Poster Abstract: Benefits of GPU-CPU Task Replacement for Edge Device and Platform

Cheng-You Lin
Dept. of Computer Science and Information Engineering
National Taiwan Normal University
Taipei City, Taiwan R.O.C.
60747047s@gapps.ntnu.edu.tw

Chao Wang
Dept. of Computer Science and Information Engineering
National Taiwan Normal University
Taipei City, Taiwan R.O.C.
cw@ntnu.edu.tw

ABSTRACT

Contemporary cyber-physical systems (CPS) applications are deployed on a networked platform with embedded devices and, like conventional workstations, each embedded device is now equipped with both CPU and GPU. In this paper, we present our on-going effort of synergizing CPU and GPU computing resources to improve application response time. We experimented on NVIDIA's Jetson Nano embedded device and RTX 2080 Ti graphics card and show that, in particular, with multiple GPU-intensive tasks running, it is possible to improve the application response time by replacing a GPU-intensive task by a corresponding CPU-intensive task. We studied several configurations of CPU-GPU task allocation and replacement, and accordingly we outlined a set of principles in leveraging such heterogeneous resources as a whole.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

AIoT, Heterogeneous, GPU, Response time, Task Replacement

ACM Reference Format:
Cheng-You Lin and Chao Wang. 2021. Poster Abstract: Benefits of GPU-CPU Task Replacement for Edge Device and Platform. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '21), May 18–21, 2021, Charlottesvle, VA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450268.3453505

1 INTRODUCTION

IoT edge computing systems integrate a large number of embedded devices. Contemporary Industrial Internet typically has a three-tier architecture [3], where the *edge tier* controls end points, the *platform tier* controls the intermediate services, and the *enterprise tier* hosts applications. Under this context, the edge computing servers are in the platform tier and edge devices are in the edge tier. The platform tier receives many messages from the edge tier, and processing these data for applications is the most important task of the edge server. The platform tier has more computing power and better hardware specification than the edge tier. The data collected by the edge tier is valuable for big data analysis.

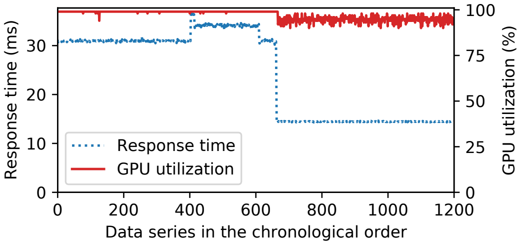

Figure 1: Image recognition by GoogleNet.

The parallel computing capability of GPU can improve the response time of IoT applications, and the computing is usually performed at the GPU-equipped platform tier and the result will be sent back to the edge tier for application execution. Currently, there are edge devices equipped with GPUs and by leveraging them the system may further improve operation efficiency. For example, NVIDIA has developed a series of GPU-equipped embedded development boards, which enable applications such as image processing and AI to achieve better performance on edge devices using GPU.

In this paper, we report our study on the use of NVIDIA Jetson Nano GPU-equipped embedded board in the edge tier. We studied the response time of computing at the platform tier and the edge tier, respectively, as well as the differences and characteristics of computing with GPUs and CPUs.

1.1 Motivations of Task Replacement

For some IoT applications, their goals can be achieved by using different computing resources. For example, suppose the goal is to identify objects in a certain image. Such an application can be implemented as either a CPU-intensive or GPU-intensive task. The GPU-intensive task may utilize a CNN (Convolutional Neural Networks) for object detection, while the CPU-intensive task may utilize an OpenCV cascade classifier.

It suggests that, unlike CPU, a GPU may accelerate task execution by utilizing spare computing cycles. We observed that running multiple GPU-intensive tasks at the same time will multiply the response time of each task, and by removing some of them we may greatly improve the response time of the rest. Figure 1 shows an empirical result of repeatedly running two identical tasks using GoogleNet [5]. We terminated one of the tasks after ~650 repetitions and observed that while the response time of the remaining task is reduced by half, the GPU utilization remains above 90%.

IoTDI '21, May 18–21, 2021, Charlottesvle, VA, USA

ACM ISBN 978-1-4503-8354-7/21/05…$15.00
https://doi.org/10.1145/3450268.3453505

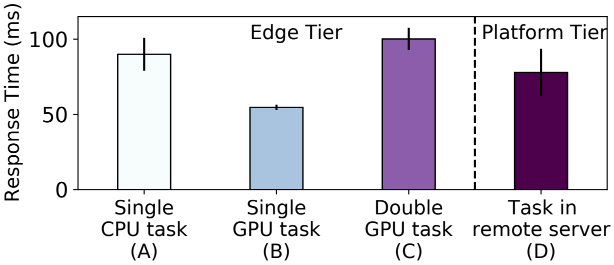

Figure 2: Response time of different type.

