Green Software Lab*

Orlando Belo[1,3], Marco Couto[2,3], Jácome Cunha[2,4], João Paulo Fernandes[5], Miguel Guimarães[2,3], Rui Pereira[2,3], and João Saraiva[2,3]

[1]ALGORITMI, [2]HASLab / INESC TEC
[3]Universidade do Minho, [4]Universidade Nova de Lisboa
[5]RELEASE, Universidade da Beira Interior
{obelo,mcouto,mguimaraes,ruipereira,jas}@di.uminho.pt
jacome@fct.unl.pt, jpf@di.ubi.pt

Abstract. This document describes the green software laboratory where we develop techniques and tools to help software engineers in green decision making. This laboratory aims at providing a software engineering discipline for energy-aware software. Thus, it both adapts and develop new engineering techniques to help developers to monitor, to analyze and to optimize their software in terms of energy consumption. This short papers briefly describes such techniques and tools.

1 Introduction

The current widespread use of non-wired but powerful computing devices, such as, for example, smartphones, laptops, etc., is changing the way both computer manufacturers and software engineers develop their products. In fact, computer/software performance (ie, execution time), is no longer the only and main concern. Energy consumption is becoming an increasing bottleneck for both hardware and software systems. In average, close to 50% of the energy costs of an organization can be attributed to the IT departments [1].

Step-by-step, some businesses have begun to promote "green" initiatives, in hopes of reducing the emissions and energy costs. For example, looking at the data center sector, Symantec Corporation decided to reduce CO_2 emissions by 15% by 2012. To accomplish it, they decided to consolidate a data center, and after finding out 60% of end user's computers were left powered on overnight, they decided to place users' computers in stand-by mode after four hours of inactivity [2]. These steps helped reduce approximately $2 million and over 6 million kilowatts of energy.

Sometimes businesses can not just physically reduce consumption as Symantec did. Recently, hardware manufacturers and researchers have developed techniques to reduce energy consumption mainly focused on developing and optimizing their hardware. However, very much like how a driver operating a car

* This work is partly funded by the Innovation Agency, SA, Northern Regional Operational Programme, Financial Incentive Grant Agreement under the Incentive Research and Development System, Project No. 38973.

can heavily influence its fuel consumption, the software which operates such hardware can drastically influence its energy consumption too! This results in energy-efficiency in the hardware level to be canceled out by inefficient energy consumption of software, where "Up to 90% of energy used by a computer can be attributed to software" (the other 10% is pure hardware energy usage) [3].

Recent research in software engineering has defined powerful techniques to improve software developers productivity. Providing advanced type and modular systems, powerful query mechanisms for Data Base Systems (DBS), and integrated development environments (IDE) are some examples of that, as well as compiler construction techniques to improve execution time of software.

Unfortunately, none of these techniques nor tools have been adapted to support greenware software development. Indeed, there is no software engineering discipline providing techniques and tools to help software developers to analyze, understand, query nor optimize the energy consumption of their software! As a consequence, if a developer notices that their software is responsible for a large battery drain, they get no support from the environment being used.

This short paper describes the Green Software Laboratory (GSL) where techniques and tools are being developed to offer software developers mechanisms to reason about their software in terms of energy consumption as they are already able to reason in terms of execution time and memory consumption. Thus, GSL reuses, adapts and develops techniques to monitor, analyze and optimize the energy consumption of software systems. Next section presents GSL and briefly describes the techniques used to analyze/develop energy-aware software. After that we present the tools that implement such techniques and support software developers in green decision making.

2 The Green Software Lab

The Green Software Lab is a team consisting of various Postdoctoral and PhD researchers spanning several universities and research centers throughout Portugal. We specifically focus on the software, where our mission is to apply source code analysis techniques to detect anomalies in energy consumption and to define transformations to reduce such consumption. We aim to develop methods to analyze *energy leaks* in software source code. Thus, the focus of our lab is to reason about energy consumption at the software level. In this context, we define energy leaks as an excessive consumption of energy by a software system.

We have begun adapting well known techniques for fault localization and program debugging in software source code, to locate energy-leaks in software and to relate such leaks to the software source code. We have also begun analyzing database queries, and through an analysis of query execution plans, can estimate the expected energy consumption of queries.

Using these and other analysis techniques, we will identify what programming practices, design patterns, and other factors contribute to high energy consumption. Being able to locate such energy leaks in the developer's code, we

will construct both a catalog of software energy metrics and a catalog of red smells (i.e., energy inefficient smells in source code).

