Report from Dagstuhl Seminar 22341

Power and Energy-Aware Computing on Heterogeneous Systems (PEACHES)

Kerstin I. Eder[*1], Timo Hönig[*2], Daniel Mosse[*3], Max Plauth[*4], and Maja Hanne Kirkeby[†5]

1 **University of Bristol, GB.** cskie@bristol.ac.uk
2 **Ruhr-Universität Bochum, DE.** timo.hoenig@rub.de
3 **University of Pittsburgh, US.** mosse@cs.pitt.edu
4 **Hasso-Plattner-Institut, Universität Potsdam, DE.** max.plauth@hpi.de
5 **Roskilde University, DK.** majaht@ruc.dk

Abstract

This report documents the program and outcomes of the Dagstuhl Seminar 22341 – Power and Energy-Aware Computing on Heterogeneous Systems (PEACHES). The seminar was held on Aug 21 – Aug 26, 2022, and brought together 35 international experts from different domains across the entire system stack – from system designers to programmers and operators. We present the abstracts of 18 talks and 5 summaries of discussions and active sessions on the principal topic areas: Energy Transparency from Hardware to Software, Energy Optimisation and Management, Computing for Sustainability, Green Computing Hackathon, and Disruptive Paradigms.

Seminar August 21–26, 2022 – http://www.dagstuhl.de/22341

2012 ACM Subject Classification Computer systems organization → Heterogeneous (hybrid) systems; General and reference → Evaluation; General and reference → Design; General and reference → Measurement; General and reference → Metrics; Hardware → Power and energy; Software and its engineering → Operating systems

Keywords and phrases energy, heterogeneous computing, operating systems, power, systems

Digital Object Identifier 10.4230/DagRep.12.8.31

1 Executive Summary

Kerstin I. Eder (University of Bristol, GB)
Timo Hönig (Ruhr-Universität Bochum, DE)
Daniel Mosse (University of Pittsburgh, US)
Max Plauth (Hasso-Plattner-Institut, Universität Potsdam, DE)

More than ever, emissions, carbon footprint, and other related environmental concerns are at the forefront of society, from several different perspectives. There is an urgent need to understand how **computing** fits into the broader picture of our planet's energy consumption and what is the role of computing in reducing our carbon footprint worldwide. This requires new ways of thinking across different domains, and necessitates highly energy-efficient hardware and software designs that adapt to changing operating conditions to become more efficient. Collaboration is increasingly required across the entire system stack – from system designers to programmers and operators.

[*] Editor / Organizer
[†] Editorial Assistant / Collector

Power and Energy-Aware Computing on Heterogeneous Systems (PEACHES), *Dagstuhl Reports*, Vol. 12, Issue 8, pp. 31–59
Editors: Kerstin I. Eder, Timo Hönig, Daniel Mosse, Max Plauth, and Maja Hanne Kirkeby
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The Dagstuhl Seminar 22341 on Power and Energy-Aware Computing on Heterogeneous Systems (PEACHES) brought together experts from computer science and computer engineering that share a common vision towards reducing carbon emissions both using innovative designs for computing systems and techniques that bridge the gap between hardware and software, as well as using computing systems to manage other environment-influencing systems. Five principal topic areas were discussed in working groups during the meeting: Energy transparency from hardware to software, Energy optimisation and management, Sustainability in computing, "Green Computing" hackathons, and Disruptive paradigms.

This report documents the program and the outcomes of PEACHES.

2 Table of Contents

3 Overview of Talks

3.1 Adaptive Optimization of (some) Parallel Applications

Antonio Carlos Schneider Beck Filho (Federal University of Rio Grande do Sul, BR)

Joint work of Arthur F. Lorenzon, Charles C. de Oliveira, Jeckson D. Souza, Antonio Carlos S. Beck
Main reference Arthur Francisco Lorenzon, Charles Cardoso De Oliveira, Jeckson Dellagostin Souza, Antonio Carlos Schneider Beck: "Aurora: Seamless Optimization of OpenMP Applications", IEEE Trans. Parallel Distributed Syst., Vol. 30(5), pp. 1007–1021, 2019.
URL https://doi.org/10.1109/TPDS.2018.2872992

Efficiently exploiting thread-level parallelism has been challenging for software developers. As many parallel applications do not scale as the number of cores increases, the task of rightly choosing the ideal configuration (number of threads and/or DVFS level and/or thread/page mapping) to produce the best results in terms of performance and/or energy is not straightforward. In this talk, I show a solution that is transparent (does not demand changes in the original code) and is adaptive (automatically adjusts to applications at runtime). However, to achieve such levels of adaptability and transparency, our optimization is limited to some applications only: those implemented with OpenMP. This is the price to pay when it comes to adaptability and energy consumption.

3.2 Energy Automation: What do people want from?

Ruzanna Chitchyan (University of Bristol, GB)

Joint work of Jan Marc Schwidtal, Proadpran Piccini, Matteo Troncia, Ruzanna Chitchyan, Mehdi Montakhabi, Christina Francis, Anna Gorbatcheva, Timothy Capper, Mustafa A. Mustafa, Merlinda Andoni, Valentin Robu, Mohamed Bahloul, Ian Scott, Tanaka Mbavarira, Juan Manuel Espana, Lynne Kiesling
Main reference Jan Marc Schwidtal, Proadpran Piccini, Matteo Troncia, Ruzanna Chitchyan, Mehdi Montakhabi, Christina Francis, Anna Gorbatcheva, Timothy Capper, Mustafa A. Mustafa, Merlinda Andoni, Valentin Robu, Mohamed Bahloul, Ian Scott, Tanaka Mbavarira, Juan Manuel Espana, Lynne Kiesling: "Emerging Business Models in Local Energy Markets: A Systematic Review of Peer-To-Peer, Community Self-Consumption, and Transactive Energy Models" SSRN, 8 Mar 2022.
URL https://doi.org/10.2139/ssrn.4032760

The different Peer-To-Peer, Community Self-Consumption, and Transactive Energy Models gives rise to new configurations of business models for local energy trading among a variety of actors. Pragmatically, as software engineers, we must view social, technical, and environmental concerns as closely interrelated. Neither of these dimensions can be ignored in the software project and product. It is a challenge to develop software tools, methods, and applications that remedy the environmental impact of human activity while improving or maintaining the social and economic standing of the system stakeholders. Inevitably, this leads to a socio-technical systems engineering approach, where focus on the human and technical elements are equally important.

(Edited by: Maja Hanne Kirkeby, Collector).

3.3 Towards a hybrid (static and statistical) worst-case execution time and worst case energy consumption estimation

Liliana Cucu-Grosjean (INRIA – Paris, FR)

Joint work of Liliana Cucu-Grosjean, Marwan Wehaiba el Khazen, Hadrien Clarke, Kevin Zagalo, Adriana Gogonel, Yves Sorel, Avner Bar Hen, Yasmina Abdeddaim, Slim Ben Amor, Kossivi Kougblenou, Rihab Bennour
Main reference Marwan Wehaiba el Khazen, Kevin Zagalo, Hadrien Clarke, Mehdi Mezouak, Yasmina Abdeddaïm, Avner Bar-Hen, Slim Ben-Amor, Rihab Bennour, Adriana Gogonel, Kossivi Kougblenou, Yves Sorel, Liliana Cucu-Grosjean: "Work in Progress: KDBench – towards open source benchmarks for measurement-based multicore WCET estimators", in Proc. of the 28th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2022, Milano, Italy, May 4-6, 2022, pp. 309–312, IEEE, 2022.
URL https://doi.org/10.1109/RTAS54340.2022.00035

Since the work of Edgar and Burns in 2000s [1], the real-time community has showed increased interest in using statistical estimator for the problem of the worst-case execution time estimation (WCET) of programs on embedded processors. We start this talk by building a common vocabulary on the real-time notions like deadline, release and worst-case execution time while scheduling algorithms are underlined by priority-based. The presented figures are obtained from measurements of an open data benchmark calls PX4-RT [2], while the data are collected either real flights or Gazebo-based simulation. We remind that for the sake of the reproducibility such information is important and their lack is one identified drawback of existing work on statistical methods for the WCET estimation. We then present the two main classes of WCET estimators: static analysis based and measurement-based and we present a statistical WCET definition to allow hybridizing these two classes, while the common understanding of WCET as a bound stays coherent. We present a hybrid WCET estimator mixing statistical and static analyses. Results indicate that this facilitates the identification of relevant paths as well as the construction of a WCET bound for complex systems. This is explained by the fact that the probabilistic description of the WCET bound is an excellent basis for time composabillity.

