

Coding green: Software engineers strive for sustainable and robust solutions

April 20 2022



Credit: Pexels/CC0 Public Domain

Climate change has forced humankind to think differently in order to preserve the planet and every living being. A wide variety of technologies and services that preserve natural resources have emerged

worldwide, as a result of research and innovation efforts. We now have zero-emission vehicles, smart and more efficient appliances, green packaging made from biodegradable plastic, and even lab-grown meat. How about software? Can an app be green too?

Apps—and anything else that is programmed—can actually be more or less efficient and require more or less power to perform their tasks. And if an inefficient app does not shock you, imagine if all the Google servers were consuming twice the power they do today, because of poor optimization. It's all about scale.

A group of researchers from the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC) and the University of Minho focuses on that exact point: making sure the software on everything that surrounds us is coded as efficiently as possible. "We want to raise awareness and help programmers build more sustainable and robust solutions," stated João Saraiva, a researcher at INESC TEC and professor at the University of Minho.

Writing more efficiently

The group has released a [scientific publication](#) where they compared the efficiency of different Android keyboard applications, such as Google Keyboard (GBoard) and Microsoft SwiftKey. Because smartphones run on batteries, the whole system must be as efficient as possible to extend [battery life](#) to the max. "In fact, replacing the most energy-greedy keyboard with the greenest one has reduced [energy consumption](#) by 18%, and when advanced features of those keyboards (word prediction or animations) were turned off, the energy consumption was also reduced, in this case, up to 9.3%", wrote the researchers Rui Rua, Tiago Fraga, Marco Couto, and João Saraiva.

What if we consider the smartphone as a whole, instead of one specific

app? With that in mind, the team has also developed GreenHub, an Android application that is able to measure the power consumption of smartphones. The data collected from the application allowed the analysis of the battery consumption of 23 million samples, spread across more than 1,600 device brands, 11,800 smartphone models, and more than 50 Android versions, which resulted in this [publication](#).

The authors were able to find charge/discharge tendencies across different countries, observable battery tendencies across brands and models, and battery usage improvements between Android versions. They have also studied how some of the most popular applications such as Facebook, Facebook Messenger, and Facebook Lite (now called Meta) behave in terms of battery consumption tendencies.

In addition to studies focused on analyzing energy consumption, the group is also dedicated to developing tools and methodologies to estimate software energy consumption. "The development of such artifacts is motivated by the need to provide tools for developers and researchers to detect [source code](#) critical energy hotspots," explained researcher Rui Rua. The latest artifact developed within this scope is a tool called E-MANFA, which allows monitoring and estimating the energy consumption of software on Android devices.

Programming more efficiently

Programming languages are the backbones of mobile applications, websites and so much more. And there are currently hundreds of programming languages available that solve different problems programmers need to address. Python, for example, is often used in Machine Learning and Artificial Intelligence, and JavaScript for Web development, just to name a few. The group of researchers has placed [programming languages](#) under the microscope to find the ones that require more energy to perform similar tasks. The result is a [scientific](#)

[publication](#) featuring the rank of 27 languages from power-saver to energy-drainer. "Developers can use this information to decide what programming language is the most appropriate for their target scenario, which can vary according to several constraints, such as limits on battery, time, and memory," said João Saraiva.

Google's energy consumption has increased over the last few years, reaching 15.4 terawatt hours in 2020, which would be enough to power 9.6 million European households in 2019, and it is very likely to keep on increasing, not only in Google's case, but in most industries. Considering that the world is using more and more power, it is of the highest importance to optimize energy. Although it might not be visible to the end-users, power efficiency in software should be taken into account if the world aims to address [climate change](#), as 80% of the world's produced energy still originates from fossil fuels.

Automatic debugging

Coding will inevitably result in bugs, and the act of debugging is, by itself, a way to improve the energy efficiency of software. The group is also focused on Software Fault Localization (SFL), one of the most expensive and time-consuming activities in program debugging. "We have developed a technique that provides semantical information about the modifications that led to a software bug. The tool that implements this technique is publicly available and it is called [Morpheus](#)," said Francisco Ribeiro, researcher at INESC TEC. In this publication, the authors showed that automated repair techniques can benefit from this new information and fix programs more effectively.

But isn't there a way to debug automatically? That is what Automated Program Repair (APR) is all about: fixing faulty programs with as little human intervention as possible. In fact, companies like Meta clearly acknowledge such benefits and actively conduct research in this area.

Regarding APR, the Portuguese group of researchers obtained a repair rate of 27% in their studies with more than 6,000 faulty programs.

The energy crisis has begun, and climate change is right around the corner. In an ever-growing digital society, being digitally efficient is a must. Whatever the method is, it is good to know software engineers are doing their part.

More information: Rui Rua et al, Greenspecting Android virtual keyboards, *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems* (2020). [DOI: 10.1145/3387905.3388600](https://doi.org/10.1145/3387905.3388600)

Rui Pereira et al, GreenHub: a large-scale collaborative dataset to battery consumption analysis of android devices, *Empirical Software Engineering* (2021). [DOI: 10.1007/s10664-020-09925-5](https://doi.org/10.1007/s10664-020-09925-5)

Francisco Ribeiro et al, On Understanding Contextual Changes of Failures, *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)* (2022). [DOI: 10.1109/QRS54544.2021.00112](https://doi.org/10.1109/QRS54544.2021.00112)

Provided by INESC Brussels HUB

Citation: Coding green: Software engineers strive for sustainable and robust solutions (2022, April 20) retrieved 1 June 2023 from <https://techxplore.com/news/2022-04-coding-green-software-sustainable-robust.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.