Auto-Tuning for Green Computing on GPUs

Ehsan Sharifi Esfahani [1], Ana Lucia Varbanescu [2], Rob van Nieuwpoort [2], Alessio Sclocco [3]

[1] Dimenco, De Run 4281, 5503LM Veldhoven, The Netherlands
[2] Faculty of Science, Informatics Institute, University of Amsterdam, The Netherlands
[3] eScience Center, Amsterdam, The Netherlands

Abstract

Green GPU (Graphics Processing Unit) computing is an important research topic in the context of supercomputing because of the large role that GPUs have today in accelerating applications. In this work, we demonstrate how auto-tuning can be used to improve energy efficiency for GPU computing, by including core and memory frequency, as well as power capping, as tunable parameters.

Methodology

We identify two sets of tunable parameters in the context of GPUs [2]:

1. The set of kernel parameters (e.g. block size and grid size)
2. The set of GPU parameters (e.g. core and memory frequency)

The combination of possible values for these tunable parameters forms a valid configuration. We propose three different configuration-search strategies for the set of GPU parameters as:

$$S1: \{f_{core}, f_{memory}, p_c\}$$
$$S2: \{f_{core}, f_{memory}\}$$
$$S3: \{p_c\}$$

Where we use f_{core} for core frequency, f_{memory} for memory frequency, and p_c for power capping.

We define the following additional parameters:

- ***Search space size*** = the number of all possible configurations for a given program.
- ***Power capping size*** = the number of configurations with average power consumption close to the power capping level. To do so, we use a 10-watt interval in our experiments. For example, if the power capping level is 100 watts and the average power consumption of our application for a given configuration is in the range of 90 to 100 watts, then we say that its average power consumption is close to the power capping level.
- ***Power capping ratio*** = the power capping size over the search space size.

Experimental setup

We performed a series of auto-tuning experiments with a brute-force search, combining:

- Five different GPUs, namely Nvidia Tesla K20m (Kepler), Nvidia GTX Titan (Kepler), Nvidia GTX TitanX (Maxwell), Nvidia GTX980 (Maxwell), Nvidia GTX TitanX (Pascal); available on the VU cluster of the Distributed ASCI Supercomputer 5 (DAS-5) [1]
- Three objective functions: energy consumption, energy-efficiency, and performance.
- Three case-study applications with fixed input sizes: vector-add, stencil, and matrix multiplication with one, two and four different kernel parameters, respectively.
- Three different GPU configuration strategies: S1, S2, S3.

Experimental results and analysis

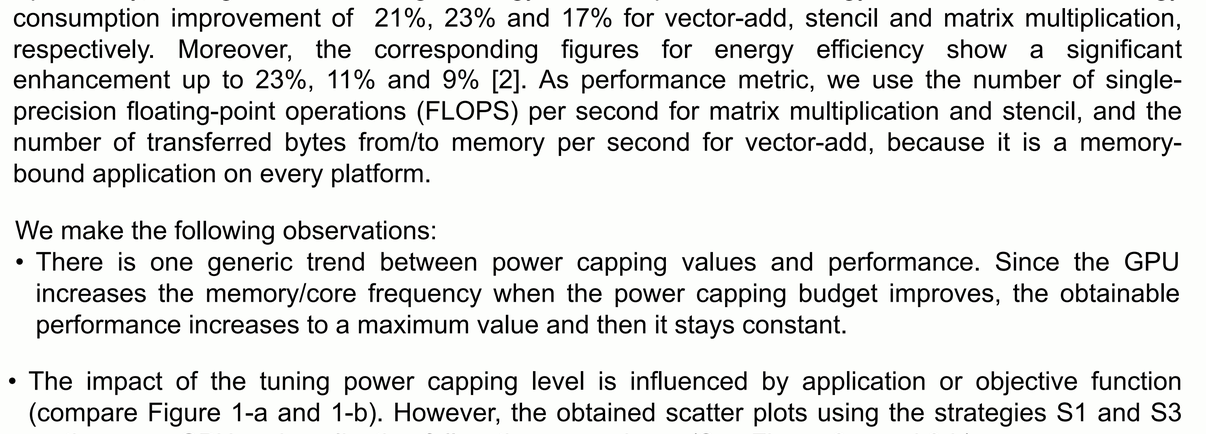
Among the five different GPUs we have performed our analysis on, the Nvidia GTX980 (with supported power capping in the range of 100W - 225W) provides the most interesting results. Specifically, the gain when using strategy S1 compared to strategy S2 leads to an energy consumption improvement of 21%, 23% and 17% for vector-add, stencil and matrix multiplication, respectively. Moreover, the corresponding figures for energy efficiency show a significant enhancement up to 23%, 11% and 9% [2]. As performance metric, we use the number of single-precision floating-point operations (FLOPS) per second for matrix multiplication and stencil, and the number of transferred bytes from/to memory per second for vector-add, because it is a memory-bound application on every platform.

