Emanuelle Noel Crespi
University of Maryland Undergraduate Computer Engineering
James A Clark's School of Engineering - Class of 2016
ecrespi@terpmail.umd.edu
(301) - 366 - 2941

Optimizing the CPU-GPU Interactions to Improve Performance

Research Objectives:

Primary

The primary focus of this project is to analyze a set of particular instructions that can be more efficiently computed by the hardware GPU. If the CPU can subsequently pass a common set of instructions to be computed on the GPU, there should be an increase in performance, especially when considering an operating system that is running simultaneous threads. As a result, the CPU would experience less overhead for a thread's execution time, and show that there can be a fair speedup overall by accounting for this particular CPU-GPU interaction. Given that parallelizable computationally extensive code allows for the possibility of faster execution time, the CPU-GPU interaction can be further optimized to acquire a significant speedup.

Secondary

The secondary objective is to gain a better understanding of any shortcomings that the GPU has in processing a set of instructions, while gaining experience with NVIDA's popular GPU architecture, Tegra K1. We have a well-defined understanding of the CPU, including how instructions can be pipelined, after focusing on the MIPS ISA in our computer architecture class (ENEE446), so shortcomings and advantages of the general purpose GPU will be researched thoroughly in this project. The concept of how instructions are pipelined on the GPU will be studied, and the actual implementation of the GPU to run code in parallel will lead this study. Optimizing the GPU for heavy workloads will be necessary for understanding the performance of our hardware, the Jetson TK1, including power consumption and heat dissipation. It will be beneficial to become familiar with parallel programming for the GPU (i.e. CUDA), along with testing on the gem5GPU simulator, while taking a more practical approach by developing test code for the Jetson TK1 board.

Inspirations for this project are listed and described below.

Conducted and Proposed Research (Overview/Methodology):

Background/Studies/Resources

- Maryland CPU-GPU Cluster Infrastructure

  The Maryland CPU-GPU Cluster Infrastructure can be seen as the main inspiration for this project. The research conducted in the UMIACS laboratories demonstrates the capabilities of CPU-GPU paradigms for high performance computing. Their primary motivation is how the GPU has become powerful enough to be useful coprocessor for the CPU, and their goal is to assist in the advancement of the capabilities for the CPU-GPU paradigm.

- Cholesky Decomposition and Linear Programming on a GPU

  Proposes linear programming algorithms that can be applied to the GPU, thus benefitting the processing of a large set of computational instructions. Cholesky algorithms are emphasized in this research to show that the GPU can be optimized for algorithms involving matrix-matrix multiplication and other linear models. This gives us an idea for how time extensive computations may be processed by the GPU to improve overall performance.

Emanuelle Noel Crespi
University of Maryland Undergraduate Computer Engineering
James A Clark’s School of Engineering - Class of 2016
ecrespi@terpmail.umd.edu
(301) - 366 - 2941

Project Description

This project design will study possible optimizations for the CPU-GPU interactions on the Jetson TK1, satisfying the ECE undergraduate category D for 3 credits in ENEE499L. It will first consider the questions that guide this topic, including any outside sources that inspired this approach.

1. Can a GPU be used to improve system performance and reduce power consumption when given the same computational workload as the CPU?
2. How can we modify/optimize a GPGPU to allow for better general-purpose computation?
3. What are the benefits and drawbacks of making these modifications for the CPU-GPU interactions?

The work will consist of preparatory steps for experimentation, a set of actual experiments, and the compilation of data for each experiment. A formatted results section describing findings and conclusions will be included. The ideal outcome would be that others use these results to further optimize the CPU-GPU interactions of a system, by either applying this to the Jetson TK1 or some other hardware.

Intended/Proposed Research

This project will analyze the performance of the Jetson TK1 CPU-GPU system while making use of it’s gpgpu capabilities. Computationally extensive algorithms, such as prime factorization of large numbers and smoke pattern rendering, will be developed and tested on the Jetson TK1. The idea is to take these as coding examples, and optimize them by sending the parallelizable sections to the available cores of the Jetson’s Kepler GPU. The runtime of each program will be tested both before and after any optimizations have been made. This information will be gathered to show whether a relative speedup has been observed. The development and debugging stages will involve a legitimate understanding of the parallel computing platform, CUDA, and it's compatible programming languages; C, and the ARM Cortex-15 instruction set. The end result should leave the audience, or any potential reader, with a new outlook on the usefulness of GPUs, and how they have potential to be optimized further for general-purpose requests. Another portion of this research will be gathering information such as power output and energy dissipation for the TK1 system (please refer to Tolga Keskinoglu’s proposal for more information).

Student Involvement

The work will be organized using github or dropbox as the repository for file management. Any research/lab work will be conducted 4:00pm – 7:00pm Monday – Friday during the week to gather/develop the potential source code for testing on the Jetson TK1, while gathering the data for analysis. The debugging/understanding of CUDA for NVIDIA’s Kepler GPU and NVIDIA’s ARM Cortex-A15 CPU instruction set will be necessary for the progression of this research lab. Other work would include the preparatory research/planning/simulating experiments such as test runs on gem5GPU(a heterogeneous CPU-GPU simulator). By the final weeks of the semester, the members will pull their findings together to develop a conclusion on any drawbacks and advantages between processing and power consumption on the Tegra K1. The members, along with an explanation of any involvement for the distributed work is listed on the following page.

