Power/Performance Analysis and Optimization for Deep Learning on CPU-GPU Platforms

Ahmet Fatih Inci and Ting-Wu Chin
Carnegie Mellon University
Department of Electrical and Computer Engineering
{ainci, tingwuc}@andrew.cmu.edu

Abstract

Due to the intrinsic data parallelism characteristic of deep learning, GPU is a much better platform for deep learning applications compared to CPU. This works aim at understanding what is left to be done for CPU given that GPU has to run deep learning applications on embedded platform. In modern mobile SoC, e.g. Nvidia Tegra X1, GPU and CPU shares same power budget and memory budget, and hence affects each other cohesively. By characterizing different deep learning workloads and CPU workloads under different kind of CPU-GPU frequencies, we wish to understand what is left to be done for CPU and how those workloads affect the tasks on GPU.

1. Introduction

With the recent success of machine learning workloads, there are growing interests in understanding the speed/power consumption trade-offs of CPU, GPU, FPGA, and ASIC while running those intelligent workloads [4, 5]. However, there is no prior art that analyzes the workloads for CPU given that GPU is running a deep learning workload in the meantime. We find this question interesting and easily neglected since people focus on deep learning on GPU. Our argument is that even though deep learning task is considered the top priority task in the whole system, there is still a lot of CPU resources to utilize. For example, suppose the mobile device that equips with both CPU and GPU has to run deep learning model in the background all the time to analyze the context of users, we expect users to run applications in the meantime. However, it is not clear, given that both CPU and GPU share the same power and memory budget, what kind of tasks are allowed to utilize the CPU without affecting the performance of the deep learning task.

In this work, we try to understand and analyze what type of tasks, under what kind of frequencies can the CPU run under various GPU constraint. We will consider the deep neural network that run on GPU the top priority task of the overall system. Hence, the performance of it acts as the constraints in our analysis. We specifically focus on the embedded platform and we use Nvidia's Tegra X1 throughout our study. In embedded MPSoC where CPU-GPU are integrated on the same chip, it is shown that global power management that controls the frequency of both CPU and GPU is essential due to shared power budget [6]. Hence, if one wants neither under-utilize the quad-core CPU nor performance degradation for the deep neural network, analysis is required to understand the sweet spot.

We study the 3 types of CPU benchmarks, 3 types of deep neural network, and different CPU-GPU frequency. For CPU, we mainly investigate the workloads in SPLASH2 benchmark suite [8] and try to come up with memory-intensive, compute-intensive, and mid-level workload in both compute and memory. For GPU, we target specifically on image classification task and includes three types of deep neural network from small to large. We also consider 3 different frequencies for CPU and 3 different frequencies for GPU.

2. Related Work

In terms of the performance and power analysis for deep learning tasks, prior art focuses on characterizing the difference between various platforms running different deep learning workloads. Nurvitadhi et al [5] compares the power/performance characteristics of binarized neural network on CPU, GPU, FPGA, and ASIC, which shows FPGA implementation have much better performance per watts compared to both CPU and GPU. Malik et al [4] shows that GPU and FPGA can compete in energy delay product and depends on the input size, or the level of data parallelism.

This work obviously diverges from prior works since we focus on what tasks can be done on CPU while GPU is running deep learning workloads. In other words, since it is shown that GPU is almost always better than CPU in deep learning workloads, we want to understand the role for CPU in the big deep learning era.

3. Methodology

In this project, Nvidia Jetson TX1 embedded platform will be used for power and performance analysis. We will run various deep neural network architectures for image classification task on GPU. We will use small, medium, and large deep neural networks to make a thorough power and performance analysis with different network characteristics. Caffe framework [2] will be used to implement deep neural networks on GPU. Similarly, we will use SPLASH2 benchmark suite for CPU. Various SPLASH2 benchmarks will be used to cover different workload characteristics such as memory-intensive and compute-intensive workloads. Furthermore, we will run these CPU-GPU benchmarks with using three different frequency values for both CPU and GPU. First, we will run them individually to obtain power, performance, and temperature results for the baseline. Secondly, we will run them jointly to do analysis and optimization for deep learning on embedded CPU-GPU platforms.