We present the results of a commercial tool, RocqStat, implementing the presented hybrid estimator and conclude by providing our current stream of work of using hybrid estimators for the identification of relevant paths for the problem of worst-case energy consumption.

References

1 Robert I. Davis and Liliana Cucu-Grosjean. *A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems*. Leibniz Trans. Embed. Syst., 6(1):03:1–03:60, 2019
2 Marwan Wehaiba el Khazen et al. *Work in Progress: KDBench – towards open source benchmarks for measurement-based multicore WCET estimators*. IEEE Real-time systems and applications symposium, 2022

3.4 Performance Isolation for Power-Limited CPUs

Mathias Gottschlag (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Mathias Gottschlag, Philipp Machauer, Yussuf Khalil, Frank Bellosa
Main reference Mathias Gottschlag, Philipp Machauer, Yussuf Khalil, Frank Bellosa: "Fair Scheduling for AVX2 and AVX-512 Workloads", in Proc. of the 2021 USENIX Annual Technical Conference, USENIX ATC 2021, July 14-16, 2021, pp. 745–758, USENIX Association, 2021.
URL https://www.usenix.org/conference/atc21/presentation/gottschlag

Since the breakdown of Dennard scaling, modern CPUs have become power limited to the point where they have to reduce their frequency when executing power-intensive code. This behavior poses a problem for performance isolation: When one task executes power-intensive instructions such as AVX2 or AVX-512, the resulting frequency reduction often affects other less power-intensive tasks.

We propose modifying the CPU accounting of existing fair schedulers to counteract this performance impact. We allocate more CPU time to low-power tasks according to the frequency reduction during execution of these tasks. While the resulting scheduling policy greatly improves performance isolation for many workloads, some workloads present a challenge as the CPU does not provide sufficient information on the power characteristics of individual tasks.

3.5 Security costs Energy – because we're doing it wrong!

Daniel Gruss (TU Graz, AT)

Joint work of Jonas Juffinger, Lukas Lamster, Andreas Kogler, Maria Eichlseder, Moritz Lipp, Daniel Gruss
Main reference J. Juffinger, L. Lamster, A. Kogler, M. Eichlseder, M. Lipp, D. Gruss: "CSI:Rowhammer – Cryptographic Security and Integrity against Rowhammer", in Proc. of the 2023 2023 IEEE Symposium on Security and Privacy (SP) (SP), pp. 236–252, IEEE Computer Society, 2023.
URL https://doi.org/10.1109/SP46215.2023.00014

As we are running into a global energy crisis, saving energy in ICT is more important than ever. However, today we just patch security on top of system designs – an approach that inherently introduces performance and energy overheads. The Meltdown patch alone caused performance overheads of roughly 5%, meaning an overhead on greenhouse-gas emissions of up to 0.09% in 2018, a similar single patch in 2030, would cause an overhead of up to 0.4%. This is unsustainable and forces us to rethink security and question how we design systems. In the talk, we argue that the curve we use to optimize systems goes from reliable to unreliable to completely unusable where the system freezes and crashes so frequently that it is not useful for any task. We then argue that we need to rethink how we approach these problems and introduce cryptography-grade error detection combined with correction mechanisms to adjust this curve to make it continuous such that optimizing *too far* leads only to a loss in performance but never to an uncorrected system error or silent data corruption. If we achieve this, we can optimize for the sweet spot of system efficiency, that has been far out of our reach so far, and while increasing the security of the system.

3.6 Turning the knobs – Automatically determine energy-efficient process configurations and clock frequencies on Linux

Benedict Herzog (Ruhr-Universität Bochum, DE), Sven Köhler (Hasso-Plattner-Institut, Universität Potsdam, DE)

Operating systems offer numerous configuration parameters to tune the system behavior in general and the energy efficiency in particular. One keystone for an energy-efficient system is the right configuration tailored to the currently running application. Finding the right configuration, however, proves a challenging task. We independently pursued and developed two different approaches, Polar and memutil, to automatically find such an energy-efficient configuration.

Polar is based on a neural network receiving the application profile as input and provides an energy-efficient configuration as output. With this blackbox approach we found that the average energy efficiency (in terms of the energy delay-squared) can be improved by 11.5% for typical applications.

In an independent work, we implemented memutil – a Linux CPU frequency governor – automatically adapting each core's clock frequency based on live readings from performance measurement units. The most promising heuristic we found to cut idling cycles on high clock frequencies was the number of L2 cache misses. Using a linear frequency interpolation, memutil reduces reduce the energy demand of all tested benchmarks compared to the default governor at minimal execution time penalty of all tested benchmarks from the NPB suite compared to the default.

Although started with different assumptions and application scenarios Polar and memutil show comparable insights and foster future investigation and collaboration.

3.7 Energy-awareness Amplifies Side Channels

Henry Hoffmann (University of Chicago, US)

Joint work of Tejas Kannan, Henry Hoffmann
Main reference Tejas Kannan, Henry Hoffmann: "Protecting adaptive sampling from information leakage on low-power sensors", in Proc. of the ASPLOS '22: 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 28 February 2022 – 4 March 2022, pp. 240–254, ACM, 2022.
URL https://doi.org/10.1145/3503222.3507775

This talk examines recent work on energy-awareness in sensing systems. This includes both adaptive sampling and adaptive neural networks. Both techniques save energy by adapting execution to inputs or sequences of inputs. Adapting sampling reduces energy by intelligently determining when to take a sample, saving energy through reduced usage of both sensor and radio. Adaptive neural networks save energy by exiting the network early when additional computation is unlikely to affect accuracy.

Unfortunately, both approaches leak information about the inputs (i.e., to either the sensor or the neural network). This talk is meant to spur discussions about these privacy/energy tradeoffs and discuss potential solutions to the problem.

3.8 Interoperability of Energy-aware Systems

Henry Hoffmann (University of Chicago, US)

Main reference Henry Hoffmann: "JouleGuard: energy guarantees for approximate applications", in Proc. of the 25th Symposium on Operating Systems Principles, SOSP 2015, Monterey, CA, USA, October 4-7, 2015, pp. 198–214, ACM, 2015.
URL https://doi.org/10.1145/2815400.2815403

Energy-awareness has become an important topic and has been addressed by researchers working at various levels of the system stack. Energy-aware solutions adjust parameters at their level of the stack to meet energy constraints or optimize energy efficiency. It has recently been observed that solutions working at different levels can interfere with each other, reducing both performance and energy efficiency. One solution for this interference is preventing individual solutions and instead having a central energy-aware authority that manages the options available at all levels simultaneously.

This talk highlights the benefits of coordinating energy-aware solutions across the system stack. It also points out drawbacks of existing, centralized approaches (e.g., [2, 1, 3, 4]). It then highlights several challenges to be addressed to achieve a modular, interoperative, and composable energy-aware system stack.

References

1 H. Hoffmann, "JouleGuard," in Proceedings of the 25th Symposium on Operating Systems Principles, Oct. 2015, pp. 198–214, doi: 10.1145/2815400.2815403.
2 M. Maggio, A. V. Papadopoulos, A. Filieri, and H. Hoffmann, "Automated control of multiple software goals using multiple actuators," in Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, Aug. 2017, pp. 373–384, doi: 10.1145/3106237.3106247.
3 C. Wan, M. H. Santriaji, E. Rogers, H. Hoffmann, M. Maire, and S. Lu, "ALERT: Accurate Learning for Energy and Timeliness," in 2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020, 2020, pp. 353–369, [Online]. Available: https://www.usenix.org/conference/atc20/presentation/wan.
4 A. Filieri, H. Hoffmann, and M. Maggio, "Automated multi-objective control for self-adaptive software design," in Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, Aug. 2015, pp. 13–24, doi: 10.1145/2786805.2786833.

3.9 Secure Energy-Aware Operating Systems

Henriette Hofmeier (Ruhr-Universität Bochum, DE), Benedict Herzog (Ruhr-Universität Bochum, DE), and Timo Hönig (Ruhr-Universität Bochum, DE)

Software mitigations of hardware vulnerabilities come with costs in terms of runtime and energy. Especially, when an application heavily interacts with the operating system, the mitigation-induced costs can exceed 25% for the Meltdown and Spectre vulnerabilities. Similar results can be observed when disabling SMT/Hyperthreading to mitigate, for example, Microarchitectural Data Sampling (MDS) cross-HyperThread attacks. Hence, it is worth to analyse on the one side whether an application requires the full protection by the mitigations and if not disable them dynamically. On the other side the appropriate mitigation can be

selected depending on the current hardware platform and system state for which we propose an approach. So questions that remain are:

- What are the potentials for such an approach?
- How to determine the protection requirements of an application?
- Who's in control?
- Is it beneficial to move the configuration from hardware into the operating system?

