Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

GPU energy efficiency

An analysis of energy consumption, usage patterns and energy saving strategies

---

Author:

Quincy Bakker

q.bakker@students.uva.nl

q.bakker@student.vu.nl

1st supervisor:	Ana Lucia Varbanescu	
daily supervisor:	Sagar Dolas	(SURF)
2nd reader:	N/A	

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

July 19, 2020

"I am the master of my fate, I am the captain of my soul"
from Invictus, by William Ernest Henley

Abstract

Here goes the abstract of this thesis.

To ...

Acknowledgements

TODO

Contents

List of Figures

List of Tables

1

Introduction

1.1 Context

TODO

1.2 Objective

TODO

1.3 Research Questions

To address the goal of this study some research questions were formulated which help guide the extraction of data and conclusions from the literature that is reviewed. This section outlines those research questions and their underlying motivation.

1.3.1 CPU-GPU Workload Characterization

TODO

RQ1: What are the different types of CPU-GPU workloads?

RQ2: How can CPU-GPU workloads be detected?

1.3.2 Energy Saving When Running CPU-GPU Workloads

TODO

RQ3: Can information about the CPU-GPU workload be used to save energy?

1.4 Research Method

This section describes the research method that was used.

1.4.1 Literature Study

In this section a description of the process by which literature was collected for the purpose of this study is provided to ensure its replicability.

1.4.1.1 Search

Google Scholar[1] was used as the main tool for discovering relevant literature. Google Scholar works similarly to Google in that it uses a search query and presents relevant results depending on the input.

Some keywords were extracted from the research questions to be used as the base for the search query:

- CPU
- GPU
- workload/usage pattern
- analysis/analyzing/characterize/characterization/model/modeling
- energy/power
- saving/conservation

From these keywords, and the support Google Scholar has for advanced search operators such as the Boolean OR operator to provide constraints to the search query [11], the following search queries were then constructed:

SQ1: GPU OR "CPU GPU" OR "GPU CPU"workload OR "usage pattern" analysis OR analyzing OR characterize OR characterizing OR characterization OR model OR modeling

SQ2: GPU OR "CPU GPU" OR "GPU CPU"energy OR power saving OR conservation

[1] `www.scholar.google.com`

1.4.1.2 Gathering Literature

To collect literature a software tool, Mendeley [1], was used. To organize the literature I subdivided the literature into four categories, as follows:

Unread: Literature that was gathered from a search query but that has not yet been read.

Related: Literature that has been read and is indirectly related to this literature review.

Selected: Literature that was read and that matches the inclusion criteria.

Not selected: Literature that was read but that does not match the inclusion criteria.

Snowballing To gather more relevant literature the snowballing technique was used, which is the process of gathering additional literature from the references of a paper.

1.4.1.3 Application of Selection Criteria

I used a set of selection criteria to filter out any irrelevant papers from the search. These criteria can be subdivided into inclusion and exclusion criteria. This section outlines those criteria and provides the reasoning behind them.

Inclusion Criteria At least one of these inclusion criteria must be fulfilled by each of the papers selected:

IC1: The study covers methods of CPU-GPU energy consumption measurement

IC2: The study covers methods to characterize CPU-GPU workloads

IC3: The study covers methods to save energy when running CPU-GPU workloads

Exclusion Criteria None of these exclusion criteria must be fulfilled by each of the papers selected:

EC1: The study does not make use of the GPU

1.4.1.4 Data Extraction

TODO

[1] `www.mendeley.com`

1.4.1.5 Data Synthesis

TODO

1.4.2 Experimentation

Experimentation for the project was done using the DAS-5 [2].

2

Background

This chapter outlines some of the research and other resources that are relevant to the topic of Graphics Processing Unit (GPU) energy conservation.

2.1 Energy Consumption

This section outlines some of the work that has been done to measure and predict energy consumption in Central Processing Unit (CPU)-GPU heterogenous computing.

2.1.1 Measuring

Measuring live energy consumption and other kernel characteristics is an important aspect of many power saving strategies. There exist tools that can perform these types of measurements, the most important of which are outlined in this section.

2.1.1.1 NVIDIA System Management Interface

NVIDIA's System Management Interface (SMI) tool is a command line utility that is able to query the GPU device state [10]. Support is limited to NVIDIA GPUs. What makes this tool useful to this research is the fact that it can retrieve the current power consumption from the GPU as it is running and that it can output this information to the console, which makes it possible to easily integrate the output programmatically.

2.1.1.2 GPGPUSim

GPGPUSim is a tool that can be used to simulate a GPU and run synthetic workloads. It offers a lot of detailed insights that can be used for workload analysis [1].

