Impact Objectives

- Help software developers to analyse, understand and optimise the energy consumption of their software
- Define an engineering discipline to support green decision making while developing software

Going green with software

If not compiled properly, software can be energy-hungry. ***Dr João Saraiva*** *explains how his project aims to provide code writers with tools to ensure their software is as energy efficient as possible*

What are the primary aims and objectives of the GreenSoftwareLab project?

In this project, we aim to study, develop and apply methods to analyse abnormal energy consumption in software source code. The focus of the project is to examine and consider the energy consumption at the software level. To achieve this, we will define energy leaks as an abnormal and excessive consumption of energy by a software system. By locating these energy leaks in the developer's code, we will be able to construct both a catalogue of software energy metrics and a catalogue of red smells (energy inefficient practices in source code), and provide a set of source code refactoring and supporting tools that help developers in green decision making and in optimising the code.

Who do you hope will benefit from the research, and how?

Programming languages and their ecosystems provide powerful abstractions and tools for programmers to develop their software such as integrated programming environments (IDE), program debuggers, program profilers, catalogues of code smells and associated program refactoring. All these systems aim at helping programmers consider the software systems' source code and, as a consequence, to be more productive. Unfortunately, several studies show that there are no similar tools to examine the energy (in)efficiency of software systems. In fact, a recent paper showed that software developers are actively discussing energy efficiency of the programs in software developer's forums, and they are asking for help regarding how/when/where to make their programs more energy efficient. Thus, software developers concerned with energy efficiency will benefit from the techniques and tools developed in this project. Moreover, studies show that a large part of the energy costs of an organisation can be attributed to the IT departments. Other studies have shown that most of the energy consumed by the hardware can be attributed to software. In fact, a large part of the energy consumed by a company can be attributed to software. By incorporating green decision making mechanisms in the languages' ecosystems, our project will help programmers and organisations produce greener software, and contribute to both a more sustainable environment and reducing their costs.

Can you tell us about the partners in this project and the importance of their contributions?

We are working in close collaboration with several partners in green software. Funded by a Luso-American Foundation/National Science Foundation (FLAD/NSF) grant, we have a project with the University of California, Irvine in the USA, where we are developing a large repository of green software applications. The idea is to have a large corpus of applications that researchers in green computing can use to validate their results. Also, we have a bilateral project running with Kosice University in Slovakia, where we are identifying software patterns that contribute to abnormal energy efficiency. These collaborations with our partners are producing several results that we are convinced will contribute towards helping software developers in green decision making, and thus for greener software systems.

How will the results of the project be communicated?

The green software laboratory is a research project whose main goal is to define techniques and tools that will contribute to make greener and more sustainable software. The main results of the projects will be PhD thesis and research papers reporting our research findings, which we are publishing in well-established conferences and journals. We are also involved in organising the 5th International Workshop on Green and Sustainable Software (GREENS) in May 2018, which is a forum where researchers in green software meet and discuss their research results and directions for future research. People can learn more about this forum at http://greens.cs.vu.nl/.

The future for computing is green

*The **GreenSoftwareLab** project is seeking to redefine the computing discipline by pioneering efforts to improve the software energy efficiency at the programming stage*

Several studies have shown that, on average, almost half an organisation's energy costs can be attributed to the Information Technology (IT) department. Armed with this information, many organisations have begun to focus on the energy efficiency of their IT departments. In the USA, the Environmental Protection Agency (EPA) has started to identify ways to measure, analyse, and implement energy efficient alternatives. Google have begun investing into this domain, and have frequently discussed the topic on their official blog.

The GreenSoftwareLab project was launched to support these efforts. 'The goal of green software engineering is to apply green principles to the design and operation of software-intensive systems,' explains Dr João Saraiva, the project coordinator. 'We aim to provide theories, techniques and tools to help software engineers make decisions about the energy consumption of their software, similar to how languages and their compilers provide language abstractions and tools such as debuggers, profiler and testing frameworks to make both developers more productive and programs more time efficient.'

CHANGING SOFTWARE NEEDS

The problem has been exasperated by the widespread use of non-wired devices. The advent of the Internet of Things (IoT) means that many consumer electronics are now powerful computing devices. This is changing the way many software engineers work. Software must run in a variety of devices and energy consumption is becoming an important bottleneck in terms of software performance.

