An Approach to Estimating Energy Consumption of Web-based IT Systems

Maja H. Kirkeby[1], John P. Gallagher[1] and Bent Thomsen[2]

[1]Roskilde University, Roskilde, Denmark
[2]Aalborg University, Aalborg, Denmark

In this short paper we motivate and present an approach to estimating the entire energy consumption of a web-based software system. The purpose of the estimation is to be able to document – and in the future also predict – the impact of changes to the platform on energy consumption, e.g., energy-reducing initiatives or a growth in the number of users.

This investigation is a part of a preliminary project on energy certified software development and development operations; we are focusing on a single web-based IT system, which will be presented after the motivation.

1. Motivation

Information and Communication Technologies (ICT) are estimated to consume 6-9% of the world's electricity at present, and this is expected to increase to 20.9% in 2030. Software regulates ICT's energy consumption; thus, our approach focus on the software together with the hardware it executes on.

Different approaches to reducing the energy consumption of software have been studied, e.g., by exploiting compilers [1, 2], by using a less energy-consuming programming language [3, 4], by providing developers with models that relate energy consumption to source code [5, 6, 7, 8, 9, 10], or help detecting energy leaks of developed source code [11, 12].

For distributed communication systems, it is crucial to also consider the relationship between software running on different devices to understanding the energy consumption of the whole system. This is because several studies that estimated the energy consumption of ICT [13, 14, 15, 16] found that there are three major energy-consumers: user-devices (42%), data transmission via networking (27%) and servers perhaps in data centres (31%) [14, tab.6]. Thus, an energy-reducing initiative focused on the computations in the source code affects mainly those components running the source code, i.e., servers and user-devices. While the energy-consumption distribution for the major components will differ for each individual program, it indicates that we have to take all three major energy-consumers into consideration when evaluating the impact of energy reducing initiatives.

CERCIRAS WS01: 1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society
kirkebym@acm.org (M. H. Kirkeby); jpg@ruc.dk (J. P. Gallagher); bt@cs.aau.dk (B. Thomsen)

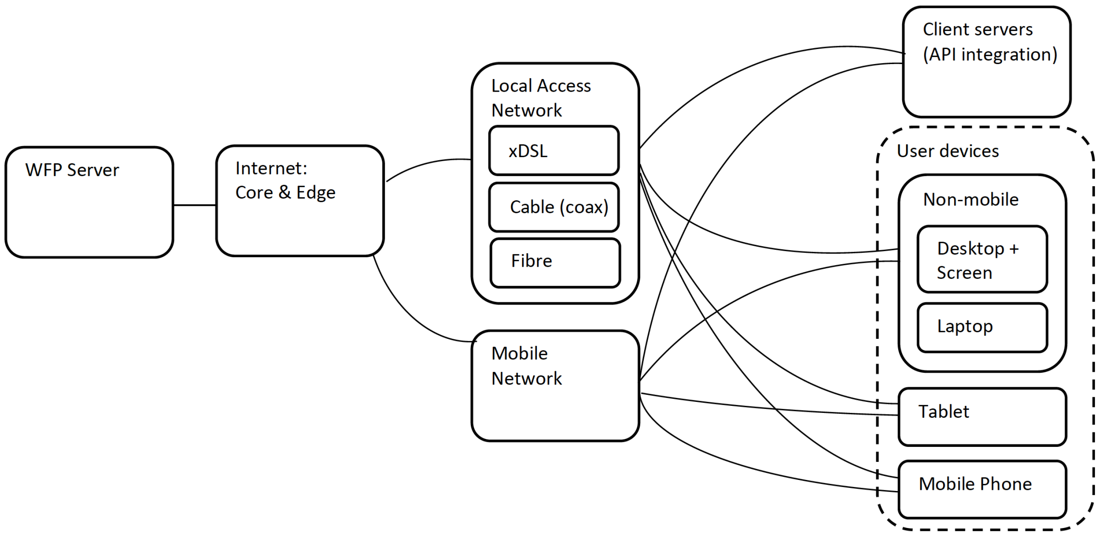

Figure 1: An overview of the WFP system. The lines indicates the possible connections between components.