Power values will be calculated by using current sensors in Nvidia Jetson TX1 platform. Temperature results also will be collected by using thermal sensors in TX1. We may also take off the heat sink to simulate embedded platforms which do not have heat sink. It will significantly increase temperature values. However, CPU and GPU will throttle themselves to not exceed thermal design power (TDP) constraint. Performance results for CPU will be calculated by using performance counters as an instruction per cycle (IPC) metric. Moreover, we will obtain CPU utilization results using system calls. Execution time of deep neural network will be calculated using Caffe framework as a performance metric.

4. Objectives and Deliverables

We try to understand how various CPU workloads and frequencies affect the performance of the deep learning task and system power under thermal design power (TDP) constraint. We fix GPU to do inference for a image classification task. Meanwhile, we try to run meaningful tasks on CPU to use the remaining resources of the embedded platform. Hence, we can better understand what is acceptable to run CPU without affeeting the performance of the deep learning task. Moreover, we try to analyze how CPU-GPU workloads affect temperature values where CPU and GPU share the same power budget. Last but not least, we hope to understand the trade-off between the performance of the CPU workload and the performance of the deep learning task by investigating the instruction per cycle (IPC) for CPU as well as the throughput for for the deep learning task (frame per second).

5. Deep Neural Networks on GPU

In this section, we will first elaborate on the benchmarking deep neural networks that we choose, then we run them in 3 kinds of GPU frequencies to obtain the power, performance, and temperature profiles. These profiles act as the baseline for the study of the CPU-GPU execution scenario.

Benchmarks We try to pick small, medium, and large deep neural networks in terms of the depth of the network. Specifically, we pick 3 neural networks from those that are used to tackle image classification task of the ImageNet dataset. We summarize the benchmark we choose in Table. 1. We measure the memory overhead by using *tegrastat* provided by NVIDIA while inferencing the image.

Notice that there is only 4GB of RAM available on TX1, which is shared between CPU and GPU, with 1GB reserved by the operating system (Ubuntu). Hence, in the case of ResNet-152, it almost consumes all the available memory of the system that might leave barely nothing for the CPU. On the other hand, the shallowest neural network we have is AlexNet, and it still consumes a large amount of memory, i.e. 720 MB.

	Memory (MB)	# Layers	Top-1 Acc.
AlexNet[3]	720	7	57.2%
GoogLeNet[7]	820	22	68.7%
ResNet-152[1]	2224	152	77.0%

Table 1. The descriptions of the deep neural networks we choose, including the memory overhead during inference (single image), number of layers, and the reported top-1 accuracy on ImageNet.

We choose 3 kinds of GPU voltage/frequency pairs, i.e. the smallest (0.82 V, 76.8 Mhz), the medium (0.85 V, 537.6 Mhz), and the largest (1.1 V, 998.4 Mhz), from the 13 available voltage and frequency pairs on NVIDIA Tegra X1 to further investigate the power, latency, and temperature profiles of the aforementioned benchmarks.

Profiling We sweep through 3 different voltage and frequency pairs for each of the benchmark that we study and collect the temperature of both CPU and GPU, the power consumption of the GPU, and the latency of the tasks.

Fig. 1 and Fig. 2 show both the power and latency profile of the aforementioned DNN benchmarks running under

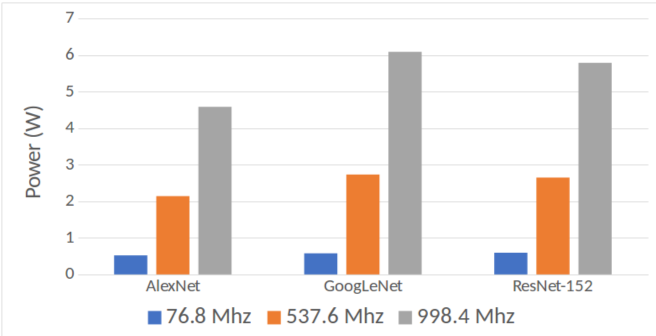

Figure 1. The power profile when running DNN benchmarks under different GPU voltage/frequency pairs.