3.10 The (in)efficiency of the Internet

Romain Jacob (ETH Zürich, CH)

Joint work of Romain Jacob, Laurent Vanbever
Main reference Romain Jacob, Laurent Vanbever: "The Internet of tomorrow must sleep more and grow old". In HotCarbon, 2022.
URL https://hotcarbon.org/pdf/hotcarbon22-jacob.pdf

Today, the ICT industry has a massive carbon footprint (a few percent of the worldwide emissions) and one of the fastest growth rates. The Internet accounts for a large part of that footprint while being also energy inefficient; i.e., the total energy cost per byte transmitted is very high. Thankfully, there are many ways to improve on the current status; we discuss two relatively unexplored directions in this paper. Putting network devices to "sleep," i.e., turning them off, is known to be an efficient vector to save energy; we argue that harvesting this potential requires new routing protocols, better suited to devices switching on/off often, and revising the corresponding hardware/software co-design. Moreover, we can reduce the embodied carbon footprint by using networking hardware longer, and we argue that this could even be beneficial for reliability! We sketch our first ideas in these directions and outline practical challenges that we (as a community) need to address to make the Internet more sustainable.

3.11 Embodied Carbon, ICT's dirty little secret

Alex Jones (University of Pittsburgh, US)

First steps to address the challenge of sustainable computing are naturally to consider the energy efficiency of information and communication technologies (ICT) during their use phase (i.e., after they are deployed into service). This includes reducing energy consumption in processors, memory systems, peripheral devices, cooling systems and a host of other components that are used in deployed systems. However, for computing to be truly sustainable, all phases of the system life-cycle, from manufacturing to disposal, must be considered. In particular, there is limited awareness to the considerable fraction of the total life-cycle from "embodied" energy and carbon impacts of computing systems from the fabrication of the integrated circuits (ICs) that are used in those devices. These impacts can be predicted from "life-cycle assessment" techniques. Using tools like GreenChip it is possible to holistically evaluate energy consumption and other environmental impacts from computing. Using these tools, I show examples of how machine learning applications and in-memory compression can impact design choices for systems. From this we can find disruptive new ways to think about sustainability and even conservation when designing next generation ICT.

3.12 Task scheduling: What should we aim for in terms of reducing energy consumption?

Julia Lawall (INRIA – Paris, FR)

Joint work of Julia Lawall, Himadri Chhaya-Shailesh, Jean-Pierre Lozi, Baptiste Lepers, Willy Zwaenepoel, Gilles Muller
Main reference Julia Lawall, Himadri Chhaya-Shailesh, Jean-Pierre Lozi, Baptiste Lepers, Willy Zwaenepoel, Gilles Muller: "OS scheduling with nest: keeping tasks close together on warm cores", in Proc. of the EuroSys '22: Seventeenth European Conference on Computer Systems, Rennes, France, April 5 – 8, 2022, pp. 368–383, ACM, 2022.
URL https://doi.org/10.1145/3492321.3519585

The task scheduler decides when each task will run, and on which core, and thus can impact the machine energy consumption. In the context of large Intel servers (the case of a 2-socket Intel 5218 was illustrated in the talk), we first observe that making the machine fully idle saves more energy than leaving one thread running somewhere on the machine, due to the ensuing uncore costs. Thus, running an application faster, to finish sooner, may reduce energy consumption. We thus present the Nest scheduler, published at EuroSys 2022 [1], that concentrates tasks on a minimal number of cores. Nest make these cores show higher utilization and thus achieve higher core frequencies, causing the application to finish sooner. Finally, we next study schedutil, Linux's new power governor for reducing energy consumption, and illustrate how, on tasks that frequently sleep for short periods of time due to synchronization, it greatly increases execution time without saving energy. We then wonder how to move forward. Should the operating system impose strategies that impact energy consumption on the hardware, which does not seem to be very successful in the schedutil case, or should it try to work with hardware features, as in Nest?

References

1 Julia Lawall, Himadri Chhaya-Shailesh, Jean-Pierre Lozi, Baptiste Lepers, Willy Zwaenepoel, and Gilles Muller. OS scheduling with Nest: keeping tasks close together on warm cores. In *EuroSys*, pages 368–383. ACM, 2022.

3.13 Minimizing the energy consumption of Federated Learning on heterogeneous devices

Laércio Lima Pilla (University of Bordeaux, FR)

Main reference Laércio Lima Pilla: "Optimal Task Assignment for Heterogeneous Federated Learning Devices", in Proc. of the 35th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2021, Portland, OR, USA, May 17-21, 2021, pp. 661–670, IEEE, 2021.
URL https://doi.org/10.1109/IPDPS49936.2021.00074

Federated Learning is a distributed machine learning technique focused on data privacy and security. In a nutshell, Federated Learning involves a group of heterogeneous devices working together to iteratively train a machine learning model under the coordination of a central server. The server chooses a subset of devices for training and sends them the model's weights for the round. These devices train the model with their own data (which is never shared) and send the updated weights to the server, which averages them before the next training round. The energy consumption of Federated Learning devices is a subject of interest

both for environmental reasons and due to the limited energy available on battery-powered devices. In this context, this talk discussed some of the efforts behind performance and energy optimizations on Federated Learning models, including a new idea for an optimal scheduling algorithm for minimizing the energy consumption of Federated Learning devices when controlling how much data each device should use for training.

3.14 Power-Aware Computing at Scale

Frank Mueller (North Carolina State University – Raleigh, US)

This talk focuses on power and energy control for cloud and high-performance computing facilities. It promotes hierarchical control systems dynamically adapting to application characteristics in a multi-tenant environment. Controls need to coordinate constraints at application/job level as well as center level to balance multiple objectives while exploiting heterogeneous memory and compute resources to the best of their ability. This leads to a number of challenges with open problems regarding trade-offs between objectives such as energy, performance, QoS, sustainability and profitability subject to future work.

References

1 *Uncore Power Scavenger: A Runtime for Uncore Power Conservation on HPC Systems* by Neha Gholkar, Frank Mueller, Barry Rountree, in Supercomputing (SC), Nov 2019.
2 *PShifter: Feedback-based Dynamic Power Shifting within HPC Jobs for Performance* by Neha Gholkar, Frank Mueller, Barry Rountree in High-Performance Parallel and Distributed Computing (HPDC), Jun 2018, pages 106-117.
3 *Power Tuning HPC Jobs on Power-Constrained Systems* by Neha Gholkar, Frank Mueller, Barry Rountree in International Conference on Parallel Architecture and Compilation Techniques (PACT), Sep 2016.
4 *PEARS: A Performance-Aware Static and Dynamic Framework for Heterogeneous Memory* by Onkar Patil, Latchesar Ionkov, Jason Lee, Frank Mueller, Michael Lang, in International Symposium on Memory Systems (MEMSYS), Oct 2021.
5 *NVM-based energy and cost efficient HPC clusters* by Onkar Patil, Latchesar Ionkov, Jason Lee, Frank Mueller, Michael Lang, in International Symposium on Memory Systems (MEMSYS), Oct 2021.
6 *Performance characterization of a DRAM-NVM hybrid memory architecture for HPC applications using Intel Optane DC Persistent Memory Modules* by Onkar Patil, Latchesar Ionkov, Jason Lee, Frank Mueller, Michael Lang, in International Symposium on Memory Systems (MEMSYS), Sep/Oct 2019.

3.15 Power Resilient NextG Data Centers

Simon Peter (University of Washington – Seattle, US)

This talk proposes to build and evaluate a power control plane (PCP) for NextG data centers that operate under tight and variable power envelopes. PCP can control power demand at a fine granularity and over short timescales by making it software-defined. The key is

to gracefully trade off power and quality of service over time. This allows PCP to shed or consolidate load to less power-intensive processors to conserve power during a power event. I show a prototype power resilient distributed file system (DFS) that establishes the viability of the idea. The DFS provides low-latency fail-over and recovery for load shedding events enabling fine-grained load control by PCP. I outline the important research questions that must be answered for PCP to become practical.

3.16 On energy awareness in NVRAM-based operating systems – NEON and PAVE

Wolfgang Schröder-Preikschat (Universität Erlangen-Nürnberg, DE), Timo Hönig (Ruhr-Universität Bochum, DE)

Joint work of Wolfgang Schröder-Preikschat, Timo Hönig, Jörg Nolte

This abstract deals with the recently launched projects NEON (non-volatility in energy-aware operating systems) and PAVE (power-fail aware byte-addressable virtual non-volatile memory), both of which have byte-addressable non-volatile main memory (NVRAM) at their core. Motivation is, on the NEON side, an energy-aware operation of a computing system by using modern NVRAM technology for the operating system itself, namely (1) to save power and (2) to survive power failures, and, on the PAVE side, an operating-system abstraction that should pave the way for a scalable and fail-safe execution of programs (esp. legacy software) virtually directly in NVRAM.

