Master Project Proposal

Lukas Koedijk (10783687)

March 2019

1 Project details

Project title: Green software
Student Name: Lukas Koedijk
Host organization or research group: KPMG Digital Enablement, `www.kpmg.nl`, Laan van Langerhuize 1
Contact person:

2 Project summary

Currently more and more people are concerned with global warming. Global warming is a result of the emission of greenhouse gasses during energy generation of not green energy options. Because of this a lot of people want to change to green energy generation to help solve this problem. Another solution is to decrease the energy consumption. The energy consumption of communication networks, personal computers and data centers world wide is increasing over the years [1]. This happens with a growth rate of 10%, 5% and 4% respectively [1]. Therefore it is important to research ways of decreasing the energy consumption. In the field of hardware there is, according to Koomey's law, an increase of the number of computations per Joule. However this is not enough, because tasks need more computations to complete due to the confidence in the improvement of hardware [2]. For this reason we need to look at possibilities in decreasing the energy consumption from a software perspective.

During this research I want to answer the following two existence questions: (1) **Is there a difference in the energy consumption of software projects in different programming languages that have the same functionality** and (2) **Is there a difference in energy consumption of different software projects (in the same programming language) that have the same functionality**? To answer these questions I first need to answer the three description and classification questions listed below.

- How can the energy consumption of a software project be measured?
- How do we proof if two programs have the same functionality?

- When is a difference in energy consumption big enough to be called a difference?

Based on the results I find when answering the second existence question, I want to look into what this difference is on code level. To do this I want to answer the descriptive/comparative question **what is the difference on code level**? This question is only useful when there is a difference in the energy consumption of software project in the same programming language that have the same functionality. Otherwise we can't link the difference in energy consumption to the difference in the code.

3 Problem analysis

The energy consumption of communication networks, personal computers and data centers are increasing yearly. There are two ways of decreasing the energy consumption, by decreasing the energy consumption of hardware or software. Scientific research is mostly focused on decreasing the energy consumption of hardware. There are also some papers about reducing the energy consumption in the software process and the decision making process. A small bit of research is done on the energy consumption of software, but their research goal is to estimate the energy consumption. Therefore what I miss and want to look into is if there is a good way of writing code regards the energy consumption.

4 Research method

As method I want to use a controlled experiment. In this experiment I will run different software projects to measure the energy consumption. Here the projects are the variable input, the energy consumption the output and everything else like compiler and energy consumption calculations should be constant. For the first research question the hypothesis is that there is a difference in the energy consumption based on the programming language chosen. The hypothesis of the second research question is that there is a difference in the energy consumption of different software projects (in the same programming language) that have the same functionality.

4.1 Data

For the first research question I need programs that have the same functionality, but are written in a different programming language. I want to use library code for this, because they are viewed as well-thought-out solutions to a problem. For the second research question I want to use the library as a baseline. The other software projects I use need to have the same functionality as the library. Thus I need to find implementations of the functionality the library has. Options for finding software projects with the same functionality are: student code of the same assignment, interview code and competition code. I already looked at some competition code on git from the Euler project. This seems to be the best option, because these projects are publicly available and can therefore also be tested by others.

When proving that all the projects have the same functionality I will define properties for the input and output and test if these properties hold for all the projects [3].

To proof that the projects have different values for the energy consumption we will preform the independent two sample t-test. This test calculates if two distributions are in the same distribution. This test can only be use if the both distributions are normally or close to normally distributed, have the same size and have the same variance. To check if they are normally distributed we can use the Chi-square test and to check if the variance is the same we can use the f-test.

4.2 Measuring energy consumption

When measuring the energy consumption of a project you have to take into account the energy consumption of CPU, memory and disk [4]. The measurements can be done with a hardware or software approach. A hardware approach is more accurate but also more expensive [4]. I want to use a hardware approach, luckily as a student of the UvA I have access to the DAS-4 and the DAS-5. The DAS is a distributed system that can also do energy measurements [5]. You can connect to the DAS using ssh with an username and a password. The nodes that can measure the energy are located at the VU. To measure the energy I need to run a job on the DAS-5 and use the DAS-4 to measure the energy consumption. When running a job I need to specify witch node I want to run the job on, because not all nodes can be measured for the energy consumption. This is done by using the *-w* flag while using the command *srun*. The measuring can then be done by using the *smnpwalk* command and then filtering out the information by using *fgrep Energy*.

4.3 Interpret results

When there is a difference in the energy consumption of different software projects in the same programming language I will look at the different projects on code-level. Here I will try to find what is causing a project to have a lower or higher energy consumption. These findings will then be tested by letting myself write two versions, where I think one is written badly regards the energy consumption and one that is written good. I will then run the two version and check if the good version has a lower energy consumption.

