (1-3) can apply since the power consumption depends on the can apply equations (1-3) since the power consumption depends on the squared of the voltage, it is possible to reduce the power by a given factor, so that the frequency (and therefore the speed of the circuits) is affected by a smaller reduction factor. This allows for a trade-off where the speed of the circuit is reduced somewhat to achieve an increase in MIPS/W. In addition to the energy consumed by the processor or processors, there are other factors that affect the energy consumption in the computer that are related to the operation of the memory, the buses and the interconnection systems, the input/output devices, etc. Therefore, efficient computing from the energy point of view implies techniques based not only on the design of the microarchitecture, but also on the computer system and operating system levels [5].

For example, memory consumes a significant portion of power. This consumption depends on two fundamental causes: the access frequency and the loss current ( $I_{leak}$  in expression (1). The organization of the memory in such a way that only the part of it that is accessed is activated (affecting the coefficient  $A$  in expression (1)) contributes to reducing

consumption. Thus, if the memory is distributed among banks that can be activated independently, consumption can be reduced by activating the necessary banks at each moment. The locality of data and instruction cache accesses determines the efficiency with which this technique can be applied. Another strategy based on the locality of the accesses consists of placing a small cache in front of the L1 cache to reduce the number of accesses to the L1 cache. If hits to this cache occur only 50% of the time, half the difference between the consumption of the L1 cache and that of the small cache would be saved.

The buses and other interconnection elements are also responsible for an important part of the energy consumption, especially the buses between integrated circuits, with a considerable number of lines, whose drivers can represent 15%-20% of the power of the integrated circuit. There are techniques that implement bus controllers together with the appropriate encoding/decoding circuitry. Thus, gray coding bus addresses can reduce level transitions on bus lines when addresses change sequentially, for example on cache line transfers. The transmission of the difference between addresses that are requested successively and the compression of the information of the address lines to minimize the number of bus lines also improves the energy consumption in the buses.

Efficient use of parallelism is an important way to reduce the power consumption of applications. According to expression (2) it is possible to reduce the clock frequency by reducing the voltage and, furthermore, from (3), a reduction of the voltage has important advantages in terms of energy consumption. Figure 1 shows the effect of distributing a workload  $W$  between  $p$  processors that operate at a frequency  $p$  times lower than that of the starting processor. Each of the processors would consume  $p^2$  times less power. Since  $p$  processors are used, it would take  $p$  times less total power to complete the workload at the same time.

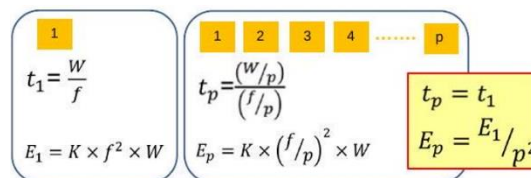


Figure1: Parallel processing and energy consumption.

Also, the processor does not need to always run at its maximum frequency. Thus, as shown in Figure 2, depending on the characteristics of the time synchronization between tasks that are processed in parallel, it is possible to reduce the frequency at which one of the processors works and reduce the energy consumed without increasing the processing time. The runtime system can control the voltage at which the processor works, and applications can also be made to use operating system functions to control the voltage values they require.

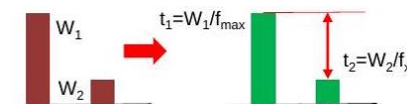


Figure2: Adjusting frequencies to reduce energy consumption in tasks with different workloads.

According to this, to reduce the energy consumption of the application, the frequency at which the cores work (DVFS, for Dynamic Voltage and Frequency Scaling) could be adjusted. In any of the cases, the effect that the reduction in energy consumption may have on the speed of the application must be taken into account. Especially in those cases in which a parallel program is being developed precisely to gain speed, or there are real-time restrictions to satisfy. Therefore, as the reduction of energy consumption appears as one more objective among the benefits of the codes, the need arises to reach a compromise between the benefits of the codes, the need arises to reach a compromise between energy consumption and speed, and performance optimization becomes a multi-objective problem and two questions arise related to this fact: (1) what is the configuration of cores and/or operating frequencies that allows the minimum energy consumption and that the application ends within the requirements set time, and (2) what is the configuration that provides the maximum speed with a power consumption less than a maximum consumption limit.

