Doing Stuff to Windows Yet Again

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

February 17, 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 Works

In the work by Hassan et al.[1] the language C++ and different compilers are explored and compared in order to find the impact of using different coding styles and compilers, where the goal is to find a balance between performance an energy efficiency. The different coding styles introduced explores 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.[1] 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 very machine and code dependent[2]. An example of this could be the work by Lima et al.[2], where they find how mainstream compliers will apply multiple optimizations to the final code, where these optimizations in a worst case will result in a worse performance and an increased energy consumption. The issue of optimizations being very machine dependent was also shown in the work by Cooper et al.[3], where analysis and optimizations was done on an Texas Instruments C6200 DSP CPU, where it was found that a large portion of energy is used by fetching instruction. They address this by introducing a fetch packet mechanism, and also find loop-unrolling to reduce the 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.[1], where they find that when choosing compiler and coding style it depends on the nature of the target machine and application. Based on the test case used, this being a Selection Sort algorithm, they find the best performance with the Borland complier, 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 Experiments

4 Results

5 Discussion

6 Conclusion

Acknowledgements

7 Future Works

References

1. 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).
2. Lima, E., Xavier, T., Faustino, A. & Ruiz, L. *Compiling for performance and power efficiency* in (Sept. 2013), 142–149.
3. Cooper, K. & Waterman, T. Understanding Energy Consumption on the C62x (Jan. 2004).