5 Expected results of the project

- Answers to the research questions
- A script that measures the energy consumption

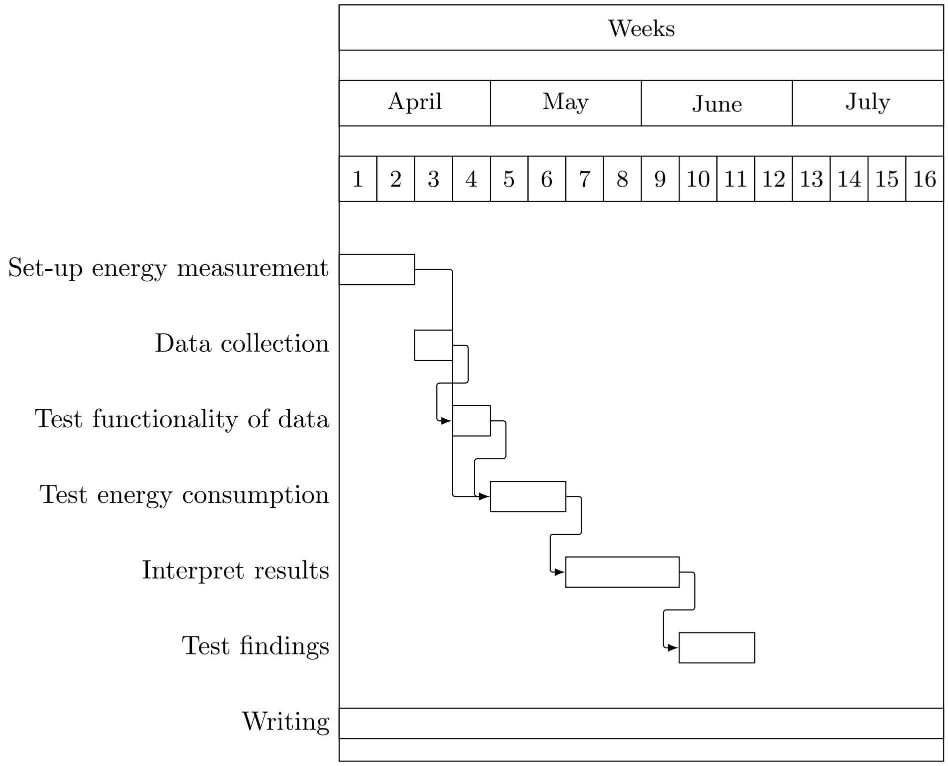
- A dataset of software project with the same functionality
- Rules for writing good software regards the energy consumption

6 Required expertise for this project

- Programming (probably Java, Python or C) - Advanced
- DAS-4/5 - Intermediate
- Statistics - Intermediate
- Testing - Intermediate
- Reading - Intermediate
- Thesis writing - Intermediate

7 Timeline

To make a structured planning I made a chart shown below. This chart shows which activities I should have completed before going to the next. In the chart the timespan is shown in weeks, where the first week start on the first of April. In the first two weeks I want to figure out how to measure the energy consumption, implement it and check if it works. The following two weeks I want to find data and check if the projects have the same functionality. In week five and six I want to measure the energy consumption of all the projects. In the three weeks after that I want to look at and interpret the results. Here I want to check if there is a difference and if so look into the source code. Based on what I find in the source code I want to test if my findings are true in week ten and eleven. The rest of the time I want to spend on writing my thesis.



8 Risks

One risks is that my dataset might be too small. I already looked at the Euler project where they list a lot of math problems that need to be solved by writing code. Their site does not let you see others result, but a quick search on git gave some solution. I haven't checked if there is enough projects available on git of the same Euler project problem in the same programming language. Another risk is that I have trouble implementing measuring of the energy consumption or that the implementation doesn't work accurately.

9 Literature survey

To get an overview of the related research I made a table, table 1, comparing the different papers. Most research is about reducing the energy consumption of a specific piece of hardware, for example scheduling on a multi-core processor. There are also some papers about reducing the energy consumption in the software and the decision making process. There is also some research done on the energy consumption of software, but their research goal was to estimate the energy consumption. Therefore what I miss and want to look into is to research if there is a good way of writing code regards the energy consumption.