An advantage of the emerging NVRAM technology is the ability to eliminate the need for conventional persistence measures within an operating system and the thus resulting reduction in its space, time and energy requirements as well as a smaller attack surface. Further benefits of NVRAM are its higher speed compared to conventional storage, its rather high capacity compared to conventional DRAM and its ability to keep its state persistently without energy costs.

Unfortunately, today's CPU cannot (yet) do without volatile memory when registers and caches are considered. Furthermore, NVRAM is much slower than conventional main memory technology such as DRAM and changing the persistent state in NVRAM by writing results in more power consumption than corresponding state changes in DRAM [2]. In addition, fail-safe guarantees are now required from the system, since power failures when writing to the NVRAM, for example, can lead to control flows that unexpectedly convert a sequential process into a non-sequential one [3]. These problems can be solved by an operating system by (1) a suitable event-based, sporadically triggered checkpointing mechanism integrated in its exception- handling subsystem and (2) a suitable integration of NVRAM into the memory hierarchy via its virtual-memory subsystem.

The idea is that a trap or power failure interrupt (PFI) results in a micro-checkpoint request that is handled with strict time guarantee in the operating system: This checkpoint event basically preempts the running process from its volatile environment into the NVRAM. The specified *residual energy window* as a characteristic feature of the power-supply unit (PSU) determines the upper time limit for this procedure [1], the *worst-case execution time* (WCET) of which must never exceed it. In program areas where this mechanism cannot be used, particularly for the backup procedure itself, transactional programming comes into play.

A main problem are *non-maskable interrupt* (NMI) nesting and critical sections that block an *interrupt request* (IRQ): both endanger the timely handling of a possible checkpoint request. Respective sections are localised by static program analysis and then rearranged based on program transformation tools so that IRQ locks are either eliminated or at least canceled again in good time. This measure ultimately allows also the intended elimination of the persistence measures in the operating system that are superfluous due to NVRAM. For memory-hierarchy integration appropriate is a two-level hierarchy of software-managed caches that uses NVRAM as a buffer for data in conventional storage and DRAM as a buffer for NVRAM pages. The buffering of the pages in the DRAM is subject to strict time guarantees so that this logically persistent data can be reliably consolidated in the NVRAM again in exceptional cases. That is, in order to survive power failures, the maximum size of this DRAM page cache is to be aligned to the size of the remaining energy window in the power supply. Last but not least, energetic methods are designed that improve the mutual interaction of the operating system and NVRAM in such a way that the persistence properties of NVRAM are used to increase energy efficiency. Static NVRAM sleep modes are provided that actively reduce power consumption orthogonally to dynamic runtime improvements by NVRAM governors in the operating system.

In summary, the starting point of both projects is existing knowledge about NVRAM-based persistence of intermittently powered mobile devices on the one hand and persistence measures in stationary computing systems on the other. The basic assumption is that an NVRAM-based operating system manages computing more efficiently than a functionally identical DRAM-based twin. Subject of the study is the specific relationship between energy efficiency, computing power and latency in operating-system variants based on Linux, as far as NEON is concerned, and NVRAM-based capacity scaling and NVRAM virtualisation in FreeBSD, as PAVE contribution. Building on this, the general relationship in terms of applicability, transferability, and generalisation of the techniques developed for NEON and PAVE are examined.

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Individual Research Grants 465958100 (NEON), 501993201 (PAVE), and 502228341 (Memento) as part of the Priority Programme on Disruptive Memory Technologies (SPP 2377).

References

1 Dushyanth Narayanan, Orion Hodson. *Whole-System Persistence*. ASPLOS XVII: Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems, ACM, 2012

2 Ivy B. Peng, Maya B. Gokhale, Eric W. Green. *System Evaluation of the Intel Optane Byte-addressable NVM*. MEMSYS '19: Proceedings of the International Symposium on Memory Systems, ACM, 2019

3 Benjamin Ransford, Brandon Lucia. *Nonvolatile Memory is a Broken Time Machine*. MSPC '14: Proceedings of the workshop on Memory Systems Performance and Correctness, ACM, 2014

3.17 Heterogeneous, High-Performance Serverless Computing: Energy Implications and Opportunities

Devesh Tiwari (Northeastern University – Boston, US)

The next wave of cloud computing – the serverless computing model – is enjoying adoption at scale by different cloud computing vendors. The serverless computing model is already rapidly accelerating the development and deployment of enterprise applications. Unfortunately, the HPC community appears to be left behind in the revolution and it is not clear what are the energy implications/opportunities for HPC community if they adopted the serverless computing model. First, I'll discuss how HPC applications and workflows could attain higher performance and energy efficiency via hybrid execution [1, Mashup]. Second, I'll demonstrate how we can leverage server heterogeneity to reduce the overall energy consumption [2, IcBreaker]. Finally, I will discuss some predictive techniques to mitigate the bottlenecks of serverless computing [3, DayDream], and a brief introduction to a novel AI-workload dataset to enable carbon/aware, sustainable data center computing efforts [4].

References

1 Roy, Rohan Basu, Tirthak Patel, Vijay Gadepally, and Devesh Tiwari. "Mashup: making serverless computing useful for HPC workflows via hybrid execution." In Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), pp. 46-60. 2022.

2 Roy, Rohan Basu, Tirthak Patel, and Devesh Tiwari. "IceBreaker: warming serverless functions better with heterogeneity." In Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 753-767. 2022.

3 Rohan Basu Roy, Tirthak Patel, and Devesh Tiwari. "DayDream: Executing Dynamic Scientific Workflows on Serverless Platforms with Hot Starts." In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC). 2022.

4 Li, Baolin, Rohin Arora, Siddharth Samsi, Tirthak Patel, William Arcand, David Bestor, Chansup Byun et al. "AI-Enabling Workloads on Large-Scale GPU-Accelerated System: Characterization, Opportunities, and Implications." In 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pp. 1224-1237. IEEE, 2022.

3.18 How can we avoid ICT becoming the next Greenpeace target?

Samuel Xavier-de-Souza (Federal University of Rio Grande do Norte, BR)

Joint work of Samuel Xavier-de-Souza, Kerstin Eder

Information & Communication Technologies consume more than 10% of the energy produced on Earth, about twice as much as the aviation industry. According to an article from Nature, this number might rise above 20% by 2030. This talk discussed alternatives that might save the world from the digital revolution without stopping it. We discussed the historical facts that have led us to rely on energy-inefficient computing, the possible paths forward, and necessary actions to revert this. As takeaways we established that computing plays

a significant and active role in the present and future environmental challenges facing the world; that much of what we know and rely about computing changed in mid 2000's; that software, especially parallel software, is now even more important to the course of the digital revolution; and that software operation becomes a key concept to ensure optimisations are not wasted.

4 Working groups

4.1 Disruptive Paradigms

Michael Engel (Universität Bamberg, DE), Ahmed Ali-Eldin Hassan (Chalmers University of Technology – Göteborg, SE), Timo Hönig (Ruhr-Universität Bochum, DE), Alex Jones (University of Pittsburgh, US), Frank Mueller (North Carolina State University – Raleigh, US), and Wolfgang Schröder-Preikschat (Universität Erlangen-Nürnberg, DE)

Power and energy consumption depend on a large number of parameters on the hardware as well as software level of systems as well as on the interaction of hardware and software. With respect to disruptive paradigms, we can consider the impact of disruptions in hardware and software technology on power and energy and, conversely, the impact of focusing on a reduction of power and energy (on local device as well as global scale) on the design of technologies that form the foundations of future systems.

First, we identified a number of recent disruptions in hardware and software which have significant impact on power and/or energy. Trends such as the ubiquitous application of (deep) machine learning algorithms to all sorts of problems as well as the rise of digital currencies and blockchain technology have significantly increased power and energy requirements of typical applications. The impact of machine learning approaches can not only be identified on the training side, but also on the inference side [1]. Applications that make intensive use of machine learning inference, e.g. autonomous cars, exacerbate the problem since all on-board computing systems of autonomous vehicles compete for battery capacity with the electric drive of the unit. Here, it can also be observed that the complexity of related solutions is increasing rapidly, one example that was discussed is Google's TensorFlow software [2].

