Optimization in code generation to reduce energy consumption

David Branco
Mestrado em Engenharia Informática

Supervisor:
Pedro Rangel Henriques

University of Minho
November 2018

Outline

<u>Covered Topics</u>:

1. Green Computing;
2. Project Objectives;
3. Microprocessors;
4. Compilers Design;
5. Integrated Development Environments;
6. Experimental Study 1;
7. Experimental Study 2;
8. Project Contribution;

Problem and Motivation

The rapid development & high demands in computer technology, produce:

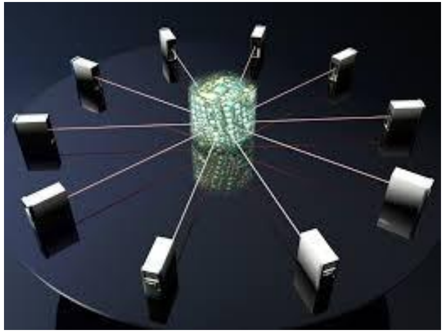
- Increasing energy costs:
  - 10% - 50% of big IT companies budget;
  - 30% of energy consumption of UK in 2020.
- Increasing cooling requirements;
- Restrictions on energy supply and access;
- Increasing equipment power density;
- Growing awareness of IT impact on the environment:
  - Servers CO_2 emissions equivalent to France and Australia in 2020;
  - Impact on marketing and public image.

Green Computing | Paradigm Definition

“ *Study and practice of the design, development, implementation, utilization and disposal of IT infrastructure efficiently and effectively with low or zero impact on the environment while reducing operating costs.*”

Green Computing Initiative

Green Computing | Measures and Solutions



Our Project

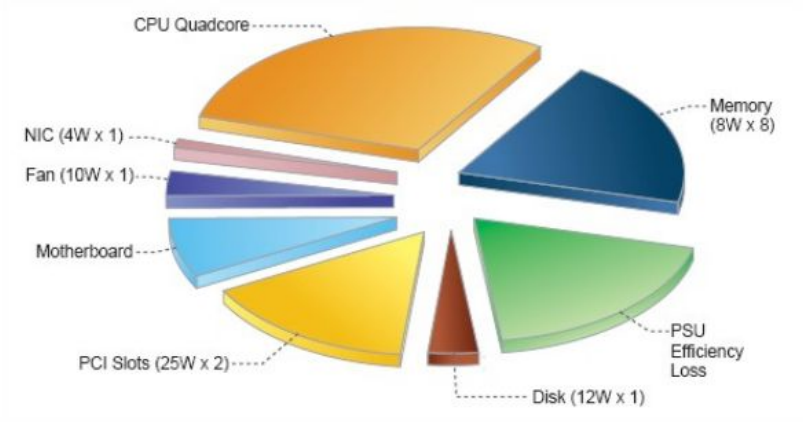
Main objectives:

1. Study the Green Computing paradigm;
2. Investigate IT elements from the energy efficiency perspective;
    2.1. Hardware;
    2.2. Software;
3. Develop an experimental study related to the triplet hardware-software-energy;
4. Produce suitable elements for other green oriented research.