The widespread use of different computer devices, from consumer electronics, to smartphones, to desk computers, is also influencing how software is developed. In fact, software engineers must structure their programs so that they can easily be adapted or ported into different computer architectures. 'These software programs are structured in a Software Product Line (SPL), and we have developed a novel, static analysis technique to compare energy use in a set of similar software products that share common computing features,' Saraiva adds. 'For example, we can estimate the energy consumed by the different software products/programs in a line, and then generate the most efficient one for a particular architecture that contains a specific set of features.'

ABNORMAL ENERGY USE

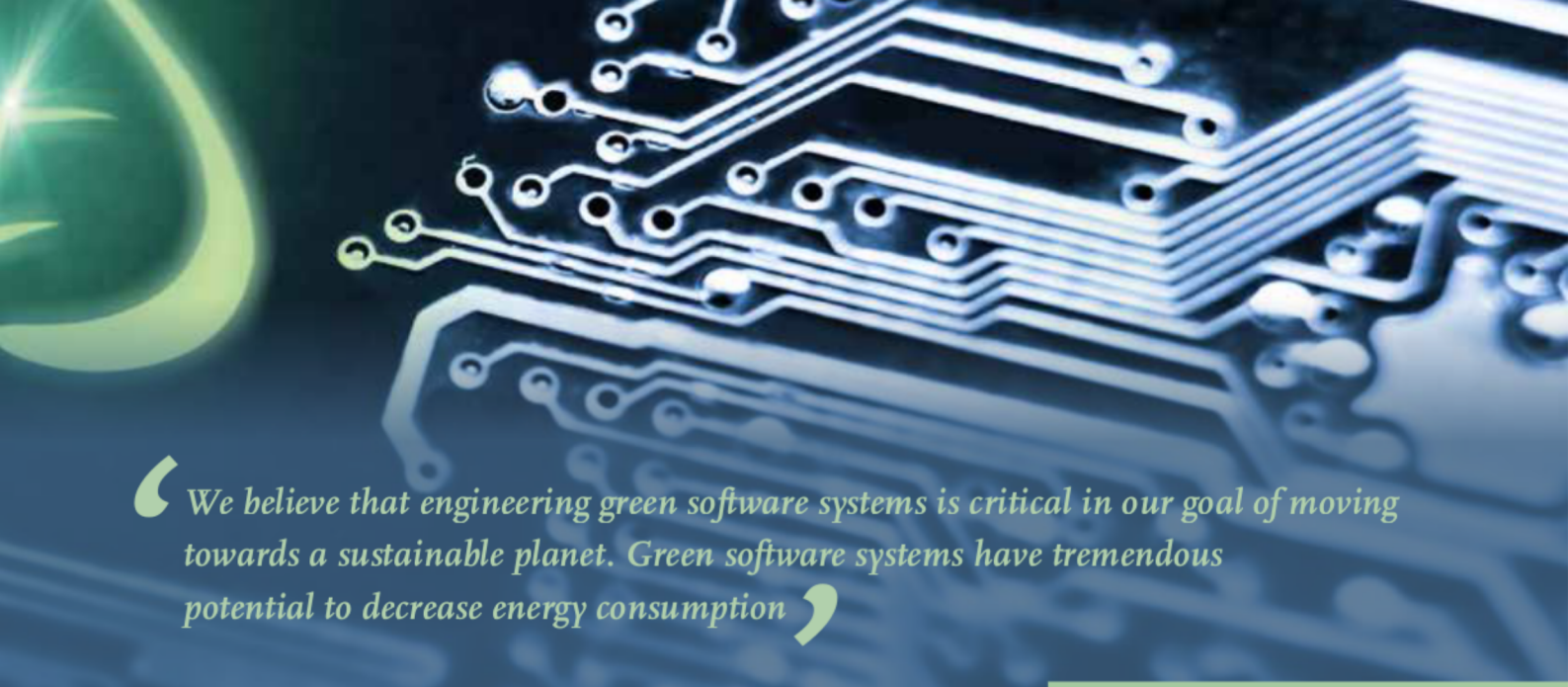
Monitoring, measuring, analysing and optimising the energy consumption of software is critical for sustainable and green software. Although energy and runtime are related, and many programmers understand that making a program faster also makes it greener, there is another variable in this equation: power. 'There are several studies that show that slower programs can be greener,' Saraiva elucidates. 'Although well-known and widely used compiler optimisation may also have a positive impact in the energy consumed by software, the programming language community need to define energy specific analysis and optimisation techniques to help programmers locate abnormal energy consumption in their software systems and to perform semi-automatic optimisations to reduce energy in software, and not only to optimise the runtime and memory consumed as language ecosystems offer today.'