Papers	Type of Research	Unit of Analysis	Goal
[2]	Controlled Experiment	Deployment strategies, releases and use case scenarios	Finding optimal energy consumption
[6]	Case Study & Controlled Experiment	HPC bag of task applications	Finding optimal energy consumption
[4]	Case Study	Small functions	Estimating energy consumption
[7]	Case Study & Controlled Experiment	Small functions	Estimating energy consumption
[8]	Case Study	Task on complex micro-architectures	Estimating worst-case energy consumption
[9]	Empirical Study	Six commonly used refactorings	Finding impact of refactorings on energy consumption
[10]	Controlled experiment	Programming languages	Rank programming languages based on speed, memory usage and energy consumption
[11]	Case Study	Multi-core processor scheduling	Efficient workload partitioning
[12]	Case Study	Cache storage management algorithms	Power aware cache management
[13]	Case Study	Java applications	Framework to automate decision-making support regarding energy consumption
[14]	Case Study	Software process	Two level green software model

Table 1: Overview of related research

9.1 Estimating energy impact of software releases and deployment strategies: the KPMG case study [2]

This paper looks at the impact of releases and deployment strategies of a software product on the energy consumption. They used a controlled experiment where the variables they changed where deployment strategies, releases and use case scenarios. The variables they measured during their tests were power consumption and execution time. They saw that both the releases and deployment strategies impacted the energy consumption and that this impact was influenced by which use case scenario they used. Therefore they concluded that there is no absolute optimal option for releases and deployment strategies with respect to energy consumption. They also found that the execution time had a bigger impact on the energy consumption than the power consumption, because of the low variability in power consumption.

9.2 The Impact of Source Code in Software on Power Consumption [4]

This paper looks at different techniques to measure the power consumption. Then they propose a model to measure the power consumption and they used this model in their implementation named *Tool to Estimate Energy Consumption* (TEEC). They test their implementation against a power meter, but they do not mention how accurate their implementation is. The figure they use at the validation is also not clear, they just state that it shows the same behaviour as TEEC. They find that the power consumption of unoptimized code is higher and has a longer execution time than the optimized code. They do not mention it but looking at their graphs, the unoptimized and optimized code seem to have the same peaks only on different time steps for the optimized code is faster.

9.3 Energy Consumption Analysis of Programs based on XMOS ISA-Level Models [7]

This paper estimates the energy consumption by developing a model which can be applied at instruction set simulation level. This was done by designing a translation from instruction set architecture code to Horn- clause representation and this model is called in the paper *Instruction Set Simulation* (ISS). They also use the CiaoPP general resource analysis framework, which is low level, to model the energy consumption. They named it *Static Resource Analysis* (SRA) in the paper. In their experiments they only use small functions to test and the results were compared to a mathematical equation. The found that the ISS function is less accurate when the value of N increases and that the SRA function is only not accurate for small value of N. Here N is the input value of the function that is tested for its energy consumption.

9.4 E-BATS: ENERGY-AWARE SCHEDULING FOR BAG-OF-TASK APPLICATIONS IN HPC CLUSTERS [6]

This paper looks at the scheduling of bags of task application in High performance Computing (HPC). They delved into the trade-off between energy consumption and performance by finding a optimal point for both variables. This was calculated by designing a dynamic Hill Climbing algorithm. Their algorithm uses less then 12% of the resources an exhausted search would use to find a majority of points close to the optimal for the trade-off in 10 of the 12 scenarios. They validated their algorithm by implementing it and running a wide range of HPC bag of task applications. They found that the estimations of their algorithm have an error below 5%.

9.5 SEEDS: A Software Engineer's Energy-Optimization Decision Support Framework [13]

This paper implements a framework that automatically optimizes the energy consumption of a Java software project. The framework does this by running multiple different given options and testing which option consumes the least amount of energy. Thus as input the framework needs a list of possible changes. Because the framework needs possible changes we don't know if the resulting code is the most energy efficient version, only that it is more energy efficient then the input. They showed that by letting their framework chose which library to use they reduced their energy consumption by 17%.

9.6 How Do Code Refactorings Affect Energy Usage? [9]

This paper addresses the lack of information about the energy impact of code refactorings. They did this by testing the energy impact of 197 projects when the using six commonly used refactorings. From this test they found that refactorings can influence the energy consumption. Also they find that one refactoring does not necessarily have the same influence on the energy consumption when used with different projects.

9.7 Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management [12]

This paper tries multiple algorithms for storage cache management to decrease the energy consumption. One algorithm is an offline greedy algorithm and they also propose an online algorithm. They evaluate their algorithms by simulating a complete storage system, enhancing the DiskSim simulator. Their greedy algorithm results in 16% less energy used then the LRU algorithm. They also find that the cache policy write-back can use 20% less energy then write-through.

9.8 A Green Model for Sustainable Software Engineering [14]

This paper makes a two level green software model. The first level is about making the software process more energy efficient. This new process is a hybrid of the sequential, iterative, and agile development processes. The second level is about the role software tools can have on improving the energy efficiency of software. They also discus the relationship between the two levels.

