Poster Abstract: ECSRL: A Learning-Based Scheduling Framework for AI Workloads in Heterogeneous Edge-Cloud Systems

Changyao Lin
Harbin Institute of Technology
Harbin, China
20S003095@stu.hit.edu.cn

Ziyang Zhang
Harbin Institute of Technology
Harbin, China
20B903026@stu.hit.edu.cn

Huan Li
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
huanli@hit.edu.cn

Jie Liu
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
jieliu@hit.edu.cn

ABSTRACT

Recent advances in both lightweight models and edge computing make it possible for inference tasks to be executed concurrently on resource-constrained edge devices. However, our preliminary experiments show that the execution of different lightweight models on edge devices may lead to a performance downgrade. In this paper, we propose a Learning-Based Scheduling Framework—ECSRL, to optimize the latency and power consumption for those inference tasks running in heterogeneous Edge-Cloud systems.

CCS CONCEPTS

• **Computing methodologies** → **Planning and scheduling**.

KEYWORDS

Heterogeneous Edge Computing; Task Scheduling; Reinforcement Learning

ACM Reference Format:
Changyao Lin, Ziyang Zhang, Huan Li, and Jie Liu. 2021. Poster Abstract: ECSRL: A Learning-Based Scheduling Framework for AI Workloads in Heterogeneous Edge-Cloud Systems. In Proceedings of *The 19th ACM International Conference on Embedded Networked Sensor Systems (SenSys), Nov 15-17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3485730.3492886

1 INTRODUCTION

As a new computing paradigm, edge AI [1] and mobile edge computing (MEC) [2] sink computing power from the cloud to edge AI devices, making it possible to inference Deep Learning (DL) workload in real-time on the edge. Compared with cloud computing cluster, edge devices have the advantages of low latency, low power consumption, low price, and easy deployment, etc. The advantages and disadvantages of both are summarized in Table 1 below.

SenSys '21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11…$15.00
https://doi.org/10.1145/3485730.3492886

Table 1: Edge Device versus Cloud Server

Paradigm	Delay	Accuracy	Power	Price
Edge + Light Model	**Low**	Low	**Low**	**Low**
Cloud + Large Model	High	**High**	High	High

In order to understand the performance when multiple DL-based workloads run concurrently on an edge device, we set up a test-bed and found that the completion time of newly arrived AI task will be affected under concurrent situation. Also, as mentioned in the reference, most of the existing cluster scheduling algorithms are either based on simulation or homogeneous virtual machine clusters [3]. Therefore, it is essential to rethink to design efficient task scheduling algorithms for edge clusters with heterogeneous devices.

In this paper, we propose to adopt reinforcement learning strategy to design a task scheduling framework for heterogeneous Edge-Cloud systems. The objective is to minimize the power consumption and completion time. Meanwhile, if the lightweight DL workload deployed on the edge device could not meet the accuracy requirement, the system will automatically offload the task to the cloud to run a high-precision model for a better inference result.

2 EXPERIMENTAL STUDY

To measure the performance of concurrent workloads execution at edge devices, in our test-bed, two lightweight models, ResNet-18 and SSD-MobileNet-v2 are deployed on NVIDIA Xaiver NX.

Figure 1 shows the performance matrix for ResNet-18 and SSD-MobileNet-v2 on edge device NX respectively. Here, m denotes the number of SSD-MobileNet-v2, r denotes the number of ResNet-18. The elements in matrix represent the ratio of the running time of the added workload concurrently with other existing workloads divided by the time when the system only executes a single such workload. For example, in Figure 1(a), the value of [1][0] represents the ratio of the inference time when ResNet-18 is deployed on NX that has already runs 1 SSD-MobileNet-v2 and 0 ResNet-18 over the inference time when ResNet-18 is deployed on the device NX alone. Since the time to complete a single inference task on a specific device is a fixed value, in this figure, the larger values indicate the performance degradation (longer completion time) as more tasks added into the system.

m \ r	0	1	2	3	4	5
0	1.000	1.277	1.754	2.252	2.762	3.155
1	1.274	1.754	2.227	2.717	3.155	∞
2	1.742	2.242	2.747	3.268	∞	∞
3	2.217	2.732	3.226	∞	∞	∞
4	2.732	3.215	∞	∞	∞	∞
5	3.185	∞	∞	∞	∞	∞

(a) ResNet-18 on NX

m \ r	0	1	2	3	4	5
0	1.000	1.517	2.232	2.899	3.636	4.255
1	1.504	2.183	2.857	3.584	4.255	∞
2	2.132	2.801	3.534	4.219	∞	∞
3	2.793	3.559	4.202	∞	∞	∞
4	3.460	4.065	∞	∞	∞	∞
5	4.065	∞	∞	∞	∞	∞