The design of application load scheduling algorithms that take into account consumption requirements in addition to speed needs to have a characterization of the energy consumption of the application on the platform on which it will be executed. This characterization is not simple. Multi-core microprocessors have a control unit, PCU (Package Control Unit), which implements certain heuristics to determine the frequencies of use of the microprocessor cores based on factors such as the number of OS calls, the expected use of the cores, temperature, cache miss rate, etc. Taking all these factors into account, the microprocessor manages the different operating modes (C and P states in AMD and Intel microprocessors, for example), and changes in these modes must be related to the type of processing performed by the application. and the energy savings achieved. There are black-box approaches that seek to characterize the energy consumption of a specific application by correlating the characteristics of the corresponding codes with the observed consumption [1,6,7]. For example, [4] illustrates the use of the power saving modes of the ACPI specification (Advanced Configuration and Power Interface) [8] implemented in the parallel resolution of sparse linear systems. It is about establishing the relationship between the characteristics of the code and the conditions in which the operating system, through ACPI in this case, promotes the changes to the states of energy saving in the cores of the microprocessors, if it cannot be caused the transition from the application. In any case, when promoting the transition to modes in which the instantaneous power consumed is reduced, it must be taken into account if there

is a penalty in the execution time that affects the final energy consumption so that no savings are produced. of energy.

Another alternative is based on the use of more or less approximate theoretical energy consumption models to design load distribution heuristics well on platforms that implement DVFS (Dynamic Voltage and Frequency Scaling) for which it is assumed that the frequency levels can be acted upon. or voltage, either on heterogeneous platforms including processors with different frequencies and processing capabilities [9-12].

### II. AN ENERGY CONSUMPTION MEASUREMENT SYSTEM

This section describes the system we have developed to measure energy consumption in real time on a computer (Figure 3). Specifically, the system makes it possible to measure individually, in each one of the nodes of a cluster, both the instantaneous power (in W) and the accumulated energy consumption (Wxh).

The measurement is carried out through sensors that provide the current intensity that circulates in the current cable that connects the equipment to the electrical network. All if the measurement is carried out at this point, it is possible to obtain the real consumption of the entire node, including both that of the active components and that due to the losses associated with the conversion in the power supply. The conclusions regarding the energy efficiency that can be reached after the study of the behavior of these consumption measures of the complete node are relevant, since they show whether the strategies that have been implemented to improve energy efficiency are observable and therefore, meaningful, and useful.



Figure 3. Arduino Mega based board and current sensors.

The system consists of an Arduino Mega board and a set of sensors connected directly to power outlets. Once connected, the current cable of each equipment passes through each sensor, which detects the intensity of the current that circulates. The sensor used is the YHDC-SCTD010T-5A, whose schematic is shown in Figure 4. It can measure up to 5 A and has a proportional output between 0 and 5 V, with an accuracy of  $\pm 2\%$ . The Arduino has an internal 10-bit A/D converter and, to better exploit its dynamic range, its internal 2.56V reference is used, which is only available on the Arduino Mega. Although this limits the measurement to 2.56V maximum input, on the Arduino Mega. Although this limits the measurement to 2.56 V maximum input, in practice up to 588 W of power could be measured. Since the nodes consume less power, this upper bound does not represent any limitation.

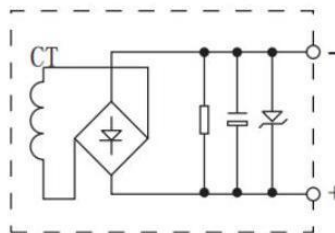


Figure 4. Internal diagram of the current sensor.

The Arduino Mega is connected via USB to one of the nodes of the cluster, which allows, in addition to supplying the corresponding power, to transmit the data through the serial port. In the case of a Linux computer, the `/dev/ttyACM0` interface is created, which allows serial data to be transmitted and received. In the first version implemented, the Arduino makes 4 captures per second of each sensor. Every second, it transmits through the port the power data (in W) of each sensor, as well as the energy consumed (in watts per hour, Wxh). The consumed energy values can be reset to 0 at any time to easily re-estimate consumption over a period of time. The capture program that the Arduino Mega runs is written in Python.