Based on the above observations, we have been exploring the possibility and performance of CPU-GPU task replacement and offloading to improve the response time of applications: With different types of tasks at the edge tier and the platform tier, why and when should we configure the system to replace GPU tasks by CPU tasks (or vice versa) or to offload tasks from the edge tier to the platform tier (or vice versa)? In the following, we report the current result of our study.

2 EMPIRICAL STUDY

We mounted NVIDIA Jetson Nano on an embedded three-wheel vehicle and run GPU-intensive/CPU-intensive tasks for a AIoT application. The Jetson Nano is wirelessly connected to an Ubuntu Linux workstation as a platform-tier server. At the edge tier, the Jetson Nano has a 128-core 921MHz GPU and ARM A57 1.43 GHz CPU. At the platform tier, the Linux workstation has RTX 2080 Ti GPU and Intel® Core™ i9-9900KF 5.00 GHz CPU.

We evaluated the following four task configurations: (A) one CPU-intensive task running OpenCV face recognition, (B) one GPU-intensive task running a SSD algorithm [4] for camera object recognition, (C) double GPU tasks, one running the SSD and another running GoogleNet [5] image recognition, and (D) one CPU-intensive task running the SSD in the Ubuntu workstation at the platform tier.

2.1 Key Observations

Figure 2 show the empirical response time using each configuration. For the case of double GPU tasks, we recorded the response time of the task running the SSD. For each configuration we repeated the experiment for one thousand times and we plotted the 95% confidence interval. Here we list four observations:

1. (*B and C vs. other configurations*) Using single GPU task we may have the fastest response time among all configurations. But if the GPU is running another task at the same time, the response time doubled and is the slowest among all configurations. This also aligns with the observation from Figure 1.
2. (*A vs. C*) The response time of a single CPU task is faster than that running double GPU tasks. This shows the benefit of task replacement: if there are two GPU-intensive tasks, one may get some response time improvement by replacing a GPU-intensive task to a corresponding CPU-intensive task.
3. (*A vs. D*) The response time of a single CPU task is slower than that of the task running in the remote workstation. When the server can afford it, the remote server CPU computing can achieve a faster response time.
4. (*D vs. B and others*) The response time of task execution in the remote workstation is faster than all the rest except the single-GPU-task configuration. Note that the response time of remote execution includes the round-trip networking delay.

2.2 Design Principles

We observe that using the edge-tier GPU one may get the best response time, under the condition that the GPU is processing a single intensive task; for multiple GPU-intensive tasks, it is preferable to either replace some of them by corresponding CPU-intensive tasks or offloading them to a remote server in the platform-tier. This may give two benefits. Firstly, the replacement will benefit the application that now is implemented by a CPU task; secondly, the replacement will also benefit the application that uses the remaining GPU task, as suggested by the observations from Figures 1 and 2.

If one needs to deploy a new application to the system, it is advantageous to consider edge-tier CPU-GPU task replacement, and one should also consider utilizing the platform-tier resources, with a grain of salt: One reason why the remote computing may have a faster response time than the single edge-tier CPU task in this work is the stable local network. In other networking configuration the result may be different.

3 DISCUSSION AND FUTURE WORK

So far, we found that work replacing and work offloading are feasible. The most important point is to avoid the edge-tier GPU having multiple tasks. It is shown that leveraging heterogeneous computing resources for multiple applications may improve application response time. Our study illustrated a limitation of edge-tier GPU processing and shows that despite the current development of GPU-equipped edge devices, the platform-tier still plays an important role. Now we are exploring the impact of networking latency to the application response time, as well as trade-offs between using platform-tier computing resources and edge-tier resources.

ACKNOWLEDGMENTS

This work is supported by MOST grant 109-2222-E-003-001-MY3. We would like to thank reviewers to help improve this work.

REFERENCES

[1] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y Zomaya. 2020. Edge intelligence: the confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal* 7, 8 (2020), 7457–7469.

[2] Keke Gai, Meikang Qiu, Hui Zhao, and Xiaotong Sun. 2017. Resource management in sustainable cyber-physical systems using heterogeneous cloud computing. *IEEE Transactions on Sustainable Computing* 3, 2 (2017), 60–72.

[3] Industrial Internet Consortium. 2019. Industrious Internet Reference Architecture. (Jun 2019). https://www.iiconsortium.org/IIRA.htm

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science* (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.