(b) SSD-MobileNet-v2 on NX

Figure 1: Performance degradation matrix of ResNet-18 and SSD-MobileNet-v2 on NX

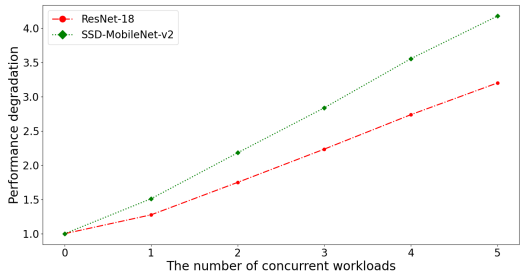

Figure 2: Performance degradation curve

The infinite value indicates that the current workload concurrency has exceeded the resource limit of the edge device (such as memory), and the scheduling algorithm will not schedule the current task to the edge device any more.

We also draw Figure 2 to illustrate the relationship as the number of concurrent workloads increases. Here, X-axis is the number of concurrent workloads, Y-axis is the average performance degradation of workloads of the same concurrent number in the performance degradation matrix obtained in Figure 1. That is, the average of all values in the same color. There are two interesting observations: 1) there exists a linear relationship between the number of workloads and the performance degradation; 2) different arrival task may have different downgrading performance even under the same system settings. These results can be used to guide our scheduling algorithm design.

3 LEARNING-BASED SCHEDULING FRAMEWORK

According to the experimental results, we propose a learning-based scheduling framework, ECSRL for scheduling tasks that execute on multiple heterogeneous edge devices. Figure 3 shows the overall architecture of ECSRL.

The performance objective is to minimize the average performance degradation and power consumption when multiple deep learning workloads are concurrently running on edge nodes. Meanwhile, the resource utilization of the cluster should be improved as much as possible. The objective function is defined as follow:

$$\min \sum_{t=0}^{T} (\alpha E_t + \beta \bar{d}_t) \tag{1}$$

In Formula 1, E_t and $\bar{d}_t$ represent the total power consumption and average performance degradation of the inference request completed in step t, respectively; α and β are adaptive weights.

ECSRL consists of three parts: Access Point (AP), Edge Cluster, and Cloud Server. The end devices will generate a series of requests of different types of inference tasks at AP. Each requests contains a deadline, accuracy requirement, service level agreement. Those tasks will randomly arrive at the AP in a k3s edge cluster (where AP and several edge devices form a k3s edge cluster).

Task scheduling: the role of AP is to receive requests, execute Learning-Based scheduling algorithm, and then dispatch requests to an edge device for inference. DL workloads on edge devices run as containers. Porting container technology to edge computing can naturally shield hardware differences and bring great convenience in deployment and management.

Task offloading: when the inference accuracy on edge node fails to meet the task requirement, the node will offload the task to the large model in the cloud for secondary inference. This task offloading mechanism can solve insufficient inference accuracy problem.

In ECSRL, we use the Actor-Critic algorithm as the scheduling strategy, and set 17,000 requests to arrive randomly within 5 minutes. Experimental results show that the throughput rate will reach more than **87**%, which demonstrates that the Learning-Based scheduling strategy has great potential to efficiently process task requests in a heterogeneous edge-cloud systems.

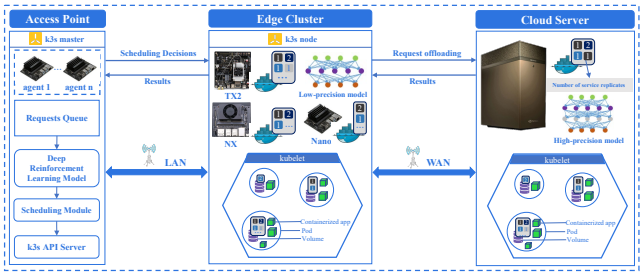

Figure 3: The system framework of ECSRL

4 DISCUSSION AND FUTURE WORK

We have implemented a prototype to prove the effectiveness of above proposed idea. Specifically, several edge devices with different computing ability, including NVIDIA Jetson Nano, NVIDIA Jetson TX2 and NVIDIA Xaiver NX, are selected in the system. Here we use Nano as the AP. NVIDIA GeForce RTX 3080×4 will be used as the cloud device to execute high-precision models. At present, we are starting with Single-Agent Reinforcement Learning on a single cluster. In the future, we will expand to multiple clusters and introduce Multi-Agent Reinforcement Learning Algorithm to expand the scale.

REFERENCES

[1] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen. 2019. Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications* 19, 1 (2019), 447–457.

[2] Pavel Mach and Zdenek Becvar. 2017. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1628–1656.

[3] Deepak Narayanan, Keshav Santhanam, Fiodar Kazhamiaka, Amar Phanishayee, and Matei Zaharia. 2020. Heterogeneity-aware cluster scheduling policies for deep learning workloads. In *14th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 20)*. 481–498.