### IV. CONTENTS ON ENERGY CONSUMPTION IN COMPUTER ENGINEERING

In the IEEE/ACM curricula for undergraduate studies in Computer Science [13] and Computer Engineering [14], reference is made to content related to energy consumption. Thus, in the Computer Science curricula, in the Architecture and Organization knowledge area, energy is included within the physical restrictions for the Digital Logic and Digital Systems courses, and within the topic of performance improvement in the Digital Components courses. and design. Also within the area of Social Issues and Professional Practice, the Computer Science curricula include content related to energy consumption in Sustainability subjects that describe the environmental impacts of computer system design decisions related to algorithms, operating systems, networks, databases etc. Regarding the Computer Engineering degree studies [14], the concepts related to

the energy consumption of Computer Engineering [14], the concepts related to energy consumption are included in the Embedded Systems area in the Techniques for low-power operation (CE-ESY-10), in the area of Systems and Project Engineering within the course of Maintainability, sustainability, manufacturability (CS-SPE-12), and in the area of Electronics, in the course of Foundations of Electronics (EL E\_2101). Apart from the inclusion of energy consumption criteria in hardware design subjects, as reflected in the Computer Engineering curricula, concepts related to energy efficiency should be taken into account in a similar way to those related to execution time and hardware complexity, for example. In fact, in the introduction of the Architecture and Organization knowledge area in [13] it is indicated that IT professionals cannot see the computer as a black box that executes programs by magic, but must have a deep knowledge of the hardware on which computing is based and the interface that provides the architecture to the higher software layers, and they must know the functional components of the computer, its characteristics, interaction and benefits, to develop high-performance programs. High performance or the feasibility of an application running on a computer is not only related to runtime constraints, but also to power consumption. Therefore, we consider interesting the inclusion of contents related to energy consumption in the subjects of the degree in Computer Engineering (and in the specialty in Computer Engineering) of the University of Granada. Given that the contents of the subjects that are currently taught are defined in the corresponding guides [15] in accordance with the credits and the period of time assigned to be taught, and with the objectives and contents of the degree report, integrate the new topics is not trivial, especially considering the time constraints and the students' workload. In subjects that do not already include the corresponding topics related to energy, these can be introduced through seminars and demonstrations in some of the practice sessions. For example, in some of the practical exercises that involve performing program performance measurements, along with the usual measurements that use the number of processors, those that consider the energy consumed can also be included.

The contents of energy consumption to be introduced in degree subjects can be classified taking into account their relationship with: (1) the measurement of energy consumption; (2) control of operating modes; and (3) the development of energy efficient programs. Regarding the measurement of energy consumption, apart from using systems such as the one described in Section 3 of this article, it is possible to make use of the performance counters (Performance Monitoring Counters) provided by the processors. For example, [16] describes an extension of the PAPI (Performance API) library to measure power and energy consumption based on performance counters. Regarding the control of operating modes, the standard interface ACPI (Advanced Configuration and Power Interface) [8] includes mechanisms for energy management and saving, conveniently controls the operation of the BIOS, and provides information about the configuration and the control of the energy saving states (C0, C1, C2, C3,..., Cn) and performance (P0, P1,..., Pn) of the processor. Along the same lines, for example, the Linux kernel implements the cpufreq infrastructure [17], which allows the operating system, either automatically from events generated by ACPI or from user program calls, to increase or lower the processor frequency to save power. Within the cpufreq are the so-called governors [18], which implement specific policies to control the speed of the processor's clock. The interface for using these services at the user level can be found in cpufreq.h [19,20,21]. In the first place, subjects such as Physical and Technological Foundations could include models of energy consumption in integrated circuits and main elements of computer electronics (buses, memories, etc.), addressing the relationship between concepts such as frequency, voltage, power and consumption. energetic. In Computer Technology and Organization, along with the criteria used to compare alternative designs of computer elements based on execution times and complexity, other criteria related to energy consumption could be included. The introduction that these core courses can provide to concepts related to energy consumption would be very useful, not only for later degree courses such as Computer Structure, Computer Architecture, and Server Engineering, but also for Operating Systems and the courses of the specialty in Computer Engineering. Precisely, in subjects related to hardware design, such as Digital Hardware Development or Embedded Systems, energy consumption criteria are important from the point of view of the autonomy of battery-powered devices.

In Computer Structure, where, among other contents, the study of assembly language is addressed, exercises related to consulting performance meters that provide information on energy consumption could surely be included in the practices carried out in this regard. As noted, microprocessor manufacturers include features that allow the clock frequency at which the processor operates to be changed to save power. Knowledge of these states and their usefulness, including references to interfaces such as ACPI, are important topics to consider in this course.