The aim of this project is to investigate how we can provide energy consumption estimates of an existing web-based software program that may be suitable for estimating the energy reductions resulting from various changes, some inspired by [17]. We expect that modelling the energy consumption of any complex software system involves interdependent analytic and empirical tasks. In this short paper, our aim is to present a starting point for a formal model, based on the existing literature, and an approach that allows iterative refinement and extension of the model, while constantly comparing the model with empirical data from the case study software.

2. The case study software

The software that we are investigating is Edora A/S's live software system Work Force Planner (WFP). WFP is a live, self-service web-based booking and resource management system with 500,000 daily interactions used by many local authorities in Denmark such as Jobcentre, A-kasser, and Borgerservice. WFP serves as an exemplary case of software since its architecture is fairly typical for web services.

The system has several entrance points serving various types of end-users, e.g., employees and managers at the jobcentre, citizens, or A-kasse consultants. It supports both standalone solutions with web-based interfaces and seamless integration to existing systems via APIs. The application is hosted at an on-premise server in a data center, i.e., it is not cloud-hosted. The server interacts with the API callers and user-devices via core internet and either fixed lines and then residential access networks, or cell networks and then cellular networks.

3. Estimating energy consumption of web-based IT systems

Much energy-related research has focused on creating energy models with the aim of providing developers with tools to reduce the energy consumption during development. A series of studies have focused on which parameters influence energy consumption. However, these are focused on individual context: User-devices: device model, settings, the internet signal strength, e.g., [18], network: the amount of data sent, the type of internet, competing traffic load, e.g., [19], and servers: the type of physical server, whether there are virtual servers on top, competing programs, and concurrent users [20, 21, 22]. These and more were included in synthesizing 30 models for cloud-based systems [23] and provides a bottom-up model on what influences the energy consumption of software.

The studies that estimate the energy consumption of ICT [16, 15, 14, 13] use the life cycle assessment model which is a top-down approach. This was refined by Preist *et al.* to a model for a single software program; they present an energy-consumption estimation method based on the life cycle assessment model and exemplify it for YouTube [24]. They estimate the electricity used by YouTube over a period of a year and calculate the green house gas emissions from these. In the single software scoped model they combine (1) user behaviour, i.e., quantify the number of users and the ways in which they use the device with (2) a process model of the service, using Environmental LCA techniques, allowing emissions associated with service use to be estimated for a given user behaviour model. It is relevant to compare these estimation results with those obtained by tools such as Argos [25], a tool that measures the environmental footprint of software.

An energy model for WFP and its requirements

Our initial attempt to build an energy model for WFP is based on the approach of Preist *et al.* [24]. WFP is hosted on a physical server in a data center and Edora A/S measures the electricity consumed by the server. The server hardware is an important factor in the server's energy consumption. Yet, because the purpose is to estimate the energy consumption and we have the true energy consumption of the server, we can treat the entire server as a black box and do not need to consider neither server software nor the server hardware to estimate the server's energy consumption. Preist *et al.* provide an approach to estimating the energy consumption of "the internet", the edge and the core internet, based on the amount of data transmitted. Thus, in order to adopt a similar approach, we need to measure the amount of data sent from and received by the server. Estimating the residential access networks [24] requires the user-hours for each kind of service, and they suggest using user analytics data (specifically IP addresses) to determine which ISP and the corresponding technology a given residential user is connected with, and therefore the time each technology is used by their service. They estimate the energy consumption of the cellular networks using the amount of data, which can be obtained by user analytics data. The energy consumption of the user-devices can be estimated by how much each different device type is used, i.e., the use-hours for the device types and the average power consumption of each device type. These numbers we expect to obtain via user analytics data and Energy Star catalogue [26]. The API callers are devices initiating the API calls. In case we are not able to determine the type of API caller device to be either user devices or servers, we suggest using a conservative energy estimation assuming them to be servers. The energy consumption of API callers can then be estimated to be the average power consumption of typical servers [26] multiplied by the total response times of the API calls. If the type of devices can be determined, the approach in [27] can be used for refining the estimate.