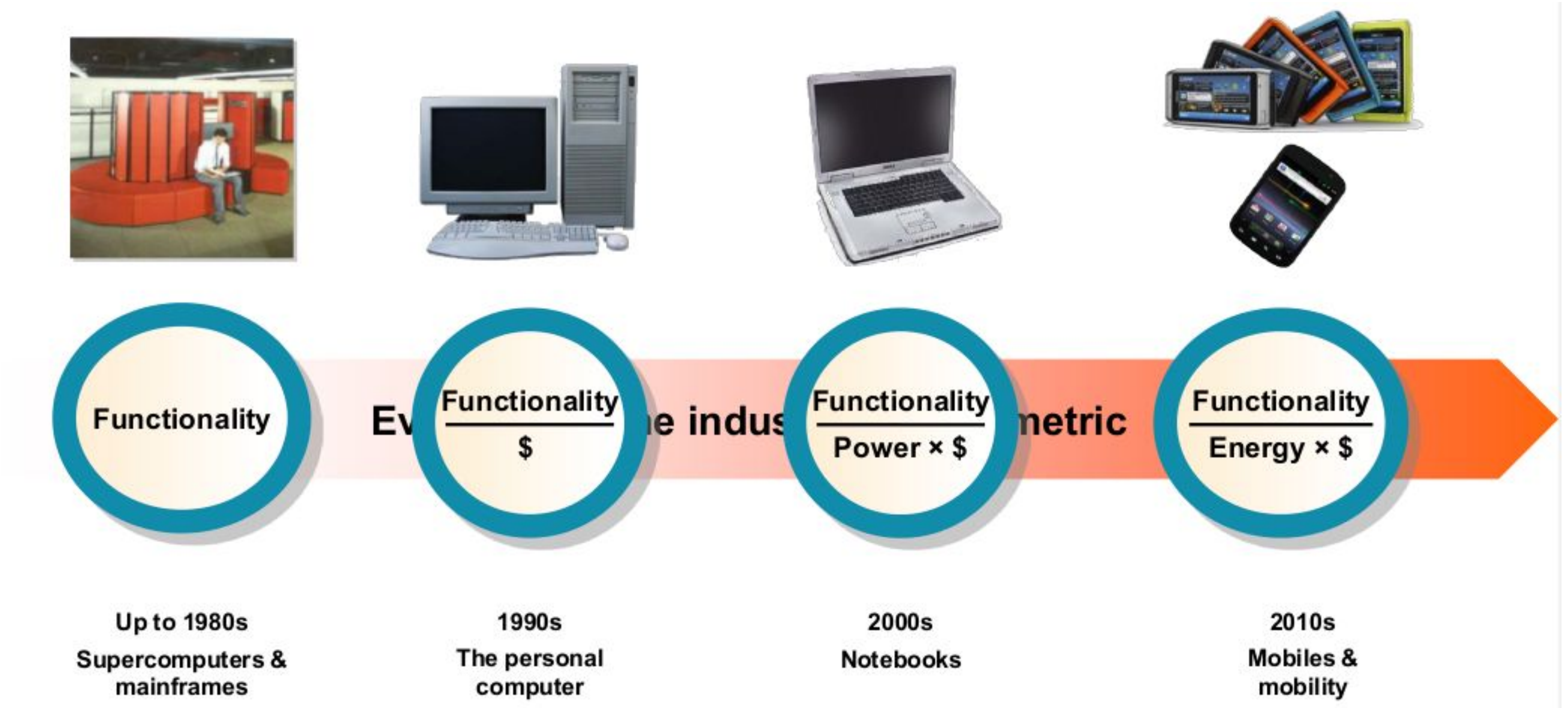
Microprocessors | Why ?

- They are everywhere !

- High energy consumption



Microprocessors | Requirements Evolution



Compilers | Why ?

- One of the most importante, mature and challenging area in IT;

- There is always room for improvement:
  - New features that need to be harnessed;
  - Optimizations of old algorithms;
  - Resurgence of techniques that regained importance.

- Potential to achieve significant improvements through optimizations:
  - ✔ Execution Time;
  - ✔ Code Size;
  - (?) Energy Consumption;

Compilers | Optimal properties of generated code

1. Correctness
   - Most important and at the same time the most fragile;
2. High Speed
   - Main focus of most of industry, optimizations and developers;
3. Small Size
   - Mainly relevant for mobile and embedded systems;
4. <u>Low Energy Consumption</u>
   - To increase the autonomy or respective costs;
   - To protect hardware limiting heat dissipation.

Compilers | Non Conventional Approaches

- Just-In-Time Compilation
  - Compile and execution time are not any more completely separated phases;
  - Tries to obtain the best of both worlds;
  - Efficiency of running object code vs inefficiency of recompiling the program.

- Green Compilers
  - Suitable for more demanding situations;
  - Set of intensive compilation techniques;
  - Lack of solutions (mainly free and open source).

IDEs | Emergence

“ *Gone are the days when a developer needed to write the code in a text editor, save it, exit the editor, run the compiler, annotate the error messages in an auxiliary pad, and finally reinspect the code.* ”

<u>Important Landmarks</u>:

- Turbo Pascal (1983): Integrated editor and compiler emerged;
- Microsoft Visual Basic (1991): Graphical Interface.

Microsoft Visual Basic

IDEs | Main Features

<u>Main Features</u>:

- Source Code Editor;
- Compiler;
- Debugger;
- Build Automation Tools;
- Extra Features:
    - Intelligent code completion and highlighter;
    - Code linting and error diagnostics;
    - Project browser and multiple output windows;
    - Integration of third-party software.

IDEs | Advantages and Disadvantages

Some Advantages:

- Project setup;
- Development tasks (diagrams, resources management, etc.);
- Project management and collaboration;
- Enforce project and company standards;
- Continual Learning.