In subjects that deal with issues related to parallelism, code optimization or performance evaluation such as Computer Architecture, Server Engineering and Architectures and High Performance Computing (the latter in the Computer Engineering specialty) criteria of code optimization and efficiency measures that take energy consumption into account. In fact, parallel processing is an effective strategy for this purpose. For example, the instantaneous powers measured by the system described in Section 3 in one of the nodes of a cluster for a sequential program (a multi-objective evolutionary algorithm for feature selection in an electroencephalogram classification problem). and two different parallel versions (one based on the paradigm of teacher-worker, MW, and another on the paradigm of islands). As can be seen, the instantaneous power of the parallel programs, which use eight processing cores, reaches higher values than the sequential program. However, taking into account that parallel programs manage to reduce processing time, in the end speed gain factors of around seven (with eight cores) are achieved, with energy consumption of around a third of that of sequential. From the measured power consumption curves, even the situation in which the parallel program is revealed: power drops in the master-worker parallel program when only one processor is working or increases in the islands parallel program when there is communication between these islands.

Proper use of the resources that are consuming energy can improve energy efficiency in addition to reducing computation time. A usual technique to improve performance in current processors that is studied in the Computer



Architecture course is speculative processing, which allows calculations to be started before the instructions on which the result of said calculations is useful or useful have finished. Right. If the prediction that has been made is correct, the processing time is reduced and the MIPS/W ratio can be greatly improved. However, since a number of energy-consuming resources are used to implement speculation efficiently, if the speculation is not correct, energy is wasted. Along these same lines, optimizing the use of computing resources such as buses or communication networks, memory or the use of the appropriate level of precision in calculations not only has an effect on computation times but also in energy consumption. It is important to introduce these energy criteria in performance optimization. On the other hand, the use of energy performance criteria makes it possible to compare the efficiency with which different current heterogeneous parallel architectures are used, where a performance measure based on the division of the gain between the number of processing cores can be debatable given that there may be cores with quite different computing capabilities.

The importance in managing the energy consumption of large computing infrastructures and the consumption requirements that currently arise in the design of supercomputers (TOP500 and GREEN500 lists) must be addressed in courses of the Computer Engineering specialty such as Data Processing Centers, Data, and Architectures and High Performance Computing. In fact, the subject of Data Processing Centers already includes the study of the technical requirements and the energy management regulations of these infrastructures.

## V.CONCLUSION

In this article we have shown that the mechanisms and strategies implemented in current architectures to improve their energy consumption, together with the tools available to make use of them, allow the elaboration of proposals for practical work and projects that facilitate the incorporation of related content. with energy consumption in Computer Engineering and Computer Engineering studies.

On the other hand, the study that has been carried out on the effects on energy consumption of certain programming strategies is contributing to the identification of efficient programming techniques from the energy point of view that can be studied together with those that seek optimization. of execution times. This situation poses a multi-objective perspective in the study of the performance of computers in terms of execution time, energy consumption, and the quality of the solution achieved, very suitable for the training of computer science professionals as engineers.