In the GreenSoftwareLab project, the aim is to develop techniques and tools that can help in this by enabling programmers to understand the energy consumed by their software and help them make more sustainable decisions. 'We are developing tools to locate energy leaks in the source code so that programmers can easily find energy inefficiencies in the programs being developed,' Saraiva adds. 'We are also developing techniques to refactor the software's source code to make it more efficient.'

LANGUAGE INDEPENDENT

Programmers use an array of different programming languages, each of which is associated with differing degrees of energy consumption. As such, it is important that any tools and techniques produced by the GreenSoftwareLab are as generic as possible, to ensure they are applicable in any context of software, independent of the programming language being used.

One way this is being achieved is through the use of Spectrum-based Energy Leak Localisation (SPELL), a language independent analysis technique that can detect energy inefficient fragments in the source code of a software system using various program simulations. Based on fault localisation principles, the technique associates different percentages of responsibility of energy consumption to different source code components. It can therefore point the developer to the most critical hot spots on a source code level and allow him to optimise the energy efficiency of their program. 'Ultimately, this has been shown to help developers become energy-aware when developing, while also helping analyse, interpret and optimise the energy consumption of a program. Our initial results have been very promising and have shown that developers using our technique were



"We believe that engineering green software systems is critical in our goal of moving towards a sustainable planet. Green software systems have tremendous potential to decrease energy consumption"

able to improve the energy efficiency by 43 per cent on average, and even outperform a profiler for energy optimisation,' Saraiva highlights. 'Another great thing about the technique is that it allows both an analysis based on one criterion, which is useful if the developer is focused on one specific optimisation such as time or energy, or a global analysis where SPELL points the developer to the most critical zones based on multiple criteria. During our studies using SPELL, we found interesting coding practices which are energy inefficient, and in turn were able to define energy efficient alternatives.'

A BRIGHTER, GREENER FUTURE

The GreenSoftwareLab project began in July last year and will run until July 2019. In the short time it has been in operation, it has already achieved several notable successes. For example, it has already established a novel methodology to statically calculate the greenness of software product lines, where it can estimate the energy consumed by each individual product and produce the greenest one with a specific set of features. It has also developed a green ranking for programming languages, in which the researchers considered 22 of the most popular languages and defined a multi-objective ranking according to their runtime, memory and energy consumption when implementing the same computer programs. This will enable programmers to make informed decisions about which language and data structure to use to make their software more energy efficient. The team also carried out a detailed study of greenness of language data structures. 'We studied the greenness of the full API of Java collection framework (JCF) and the Haskell advanced data structures, and we proposed a simple refactoring technique to optimise the energy consumption of Java/Haskell applications,' Saraiva elaborates. Gathering and analysing data from the crowd has also been considered by the project, particularly in trying to improve the battery uptime of Android devices and in the context of the GreenHub initiative (http://greenhubproject.org/). For this, a technological infrastructure has been developed. This is to periodically collect data that may relate to battery state and battery usage. 'The data is collected upon the installation of an app that we have developed, and is sent to a centralised data repository of our own', explains Saraiva. Also the team is sharing the data they have collected in their repository.

'In order to make such data accessible, we have implemented a REST API, which can be invoked a number of ways, including inside a Web App or using a command line interface', he continues.

Ultimately, the GreenSoftwareLab researchers are keen to carve out a new engineering discipline that is dedicated to supporting a greener, more sustainable approach to software development. 'We believe that engineering green software systems is critical in our goal of moving towards a sustainable planet,' Saraiva enthuses. 'Green software systems have tremendous potential to decrease energy consumption. Although it is the hardware that consumes energy, the software that operates this hardware has a main role in controlling the energy consumed, very similar to how a driver influences the fuel consumed by a car. That means that the software should be rethought to consider sustainability issues using innovative business models, processes, and incentives.' In light of the progress already made within the project, it appears the researchers are well on their way to making this vision a reality.

Project Insights

FUNDING

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project POCI-01-0145-FEDER-016718

PARTNER

Rui Abreu, IST, University of Lisbon and INESC TEC • Jácome Cunha, NOVA LINCS, NOVA University of Lisbon • João Paulo Fernandes, CISUC, University of Coimbra • João Saraiva, HASLab/INESC TEC, University of Minho

CONTACT

João Saraiva
Project Coordinator

T: +351 253604470 (ext. 4433)
E: saraiva@di.uminho.pt
W: http://greenlab.di.uminho.pt/