Especially the increase of software complexity has impacts not only on the local device level, e.g. on the battery runtime of a smartphone, but in turn on the global sustainability of devices. Since modern computing devices are no longer designed for upgradability (or repairability, though at least in the EU regulations related to repairability are currently in the making [5]), the plethora of incoming software updates results in devices being too "weak" – e.g. in terms of computing power or memory – for next year's version of the system and application software. In turn, users are forced to replace devices long before the end of their lifetime due to physical constraints. This is problematic on a global scale since a significant fraction of the CO2 (equivalent) emissions caused by electronic devices is generated by the *production* of devices, in many cases more than 50% of the overall impact. Accordingly, one simple way to reduce the CO2 impact of a device is to use it for a longer time [4].

The extended use of devices, however, is significantly constrained by the current mode in which software updates are handled by the device manufacturer. Devices connected to the Internet require frequent security updates to enable secure (and, in turn, safe) operation.

Even if updates are provided on a regular basis, there is often no way to obtain security updates independent of feature additions. Here, one *disruptive solution* would be to *require the separation of security and feature updates* and allow users to install only selected subsets of updates. However, this leads to an explosion of possible configurations which need to be tested, updated, and provided [6].

An even more disruptive approach is to start from scratch. Instead of trying to debloat software, as discussed elsewhere in this seminar, a renewed concentration on the development of *lean software*, as postulated by Niklaus Wirth already in 1995 [3], might be an approach to solve this side of the software crisis. This is also based on the observation that, as an extension of the Pareto principle, a significant fraction of the features of a software product are rarely or never used. For example, a DuPont study mentioned in an Agile 2002 keynote [7] concluded that "only 25% of a system's features are ever needed".

Another application of the Pareto principle was the basis of a followup discussion on power and energy optimization. Even though it is well known from software optimizations for performance, the question *which parts of the system to optimize* should be considered early on. Accordingly, the selection of optimizations to apply should not be based on the largest possible gain in the respective component, but on the frequency of use of a component. As with other non-functional properties, small gains in the reduction of energy consumption achieved on a hot path are more important than large gains that are only relevant for a small number of rarely executed corner cases.

When discussing the hardware-software interface, one central question is how to distribute the implementation of functionality between hardware and software components. Here, hardware can be more power and energy efficient due to customization to a specific problem as well as the larger grade of possible parallelism compared to software. Typical accelerators for often used functionality are in use in commodity systems for more than a decade, e.g. for video de- and encoding, general signal processing, cryptography, and neural networks. On the hardware-software interface layer, instruction set extensions provide a tradeoff between efficiency and flexibility, e.g. by the addition of DSP or vector instructions to conventional CPU instruction sets.

On the side of computer architecture, the possible power and energy impact of some recent developments was discussed. A significant contributor are embedded microcontroller chips. While the power and energy consumption of individual chips is low, the large number of deployed devices (6.7 billion ARM-based chips alone in the fourth quarter of 2020 [14]) make microcontrollers an interesting target for disruptive optimizations. The recently introduced Raspberry Pi Pico 2040 microcontroller does not follow the common approach to integrate flash memory for persistent storage on chip, but rather uses an external flash chip. Since integrating flash on-chip requires the use of relatively coarse semiconductor feature sizes, this also constrains the power scaling effects described by Dennard [15], whereas taking flash off-chip enables the downscaling of the remaining controller circuit. By coincidence, this also removed the reliance on specific manufacturing capacities that are in high demand during the current semiconductor supply crisis, making the Pico 2040 one of the few microcontrollers without significant supply constraints. Accordingly, the disaggregation of functionality to different chips (on a common carrier using chiplets or in separate packages) might be a – at first counterintuitive – approach to reduce power and energy consumption.

Disruptions related to *memory energy consumption* were also the focus of extended discussions. One significant contributor here is the requirement to periodically refresh DRAM contents. Several approaches to reduce the refresh overhead by using software and hardware approaches have been published recently [16]. A more disruptive approach is the use of

persistent, byte-addressable main memory, which is also the focus of the recently started German coordinated research project on *Disruptive Memory Technologies* [17] and was the subject of a presentation at this seminar [23]. Here, the discussion in the group diverged significantly from power- and energy-related topics to general questions about the hardware and software implications of persistent memory technologies, which are omitted here for brevity and focus.

However, one notable and possibly disruptive idea, related to lean software discussed above, was *cache-only computing*. Here, the idea is that CPU caches in today's systems are so large that significant parts of a software product (e.g., a microkernel of a system, see also the Pareto discussion above) can always remain in cache on a certain level, which could result in a significant reduction of memory traffic. Similar approaches to *cache-locking* of code have been employed in hard real-time systems for some time already [18]; investigating their energy impact will be interesting. Here, the discussion diverged into topics related to *predictable computing*, e.g. through the use of scratchpad memories instead of caches [19] and the creation of predictable computer systems out of unpredictable components, similar to von Neumann's early ideas on building dependable systems from unreliable components [20]. This, in turn, can be of relevance for improved worst-case energy consumption (WCEC) analyses [21], which e.g. are relevant to implement effective and efficient *intermittent computing* for IoT devices [22].

Another interesting *disruptive approach* could be the combination of accelerators and *approximate computing* [8]. It can be shown that in many applications, such as signal processing, perfect numerical precision is not required [9] – especially if the noise introduced by the signals to be processed is larger than the noise introduced by the computational approximation. The disruptive factor here could be to revisit analog computing components as parts of function-specific accelerators [10] and the introduction of hybrid digital/analog ("mixed signal") circuits. This approach can result in significant reductions of energy and power consumption, since certain computationally complex operations can be implemented by exploiting physical effects of electronic components. For example, integration is a numerically complex operation which is intrinsically implemented by a capacitor.

This move away from digital computing can be taken even further. A recent research direction is to attempt building computing systems that do not run on electricity, but instead are based on (synthetic) biological circuits [11]. For example, it is often stated that the human brain only consumes about 20 W of power while (sometimes) providing capabilities unmatched by any digital computer. First approaches to model brains using complex digital electronic systems are already in existence [12], while the development, programming, and integration of biological circuits into digital systems still seems to be a bit further away. Orthogonal to biological computing, implementing *storage using biological components such as DNA* [13] could be another disruptive approach.

In summary, the workgroup on disruptive paradigms discussed a large number of approaches, which in turn shows the vast size of the possible design space related to power and energy efficiency. Some of the discussed topics, such as software complexity and upgradability, also made the connection between localized, short-term power and energy effects and long-term global sustainability challenges. Overall, a single localized solution will not be sufficient – one major takeaway of the discussions in the related session is that a combination and cooperation of disruptive approaches on all levels of the hardware and software stack is required in order to have a positive impact on power and energy consumption. However, we have also noticed that "blind" optimizations without considering Pareto effects can be wasted efforts. Accordingly, power and energy are system-wide challenges which have to be optimized using carefully coordinated approaches.

References

1 Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. *Compute and energy consumption trends in deep learning inference*. arXiv preprint arXiv:2109.05472, 2021

2 Zejun Zhang, Yanming Yang, Xin Xia, David Lo, Xiaoxue Ren, and John Grundy. *Unveiling the mystery of API evolution in deep learning frameworks: a case study of Tensorflow 2*. In Proceedings of the 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '21). IEEE Press, 238–247. https://doi.org/10.1109/ICSE-SEIP52600.2021.00033

3 Niklaus Wirth. *A Plea for Lean Software*. IEEE Computer, vol. 28, no. 02, pp. 64-68, 1995. doi: 10.1109/2.348001

4 Peter Marwedel and Michael Engel. *Plea for a Holistic Analysis of the Relationship between Information Technology and Carbon-Dioxide Emissions*. 23th International Conference on Architecture of Computing Systems 2010, pp. 1-6

5 Sahra Svensson et al. *The emerging 'Right to repair' legislation in the EU and the US*. Proceedings from Going Green–Care Innovation (2018): 27-29

6 Reinhard Tartler, Daniel Lohmann, Julio Sincero, and Wolfgang Schröder-Preikschat, W. *Feature consistency in compile-time-configurable system software: Facing the Linux 10,000 feature problem*. Proceedings of the sixth conference on Computer systems. 2011

7 Martin Fowler – report from Agile 2002, quoting DuPont study `http://martinfowler.com/articles/xp2002.html`

8 Sparsh Mittal. *A survey of techniques for approximate computing*. ACM Computing Surveys (CSUR) 48.4 (2016): 1-33.

9 Andreas Heinig, Vincent J. Mooney, Florian Schmoll, Peter Marwedel, Krishna Palem, and Michael Engel. *Classification-based improvement of application robustness and quality of service in probabilistic computer systems*. In International Conference on Architecture of Computing Systems (pp. 1-12). Springer, Berlin, Heidelberg

