Energy Consumption Detection in LabVIEW

Tiago Carção[*], Jácome Cunha[*†], João Paulo Fernandes[*‡], Rui Pereira[*], and João Saraiva[*]

[*] HASLab/INESC TEC & Universidade do Minho, Portugal
[†] CIICESI, ESTGF, Instituto Politécnico do Porto, Portugal
[‡] RELEASE, Universidade da Beira Interior, Portugal
{tiagocarcao,jacome,jpaulo,ruipereira,jas}@di.uminho.pt

I. INTRODUCTION

For the past few years, we've begun to witness an exponential growth in the information and communication technologies (ICT) sector. While undoubtedly a milestone, all of this occurs at the expense of high energy costs needed to supply servers, data centers, and any use of computers [1]. Associated with these high energy costs is the emission of greenhouse gases. These two issues have become major problems in society [2]. The ICT sector contributes to 8% of the overall energy consumption [3], with 50% of the energy costs of an organization being attributed to the IT departments [4].

Since energy-efficient hardware is canceled out by inefficient software, where "Up to 90% of energy used by ICT hardware can be attributed to software" [5], this is where we should begin looking. This isn't shocking, since software development has always focused on response efficiency/functionality, not minimization of power consumption [6], making software design/construction energy-unaware.

We would like to present our work, where using advanced Spectrum-Fault Localization techniques, we detect energy-consumption faults (Red Smells) in software code. In doing so, we can detect which fragments of code consume the most/least, and using this information, offer ways to refactor source code to become green-aware.

II. ENERGY CONSUMPTION DETECTION

While analyzing source code, we can consider the excess energy consumption to be energy-leaks. Considering this, we apply a fault-localization techniques in source code to detect these energy-leaks.

This technique, Spectrum-based Fault Localization (SFL) [7], [8], is a statistical analysis based on the program's execution with the goal of locating program faults. We easily adapt SFL to localize energy-leaks in a program, giving us a thorough analysis of where excess energy is being consumed so one may be able to resolve this problem.

Currently, using Intel's Power Gadget framework[1], we obtain the energy-consumption for each function in a C-language program. Using an adapted SFL model for energy-consumption, we can easily detect where energy-leaks occur.

Using this information, we are beginning to construct a "Red Smell" catalog for software code, and the appropriate refactoring to turn the software more "Green-Aware".

---

[1]Intel's Power Gadget: http://software.intel.com/en-us/articles/intel-power-gadget-20

III. APPLICATION IN LABVIEW

As shown by Chambers and Scaffidi [9], alerting an end-user of a bad smell, and subsequently advising them how to fix those smells, increases success rates at finding and fixing those problems.

Since applying SFL for energy analysis is program language independent technique, we can exercise the same principal to identify "Red Smells" in the LabVIEW environment. Along with "Red Smell" detection, we can also define a green/red catalog and appropriate suggested refactor, to help end-users find and fix these energy-leaks. These changes will allow LabVIEW to be more energy-aware, reducing energy and monetary costs.

IV. INDUSTRIAL EXPERIENCE

Our research group has relevant experience applying research results in industry in the following funded projects:

- In the Spreadsheets as a Programming Paradigm (SSaaPP) project, we have applied model-driven engineering approaches in projects with *Bosch*, *Primavera Software*, and *National Food Bank*.

- In the context of green-computing, we are currently working in the GreenSSMC project, alongside *VisionSpace* where we use techniques to detect high energy consumption in the mission control software system of the European Space Agency (ESA).

REFERENCES

[1] T. Guelzim and M. S. Obaidat, "Chapter 8 - Green Computing and Communication Architecture," in *Handbook of Green Information and Communication Systems*, M. S. Obaidat, A. Anpalagan, and I. Woungang, Eds. Academic Press, 2013, pp. 209–227. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124158443000085

[2] S. Ricciardi, F. Palmieri, J. Torres-Viñals, B. D. Martino, G. Santos-Boada, and J. Solé-Pareta, "Chapter 10 - Green Data center Infrastructures in the Cloud Computing Era," in *Handbook of Green Information and Communication Systems*, M. S. Obaidat, A. Anpalagan, and I. Woungang, Eds. Academic Press, 2013, pp. 267–293. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124158443000103

[3] H. T. Mouftah and B. Kantarci, "Chapter 11 - Energy-Efficient Cloud Computing: A Green Migration of Traditional {IT}," in *Handbook of Green Information and Communication Systems*, M. S. Obaidat, A. Anpalagan, and I. Woungang, Eds. Academic Press, 2013, pp. 295–330. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124158443000115

[4] R. R. Harmon and N. Auseklis, "Sustainable IT services: Assessing the impact of green computing practices," *PICMET '09 - 2009 Portland International Conference on Management of Engineering & Technology*, pp. 1707–1717, Aug. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5261969

[5] R. Standard, "GHG Protocol Product Life Cycle Accounting and Reporting Standard ICT Sector Guidance," in *Greenhouse Gas Protocol*, 2013, no. January, ch. 7 - Guide.

[6] N. Wirth, "A plea for lean software," *Computer*, vol. 28, no. 2, pp. 64–68, 1995.

[7] R. Abreu, "Spectrum-based Fault Localization in Embedded Software," {PhD} in Computer Science, Delft University of Technology, 2009.

[8] R. Abreu, P. Zoeteweij, and A. J. V. Gemund, "Spectrum-Based Multiple Fault Localization," *2009 IEEE/ACM International Conference on Automated Software Engineering*, pp. 88–99, Nov. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5431781

[9] C. Chambers and C. Scaffidi, "Smell-driven performance analysis for end-user programmers," in *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*. IEEE, 2013, pp. 159–166.