We make the following observations:

- There is one generic trend between power capping values and performance. Since the GPU increases the memory/core frequency when the power capping budget improves, the obtainable performance increases to a maximum value and then it stays constant.
- The impact of the tuning power capping level is influenced by application or objective function (compare Figure 1-a and 1-b). However, the obtained scatter plots using the strategies S1 and S3 on the same GPU and application follow the same shape (See Figure 2-a and 2-b).

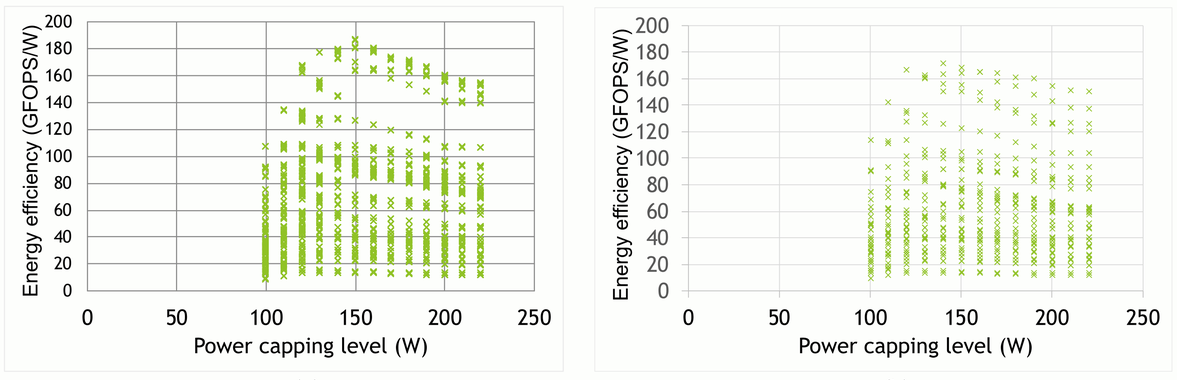
		Benchmark								
		Vector-add			Stencil			Matrix Multiplication		
	Strategy	S1	S2	S3	S1	S2	S3	S1	S2	S3
Objective functions	Energy Consumption(mJ)	61	78	70	145	189	145	9887	11956	9925
	Energy-efficiency (performance/watt)	3185	2470	2364	200	177	198	187	171	171
	Performance (FLOPS or Byte/Second)	194	193	167	33.6	33.5	33.2	2114	2051	2080

Table 1: The optimal values of each objective function for the all benchmarks with different GPU configuration strategies on Nvidia GTX980 (empirically obtained)



(a) (b)

Figure 1: The impact of tuning the power-capping level on energy consumption (a) and performance (b) using the S1 for matrix multiplication. **The values represent the obtained total energy consumption in mJ (a) and performance in GFLOPS (b) under a specific power budget for a valid configuration**



(a) (b)

Figure 2: The impact of tuning the power-capping level on energy efficiency (performance/watt) for matrix multiplication using S1 (a) vs S3 (b). **The values represent the obtained energy efficiency under a specific power budget for a valid configuration.**

Which GPU configuration strategy to choose?

- By having a high power capping ratio, the chance of getting better results for any given objective function using the S1 strategy also increases.
- The S3 strategy tunes the GPU parameters faster, because it has a smaller search space with only one parameter.
- With small power capping ratio, we can only use the S2 strategy since power capping does not have any influence on our results.

Power capping as a new tunable parameter

GPU-level power capping is a new tunable parameter which depends on:

- **Objective function**
- **Application**
- **GPU architecture**
- **GPU configuration strategy**

Figure 3 shows the relationship between these parameters and the obtained optimal power capping values on two different GPUs. For example, we can observe that the power capping value to reach to the lowest total energy consumption for matrix-multiplication using the S1 strategy on Nvidia GTX980 is 120 watts. However, the corresponding figure to achieve the highest energy efficiency is 150 watts.