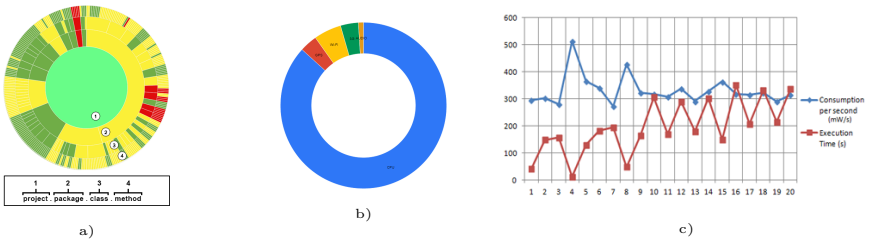
These techniques are the main building blocks for providing a set of source code refactorings and supporting tools that help developers in green decision making and in optimizing the code they write. A source code refactor is a source-to-source transformation that does not change the semantic behaviour of a program. Our energy-aware catalog of refactorings, named *green refactorings*, will be used to improve software energy consumption.

The team is developing libraries and tools to support green decision making. To analyze and locate energy leaks, an energy profiler, and other energy monitoring tools, will be developed. To optimize energy leaks, a framework implementing the red smells/green refactorings will be defined as IDE plugins. Such a framework will localize a red smell in the source code, while also providing the programmer information, and automatically optimize the energy usage. This will allow programmers to finally become energy-aware and have ways to support green decision making.

3 Tools

Green Droid [4] is an Android source code analyzer which relates anomalous energy consumption to source code in an application by analyzing different executions (tests). It uses an API adapted from an existing energy profiler for Android applications (Power Tutor [5]) to obtain the power consumption values.

The tool uses information about program trace, energy consumption and execution time to classify the application's methods accordingly, generating 3 graphs for the analysis: a sunburst diagram, a pie chart and a line chart comparing execution time with consumption per second. These graphs are shown in Figures 1.a, 1.b and 1.c, respectively.



a) b) c)

Fig. 1: Different types of graphs generated by Green Droid

SPELL - SPectrum-based Energy Leak Localization [6] is a language independent source-code analysis tool, based on a spectrum-based fault localization technique [7], a technique originally used for detecting bugs within programs, which detects energy leaks from various levels ranging from packages, functions, and source line level. Using the RAPL [8] framework to obtain accurate energy consumption values, and a series of tests cases, we can detect which sections of the

program (components) contain energy leaks. An abstract representation of this technique is shown in Figure 2.

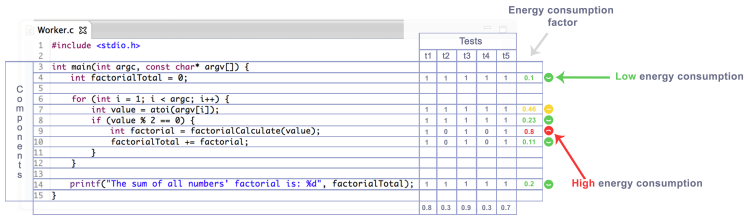


Fig. 2: Spectrum-based Fault Localization technique

gSQL [9] is a technique developed with the purpose of creating energy consumption plans for SQL queries. With the help of an external energy monitoring device, we were capable of measuring the energy consumption of query executions. Using these values, we were able to estimate the consumption costs of certain SQL operators, and have created an energy consumption model which is embedded within the PostgreSQL kernel.

References

1. Harmon, R.R., Auseklis, N.: Sustainable it services: Assessing the impact of green computing practices. In: Management of Engineering & Technology, 2009. PICMET 2009. Portland International Conference on, IEEE (2009) 1707–1717
2. Thompson, J.: Environmental progress and next steps. Email to Everyone Symantec (Employees) (2008)
3. Standard, R.: GHG Protocol Product Life Cycle Accounting and Reporting Standard ICT Sector Guidance. In: Greenhouse Gas Protocol. Number January. (2013)
4. Couto, M., Carçao, T., Cunha, J., Fernandes, J.P., Saraiva, J.: Detecting anomalous energy consumption in android applications. In: Programming Languages. Springer (2014) 77–91
5. Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R.P., Mao, Z.M., Yang, L.: Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: Proceedings of the eighth IEEE/ACM/IFIP international conference on hardware/software codesign and system synthesis, ACM (2010) 105–114
6. Carcao, T.: Measuring and visualizing energy consumption within software code. In: Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on, IEEE (2014) 181–182
7. Abreu, R., Zoeteweij, P., Van Gemund, A.J.: Spectrum-based multiple fault localization. In: Automated Software Engineering, 2009. ASE'09. 24th IEEE/ACM International Conference on, IEEE (2009) 88–99
8. Rotem, E., Naveh, A., Ananthakrishnan, A., Rajwan, D., Weissmann, E.: Power-management architecture of the intel microarchitecture code-named sandy bridge. IEEE Micro (2) (2012) 20–27
9. Goncalves, R., Saraiva, J., Belo, O.: Defining energy consumption plans for data querying processes. In: 2014 IEEE Fourth International Conference on Sustainable Computing and Communications, SustainCom 2014. (Dec 2014) 641–647