Members

⟶ Emanuelle Crespi (Embedded Software Development/Performance Testing)

- Gathers and modifies code from reliable sources (i.e. academic sources and studies) or writes this code independently to develop for testing the CPU-GPU performance on the Jetson TK1. Computationally extensive code will be tested on the board. Parallelizable sections of this code will be developed to follow the format of NVIDIA's CUDA API to allow for GPU processing. The compilable code must be compatible with the Jetson's ARM Cortex-A15 instruction set.

- Makes use of any available peripherals for analyzing the system execution time (including data transfer, and any other needed components). Successful runs will be used to organize the performance diagnostics for that file in a well-formatted document.

- Will make sure that the developed code is well commented with an explanation of the significance of this test. The head of these documents (includes date, time, name/s, and any external sources) will describe what has been tested, along with an explanation of the performance diagnostics, and suggestions for further development.

⟶ Tolga Keskinoglu (Embedded Software Development/Power Consumption Testing)

- Gathers code from reliable sources or develops code from scratch to test power consumption on the Jetson TK1

- Experiments with methods of testing power consumption for the Tegra K1 such as using a multi-meter or other necessary equipment to determine which types of processing workloads can be computed with greater power efficiency on the TK1

- Will make sure that the developed/tested code is well commented with an explanation at the head of the file describing its intent for what has been tested (includes date, time, name/s, and any external sources).

Project Schedule:

→Weeks 1-3: Download, install, and test CUDA libraries and any necessary drivers [Aug 29 - Sep 16]

- Running tests with the simulator to gain a general understanding of how it works
- Familiarizing with languages/libraries that will be necessary for project development
- ***Coding 2-3 test programs both on/off board for experience using C, or any other languages including CUDA libraries (Sep 16)***

→Week 4-5: Research/Planning for future experiments [Sep 19 - Sep 30]

- Accessing scholarly resources to plan for future experiments
- Discussing plans with ECE faculty to narrow down goals and validate approach
- Documented plan for our primary focus including what exactly will be tested

→Weeks 6-13: Experimenting and testing applications on Jetson TK1 [Oct 3 - Nov 25]

- ***1-2 documented successful experiments on Jetson TK1 (Nov 16)***
- ***Mid-semester report showing the results thus far and plans to move forward (Nov 23)***
- More experiments using Jetson TK1

→Week 14-16: More in-depth experimenting and testing [Nov 28 - Dec 16]

- ***3 documented successful experiments on Jetson TK1 (Dec 2)***
- Reviewing/optimizing and bringing our tests together to draw conclusions for the TK1 system
- ***Rough draft of final report (Dec 7)***
- **Final report including documented experiments, source code, analysis, results, suggestions for further development and conclusions (Dec 16)**

Emanuelle Noel Crespi
University of Maryland Undergraduate Computer Engineering
James A Clark's School of Engineering - Class of 2016
ecrespi@terpmail.umd.edu
(301) - 366 - 2941

Conclusion:

Student Learning Outcomes

Regardless of the result, I expect to acquire the following knowledge base by the end of this project. This experience will prove useful in the embedded industry (for the popular ARM32 ISA), with GPU programming as an added bonus.

⟶ more in depth understanding of parallel computing for how instructions are sent to the GPU

⟶ insight for the CPU-GPU interactions, including benefits and drawbacks, of the GPU in modern day hardware

⟶ experience coding with the following hardware/components for NVIDIA's Jetson TK1

a. Kepler GPU (192 cores for parallel GPGPU programming/development)

b. Quad-core ARM Cortex-A15 CPU

⟶ experience coding with the following API/software

c. CUDA (for parallel GPU programming)

d. gem5GPU (for heterogeneous CPU-GPU simulating)

⟶ experience coding for the following languages

a. C, and any other necessary languages for this project

The Maryland CPU-GPU infrastructure shows that modern day hardware allows for an incredible boost in speed for high performance computing. Their research has proved that in the general case, the GPU is a necessary co-processor for speeding up the execution of parallelizable instructions on the CPU. We use this knowledge as our motivation to find a way to further optimize this interaction in computers, and hopefully make it even easier to perform small tasks on the CPU while fully utilizing the hardware capabilities of the GPU for other tasks.

Emanuelle Noel Crespi
University of Maryland Undergraduate Computer Engineering
James A Clark's School of Engineering - Class of 2016
ecrespi@terpmail.umd.edu
(301) - 366 - 2941

Bibliography:

1. "A view of the parallel computing landscape". *Commun. ACM*. 52 (10): 56–67

2. <https://gem5-gpu.cs.wisc.edu/wiki/Main_Page>
gem5-gpu: A Heterogeneous CPU-GPU Simulator Jason Power, Joel Hestness, Marc S. Orr, Mark D. Hill, David A. Wood. Computer Architecture Letters vol. 13, no. 1, Jan 2014

3. <http://elinux.org/Jetson/Graphics_Performance>
"Jetson/Graphics Performance." *ELinux.org*. Web. 06 Sept. 2016.

4. <http://www.umiacs.umd.edu/research/GPU/publications.html>
J. H. Jung and D. P. O'Leary. Cholesky decomposition and linear programming on a gpu. In *Workshop on Edge Computing Using New Commodity Architectures (EDGE)*, Chapel Hill, North Carolina, May 2006.

4. <http://www.nvidia.com/object/cuda_home_new.html>
"Parallel Programming and Computing Platform | CUDA | NVIDIA | NVIDIA." *Parallel Programming and Computing Platform | CUDA | NVIDIA | NVIDIA*. Web. 18 Aug. 2016.

5. <http://www.umiacs.umd.edu/research/GPU/research.html>
"Research Areas." *Maryland CPU-GPU Cluster Infrastructure*. Web. 18 Aug. 2016.