In summary, Preist *et al.*'s model can be conceptually seen as involving functions relating transmitted data and energy consumption, of general form $f(D) = E$, where D is a measure of data and E is energy, for the network part of the system. That is, if D amount of data is transmitted from a server to a client, then the model predicts that $f(D)$ amount of energy will be used, distributed in some way over the server, client and network. Secondly, the amount of energy consumed by clients and servers are expressed as functions with parameters such as the number of users and the time spent.

Our first goal is to capture these relationships, specifically for the WFP system, in a dynamic model such as a priced timed automaton [28], so that we could perform experiments and simulations using a tool such as UPPAAL [29] with different amounts of data and configurations of the system.

In the medium term, we aim to build on this model, extending and refining it in two directions.

- The modelling based on Preist *et al.*'s approach can be refined, taking into account other details of the architecture, hardware, network, routing algorithms, etc. This would entail taking further measurements of data traffic and power consumption at different points of the system as input to the model construction. Such refinements would support predicting the impact of system changes, e.g., changes to the server architecture and hardware or changes in the data transmission.
- We aim to extend the model using a "software based" approach, constructing functions based on the analysis of the algorithms used in the client and server. This will draw on the large literature on resource analysis of software, using techniques such as abstract interpretation (e.g. [5, 6]). Such models will also help to identify code "hot spots" that consume much energy and are targets for optimization. This can be done statically, as opposed to being based on dynamic energy measurements, and can assist a developer to be more "energy-aware".

The WFP system is a typical web-based system, and the approach does not yet cover other typical or new system design possibilities. When going beyond the WFP system, it would be useful to extend the approach to cover, for instance, the use of cloud environments and more complex server architectures including cloud front-ends, load-balancers, S3 buckets (for static parts of the web-sites); the use of multiple and perhaps automated scaling of web-servers, databases; the use of edge-servers; and the use of apps or other software front-ends to publish the service instead of using browsers. Such extensions would also provide the basis for a holistic energy consumption analysis of the full spectrum of software system design possibilities, e.g., comparing thin clients and heavy cloud computing to client-heavy computing with reduced data transmission or to edge-heavy computing.

4. Discussion

The initial energy model based on Preist *et al.*'s work can only reflect some energy-reduction initiatives and only to some extent. It can reflect initiatives that reduce the amount of transmitted data, e.g., by reducing the amount of JavaScript Libraries, reducing the user-hours, e.g., by improving the user flow, and of course reductions of computations on the server side.

However, since the user-device estimation does not reflect the actual computations in its browser, reductions of these would not be reflected. This may cause an imprecision in the modelled energy consumption. For instance, consider an initiative where the server stores and sends compressed data – instead of storing and sending ordinary data– to the user-device which then have to decompress it and show it in the browser – instead of just showing it. The reduction in energy consumption could be overestimated, since transmission of less data causes a reduction of energy consumption and the extra computations caused by the decompression on the user-device are not taken into account. Therefore, it would be of interest to establish the limitations of the model, and investigate how to provide more accurate estimates of the user-devices' energy consumption. Such investigations could consider the energy model parameters and are the topic of the proposed model enhancements proposed above.

As mentioned above, we envision that a formal model based on the parameters established experimentally can be captured in the form of a priced timed automata [28] and can then be subjected to automatic analysis by the UPPAAL model checker [29]. We envision that it will be possible to use UPPAAL to find a balance between the system's response time requirements and its total energy consumption following ideas presented in [30]. A domain specific language (MobiCa) which can be used to model Mobile Cloud Computing systems and translate such descriptions into UPPAAL models is presented in [31, 32]. It is likely that this work could be extended or modified to describe web based systems such as WFP, thereby facilitating reasoning about the effects of single server systems and cloud based systems, even reasoning about optimal placement of functionality and scheduling of tasks. However, until a more precise model is developed it is important to evaluate whether new initiatives can be estimated accurately within the boundaries of the model.

Acknowledgments

We thank the reviewers for the insightful recommendations. This work is partially supported by CERCIRAS COST action CA19135 funded by the COST Association and partially by the project *Energy certified software development and development operations* funded by DigitalLead.

