International Journal of Technology and Emerging Sciences (IJTES)

www.mapscipub.com

Volume 05 || Issue 02 || July 2025 || pp. 01-03 E-ISSN: 2583-1925

Practicing of Green Coding: A Sustainable Software Development Approach

Ms. Jasmeen Kaur

Assistant Professor, Dept. of Computer Science & Engineering, PTU, Punjab

---

Abstract - *With the growing environmental impact of computing, adopting more sustainable practices—referred to as green computing—has become increasingly essential. Green coding principles encompass intelligent workload orchestration, aiming to minimize embedded carbon by optimizing workload management and decreasing reliance on new hardware. A key challenge in sustainable coding involves addressing poor coding standards and a lack of knowledge in green software development. The literature survey explores energy consumption in different languages and adopt a two layered architecture for sustainable software development.*

Key Words: Green Coding , Green Programming Languages, Carbon footprints, Sustainable Software Architecture ,Energy Consumption

1. INTRODUCTION

When most people think about sustainability, they consider conserving energy or recycling. But what about sustainability in the digital world? Green software is a relatively fresh perspective that refers to software practices and architecture, using software to reduce power consumption, use hardware components built to lower emissions, data center design, and climate science intersect to try and make software engineering more sustainable. The Principles of Green Software Engineering are a core set of competencies needed to define, build and run green sustainable software applications. [1]

2. GREEN CODING PRINCIPLES

Carbon Efficiency: Build applications that are carbon efficient.

Electricity Efficiency: Build applications that are energy efficient.

Carbon Awareness: Consume electricity with the lowest carbon intensity.

Hardware Efficiency: Build applications that are hardware efficient.

Measurement: Improve sustainability through measurement.

Climate Commitments: Defining the exact mechanism of carbon reduction.

3. CARBON FOOTPRINTS

Recent research indicates that by the end of 2020, carbon emissions could increase by up to 20%. A significant contributor to this rise is the growing reliance on data centers that support cloud computing architectures. These data centers, along with the adoption of emerging technologies, are primarily responsible for the surge in carbon footprints. [2] As they process vast amounts of user-generated requests, they consume substantial energy most of which is derived from non renewable sources. This high energy consumption is a key driver of increased carbon emissions. The findings indicate that large language models (LLMs) currently lack the sustainability awareness necessary to help reduce carbon emissions during their use.[3] Considering the substantial carbon footprint involved in training these models, it would be advantageous to minimize emissions during their deployment.

4. GREEN DATA MINING AND INFORMATION RETRIEVAL

Green data mining and eco-friendly information retrieval focus on minimizing the environmental impact of data processing activities while still extracting valuable insights. [4] These practices involve implementing strategies for managing carbon emissions and integrating renewable energy sources into operational frameworks. Additionally, the responsible disposal of electronic waste generated through data mining processes is considered an essential aspect of sustainable information retrieval and analytics.

Now a days , Python has quickly become the go-to language for data analysis due to its versatility and ease of use but a paper cited by Electronic Trading Technology states that Python can use significantly more energy (e.g., 45 times as much as C++) to perform the same task.

5. CHOOSE THE RIGHT PROGRAMMING LANGUAGE

Another important factor influencing computing emissions is the choice of programming language. The energy efficiency of a language can vary based on its design and implementation. [5] For instance, certain languages are specifically built to support concurrent programming or parallelism, allowing multiple tasks to be executed at the same time. This capability helps lower energy consumption by making more efficient use of CPU resources.

International Journal of Technology and Emerging Sciences (IJTES)

www.mapscipub.com

Volume 05 || Issue 02 || July 2025 || pp. 01-03 E-ISSN: 2583-1925

Energy efficiency should be a key consideration during code development. This includes measuring the program's energy consumption, striking a balance between performance and energy savings, and selecting programming languages known for lower energy usage (such as Java, Go, C#, C++, etc.).[6]

6. Layered Model for Environmentally Conscious Software Development

Mahmoud and Ahmad proposed a two-level model which aimed at promoting sustainability throughout the software development life cycle [7]. The first level outlines guidelines for creating green software by integrating sequential, iterative, and agile development approaches. This level encompasses several stages:

Requirements gathering, where developers assess the necessity and potential energy efficiency of proposed systems;

Design and implementation, which prioritize the use of efficient data structures, algorithms, and system architectures;

Testing, which not only verifies functionality but also evaluates environmental impact and fault tolerance;

Green analysis, focusing on metrics such as CPU usage to measure energy consumption;

Usage, emphasizing energy-efficient resource management influenced by user interactions;

Maintenance, which supports sustainability as new features are introduced;

Disposal, encouraging environmentally responsible recycling of outdated software and hardware.