10 Glenn ER Cowan, Robert C. Melville, and Yannis P. Tsividis. *A VLSI analog computer/digital computer accelerator*. IEEE Journal of Solid-State Circuits 41.1 (2005): 42-53

11 Stuart A. Kurtz et al. *Biological computing*. Complexity theory retrospective II (1997): 179-195

12 Steve B. Furber et al. *The Spinnaker project*. Proceedings of the IEEE 102.5 (2014): 652-665

13 Yiming Dong et al. *DNA storage: research landscape and future prospects*. National Science Review 7.6 (2020): 1092-1107

14 Arm, Inc. *The Arm ecosystem ships a record 6.7 billion Arm-based chips in a single quarter*. `https://www.arm.com/company/news/2021/02/arm-ecosystem-ships-record-6-billion-arm-based-chips-in-a-single-quarter`

15 Robert H. Dennard et al. *Design of ion-implanted MOSFET's with very small physical dimensions*. IEEE Solid-State Circuits Society Newsletter 12.1 (2007): 38-50

16 Ishwar Bhati et al. *DRAM refresh mechanisms, penalties, and trade-offs*. IEEE Transactions on Computers 65.1 (2015): 108-121

17 Olaf Spinczyk and Jörg Nolte. *Disruptive Memory Technologies – DFG Priority Program 2377*. `https://spp2377.uos.de`

18 Isabelle Puaut, and Alexis Arnaud. *Dynamic instruction cache locking in hard real-time systems*. Proc. of the 14th Int. Conference on Real-Time and Network Systems. 2006

19 Rajeshwari Banakar, Stefan Steinke, Bo-Sik Lee, M. Balakrishnan, and Peter Marwedel. *Scratchpad memory: A design alternative for cache on-chip memory in embedded systems*. In Proceedings of the Tenth International Symposium on Hardware/Software Codesign. CODES 2002 (IEEE Cat. No. 02TH8627) (pp. 73-78)

20 John von Neumann. *Probabilistic logics and the synthesis of reliable organisms from unreliable components*. Automata studies 34 (1956): 43-98

21 Ramkumar Jayaseelan, Tulika Mitra, and Xianfeng Li. *Estimating the worst-case energy consumption of embedded software*. 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06). IEEE, 2006

22 Brandon Lucia et al. *Intermittent computing: Challenges and opportunities*. 2nd Summit on Advances in Programming Languages (SNAPL 2017) (2017)

23 Timo Hönig and Wolfgang Schröder-Preikschat. *On Energy-Awareness in NVRAM-based Operating Systems – NEON and PAVE*. In Power and Energy-aware Computing on Heterogeneous Systems (PEACHES), Dagstuhl Seminar 22341, 2022

4.2 Energy Optimization and Management

Maja Hanne Kirkeby (Roskilde University, DK), Ahmed Ali-Eldin Hassan (Chalmers University of Technology – Göteborg, SE), Liliana Cucu-Grosjean (INRIA – Paris, FR), Benedict Herzog (Ruhr-Universität Bochum, DE), Henry Hoffmann (University of Chicago, US), Fiodar Kazhamiaka (Stanford University, US), Laércio Lima Pilla (University of Bordeaux, FR), Simon Peter (University of Washington – Seattle, US), George Porter (University of California – San Diego, US), and Samuel Xavier-de-Souza (Federal University of Rio Grande do Norte, BR)

In this break-out session we discussed the aspect of Energy optimization and management for Power and Energy-aware Computing on Heterogeneous Systems. Our discussion sought to answer a question: How can we create energy optimal or near optimal software solutions? We propose a classification of open problems and associated challenges to identify relevant energy models such that multiple objectives and constraints like price, security, accuracy, time, memory, number of kernels, flexibility, and energy budget are achieved.

The computer stack has several layers, e.g., a simple stack covers developed applications, compilers creating the binaries which are executed by the operating system managing the hardware. Considering the stack from hardware and upwards; the higher the layer, the greater the abstraction, or seeing it top-down: when a layer provides its results to lower layers, it gives control to the lower layers. While the layered structure reduces complexity of the individual task and allows great flexibility, the layers obfuscate how different actions in one layer cause changes in the system's energy consumption. In an effort to improve transparency it is increasingly common to have the lower layers provide parameters to higher layers of the stack. However, due to the traditional abstractions, the higher layer may not be aware or take advantage. The aspect of energy transparency over the layers is discussed by another breakout session, so in this session we focus on how we can consider energy optimization and management over the whole system and on identifying open challenges.

Different studies show that individual layers can optimize for energy, however no individual layer can provide energy optimal solutions. Additionally, trying to optimize from several layers at the same time can cause energy inefficiencies; as layers that work without knowledge of the others cause destructive interference [1, 2] . Objectives for the optimization are given by the context and the user; prioritizing wanted effect (the output) to the controller(s). Each layer has different optimization opportunities and while a layer may not be able to find an optimal solution by itself, it may expose "knobs", i.e., variables, and constraints, e.g.,

max frequency, to controllers. We expect these variables to be domain dependent and, here, we keep the definition of controllers to a semantic definition where a controller decides the values of the variables. A controller could be a full-stack controller, the next system layers (upwards or downwards), or another type of control that governs the values.

In the following, we propose a general formalization of the Energy Optimization and Management Problem. Let there be a set of metrics (non-functional properties) such as price, security, accuracy, time, latency, memory, flexibility, energy budget, green house gas emissions. Some of these metrics will be objectives and some constraints, and over time their role may change. The different layers of the system expose configuration variables and their constraints. We can then formulate the global optimization problem covering the entire stack in the general form of:

```
optimize f(metrics, t)        // objective
    subject to                //constraints
    G(variables, t) <= b(t)     // this is a system of inequalities
                               // that express the constraints on
                               // the non-functional properties
                               //  and variable values
```

In general, energy (or power) could appear as part of the objective function or part of the constraints. For example, embedded systems might want to minimize energy given a target latency [3], while an HPC system might have a power constraint (dictated by the facility) and work to maximize application throughput given that constraint [4, 5]. The decision variables represent options provided by different layers of the stack. For example, the embedded application could expose different configuration variables representing the algorithm to use for detecting targets in a signal, with more accurate algorithms requiring more energy. The HPC system could use voltage and frequency scaling as variables that govern power and performance tradeoffs.

This form is a classical optimization form and can be used to express situations where a single controller is given a complete system model and full knowledge of the system's knobs, or a distributed approach such as when each layer has a separate controller that optimizes over the limited set of knobs in its domain. It can also express problems that are solved once apriori for a workload, or that are frequently re-solved for dynamic workloads that benefit from the system being tuned over time. Given the complexity of modern computing systems, a centralized controller approach is likely to be intractable, although it can be made tractable by identifying and removing variables that have little or no impact on the objective function, being able to find optimal solutions considering a reduced vertical design space. Depending on the problem, the design space is reduced in different ways, i.e., different parts of the solution are fixed:

1. from hardware to solution: given a fixed hardware, how may I write my software to solve a specific task while optimizing for a specified objective?
2. from software to platform: given my software, how can I find the hardware and/or software system that enables me to solve my specific task while optimizing for a specified objective?
3. HW/SW co-design or co-optimization: given this general scenario for my software solving a class of tasks on a family of platforms, how can I optimize for a specified objective?

Formalizing the problem in this way can help us understand the computational complexity of the optimization problem, as well as reveal how objectives can be incorporated into the application development, e.g., accuracy, and a future challenge lies in finding the best ways to build software that can be optimized for energy.

To enable the use of this formulation in practice, we identified the following open challenges:

1. Identifying which variables are important to optimize for a given problem class, e.g., a given application or class of applications, or specific hardware, or family of hardware. The fewer variables exposed, the more tractable the problem. It is important that the exposed variables are the significant ones.
2. Developing effective interfaces that allow each layer to expose the variables and receive information from other layers.
3. Developing energy models that express the relationship between variables and objectives. We speculate that models could be both white-box models such as sets of equations or black-box learned models.

References

1 Henry Hoffmann. 2015. JouleGuard: energy guarantees for approximate applications. In Proceedings of the 25th Symposium on Operating Systems Principles (SOSP '15). Association for Computing Machinery, New York, NY, USA, 198–214. https://doi.org/10.1145/2815400.2815403

2 H. Hoffmann, "CoAdapt: Predictable Behavior for Accuracy-Aware Applications Running on Power-Aware Systems," 2014 26th Euromicro Conference on Real-Time Systems, 2014, pp. 223-232, doi: 10.1109/ECRTS.2014.32.