Some Disadvantages:

- Require an initial investment;
- Necessary to choose an appropriate tool;
- Facilitates the creation of heinous code.

IDEs | Differentiation Factors

- Open Source communities, vendors and software companies;
- Different pricing and licensing;
- Different target audiences;
- Supported operating systems;
- System model;
- Different target machine and purpose;
- Supported programming languages;
- Features that provide (profiler, static code analysis, etc.);
- Allow or not plugins and extension integration.

IDEs | Software-as-a-Service

Allows to use the web browser as a client and to access a good range of cloud-based applications and services.

Some Advantages:

- Requires virtually no download or installation;
- Compatible with a greater number of devices;
- Access from anywhere in the world;
- Better collaboration between people in different locations;
- Better management of computational resources.

Study 1 | Objectives

Impact of GCC optimization levels in energy consumption during program execution

Main Objectives:

- Impact of optimized code on CPU energy consumption;
- Ascertain the strategy adopted by GCC regarding energy efficiency.

Study 1 | Experimental Elements

Main Elements:

- Execution time and energy consumption;
- *3* hardware components: CPU, RAM e GPU;
- *12* benchmarks in *4* programming languages: C, C++, Objective-C and Go;
- GCC optimization levels:
    - O0: Default Level (disables all optimization flags);
    - O1: Basic Level (up to 39 flags);
    - O2: Recommended Level (up to 73 flags);
    - Os: Reduced Code Size (up to 65 flags);
    - O3: Highest Level (up to 82 flags);
    - Ofast: Disregard strict standards compliance (up to 85 flags).

Study 1 | Measurement Elements

Testing Platform:

- Intel® Core i7-4710HQ up to 3.5 GHz (Haswell Family);

Measurement Software:

Running Average Power Limit (RAPL) by Intel®

- Measure time and energy consumption of a certain operation;
- Performs readings of CPU, RAM and GPU;
- Receive as argument the path for program makefile/executable;
- Managed through a mechanism of flags;
- Set the number of cores, read through perf, etc.;

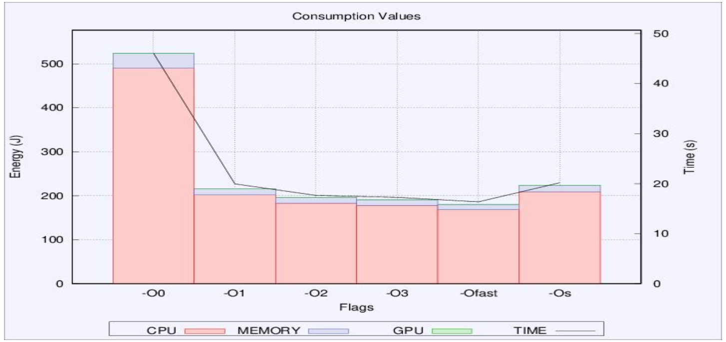
Study 1 | Methodology

Measurement Process:

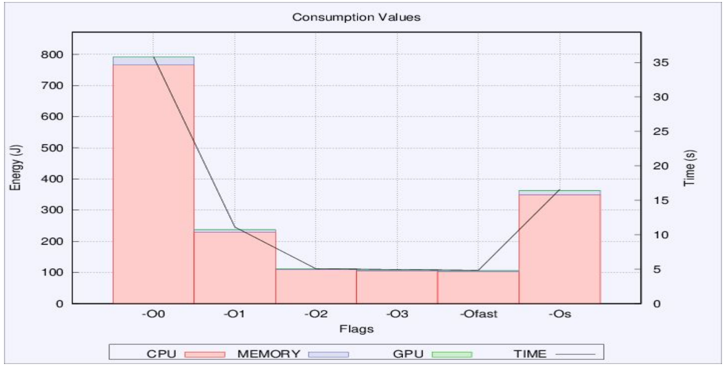
1. Select a program;
2. Select an optimization level;
3. Apply the measurement tool *100* times for the selected components;
4. Process the output generated:
   4.1. Exclude the *20* extreme cases;
   4.2. Calculate the intended average values;
5. Repeat the process for all the remaining programs and compilation profiles;
6. Output process to charts, tables and HTML pages.

