Energy-Efficient Software

Procaccianti, G.

2015

document version
Publisher's PDF, also known as Version of record

English Summary

The energy consumption of ICT is growing at an unprecedented pace. The main drivers for this growth are the widespread diffusion of mobile devices and the proliferation of datacenters, the most power-hungry IT facilities. In addition, it is predicted that the demand for ICT technologies and services will increase in the coming years. Finding solutions to decrease ICT energy footprint is and will be a top priority for researchers and professionals in the field.

As a matter of fact, hardware technology has substantially improved throughout the years: modern ICT devices are definitely more energy efficient than their predecessors, in terms of performance per watt. However, as recent studies show, these improvements are not effectively reducing the growth rate of ICT energy consumption. This suggests that these devices are not used in an energy-efficient way. Hence, we have to look at software.

Modern software applications are not designed and implemented with energy efficiency in mind. As hardware became more and more powerful (and cheaper), software developers were not concerned anymore with optimizing resource usage. Rather, they focused on providing additional features, adding layers of abstraction and complexity to their products. This ultimately resulted in bloated, slow software applications that waste hardware resources – and consequently, energy.

In this dissertation, the relationship between software behavior and hardware energy consumption is explored in detail. For this purpose, the abstraction levels of software are traversed upwards, from source code to architectural components. Empirical research methods and evidence-based software engineering approaches serve as a basis. First of all, this dissertation shows the relevance of software over energy consumption. Secondly, it gives examples of best practices and tactics that can be adopted to improve software energy efficiency, or design energy-efficient software from scratch. Finally, this knowledge is synthesized in a conceptual framework that gives the reader an overview of possible strategies for software energy efficiency, along with examples and suggestions for future research.