2.1.2 Workload Analysis

An important component in any energy saving strategy is to perform a workload analysis, since the decisions that are made often depend on the type of workload that is running [3].

2.1.2.1 Predicting Energy Consumption

Ma and Zhong [8] developed a method to statistically analyze and model the power consumption of a mainstream GPU. To achieve this they make use of the fact that there exists an innate coupling among the power consumption characteristics, runtime performance and dynamic workloads. They found that their model is capable of robustly and accurately predicting the dynamic power consumption estimation of a target GPU at runtime, especially for graphics applications.

Ma and Zhong [8] state that due to the relatively simpler cache hierarchy, higher level of parallelism, less complex control requirements, and more computation units, GPU power modeling differs from general-purpose processing units. Some limitations of their approach they state are that micro architectural knowledge of the GPU is needed to provide more complex and accurate modeling approaches, and that quantitative analysis of GPU workloads and statistical selection of the power consumption correlated workloads are necessary in the data preprocessing step.

Hong and Kim [4] developed an Integrated Power and Performance (IPP) prediction model that predicts application execution time and access rate and performance per watt. The model is able to predict the power consumption and execution time with an average of 8.94% error.

Jiao et al. [5] systematically characterized the energy efficiency of GPU computing by investigating the correlation between power consumption and different computational patterns under various voltage and frequency levels. They used three different applications with various degrees of compute and memory intensiveness and found that the GPU application kernels' performance and power consumption are largely determined by the rate of issuing instructions and the ratio of global memory transactions to computation instructions.

Nagasaka et al. [9] developed a statistical model to estimate the power consumption of GPU kernels. To achieve this they use the performance counters exposed for Compute Unified Device Architecture (CUDA) applications to train a linear regression model using them as independent variables and power consumption as dependent variable. They found a linear correlation between the performance profiles and power consumptions. Their

regression model achieves highly accurate estimates with an average error ratio of 4.7% with the applications they tested. A limitation of their model is that is does not handle kernels that perform texture reads well since there is a lack of performance counters to monitor texture access, resulting in an underestimation for those kernels. They state that the recent enhancements to the CUDA profiler can potentially remedy this issue.

Chen et al. [3] also developed a method to statistically analyze GPU power consumption. They designed a high-level GPU power consumption model using sophisticated tree-based random forest methods which can correlate the power consumption with a set of independent performance variables. Their model is able to accurately predict GPU runtime power consumption and provides insights for understanding the dependence between the GPU runtime power consumption and the individual performance metrics. To gain detailed insights they used *GPGPUSim* [1] to simulate the kernels. Their random forest model is able to identify the most influential variables in power prediction.

Komoda et al. [6] developed an empirical model of the performance and the maximum power consumption of a CPU-GPU heterogenous system to predict the execution time and total power consumption.

Li, Byna, and Chakradhar [7] used GPU performance and power models to make predictions for potential workload consolidation strategies that can optimize power usage.

2.1.2.2 Usage Patterns

TODO

2.2 Energy Saving

This section goes into detail on the research that has been done in the domain of CPU-GPU heterogenous computing energy saving methods.

2.2.1 Disabling Cores

Hong and Kim [4] developed an IPP prediction model to predict the optimal number of active GPU processors to achieve the highest performance per watt ratio for a given application. They based their model on the intuition that when an application reaches the peak memory bandwidth, using more cores does not result in a performance improvement. Their model is also able to determine the increases in power consumption that resulted from increases in temperature. Their approach requires modification to the programmer's code since they cannot change the hardware or the thread scheduler, so they limit the number

of blocks inside an application to constrain the number of active cores. By using IPP they managed to save on average 10.99% of runtime energy consumption on applications that are limited by memory bandwidth by using fewer cores.

2.2.2 Dynamic Voltage and Frequency Scaling

Dynamic Voltage and Frequency Scaling (DVFS) is a technique that allows the voltage and frequency of the GPU to be adjusted dynamically so as to reduce power usage.

Komoda et al. [6] developed a power capping technique through coordinating DVFS and task mapping to prevent a load imbalance between the CPU and GPU in heterogenous systems. Using their model to predict execution time and total power consumption based on a set of parameters, they managed to determine the optimal set of device frequencies and task mappings at the beginning of the execution, with the proposed power capping technique achieving more than 93% of performance compared to the ideal one in 24 out of 25 cases.