Study 1 | Results

Oggenc (C)

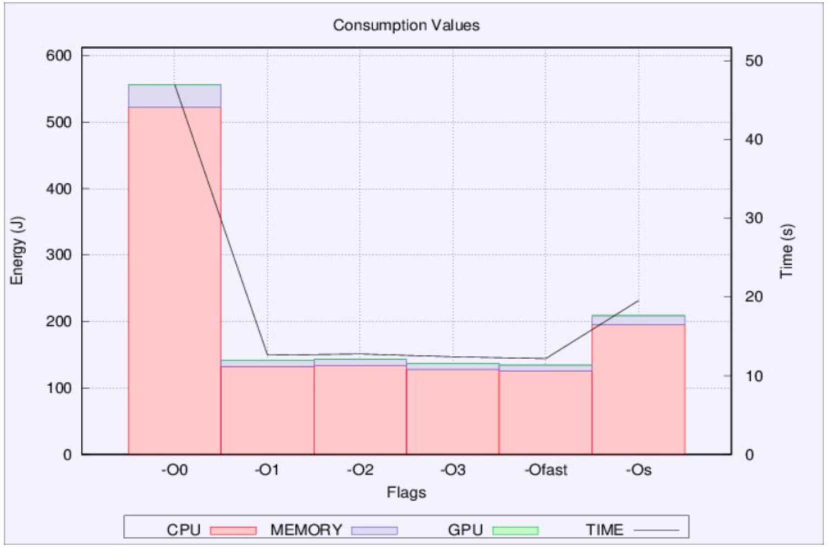


PGo (Go)



Pbrt (C++)

Flags	-O0	-O1	-O2	-O3	-Ofast	-Os
Time (s)	47.008	12.614	12.761	12.413	12.175	19.547
GPU (J)	0.028	0.006	0.005	0.096	0.220	0.009
Memory (J)	34.017	9.143	9.250	9.010	8.852	14.158
CPU (J)	522.205	132.318	133.565	127.830	125.526	195.006



Study 1 | Conclusions

Some of Study Conclusions:

- Optimization levels allow achieve considerable improvements;
- Correlation between execution time and energy consumption;
- Ofast > O2, O3 > O1, Os;
- Optimization results transverse to the different languages;
- It is not possible to conclude with certainty GCC's strategies on the matter.

Study 2 | Objectives

Impact of compilation by IDEs in energy consumption during program execution

Main Objectives:

- Deepen research on the compilation parameters;
- Study and evaluate software development tools;
- Perform measurements on more demanding benchmarks.

Study 2 | Tools Requirements

A given IDE is considered a candidate to be analyzed:

- Running under Linux environment;
- Support the C language and GCC compiler;
- Have a stable and release version;
- Capable of accomplishing the tasks under study without resorting to plugins installation;
- No cost of usage or at least have a trial version;
- Specification of the parameters used during the compilation process.

Study 2 | Analyzed Tools

IDE Name	Studied Version	Usage Model	License Type	Target Audience
Code::Blocks	17.12-1	Standalone	Free License	Beginner
Geany	1.33	Standalone	Free License	Beginner
DialogBlocks	5.15.3 (Unicode) Built Dec 13 2017	Standalone	Free for unregistered and registered account	Beginner
ZinjaI	20180221	Standalone	Free License	Beginner
Anjuta DevStudio	3.18.2	Standalone	Free License	Intermediate
GPS	20170515-63	Standalone	Free License	Intermediate
CLion	2018.1 Build #CL-181.4203.54	Standalone	Free Trial	Advanced
NetBeans IDE	8.2 Build 201609270201	Standalone	Free License	Advanced
CodeLite	12.0.0	Standalone	Free License	Advanced
Eclipse CDT	9.4.3.201802261533	Standalone	Free License	Advanced
KDevelop	5.2.1	Standalone	Free License	Advanced
Qt Creator	4.6.0 Based on Qt 5.10.1	Standalone	Free open source version and trial commercial version	Advanced
Oracle Developer Studio	12.6	Standalone	Free after account registration	Advanced
Sphere Engine	20180319.r445	Cloud	Free Trial	Intermediate
AWS Cloud9	Not specified	Cloud	Free with limited resources	Advanced

Study 2 | Profiles and Parameters

Compilation Profiles:

- Provide 2 profiles: Debug and Release;
- Little diversity and quantity:
  - *51* profiles: *29* distinct (43% repeated);
  - *144* parameters: *28* distinct (81% repeated).

Compilation Parameters:

- Configuration of the compilation process (31%);
- Management of diagnostic messages (26%);
- Produce debugging/profiling information (17%);
- Optimization of generated code (26%).

Study 2 | Experimental Elements

Main Elements:

- *12* C benchmarks;
- *18* Software Development Tools (*15* IDEs e *3* BATs);
- *51* compilation profiles consisting of *144* compilation parameters;
- Execution time, CPU and RAM energy consumption, Energy/Time ratio.

