Exploring the Energy Consumption of Highly Parallel Software

Mads Hjuler Kusk*, Jeppe Jon Holt*
and Jamie Baldwin Pedersen*
Department of Computer Science, Aalborg University, Denmark
*{mkusk18, jholt18, jjbp18}@student.aau.dk

February 23, 2023

Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1 Introduction

Research questions:

- How does the compiler, temperature and background process impact the energy consumption?
- What is the best measuring instrument for Windows?
- *How well does microbenchmarks represent a realistic usecase compared to macrobenchmarks?*
- How does parallelism affect the energy consumption
- How does P-cores and E-cores affect the execution of parallelism in a process, versus only P-Cores?
- *How can measuring instruments be calibrated to better fit a ground truth*

Hypothsis:

- The expectation for the compilers are that the performance and energy consumption will be similar, but with some deviations. Resulting from the individual compilers implementation.
- The expectation for the temperature is that it in itself would not effect the performance unless the cpu starts to thermal throttle. Though the heat of the cpu is directly related to the electrical resistance meaning that it would be less efficient in terms of joule per computation.
- We do not expect the background process to have a large impact on the energy consumption, this mostly because the non-essential background process are executed rarely and for very brief durations. We do expect that the results will become more consistent without
- We know that IPG and LHW are very similar and we expect Windows RAPL driver to be similar as well.
- We expect that using parallelism there is not a correlation between execution time and energy consumption.
- We expect if we can get the right thread to run on a E core that it could improve energy consumption slightly depending on the size of the benchmark. Where the larger the benchmark the bigger the improvement in energy consumption.
- We expect the calibration to be very situation specific, where the calibration might improve the measurements in some cases, but not in others.

2 Related Work

2.1 Previous Work

This paper builds upon the knowledge gathered in our previous work *"A Comparison Study of Measuring Instruments"*[1] where different measuring instruments were compared to explore whether a viable software-based measuring instrument was available for Windows. It was found that Intel Power Gadget

(IPG) and Libre Hardware Monitor (LHM) on Windows have similar correlation to hardware-based measuring instruments as Intel's Running Average Power Limit (RAPL) has on Linux. This chapter builds upon the related work chapter of the previous work and as such will not be repeated, however, it will be expanded upon.

2.2 Parallel Software

In the work by Abdelhafez et al.[2], the energy consumption for sequential and parallel genetic algorithms are explored, where one research question aims to explore the impact on the energy consumption when using a variable number of cores.

2.3 Compilers

In the work by Hassan et al.[3] the language C++ and different compilers are explored and compared to find the impact of using different coding styles and compilers, where the goal is to find a balance between performance and energy efficiency. The different coding styles introduced explore the impact of splitting CPU and IO operations and interrupting the CPU-intensive instructions with sleep statements. The C++ compilers used in the work by Hassan et al.[3] include MinGW GCC, Cygwin GCC, Borland C++, and Visual C++, and the energy measurements are performed using Windows Performance Analyses (WPA). All compilers are used with default settings, and no optimizations were chosen based on existing work illustrating how optimizations made by compilers are machine and code-dependent[4]. An example of this could be the work by Lima et al.[4], where they found how mainstream compilers will apply multiple optimizations to the final code, where these optimizations in the worst case will result in worse performance and increased energy consumption. The issue of optimizations being very machine dependent was also shown in the work by Cooper et al.[5], where analysis and optimizations were done on a Texas Instruments C6200 DSP CPU, they found that a large portion of the energy is used by fetching instruction. They address this by introducing a fetch packet mechanism, and also find loop-unrolling to reduce energy consumption. While these optimizations decrease the energy consumption for the Texas Instruments C6200 DSP CPU, they note that for other CPUs varying results are expected. A similar conclusion is also found in the work by Hassan et al.[3], where they find that when choosing a compiler and coding style it depends on the nature of the target machine and application. Based on the test case used, which in this case was an election sort algorithm, they find the best performance with the Borland compiler, and the lowest energy with the Visual C++ compiler, and find that both separating IO and CPU operations and interrupting the CPU-intensive instructions with sleep statements also decrease the energy consumption.