2.2.3 Load Balancing

Li, Byna, and Chakradhar [7] developed a runtime framework that dynamically consolidates workloads from multiple user processes into a single GPU workload. By using performance and power models they predict potential workload consolidation strategies that optimize power usage. They experimented on a variety of workloads that perform poorly on a GPU compared to a well optimized multicore CPU implementation and showed that their framework for GPUs can provide $2 - 22\times$ the energy benefit over a multicore CPU implementation.

3

Usage Patterns

TODO

4

Energy Saving Strategies

TODO

5

Dynamic Energy Saving

TODO

6

Discussions

TODO

7

Conclusion

TODO

Appendix A

TODO

References

[1] Ali Bakhoda et al. "Analyzing CUDA workloads using a detailed GPU simulator". In: *ISPASS 2009 - International Symposium on Performance Analysis of Systems and Software*. 2009, pp. 163–174. ISBN: 9781424441846. DOI: 10.1109/ISPASS.2009.4919648. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/02/gpgpusim.ispass09-2.pdf.

[2] Henri Bal et al. "A Medium-Scale Distributed System for Computer Science Research: Infrastructure for the Long Term". In: *Computer* 49.5 (2016), pp. 54–63. ISSN: 00189162. DOI: 10.1109/MC.2016.127. URL: https://isis-data.science.uva.nl/cgmsnoek/pub/bal-das-computer.pdf.

[3] Jianmin Chen et al. "Statistical GPU power analysis using tree-based methods". In: *2011 International Green Computing Conference and Workshops, IGCC 2011*. 2011. ISBN: 9781457712203. DOI: 10.1109/IGCC.2011.6008582.

[4] Sunpyo Hong and Hyesoon Kim. "An integrated GPU power and performance model". In: *Proceedings - International Symposium on Computer Architecture*. 2010, pp. 280–289. ISBN: 9781450300520. DOI: 10.1145/1815961.1815998. URL: http://www.cs.binghamton.edu/%7B~%7Dmillerti/cs680r/papers/GPU/AnIntegratedGPU.pdf.

[5] Y Jiao et al. "Power and performance characterization of computational kernels on the GPU". In: *Proceedings - 2010 IEEE/ACM International Conference on Green Computing and Communications, GreenCom 2010, 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing, CPSCom 2010*. 2010, pp. 221–228. ISBN: 9780769543314. DOI: 10.1109/GreenCom-CPSCom.2010.143. URL: https://research.cs.vt.edu/synergy/pubs/papers/yang-perf-power-GPU-DVFS-greencom10.pdf.

[6] Toshiya Komoda et al. "Power capping of CPU-GPU heterogeneous systems through coordinating DVFS and task mapping". In: *2013 IEEE 31st International Conference on Computer Design, ICCD 2013*. 2013, pp. 349–356. ISBN: 9781479929870. DOI: 10.1109/ICCD.2013.6657064. URL: http://www.cse.chalmers.se/%7B~%7Dsica/phd/mappingstudy/primarystudies/S114.pdf.

[7] Dong Li, Surendra Byna, and Srimat Chakradhar. "Energy-aware workload consolidation on GPU". In: *Proceedings of the International Conference on Parallel Processing Workshops*. 2011, pp. 389–398. ISBN: 9780769545110. DOI: 10.1109/ICPPW.2011.25. URL: https://www.researchgate.net/publication/224263014.

[8] Xiaohan Ma and Lin Zhong. "Statistical Power Consumption Analysis and Modeling for GPU-based Computing". In: *Proceedings of the SOSP Workshop on Power Aware Computing and Systems (HotPower '09)* (2009), None. URL: https://www.yecl.org/publications/ma09hotpower.pdf%20http://www.sigops.org/sosp/sosp09/hotpower.html.

[9] Hitoshi Nagasaka et al. "Statistical power modeling of GPU kernels using performance counters". In: *2010 International Conference on Green Computing, Green Comp 2010*. 2010, pp. 115–122. ISBN: 9781424476138. DOI: 10.1109/GREENCOMP.2010.5598315. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.232.4758.

[10] NVIDIA. *NVIDIA System Management Interface | NVIDIA Developer*. URL: https://developer.nvidia.com/nvidia-system-management-interface (visited on 07/18/2020).

[11] Daniel M. Russel. *Google Advanced Search Operators*. URL: https://docs.google.com/document/d/1ydVaJJeL1EYbWtlfj9TPfBTE5IBADkQfZrQaBZxqXGs (visited on 07/19/2020).

Statement of Originality

This document is written by Student Quincy Bakker who declares to take full responsibility for the contents of this document.

I declare that the text and the work presented in this document are original and that no sources other than those mentioned in the text and its references have been used in creating it.

The Faculty of Science is responsible solely for the supervision of completion of the work, not for the contents.