## References

- [1]. De Sensi, D.: "Predicting performance and power consumption of parallel applications". In *24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016. DOI: 10.1109/PDP.2016.41
- [2]. Mudge, T.: "Power: A first-class architectural design constraint". *IEEE Computer*, Vol. 34, No. 4, pp.52-58. April, 2001.
- [3]. Aliaga, J.I.; Barreda, M.; Dolz, M.F.; Martín, A.F.; Mayo, R.; Quintana-Ortí, E.S.: "Assessing the impact of the CPU power-saving modes on the task-parallel solution of sparse linear systems". *Cluster Computing*, 17, pp. 1335-1348, 2014. 7. Barik, R.; Farooqui, N.; Lewis, B.T.; Hu, C.; Shpeisman, T.: "A black-box approach to energy-aware scheduling on integrated CPU-GPU systems". *CGO'15*, March 12-14, Barcelona, Spain, pp.70-81, 2016.
- [4]. Lee, Y.C.; Zomaya, A.Y.: "Energyconious scheduling for distributed computing systems under different operationg conditions". *IEEE Trans. On Parallel and Distributed Systems*, Vol.22, No.8, pp.1374-1381. August, 2011.
- [5]. Manakkadu, Sheheeda, Sourav Dutta, and Nazeih M. Botros. "Power aware parallel computing on asymmetric multiprocessor." In *2014 27th IEEE International System-on-Chip Conference (SOCC)*, pp. 35-40. IEEE, 2014.
- [6]. Karthikeyan, K., R. Sunder, K. Shankar, S. K. Lakshmanaprabu, V. Vijayakumar, Mohamed Elhoseny, and GunasekaranManogaran. "Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC-BA)." *The Journal of Supercomputing* 76, no. 5 (2020): 3374-3390.
- [7]. Zhang, Xiangdong, GunasekaranManogaran, and BalaAnand Muthu. "IoT enabled integrated system for green energy into smart cities." *Sustainable Energy Technologies and Assessments* 46 (2021): 101208.
- [8]. Mariam Manakkadu, Sheheeda. "POWER-AWARE PERFORMANCE OPTIMIZATION ON MULTICORE ARCHITECTURES."
- [9]. Lin, Xiaochan, Xu Sun, GunasekaranManogaran, and Bharat S. Rawal. "Advanced energy consumption system for smart farm based on reactive energy utilization technologies." *Environmental Impact Assessment Review* 86 (2021): 106496.
- [10]. Shin, Youngsoo, and Kiyoun Choi. "Power conscious fixed priority scheduling for hard real-time systems." In *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*, pp. 134-139. IEEE, 1999.
- [11]. Shin, Youngsoo, Kiyoun Choi, and Takayasu Sakurai. "Power optimization of real-time embedded systems on variable speed processors." In *IEEE/ACM International Conference on Computer Aided Design. ICCAD-2000. IEEE/ACM Digest of Technical Papers (Cat. No. 00CH37140)*, pp. 365-368. IEEE, 2000.
- [12]. Shin, Youngsoo, Jun Seomun, Kyu-Myung Choi, and Takayasu Sakurai. "Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 15, no. 4 (2010): 1-37.
- [13]. Dorrnsoro, B.; Nesmachnow, S.; Taheri, J.; Zomaya, A.Y.; Talbi, E-G; Bouvry, P.: "A hierarchical approach for energy-efficient scheduling of large workloads in multicore distributed systems". *Sustainable Computing: Informatics and Systems*, 4, pp.252-261, 2014.
- [14]. Rotem, E.; Weiser, U.C.; Mendelson, A.; Ginosar, R.; Weissmann, E.; Aizik, Y.: "HEARtH: Heterogeneous multicore platform energy management". *IEEE Computer magazine*, pp.47-55. October, 2016.
- [15]. Dutta, Sourav, SheheedaManakkadu, and Dimitri Kagaris. "Classifying performance bottlenecks in multi-threaded applications." In *2014 IEEE 8th International Symposium on Embedded Multicore/ManycoreSoCs*, pp. 341-345. IEEE, 2014.
- [16]. Wang, Xiaodong, Yanjun Yao, Xiaorui Wang, Kefa Lu, and Qing Cao. "Carpo: Correlation-aware power optimization in data center networks." In *2012 Proceedings IEEE INFOCOM*, pp. 1125-1133. IEEE, 2012.

- [17]. Zheng, Kuangyu, Xiaodong Wang, Li Li, and Xiaorui Wang. "Joint power optimization of data center network and servers with correlation analysis." In *IEEE INFOCOM 2014-IEEE conference on computer communications*, pp. 2598-2606. IEEE, 2014.
- [18]. Chipara, Octav, Zhimin He, Guoliang Xing, Qin Chen, Xiaorui Wang, Chenyang Lu, John Stankovic, and Tarek Abdelzaher. "Real-time power-aware routing in sensor networks." In *2006 14th IEEE International Workshop on Quality of Service*, pp. 83-92. IEEE, 2006.
- [19]. Lefurgy, Charles, Xiaorui Wang, and Malcolm Ware. "Power capping: a prelude to power shifting." *Cluster Computing* 11, no. 2 (2008): 183-195.
- [20]. Kandasamy, Nagarajan, Sherif Abdelwahed, and John P. Hayes. "Self-optimization in computer systems via on-line control: Application to power management." In *International Conference on Autonomic Computing, 2004. Proceedings.*, pp. 54-61. IEEE, 2004.
- [21]. Ma, Xin, and Rui-Lan Liu. "Reactive power optimization in power system based on improved niche genetic algorithm." In *2010 International Conference On Computer Design and Applications*, vol. 3, pp. V3-413. IEEE, 2010.