References

[1] K. Georgiou, C. Blackmore, S. Xavier-de Souza, K. Eder, Less is More, in: Proceedings of the 21st International Workshop on Software and Compilers for Embedded Systems, ACM, New York, NY, USA, 2018, pp. 35–42. URL: https://dl.acm.org/doi/10.1145/3207719.3207727. doi:10.1145/3207719.3207727.

[2] K. Georgiou, Z. Chamski, A. Amaya Garcia, D. May, K. Eder, Lost In Translation: Exposing Hidden Compiler Optimization Opportunities, The Computer Journal (2020). URL: https://academic.oup.com/comjnl/advance-article/doi/10.1093/comjnl/bxaa103/5880035. doi:10.1093/comjnl/bxaa103.

[3] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, J. Saraiva, Energy efficiency across programming languages: How do energy, time, and memory relate?, SLE 2017 - Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, co-located with SPLASH 2017 (2017) 256–267. doi:10.1145/3136014.3136031.

[4] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. P. Fernandes, J. Saraiva, Ranking programming languages by energy efficiency, Science of Computer Programming 205 (2021) 102609. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167642321000022. doi:10.1016/j.scico.2021.102609.

[5] K. Eder, J. P. Gallagher, P. López-García, H. Muller, Z. Banković, K. Georgiou, R. Haemmerlé, M. V. Hermenegildo, B. Kafle, S. Kerrison, M. Kirkeby, M. Klemen, X. Li, U. Liqat, J. Morse, M. Rhiger, M. Rosendahl, ENTRA: Whole-systems energy transparency, Microprocessors and Microsystems 47 (2016) 278–286. doi:10.1016/j.micpro.2016.07.003. arXiv:1606.04074.

[6] U. Liqat, S. Kerrison, A. Serrano, K. Georgiou, P. Lopez-Garcia, N. Grech, M. V. Hermenegildo, K. Eder, Energy Consumption Analysis of Programs based on XMOS ISA Level Models, in: 23rd International Symposium on Logic-Based Program Synthesis and Transformation, LOPSTR, volume 8901 of *LNCS*, 2013, pp. 72–90.

[7] J. Pallister, S. Kerrison, J. Morse, K. Eder, Data dependent energy modelling: A worst case perspective, in: Proceedings of the Eighteenth International Workshop on Software and Compilers for Embedded Sys- tems. ACM, 2015, pp. 12–21. URL: http://arxiv.org/abs/1505.03374. arXiv:1505.03374.

[8] S. Hao, D. Li, W. G. J. Halfond, R. Govindan, Estimating mobile application energy consumption using program analysis, Software Engineering (ICSE), 2013 35th International Conference on (2013) 92–101. doi:10/4c7.

[9] D. Li, S. Hao, W. G. Halfond, R. Govindan, Calculating source line level energy information for Android applications, 2013 International Symposium on Software Testing and Analysis, ISSTA 2013 - Proceedings (2013) 78–89. doi:10.1145/2483760.2483780.

[10] S. Chowdhury, S. Borle, S. Romansky, A. Hindle, GreenScaler: training software energy models with automatic test generation, volume 24, Empirical Software Engineering, 2019. doi:10.1007/s10664-018-9640-7.

[11] R. Pereira, T. Carção, M. Couto, J. Cunha, J. P. Fernandes, J. Saraiva, SPELLing out energy leaks: Aiding developers locate energy inefficient code, Journal of Systems and Software 161 (2020) 110463. URL: https://doi.org/10.1016/j.jss.2019.110463. doi:10.1016/j.jss.2019.110463.

[12] R. Verdecchia, A. Guldner, Y. Becker, E. Kern, Code-Level Energy Hotspot Localization via Naive Spectrum Based Testing, Springer International Publishing, 2018. URL: http://dx.doi.org/10.1007/978-3-319-99654-7_8. doi:10.1007/978-3-319-99654-7_8.

[13] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. Blair, A. Friday, The climate impact of ICT: A review of estimates, trends and regulations, arXiv (2021) 1–22. URL: http://arxiv.org/abs/2102.02622. arXiv:2102.02622.