The second level focuses on software tools that improve energy efficiency, including operating system frameworks, fine-grained computing techniques, performance metrics, energy allocation algorithms, and virtualization strategies. These tools enable software developers to optimize energy consumption, thereby promoting environmental sustainability.

This comprehensive approach ensures that sustainability is embedded at every stage of the software life cycle.

7. Energy Efficiency of Algorithms in Different Languages

Research findings show that an algorithm's energy consumption increases in line with its algorithmic complexity. This means that as the input size grows, energy usage scales similarly. For example, if an algorithm has a time complexity of $O(n^2)$, doubling the input size will typically result in four times the energy consumption.

Different programming languages exhibit varying levels of energy consumption, influenced by factors such as abstraction layers, compiler optimizations, and implementation details.[8] From the result collected we found that compiled and semi-compiled languages such as C++ and Java are generally more energy-efficient than interpreted languages like Python, R, and MATLAB, with energy consumption in interpreted languages reaching up to 54 times higher in some cases and secondly the energy efficiency of programming languages can differ significantly depending on whether the task involves training or inference .As a result, selecting the appropriate programming language is crucial for energy-efficient development.[9]

8. Developer Awareness and Workplace Prioritization of Sustainable Coding Practices

The survey results indicate that the terms "green code" and "green coding" are not widely recognized, with only about 30% of respondents familiar with them. Most professional respondents reported that their workplaces do not actively discuss environmentally friendly software development. However, a significant number expressed a desire for sustainability to be given more attention. Just over one-third of participants said they consider energy efficiency when writing code, while slightly less than one-quarter take energy consumption into account when selecting tools and techniques for their projects. [10]

9. Conclusion

This research serves as a valuable reference for developers seeking to implement more sustainable coding practices. By highlighting the environmental advantages of optimized code, the study promotes a shift toward greater sustainability within the software industry. Creating energy-efficient software is a complex task that demands a mindset shift from both developers and designers. Advancing sustainability requires coordinated efforts at multiple levels. Developers can help lower carbon emissions by following best practices and understanding the environmental impact of their decisions. At the organizational level, promoting sustainability involves implementing a green coding framework and assessing performance not only by traditional metrics but also by energy efficiency.

By adopting green software development practices, developers can play a key role in reducing the carbon footprint of digital solutions through improved energy use, efficient resource management, sustainable coding methods, and user awareness. Every contribution matters in this collective mission—even a single optimization can have a meaningful environmental impact.

Disclosure: I certify that all content, data, and organizational references in this manuscript have been included with proper authorization and consent. I accept full legal and ethical responsibility for the accuracy, originality, and integrity of the work. The journal, publisher, and editorial board assume no liability for any disputes arising from the submitted material.

References

[1]. Green Software Engineering Full Guide, Feb-2023, Fuller Brand Communication.

[2]. S Thakur and A Chaurasia. Towards Green Cloud Computing: Impact of carbon footprint on environment, 2016, IEEE explore.

[3]. Vartziotis, Tina et al., 2024, Learn to Code Sustainably: An Empirical Study on Green Code Generation. Cornell University. https://doi.org/10.48550/arXiv.2403.03344
[4]. S. G. Paul et al., "A Comprehensive Review of Green Computing: Past, Present, and Future Research," in IEEE Access, vol. 11, pp. 87445-87494, 2023, doi: 10.1109/ACCESS.2023.3304332.
[5]. Mahmudova SH. (2024). Exploring international practices in the field of green coding. Danish scientific journal, 87. https://doi.org/10.5281/zenodo.13620831
[6]. Cassel, D. Which programming languages use the least Electricity?
[7]. Sara S.Mahmoud and Imtiaz Ahmad, A Green Model For Sustainable Software Engineering .2013 International Journal of Software Engineering and Its Applications Vol.7, No.4, pp.55-74.
[8]. T. Khudher, 'Green Coding : Energy consumption in different programming languages and algorithms', Dissertation, 2024.
[9]. Niccol` o Marini at el .Green AI: Which Programming Language Consumes the Most?, 2025. https://doi.org/10.48550/arXiv.2501.14776
[10]. H Ritchie, M Roser, Overview of Global Energy, Accessed 2022-01-20