An Adaptive Spatial-Temporal GPU Scheduling in Edge Cloud Computing Environment*

Taewoo Kim[0000−0003−4290−6460], Tuan Manh Tao[0009−0007−8317−2113], Khac Tuyen Dinh[0009−0004−2558−6404], Minsu Jeon[0000−0002−2739−8149], Changha Lee[0000−0003−3687−2989], and Chan-Hyun Youn[0000−0002−3970−7308]

Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
{taewoo_kim,ttmanh,tuyen.dk,msjeon,changha.lee,chyoun}@kaist.ac.kr

1 Proposed Heuristic Spatial-Temporal Scheduling with adaptive GPU Provisioning

Edge cloud computing with GPUs enables low-latency real-time applications. An efficient GPU scheduler optimizes hardware utilization and meets service latency objectives (SLOs). NVIDIA's Multi-Process Services (MPS) [1] enables concurrent execution of multiple DNN tasks.

In this work, we propose a hybrid Spatial-Temporal GPU scheduling approach with adaptive GPU partitioning to address the heterogeneous workload demand. Our approach introduces a GPU adaptive spatial partitioning scheme to improve GPU utilization and minimize GPU resource reconfiguration overhead expressed through Fig. 1.

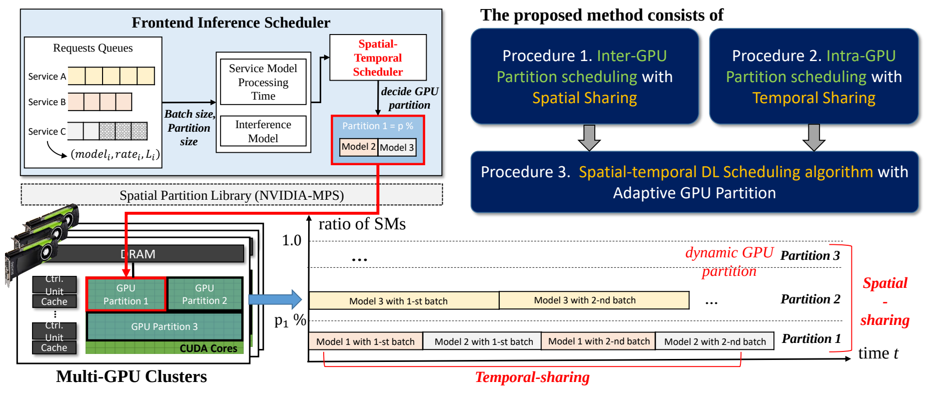

Fig. 1. An architecture of the proposed spatial-temporal sharing approach with adaptive GPU partitioning.

The objective of the GPU partitioning strategy is to optimize the overall throughput of the system. To achieve this, the scheduler prioritizes models with

* This work is supported by Samsung Electronics Co., Ltd.

high request rates by determining the most efficient GPU partition size for them. Upon collecting statistics on the service models that need to be rescheduled, the scheduler employs a greedy approach. It sorts the models to be rescheduled based on their request rate, starting with the highest request rate model. For each scheduled model, the scheduler identifies the ideal GPU partition size.

2 Experiments and Disscusion

We select 3 inference workloads: Resnet18, Resnet50[2], and VGG16[3] at a request rate of 50 requests/s with the starting time stamp 0, 50, and 100, respectively. This simulation confirms the efficiency of the GPU dynamic partitioning algorithm in preserving SLOs for concurrent services with temporal sharing.

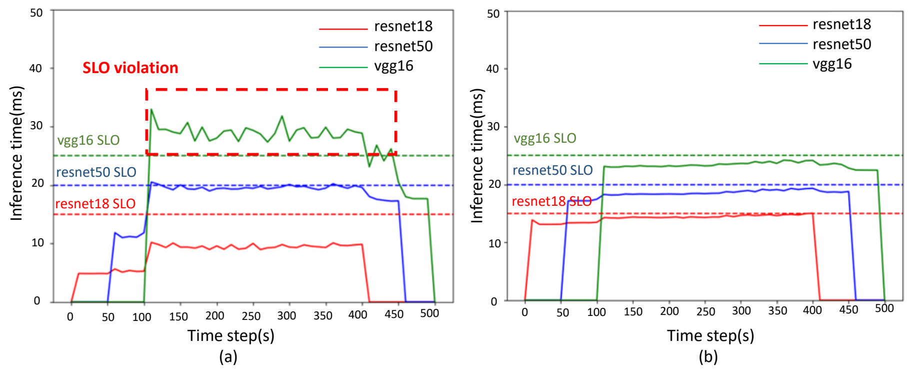

Fig. 2. Concurrent models inference time when (a) running with Default MPS as baseline and (b) proposed GPU provisioning method during 500 sec.

Fig 2 (b) shows GPU scheduler's adaptive resource partitioning for concurrent service models. Consistent inference latency is achieved throughout, demonstrating performance isolation. In contrast, the default MPS task scheduler (a) fails to ensure isolation, causing unpredictable latency fluctuations during co-execution, especially during peak periods (time step 100-400).

References

1. NVIDIA Multi-Process Services,https://docs.nvidia.com/deploy/mps/index.html. Last accessed 15 May 2023
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 770–778
3. Simonyan, K, Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.