3 C. Imes, D. H. K. Kim, M. Maggio and H. Hoffmann, "POET: a portable approach to minimizing energy under soft real-time constraints," 21st IEEE Real-Time and Embedded Technology and Applications Symposium, 2015, pp. 75-86, doi: 10.1109/RTAS.2015.7108419.

4 Huazhe Zhang and Henry Hoffmann. 2019. PoDD: power-capping dependent distributed applications. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19). Association for Computing Machinery, New York, NY, USA, Article 28, 1–23. https://doi.org/10.1145/3295500.3356174

5 Neha Gholkar, Frank Mueller, Barry Rountree, and Aniruddha Marathe. 2018. PShifter: feedback-based dynamic power shifting within HPC jobs for performance. In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing (HPDC '18). Association for Computing Machinery, New York, NY, USA, 106–117. https://doi.org/10.1145/3208040.3208047

4.3 Results and Insights from the Green Computing Hackathon

Sven Köhler (Hasso-Plattner-Institut, Universität Potsdam, DE), Benedict Herzog (Ruhr-Universität Bochum, DE), Henriette Hofmeier (Ruhr-Universität Bochum, DE), Maja Hanne Kirkeby (Roskilde University, DK), Max Plauth (Hasso-Plattner-Institut, Universität Potsdam, DE), and Lukas Wenzel (Hasso-Plattner-Institut, Universität Potsdam, DE)

Despite energy consumption of software being an omnipresent topic of discussion, knowledge about means to measure and quantify that energy demand is less widely spread. Complementing the scientific talks at the PEACHES seminar, a series of three practical sessions was held over the course of several days.

In total, 36 persons took part in this "Green Computing Hackathon". Participants were given a choice of multiple hardware platforms (Laptops, PCs and various embedded boards), as well as example workloads by the organizers. Alternatively, they were encouraged and supported in using their own hardware and investigating self-provided problems from their own research domains.

The first session started with an introductory talk on measurement facilities, hardware counters and a showcase of software tools like "PinPoint" or "likwid" for retrieving energy and power readings on multiple platforms. Furthermore, good practices and avoidance of common pitfalls in energy measurements were discussed.

Among the workloads and benchmark suites investigated by the participants, the most popular was "heatmap", a simple round-based convolution simulation. Participants compared the influence of different compiler flags, optimisation levels, parallelization strategies, task sizes, clock frequencies, hardware platforms, and vector extensions, as well as the difference between CPU- and GPU-based implementations.

In an extended session, Alex K. Jones demonstrated the "GreenChip" tool, that estimates the carbon footprint for production and application phases in the lifecycle of chip designs. Taking into account factors such as chip area population, chip operational energy, manufacturing technology node, and power grid mix, the tool is focused on indifference and break-even analyses between alternative design choices.

The participants' feedback for all three hackathon sessions was overwhelmingly positive and we highly encourage future Dagstuhl Seminars to include comparable hands-on sessions.

Tools used:

- `https://github.com/osmhpi/pinpoint`
- `https://github.com/RRZE-HPC/likwid/wiki/Likwid-Powermeter`
- `https://github.com/Pitt-Jones-Lab/Greenchip`

List of hackathon organizers:
Maja Hanne Kirkeby (Roskilde Universitet), Benedict Herzog, Henriette Hofmeier (Ruhr-Universität Bochum), Max Plauth, Lukas Wenzel, Sven Köhler (Hasso Plattner Institute Potsdam)

4.4 Energy Transparency across the Hardware/Software Stack

George Porter (University of California – San Diego, US), Ahmed Ali-Eldin Hassan (Chalmers University of Technology – Göteborg, SE), Antonio Carlos Schneider Beck Filho (Federal University of Rio Grande do Sul, BR), Ruzanna Chitchyan (University of Bristol, GB), Kerstin I. Eder (University of Bristol, GB), Christian Eichler (Ruhr-Universität Bochum, DE), Mathias Gottschlag (KIT – Karlsruher Institut für Technologie, DE), Daniel Gruss (TU Graz, AT), Maja Hanne Kirkeby (Roskilde University, DK), Sven Köhler (Hasso-Plattner-Institut, Universität Potsdam, DE), Julia Lawall (INRIA – Paris, FR), Simon Peter (University of Washington – Seattle, US), Max Plauth (Hasso-Plattner-Institut, Universität Potsdam, DE), Sibylle Schupp (TU Hamburg, DE), and Samuel Xavier-de-Souza (Federal University of Rio Grande do Norte, BR)

Five to ten years in the future, the power landscape will have fundamentally changed. The world will have a high proportion of renewables, with volatile power generation. Impact from climate change will be felt much more directly, with power supply variability due to grid disruption sharply up. Information and communication technology (ICT) will also constitute a much larger fraction of the world's demand for power, due to the end of Dennard scaling and increasing demand for more computation due to machine learning. In this world, power must be a first-class design constraint for all aspects of the systems design process.

The working group discussed how to achieve better transparency of power supply and demand across the systems hardware and software stack. We ask the question: *How can we poke through the entire stack of layers in a compute system to allow relevant energy information to flow across the layers to where it is needed? How can lower layers provide energy use information? How can higher layers communicate performance requirements?*

4.4.1 Artifacts and grand challenges

A number of grand challenges and research artifacts that are necessary for power transparency were identified:

- Carbon/PowerTop at scale. This tool would identify top power consumers across a cluster of machines. It should be able to break down power consumption into increasingly fine grain consumers, including to virtual machines, processes, users, functions, and instructions, across user and kernel modes. It should also be able to break down consumption to hardware components, including network switches, peripherals, IO devices, accelerators, network links and interfaces.
- Top-Down analysis for power. Such a tool would help pinpoint sources of power consumption at the microarchitectural level. Similar to Intel's popular Top-Down microarchitectural analysis for performance (cf. https://github.com/andikleen/pmu-tools), this tool would descend the hierarchy of microarchitectural components for progressively finer-grain views into power consumption.
- PowerLint. This tool would help find power bugs and suggest fixes via source code analysis. A potentially interesting angle is to expand the scope of this tool to a planetary database of major power consumers. It could then suggest fixes also to code that is not a power center in isolation, but becomes so by being part of popular uses that collectively consume a large amount of power via long tail effects.
- McPowerAfee. A scanner for power viruses.

4.4.2 Intellectual challenge

All of these artifacts require power attribution. Providing it requires solving a number of challenges in hardware, OS, language, compiler, and toolchains. We developed a rough order of importance of intellectual challenges to be solved to attribute power at the required fidelity and granularity.

The first challenge is to achieve better fidelity and granularity in time and space at the hardware level. Existing technologies, such as Intel's RAPL, operate at relatively coarse granularities of roughly 1ms (some down to 50-150us), making it difficult to attribute power use to functions and instructions. To do so, we need power attribution down to nanoseconds. We also need models and instrumentation for power attribution to microarchitectural components, including TLBs, caches, memory banks, and IO lanes. For example, it was shown that storing a data structure across multiple DRAM banks uses more power. To reduce power usage, we have to be able to account for these effects. Finally, power attribution into the power utility infrastructure (AC, power distribution, power storage, ...) would be necessary to determine large-scale power draw. The breakout group believes that power usage effectiveness (PUE) is not enough to characterize the power draw of these components. While Hyperscalers are near PUE = 1, the edge is not. Further, PUE is usually reported as an average. In reality, it varies based on utilization. Actuation overheads also need to come down to be able to react appropriately. For example, the latency to switch among C states, enter/exit hibernation, currently have high hardware and especially software overheads.

The second challenge is how to achieve scale of attribution. We need to trace power use across clusters of machines, switches, storage devices, memory, etc. This requires scalable, low-overhead mechanisms and data structures to identify power centers at high fidelity and over short timescales.

The third challenge is to trace power use across virtualization and abstraction boundaries. To trace power across the system stack, we need APIs and execution environments to communicate power requirements and demand among power consumers and producers. Virtualization also implies dealing with power as a virtual resource. Power may be stranded if over-provisioned to a single VM that does not use all of the power. Power may be unavailable if over-committed to many VMs that collectively use more than is available.

The fourth challenge is quality of service under power constraints. Many service level agreements include slack that may be used to trade performance for power. To do so, power consumption models are needed. Program configuration changes may also be used to trade fidelity for power. For example, deep neural networks (DNNs) can turn off layers to save power, with a quantified quality loss. To do all of this, we need interfaces for software to specify its power-QoS tradeoff space. A few example uses are power-fair scheduling and power-proportionate computing.

The fifth challenge is automation. Many developers and users do not care or do not know how to optimize for power. We need compilers and programming language tools that help us automate power optimization and tracing across the systems stack. Power could become a non-functional specification of the design and build process of modern software. The specification could be used by compilers for algorithm selection and fitting of algorithms to hardware.