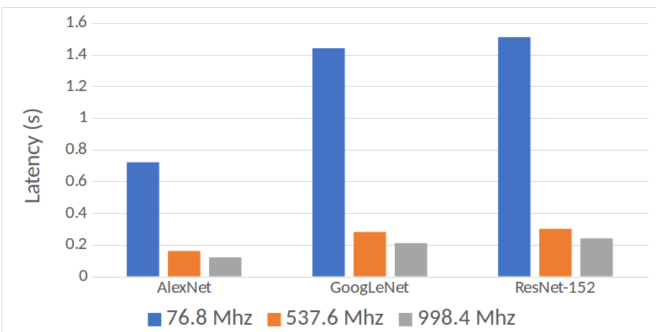

Figure 2. The latency profile when running DNN benchmarks under different GPU voltage/frequency pairs.

different GPU voltage/frequency pairs. As expected, the power consumption is higher when the GPU voltage and frequency are higher. Also, the deeper the neural network the higher the latency. Interestingly, the power consumption of GoogLeNet is higher than ResNet-152 at each voltage and frequency pair, which implies that the utilization of the GPU when runnnig GoogLeNet is higher than ResNet-152. On the other hand, the three benchmarks share similar temperature profiles, i.e. around 40C on average, which is far from the throttling temperature of the GPU on TX1, i.e. 89.5C.

To further compare and analyze the characteristics of these benchmarks, we normalize the latency, power consumption, and accuracy to AlexNet as shown in Fig. 3. From AlexNet to GoogLeNet, the major increase of the cost is latency and power consumption with slightly increase in memory overhead. Though at the first glance that accuracy improves from 57.2% to 68.7% is not much compared to the increment in cost, it is hard to judge the impact of accuracy on the quality of services. On the other hand, from GoogLeNet to ResNet-152, the major cost increase is memory, i.e. more than 2x. In terms of memory overhead and accuracy, it seems that it does not worth it to go from GoogLeNet to ResNet-152 unless there is a hard target for the accuracy since it requires much more memory with small accuracy improvement.

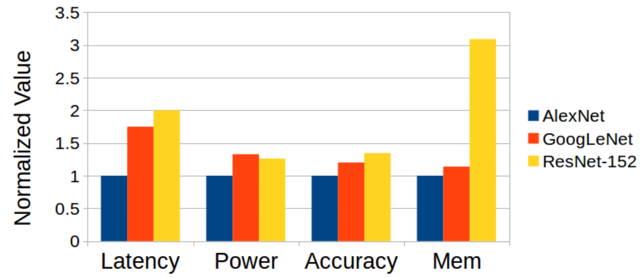

Figure 3. Compare the benchmarks in terms of latency, power consumption, memory overhead, and accuracy. The values are normalized to AlexNet. We fix the voltage/frequency pair to the highest one.

6. Timeline

Timeline of the project tasks are listed below. Group members will work closely on each task to come up with a thorough analysis of power and performance results for deep learning on a CPU-GPU platform.

M1 - Choosing three SPLASH benchmarks (memory-intensive, compute-intensive) for CPU

- Choosing three image classification deep neural networks with different scales for GPU

- Running CPU-GPU benchmarks individually to obtain the baseline for comparison purposes

M2 - Running CPU-GPU benchmarks jointly by changing frequency values for both CPU and GPU

- Analyzing power, performance, and temperature results

7. Conclusion

The authors will analyze power and performance results of a deep learning application on a CPU-GPU platform which is Nvidia Jetson TX1 SoC in this case. Power and performance analysis will identify what is left to be done on CPU while GPU is running inference on deep neural network.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[4] M. Malik, F. Farahmand, P. Otto, N. Akhlaghi, T. Mohsenin, S. Sikdar, and H. Homayoun. Architecture exploration for energy-efficient embedded vision applications: From general purpose processor to domain specific accelerator. In *VLSI (ISVLSI), 2016 IEEE Computer Society Annual Symposium on*, pages 559–564. IEEE, 2016.

[5] E. Nurvitadhi, D. Sheffield, J. Sim, A. Mishra, G. Venkatesh, and D. Marr. Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic. In *Field-Programmable Technology (FPT), 2016 International Conference on*, pages 77–84. IEEE, 2016.

[6] A. Pathania, Q. Jiao, A. Prakash, and T. Mitra. Integrated cpu-gpu power management for 3d mobile games. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6. ACM, 2014.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[8] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The splash-2 programs: Characterization and methodological considerations. In *Computer Architecture, 1995. Proceedings., 22nd Annual International Symposium on*, pages 24–36. IEEE, 1995.