9.9 Estimating the Worst-Case Energy Consumption of Embedded Software [8]

This paper estimates the worst-case energy consumption of a task on complex micro-architectures. This is important for battery-operated embedded devices, where we don't what the battery to drain empty before a critical task is completed. They test their result against a commonly used benchmark and they find that their values have at most 33% difference with the benchmark.

9.10 Energy Efficient Workload Balancing Algorithm for Real-Time Tasks over Multi-Core [11]

This paper proposes an algorithm to makes sure all cores in a multi-core processor have the same workload. This is reducing energy consumption because multiple single core processors consume more energy.

9.11 Energy Efficiency across Programming Languages [10]

This paper tries to find a connection between the speed, memory usage and energy consumption of a programming language. They do this by choosing the fastest implementation of the exact same algorithm, defined in the computer language benchmarks game, in different programming languages. From these programs they measured the execution time, memory usage and energy consumption. They used Intel's Running Average Power Limit (RAPL) tool to measure the energy consumption and for the memory usage and execution speed they used the *time* command available in Unix-based systems. They find that a faster programming language does not necessarily have a lower energy consumption and memory usage does not relate to energy consumption. A big problem with this paper is that in their threads to validity paragraph they defend their implementation instead of stating what could be wrong with their implementation. My main problem with this paper is that they compare languages based on the fastest solution in some competition and these are not comparable. Because there can be languages where the fastest solution given in the competition is not the fastest at all.

10 Bibliography

References

[1] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ict electricity consumption from 2007 to 2012", *Computer Communications*, vol. 50, pp. 64–76, 2014.

[2] R. Verdecchia, G. Procaccianti, I. Malavolta, P. Lago, and J. Koedijk, "Estimating energy impact of software releases and deployment strategies: The kpmg case study", in *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, IEEE Press, 2017, pp. 257–266. DOI: 10.1109/ESEM.2017.39.

[3] T. Mens and T. Tourwé, "A survey of software refactoring", *IEEE Transactions on software engineering*, vol. 30, no. 2, pp. 126–139, 2004.

[4] H. Acar, G. Alptekin, J.-P. Gelas, and P. Ghodous, "The impact of source code in software on power consumption", *International Journal of Electronic Business Management*, vol. 14, pp. 42–52, 2016.

[5] H. Bal, D. Epema, C. de Laat, R. van Nieuwpoort, J. Romein, F. Seinstra, C. Snoek, and H. Wijshoff, "A medium-scale distributed system for computer science research: Infrastructure for the long term", *Computer*, vol. 49, no. 5, pp. 54–63, 2016.

[6] A. V. Filip, A.-M. Oprescu, S. Costache, and T. Kielmann, "E-bats: Energy-aware scheduling for bag-of-task applications in hpc clusters", *Parallel Processing Letters*, vol. 25, no. 03, p. 1 541 005, 2015.

[7] U. Liqat, S. Kerrison, A. Serrano, K. Georgiou, P. Lopez-Garcia, N. Grech, M. V. Hermenegildo, and K. Eder, "Energy consumption analysis of programs based on xmos isa-level models", in *International Symposium on Logic-Based Program Synthesis and Transformation*, Springer, 2013, pp. 72–90.

[8] R. Jayaseelan, T. Mitra, and X. Li, “Estimating the worst-case energy consumption of embedded software”, in *Real-Time and Embedded Technology and Applications Symposium, 2006. Proceedings of the 12th IEEE*, IEEE, 2006, pp. 81–90.

[9] C. Sahin, L. Pollock, and J. Clause, “How do code refactorings affect energy usage?”, in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ACM, 2014, p. 36.

[10] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, and J. Saraiva, “Energy efficiency across programming languages: How do energy, time, and memory relate?”, in *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*, ACM, 2017, pp. 256–267.

[11] M. Zakarya, N. Dilawar, M. A. Khattak, and M. Hayat, “Energy efficient workload balancing algorithm for real-time tasks over multi-core”, *World Applied Sciences Journal*, vol. 22, no. 10, pp. 1431–1439, 2013.

[12] Q. Zhu, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao, “Reducing energy consumption of disk storage using power-aware cache management”, in *Software, IEE Proceedings-*, IEEE, 2004, pp. 118–118.

[13] I. Manotas, L. Pollock, and J. Clause, “Seeds: A software engineer’s energy-optimization decision support framework”, in *Proceedings of the 36th International Conference on Software Engineering*, ACM, 2014, pp. 503–514.

[14] S. S. Mahmoud and I. Ahmad, “A green model for sustainable software engineering”, *International Journal of Software Engineering and Its Applications*, vol. 7, no. 4, pp. 55–74, 2013.