4.4.3 Carbon versus power

We also discussed operational carbon issues of power. However, the main conclusion was that carbon simply modulates power cost, so it can be easily modeled as part of a power variability problem.

4.4.4 Finding and fixing power bugs

We lack an intuition of what uses how much energy (missing conceptual model) and our expectations sometimes break.

power bug: *implementation or hardware caused increased power consumption unrelated to the algorithm.*

How to find power bugs? This requires an energy profiler (attribution of which operation/program section/part of the system consumes which amount of energy). However, an energy linter is what we actually want. The linter processes the (expectedly huge) amounts of profiler output and identifies what part we should look out for.

How to attribute power to code, in particular under presence of interactions with "intelligent" devices (e.g., DMA, Accelerators, Hard Drives) is a prerequisite for a profiler.

Measurement: Use RAPL (Resolution: CPU Package µJ per 1ms, DRAM µJ per 1ms, power planes µJ per 50µs, core voltage V per 150µs), also include off-chip energy consumption (DRAM activity as a result of what ever happens below the last cache level, diverse IO operations affecting devices potentially across the entire data center (stuff happening on the storage network, ...); for GPUs there are similar performance counters/registers.

Mapping: we need to attribute the RAPL measurements back to an instruction/basic block granularity. 1ms/50us translates to quite a large window of possible "culprits" in the original program. We could pad functions/basic blocks/... with nop/pause/... to full RAPL windows for measurement (will be slower than regular execution but allows for measurement).

We also should go beyond the processor for true end to end evaluation. This is even more difficult on the IO level, where packets/operations from multiple sources get aggregated and share the blame. What about indirect effects like spinning up fans just because of an unfortunate simultaneous placement of two independent computationally heavy threads, that even might have happened seconds ago (heat propagates slowly)?

We might also want to account for activity in remote machines triggered by a local action. From that, attribute what part of the remote power consumption was caused by the local action. This step is controversial. The execution time profiling community hasn't solved the issue and they have been working on this for a long time. Hence, this is an overly ambitious, risky step. To resolve this issue, we should get reliable estimates of which proportion of total energy for a workload execution is consumed by the processor alone. This would require a controlled setup in an otherwise quiet datacenter, but it might be illuminating.

How to detect power bugs? We need to establish a firm understanding of which patterns/behavior cause power bugs to identify them. We can do this in a variety of ways. Empirically: run extensive benchmark suites, analyze behavior and generalize appropriately. However, there is a wide variety of architectures and implementations, so it is challenging to have a comparable results collected. Conceptually: reason from underlying micro- to system architectures, which patterns are expected to have detrimental effects. Big Data: collect large data sets from systems in regular operation and analyze those. "Bugs" may also be observed from a divergence between the energy requirements (i.e., features/properties expected by the software stakeholder or owner who pays for the software) and the delivered implementation. E.g., if the customer wants "green star/Energy Efficiency" badge, but is not delivered the required energy behavior, that is a power bug.

4.5 Sustainable Computing

Andreas Schmidt (Universität des Saarlandes – Saarbrücken, DE), Henriette Hofmeier (Ruhr-Universität Bochum, DE), Alex Jones (University of Pittsburgh, US), Daniel Mosse (University of Pittsburgh, US), Frank Mueller (North Carolina State University – Raleigh, US)

In the "Sustainable Computing" breakout session, we looked beyond energy transparency, management, and optimization (which was the focus of the parallel groups). One of our first conclusion was that the footprint of computing services should become more transparent. With computing services, we also mean the physical products involved, such as servers or smart phones. In this case, footprint is not only limited to energy usage in operation, but extends to energy required to manufacture hardware as well as other sustainability metrics related to manufacturing. The latter includes, for instance, CO2 equivalents or effects on humans, e.g. carcinogens, disability-adjusted life years (DALY), or volatile organic compounds (VOC). Generally speaking, we believe that Life-cycle Assessment (LCA) methods should be more thoroughly applied to computing systems to improve transparency.

The *Planetary Health Diet*[1] is a diet that allows a certain population (10bn by the year 2050) to life without hunger, while respecting earth's natural resources. Similarly, we came up with an idea to develop a *Planetary Digital Diet* that governs what a sustainable (and healthy) consumption of digital services would be. Analogous to the grams per day for different food categories (e.g. red meat or vegetables), one could come up with minutes per day for different digital services (e.g. office software on desktop or mobile games). We also applied the "Five R's of Sustainability" (5Rs) to computing. The idea behind the 5Rs is to provide multiple steps at which we can do something with products before we, in the worst case, put them in a landfill or burn them. *Refuse* is about avoiding a footprint in the first place, while *reduce* aims for a lower footprint, if it cannot be avoided. When we *reuse*, we continue using a product in a way that is different from the original manufacturer's idea; either because it can no longer be used for its original purpose or we do not need it anymore. *Recycle* is the (usually lossy and energy-intensive) process of turning the product into pieces and using the pieces to eventually create a new product. Lastly, *rot* is "giving back to nature", thereby keeping the resources in our ecosystem. Applied to computing, we came up with:

- *Refuse*: How can users that care about sustainability refuse digital services that would increase their footprint? Which analogue solutions are more sustainable than digital equivalents? How can we stop providers from offering free-of-charge, unlimited services? How can data minimization policies, e.g. GDPR, have a positive sustainability impact?
- *Reduce*: How can we nudge users to reduce their consumption of digital services (and hence, reduce their footprint)? How can existing and upcoming technology be altered to be more sustainable?
- *Reuse*: How can we foster the reuse of software instead of reproducing it? How can software be easily reduced in footprint (debloated)? How can software be designed to be easily reusable? How can we foster the reuse of data and models, in particular via open science/source? How can we enable a second life for (more) hardware (analogous to car batteries being reused in stationary contexts)?

[1] `https://en.wikipedia.org/wiki/Planetary_health_diet`

- *Recycle*: How can computing hardware be changed to allow for better decomposition, allowing reparation and recycling of individual components?
- *Rot*: How can computing systems be build in a biodegradable way? How can renewable materials be used to build hardware?

In summary, we can conclude that there are various steps to be taken to make computing software and hardware more sustainable as well as educate users in sustainable consumption. These steps, in synergy with improving energy-usage that our digital services rely on, can help to create a more sustainable digital future.

Participants

- Ahmed Ali-Eldin Hassan
Chalmers University of Technology – Göteborg, SE
- Antonio Carlos Schneider Beck Filho
Federal University of Rio Grande do Sul, BR
- Ruzanna Chitchyan
University of Bristol, GB
- Liliana Cucu-Grosjean
INRIA – Paris, FR
- Julian De Hoog
The University of Melbourne, AU
- Kerstin I. Eder
University of Bristol, GB
- Christian Eichler
Ruhr-Universität Bochum, DE
- Michael Engel
Universität Bamberg, DE
- Mathias Gottschlag
KIT – Karlsruher Institut für Technologie, DE
- Daniel Gruss
TU Graz, AT
- Benedict Herzog
Ruhr-Universität Bochum, DE
- Timo Hönig
Ruhr-Universität Bochum, DE
- Henry Hoffmann
University of Chicago, US
- Henriette Hofmeier
Ruhr-Universität Bochum, DE
- Romain Jacob
ETH Zürich, CH
- Alex Jones
University of Pittsburgh, US
- Fiodar Kazhamiaka
Stanford University, US
- Maja Hanne Kirkeby
Roskilde University, DK
- Sven Köhler
Hasso-Plattner-Institut, Universität Potsdam, DE
- Julia Lawall
INRIA – Paris, FR
- Laércio Lima Pilla
University of Bordeaux, FR
- Tulika Mitra
National University of Singapore, SG
- Daniel Mosse
University of Pittsburgh, US
- Frank Mueller
North Carolina State University – Raleigh, US
- Simon Peter
University of Washington – Seattle, US
- Max Plauth
Hasso-Plattner-Institut, Universität Potsdam, DE
- George Porter
University of California – San Diego, US
- Andreas Schmidt
Universität des Saarlandes – Saarbrücken, DE
- Gunnar Schomaker
Universität Paderborn, DE
- Wolfgang Schröder-Preikschat
Universität Erlangen-Nürnberg, DE
- Sibylle Schupp
TU Hamburg, DE
- Jennifer Switzer
University of California – San Diego, US
- Devesh Tiwari
Northeastern University – Boston, US
- Lukas Wenzel
Hasso-Plattner-Institut, Universität Potsdam, DE
- Samuel Xavier-de-Souza
Federal University of Rio Grande do Norte, BR