[14] J. Malmodin, D. Lundén, The energy and carbon footprint of the global ICT and E & M sectors 2010-2015, Sustainability (Switzerland) 10 (2018). doi:10.3390/su10093027.
[15] L. McMahon, Costing the Earth, Technical Report, Contact Policy Connect, 2018. URL: https://www.policyconnect.org.uk/appccg/sites/site_appccg/files/report/572/fieldreportdownload/isstayingonlinecostingtheearth.pdfhttps://www.policyconnect.org.uk/appccg/research/staying-online-costing-earth.
[16] A. Andrae, T. Edler, On Global Electricity Usage of Communication Technology: Trends to 2030, Challenges 6 (2015) 117–157. doi:10.3390/challe6010117.
[17] M. Willis, J. Hanna, E. Encinas, J. Auger, Low power web: Legacy design and the path to sustainable net futures, in: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–14.
[18] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shojafar, A. I. A. Ahmed, S. A. Madani, K. Saleem, J. J. Rodrigues, A survey on energy estimation and power modeling schemes for smartphone applications, International Journal of Communication Systems 30 (2017) 1–23. doi:10.1002/dac.3234.
[19] M. F. Tuysuz, Z. K. Ankarali, D. Gözüpek, A survey on energy efficiency in software defined networks, Computer Networks 113 (2017) 188–204. doi:10.1016/j.comnet.2016.12.012.
[20] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, K. Pentikousis, Energy-efficient cloud computing, Computer Journal 53 (2010) 1045–1051. doi:10.1093/comjnl/bxp080.
[21] A. Kipp, T. Jiang, M. Fugini, I. Salomie, Layered Green Performance Indicators, Future Generation Computer Systems 28 (2012) 478–489. URL: http://dx.doi.org/10.1016/j.future.2011.05.005. doi:10.1016/j.future.2011.05.005.
[22] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J. M. Pierson, A. V. Vasilakos, Cloud computing: Survey on energy efficiency, ACM Computing Surveys 47 (2015). doi:10.1145/2656204.
[23] Z. Li, S. Tesfatsion, S. Bastani, A. Ali-Eldin, E. Elmroth, M. Kihl, R. Ranjan, A Survey on Modeling Energy Consumption of Cloud Applications: Deconstruction, State of the Art, and Trade-Off Debates, IEEE Transactions on Sustainable Computing 2 (2017) 255–274. URL: http://ieeexplore.ieee.org/document/7967771/. doi:10.1109/TSUSC.2017.2722822.
[24] C. Preist, D. Schien, P. Shabajee, Evaluating sustainable interaction design of digital services: The case of YouTube, Conference on Human Factors in Computing Systems - Proceedings (2019) 1–12. doi:10.1145/3290605.3300627.
[25] François Zaninotto, Argos: Measure the carbon footprint of software, improve developer practices, 2020. URL: https://marmelab.com/blog/2020/11/26/argos-sustainable-development.html/.
[26] EPA, Energy Star, Office Equipment, 2012. URL: https://www.energystar.gov/products/office_equipment.
[27] S. Georgiou, D. Spinellis, Energy-delay investigation of remote inter-process communication technologies, Journal of Systems and Software 162 (2020) 110506. URL: https://www.sciencedirect.com/science/article/pii/S0164121219302808. doi:https://doi.org/10.1016/j.jss.2019.110506.
[28] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, Quantitative analysis of real-time

systems using priced timed automata, Communications of the ACM 54 (2011) 78–87.
[29] K. G. Larsen, P. Pettersson, W. Yi, Uppaal in a nutshell, International journal on software tools for technology transfer 1 (1997) 134–152.
[30] N. Guermouche, C. Godart, Timed model checking based approach for web services analysis, in: 2009 IEEE International Conference on Web Services, IEEE, 2009, pp. 213–221.
[31] L. Aceto, A. Morichetta, F. Tiezzi, Decision support for mobile cloud computing applications via model checking, in: 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, IEEE, 2015, pp. 199–204.
[32] L. Aceto, K. G. Larsen, A. Morichetta, F. Tiezzi, A cost/reward method for optimal infinite scheduling in mobile cloud computing, in: C. Braga, P. C. Ölveczky (Eds.), Formal Aspects of Component Software, Springer International Publishing, Cham, 2016, pp. 66–85.