Bober D.O., student of Kyiv National University
of Technologies and Design
(Language and scientific supervisor – Vyshnevska M.O.,
associate professor of Kyiv National University
of Technologies and Design)

THE USE OF COMPILED PROGRAMMING LANGUAGES FOR INCREASING PROFITS

The escalating concern surrounding the energy consumption of software applications exhibits a growing trend on an annual basis. This concern assumes even greater significance in the context of mobile and embedded devices, where the limited battery life imposes substantial constraints. However, it is imperative to underscore that the implications of elevated energy consumption extend beyond the mere requirement for increased recharging cycles. They also encompass a commensurate upsurge in electricity expenditures. Furthermore, it is crucial to acknowledge the finite nature of electricity as a resource, suggesting the potential for its depletion over time. Hence, the overarching reduction in energy consumption holds promise for long-term benefits across a broader spectrum. The pivotal inquiry pertains to the determinants governing the quantum of energy consumed by an application.

Numerous factors contribute to the calculation of an application's energy consumption. Nevertheless, this study restricts its focus to a specific determinant: the programming language employed in the application's development. Diverse programming languages exist, characterized by their general-purpose or specialized nature, their alignment with declarative or imperative paradigms, and, notably pertinent to this investigation, their classification as either interpreted or compiled languages. Interpreted languages, by their very nature, necessitate an intermediary translation process, as they are not amenable to direct comprehension by the computer. Consequently, this translation process becomes a recurring necessity each time an application is initiated, thereby imposing an incremental computational burden. It is pertinent to highlight that interpreters often operate without foreknowledge of the precise data with which they will engage, necessitating preparatory measures to accommodate a wide range of potential scenarios proactively.

Interpreted languages frequently manifest less verbose syntax, facilitating rapid code development and expediting idea iteration for developers. However, this very attribute renders it challenging for the interpreter to predict and prepare for the actions executed by the code. Compiled languages, in contrast, pursue different objectives. Rather than placing the interpretive burden on the machine executing the application, compiled languages eliminate this intermediary step, translating the code into machine language before it leaves the developer's environment. This approach obviates the overhead that an interpreter typically imposes. Furthermore, compiled languages optimize the code during translation, thereby reducing the computational workload of the machine executing the application. However, it is imperative to recognize that the code authored in compiled languages tends to be more verbose and syntax-intensive.

Compiled languages necessitate prior knowledge of the data types to provide optimization, and while certain modern compiled languages can autonomously infer data types to some extent, such inference is not infallible. In such instances, developers are required to manually specify data types, which can render code composition and refactoring more arduous. Nevertheless, applications written in compiled languages consistently exhibit significantly higher performance compared to those in interpreted languages. The performance differential may range from a fewfold to several orders of magnitude, where compiled code demonstrates superior execution speed.

It is imperative to acknowledge that the emphasis in this discourse has centered on speed, even though the primary focus of this paper remains energy consumption. This emphasis arises from the proven correlation between code execution speed and energy efficiency, where faster code generally entails lower energy consumption. However, it is notable that interpreted languages currently enjoy greater popularity, leading to a proliferation of power-intensive applications in the market. As previously emphasized, electricity is a finite and costly resource. While the pursuit of profits is evident, it is plausible to achieve higher profits with more energy-efficient practices. However, this does not necessarily entail an immediate and wholesale transition to compiled languages, given the inherent challenges and potential developmental barriers.

A pragmatic proposal emerges from this analysis: gradual transition. Particularly in cases where an application has achieved a substantial degree of completion, the phased substitution of components with alternative programming languages holds the potential to significantly ameliorate energy consumption, particularly in scenarios characterized by computationally intensive workloads. This approach can be implemented with minimal disruption to the development process, requiring a measured adjustment on the part of developers.

References:

1. Georgiou S., Kechagia M., & Spinellis D. Analyzing programming languages' energy consumption: An empirical study. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics.* 2017. P. 1–6.

2. Keijzers S., van Eekelen M. C., & van Gastel B. Energy Consumption Analysis of Practical Programming Languages. *Radboud University Nijmegen, Master Thesis*. 2014.

3. Pereira R., Couto M., Ribeiro F., Rua R., Cunha J., Fernandes J. P., & Saraiva J. Energy efficiency across programming languages: how do energy, time, and memory relate? In *Proceedings of the 10th ACM SIGPLAN international conference on software language engineering.* 2017. P. 256–267.

4. Pereira R., Couto M., Ribeiro F., Rua R., Cunha J., Fernandes J. P., & Saraiva J. Ranking programming languages by energy efficiency. *Science of Computer Programming.* 2021. 205, 102609.

Danylov A.D., student of Kyiv National University
of Technologies and Design
(Language and scientific supervisor – Vyshnevska M.O.,
associate professor of Kyiv National University
of Technologies and Design)

ETHICAL PROBLEMS OF USING ARTIFICIAL INTELLIGENCE AND WAYS TO SOLVE THEM

In recent years, artificial intelligence has become an integral part of our lives. Some people use it for entertainment, while others outsource some of their work to it. But as AI has become increasingly popular, it has become clear that this technology has certain ethical issues. They arise from the fact that AI is not able to independently evaluate the data it operates with. There are no moral or legal constraints that prevent it from doing so. Identifying and overcoming such ethical issues will make it possible to introduce AI-based technologies in various fields without the threat of violating legal or ethical norms.

One of the main ethical issues is data privacy. AI has the potential to significantly interfere with people's privacy due to its ability to process and analyse large amounts of personal data. This can lead to violations of privacy and confidentiality rights. People's data could end up in the hands of criminals, which would cause very serious harm to users. Or this data can be obtained by advertising agencies, even without the users' knowledge, without breaking any law.

Another issue is bias and discrimination. AI can learn from existing data that may contain biases. For example, if AI is used to hire staff and is trained on existing hiring data that has a gender or racial bias, the AI may continue this practice.

There is also the issue of AI using copyrighted data. When training large AI models, such as GPT-4 or DALL-E 3, large amounts of data are used that are not checked for possible copyright protection. This leads to the fact that people get free access to use such data due to the lack of laws for this case.

Another important issue is determining liability for the harm that AI can cause. For example, if a person is injured while using AI to perform a complex medical operation, someone must be held accountable for this. But it will be almost impossible to identify those responsible in this case at the moment. Likewise, if someone uses AI to break the law, it will be difficult to determine who should be punished: only the person who broke the law or also the company that developed the AI.

Another important issue is the replacement of humans with AI. It can be much cheaper for large companies to develop and train one AI-powered system than to hire a whole staff. This will lead to a sharp increase in unemployment and poverty. AI will take over almost all positions, and in the meantime, people will not be able to get a job anywhere.