3 Method

3.1 Measuring Instruments

This section present the different measuring instruments utilized in this work. The measuring instruments utilized in the previous work will only be briefly introduced, however more detail can be found in our previous work[1]. In this paper, four software-based measuring instruments and one hardware-based measuring instrument is used, where the hardware-based measuring instrument represents the ground truth.

Scaphandre One measuring instrument not used in the previous work is Scaphandre[6]. Scaphandre is described as a monitoring agent which can measure energy consumption and is made for Linux where it can use Powercap RAPL which is a Linux kernel subsystem where data can be read from RAPL. It also has the functionality of measuring energy consumption of some virtual machines, specifically Qemu and KVM hypervisors as of writing this. A driver also exists which allows for installing RAPL on Windows.[7] Doing so allows using Scaphandre on a Windows computer where the sensor is RAPL which is utilizing the model-specific-registers (MSR) to update its counters, so it is not using Powercap RAPL. The Windows version of Scaphandre has some limitations, but is able to report the energy consumption of the CPU package (PKG) in watts which includes energy consumption of the entire socket. Furthermore it can also reports and estimation for the energy consumption for individual processes. It does so my storing CPU usage statistics along side the values of the energy counters. Then it is able to calculate the ratio of the CPU time for each Process ID (PID). With the calculated ratio a new calculation is made to get the subset of the energy consumption which is estimated to belong to a specific PID. On Linux to get the energy consumption of an application, which in many cases would have several PID's, Prometheus can be used, which is a monitoring system that work with Scaphandre. However this functionality is not compatible with Windows.

Running Average Power Limit Intel's RAPL is the most commonly seen software-based measuring instrument seen in the literature. It uses MSRs and Hardware performance counters to calculate how much energy the processor uses. RAPL has previously only been directly accessible on Linux and Mac, and therefore our previous work we only used it on Linux. Where we found that RAPL had a high correlation of 0.81 with our ground truth on Linux.[1]

Intel Power Gadget IPG is a software tool created by Intel, which can estimate the power of Intel processors. It contains a command line version called Powerlog which allows accessing the energy consumption using callable APIs. It uses the same hardware counters and MSRs as RAPL does, therefore it is expected to observe similar measurements to that of RAPL. Which is also shown in our previous work where we found that IPG had a high correlation of 0.78 with our ground truth on Windows. IPG was also found to have a high correlation of 0.83 with RAPL, although the measurement are on different operating systems.[1]

Libre Hardware Monitor We found that LHM had a correlation of 0.76 with our ground truth on Windows.[1]

AC Current Clamp

4 Experiments

5 Results

6 Discussion

7 Conclusion

Acknowledgements

8 Future Works

References

1. Holt, J., Kusk, M. H. & Pedersen, J. B. *A Comparison Study of Measuring Instruments* (Aalborg University Department of Computer Science, 2023).

2. Abdelhafez, A., Alba, E. & Luque, G. A component-based study of energy consumption for sequential and parallel genetic algorithms. *The Journal of Supercomputing* **75,** 1–26 (Oct. 2019).

3. Hassan, H., Moussa, A. & Farag, I. Performance vs. Power and Energy Consumption: Impact of Coding Style and Compiler. *International Journal of Advanced Computer Science and Applications* **8** (Dec. 2017).

4. Lima, E., Xavier, T., Faustino, A. & Ruiz, L. *Compiling for performance and power efficiency* in (Sept. 2013), 142–149.

5. Cooper, K. & Waterman, T. Understanding Energy Consumption on the C62x (Jan. 2004).

6. Hubblo. *Scaphandre* https://github.com/hubblo-org/scaphandre.

7. Hubblo. *windows-rapl-driver* https://github.com/hubblo-org/windows-rapl-driver.