Measurement Elements:

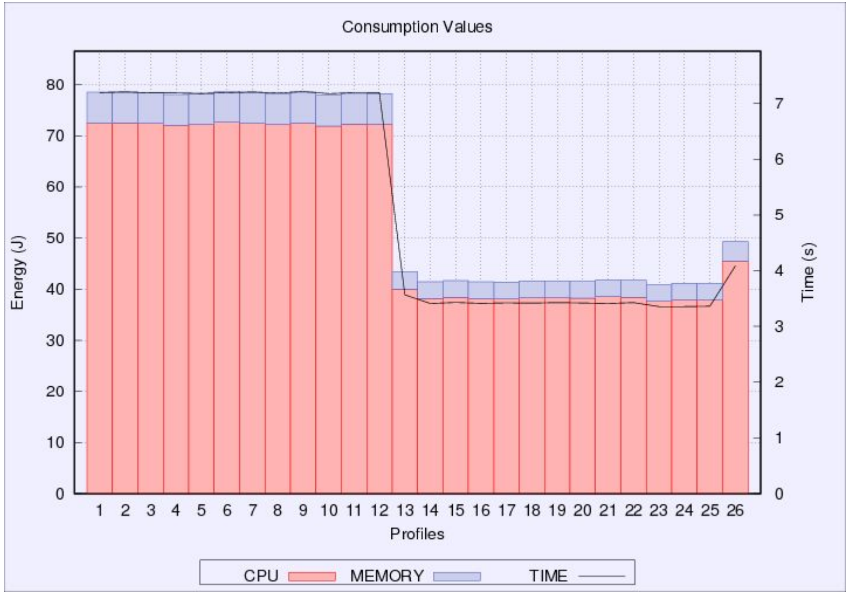
- Target Machine with Intel CPU (Haswell Family);
- Measurement Tool using RAPL.

Study 2 | Methodology

Measurement Process:

1. Select a program;
2. Select a compilation profile;
3. Apply the measurement tool *50* times for the selected components;
4. Process the output generated:
   4.1. Exclude the *20* extreme cases;
   4.2. Calculate the intended average values;
5. Repeat the process for all the remaining programs and compilation profiles;
6. Output process to charts, tables, rankings and HTML pages;

