

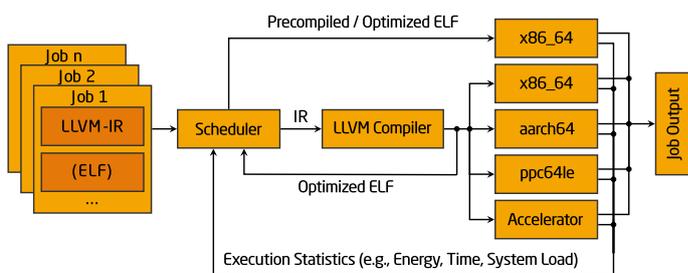
Help your Compiler to Cut the Carbon Footprint

Project Goals

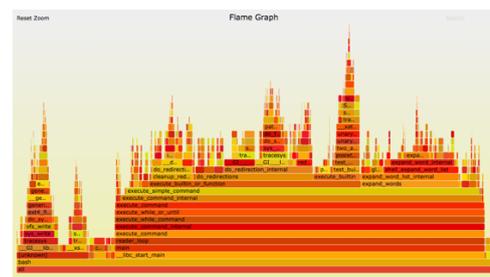
In this master project, you will explore how to use code analysis techniques to estimate the energy demand of software on a very fine-grained level (e.g. on function granularity). To achieve this goal, several metrics need to be extracted from code, such as memory access patterns, instruction profiles, and many more. Both hardware performance counters and energy measurement facilities—such as the PINPOINT tool developed at our group—will serve as important metrics for the attribution process.

As the energy properties vary heavily across systems, you will be given access to a large variety of computing devices to validate your findings. Ultimately, our goal is to use the collected data to optimize job placement in data centers in order to improve energy efficiency.

We will assist you publishing your findings in a technical report or a scientific article.



Energy-Aware Job Scheduler Architecture



Visualization of Energy Profile

Background

In recent years, energy has become a critical operating resource across all computer categories, ranging from embedded devices to powerful server systems and entire data centers. For small-scale, battery-operated equipment, the relevance of energy-awareness may be obvious. On a larger scale however, energy consumption contributes significantly to the carbon footprint and the operating costs of compute infrastructure, and power dissipation is also an important factor limiting the performance and scalability of today's systems.

Unfortunately, the impact of software on the energy efficiency of computers is still widely underestimated. Aside from this canonical worst case where a rogue process is draining the batteries of mobile devices, it is not obvious what software optimization technique may lead to energy-efficient code.

The situation is further complicated by an increasing level of heterogeneity in current system architectures. Besides traditional general purpose processors, heterogeneous computer architectures incorporate accelerators such as GPUs, FPGAs, or TPUs. As a result, some processor architectures or accelerators may be more efficient at certain workloads than others. Being able to attribute energy demand to code can reveal valuable insights regarding the efficiency of the overall system operation as well as of specific workload executions. In addition, the gathered data can be used to optimize operation procedures or adjust workload execution modalities.

Contact

If you have any further questions, please do not hesitate to contact us.

Sven Köhler (C-1.11, sven.koehler@hpi.de)

Lukas Wenzel (C-1.11, lukas.wenzel@hpi.de)

Max Plauth (C-1.6, max.plauth@hpi.de)

Prof. Dr. Andreas Polze (C-1.8, andreas.polze@hpi.de)

Related Work

[1] K. W. Cameron, Rong Ge and Xizhou Feng, “High-performance, power-aware distributed computing for scientific applications,” in *Computer*, vol. 38, no. 11, pp. 40-47, Nov. 2005, doi:10.1109/MC.2005.380.

[2] Benedict Herzog, Max Plauth, Timo Hönig, Sven Köhler, Wolfgang Schröder-Preikschat, Andreas Polze. 2019. “Bridging the Gap: Energy-efficient Execution of Software Workloads on Heterogeneous Hardware Components.” In *Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19)*, June 25–28, 2019, Phoenix, AZ, USA, doi:10.1145/3307772.3330176

[3] Pierre Olivier, A K M Fazla Mehrab, Stefan Lankes, and Mohamed Lamine Karaoui, Rob Lyerly, Binoy Ravindran. 2019. “HEXO: Offloading HPC Compute-Intensive Workloads on Low-Cost, Low-Power Embedded Systems.” In *The 28th International Symposium on High-Performance Parallel and Distributed Computing (HPDC '19)*, June 22–29, 2019, Phoenix, AZ, USA, doi:10.1145/3307681.3325408