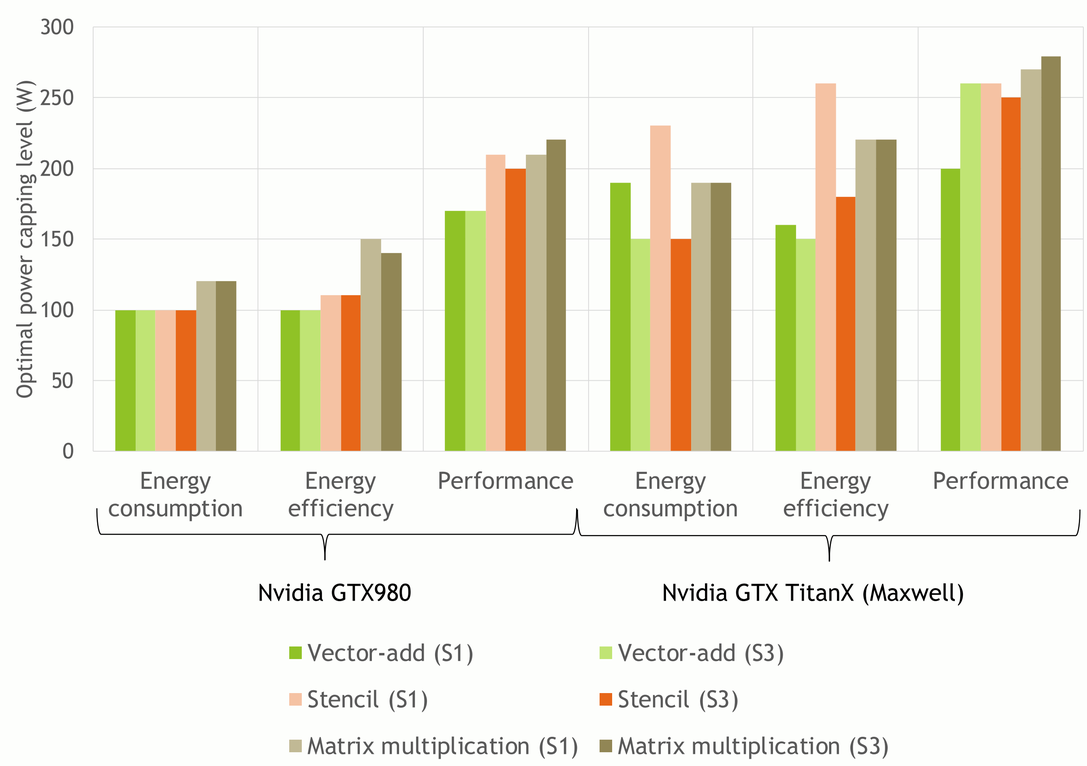


Figure 3: Comparison of the tuned power capping level to obtain the best objective function value for each benchmark on two different platforms using the S1 and S2 strategy.

Conclusions

We provided three different tuning strategies for our set of GPU parameters. We make the following observations:

- Tuning the GPU parameters using the S1 strategy, together with kernel parameters, can always lead to improved energy efficiency and energy consumption, when the power capping ratio is high.
- Power capping can be considered as a new tunable parameter for GPUs.

Future work

We also suggest the following future work:

- Find optimal power capping value faster in time-constrained systems.
- Find out the possible dependency between the problem size and the optimal power capping value.
- Perform more experimental analysis on newer GPU architectures (e.g. Pascal and Turing)
- See the power-metric impact of the S1 strategy on the Square Kilometer Array project, as a large radio telescope of the future, which the high power consumption is one of the the main issues in this project [3].

References

[1] Bal, H., Epema, D., de Laat, C., van Nieuwpoort, R., Romein, J., Seinstra, F., et al (2016). **A medium-scale distributed system for computer science research: Infrastructure for the long term**. Computer, (5), 54-63.

[2] Sharifi Esfahani, E. (2019). M.Sc. Thesis. **"Auto-tuning for energy efficiency on GPUs"**, Faculty of Science, Informatics Institute, University of Amsterdam, The Netherlands.

[3] Barbosa, D., Márquez, G. L., Ruiz, V., Silva, M., Verdes-Montenegro, L., Santander-Vela, J., ... & Kramer, M. (2012). **Power Challenges of Large Scale Research Infrastructures: the Square Kilometer Array and Solar Energy Integration; Towards a zero-carbon footprint next generation telescope**. arXiv preprint arXiv:1210.4972.