Study 2 | Profiles and Parameters Results

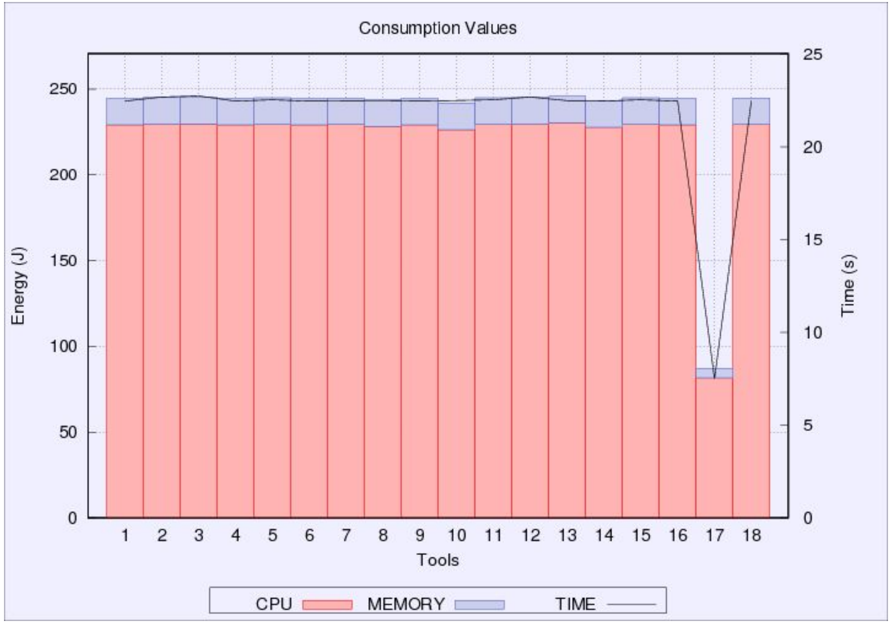


Total Profiles - Binary-trees

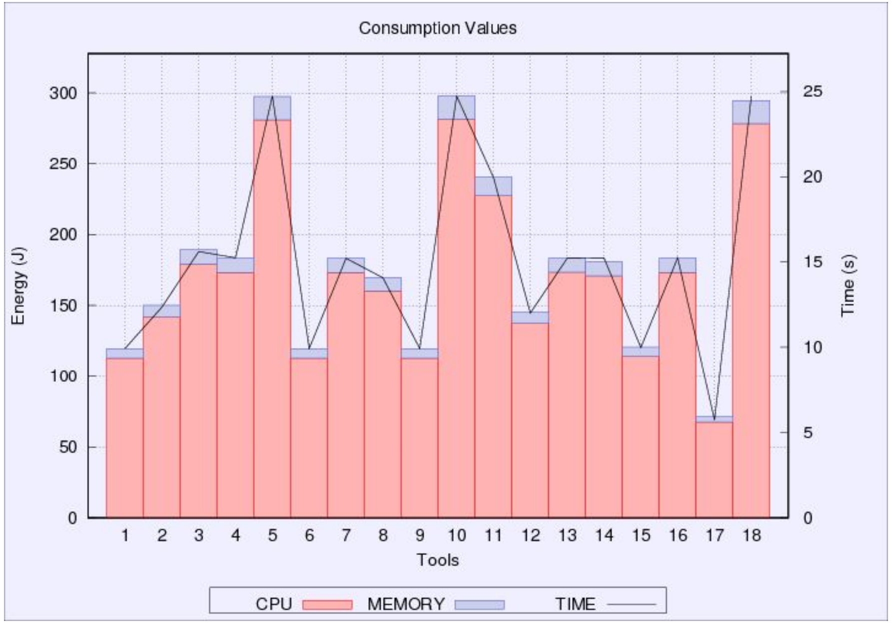
Optimization Level	Time (s)	Energy (J)	CPU (J)	Memory (J)	Energy/Time (J/s)
O0	16.862	181.611	169.872	11.739	10.582
O1	8.689 48.5%	85.511 52.9%	79.195 53.4	6.316 46.2%	10.212 3.5%
O2	8.415 50.1%	82.182 54.7%	76.063 55.2	6.119 47.9%	10.032 5.2%
O3	7.644 54.7%	74.058 59.2%	68.485 59.7	5.573 52.5%	9.872 6.7%
Os	9.408 44.2%	93.879 48.3%	87.061 48.7	6.818 41.9%	10.315 2.5%

Comparison between optimization levels

Study 2 | Tools Results



Default Profiles - K-nucleotide



Total Tools - N-body

Study 2 | Tools Results

Tool ID	Tool Name	Execution Time (s)	Total Energy (J)	CPU Energy (J)	Memory Energy (J)	Energy/Time (J/s)
1	CMake	3	3	3	3	3
2	qmake	8	4	4	6	4
3	Qbs	11	9	9	11	7
4	NetBeans IDE	10	11	9	10	7
5	Code::Blocks	15	15	14	15	11
6	CLion	3	3	3	3	3
7	CodeLite	6	7	7	7	8
8	Eclipse CDT	5	6	6	5	6
9	KDevelop	3	3	3	3	3
10	Geany	14	13	12	14	12
11	Anjuta DevStudio	12	12	11	12	9
12	Qt Creator	7	5	5	8	5
13	DialogBlocks	9	10	10	9	9
14	ZinjaI	4	8	8	4	10
15	GPS	2	2	2	2	2
16	Oracle Developer Studio	10	11	9	10	7
17	Sphere Engine	1	1	1	1	1
18	AWS Cloud9	13	14	13	13	13

Tools ranked with 1 decimal point

Study 2 | Conclusions

- Some tools have good solutions for developers;

- Measured Strands:
  - Great impact of the compiler (*86%* for the best case);
  - CPU is responsible for *90%* of energy consumption.
  - Correlations Observed:
    - Execution time and total energy consumption;
    - Total and CPU energy consumption;
    - Execution time and memory energy consumption.

- Maximum energy efficiency achieved:
  - CPU(53%) > Total energy(52%) > Execution time(47%) > RAM(46%).

Our Contribution

Some of project contributions:

1. Green oriented research of IT components;
2. Definition of experimental studies and methodologies;
3. Relevant results and conclusions;
4. Measurement Tool;
5. Green oriented workbench.

All this material is available at:

http://www4.di.uminho.pt/~gepl/OCGREC/

Optimization in code generation to reduce energy consumption

David Branco
Mestrado em Engenharia Informática

Supervisor:
Pedro Rangel Henriques

University of Minho
November 2018