
POSTER:POWERING MULTI-TASK FEDERATED LEARNING WITH COMPETITIVE GPU RESOURCE SHARING

ABSTRACT

Federated learning (FL) nowadays involves heterogeneous compound learning tasks as cognitive applications' complexity increases. For example, a self-driving system hosts multiple tasks simultaneously (e.g., detection, classification, segmentation, etc.) and expects FL to retain life-long intelligence involvement. However, our analysis demonstrates that, when deploying compound FL models for multiple training tasks on a GPU, certain issues arise: (1) As different tasks' skewed data distributions and corresponding models cause highly imbalanced learning workloads, current GPU scheduling methods lack effective resource allocations; (2) Therefore, existing FL schemes, only focusing on heterogeneous data distribution but runtime computing, cannot practically achieve optimally synchronized federation. To address these issues, we propose a *full-stack* FL optimization scheme to address both *intra-device* GPU scheduling and *inter-device* FL coordination for multi-task training. Specifically, our works illustrate two key insights in this research domain: (1) Competitive resource sharing is beneficial for parallel model executions, and the proposed concept of "virtual resource" could effectively characterize and guide the practical per-task resource utilization and allocation. (2) FL could be further improved by taking architectural level coordination into consideration. Experiments show that we could greatly enhance the GPU resource utilization, and in turn improve the overall intra-device training throughput by $2.16\times\sim 2.38\times$ and inter-device FL coordination throughput by $2.53\times\sim 2.80\times$ in complex multi-task FL scenarios.

1 INTRA-DEVICE GPU SCHEDULING WITH COMPETITIVE RESOURCE SHARING

We propose a GPU computational scheduling method with competitive resource sharing.

Fig. 1 ① indicates a fully isolated spatial resource allocation, which is a very recent GPU scheduling technique to resolve resource interference (e.g., MPS (Nvidia, 2021)).

Instead of fully spatial isolation, we could also enable different tasks to share certain resources and enable the competitive sharing. This could be done by assigning virtual resources to each individual task, which could accumulate to exceed 100% physical GPU resources. The exceeded resources come from the resource sharing between tasks, as shown in Fig. 1 ②.

However, excessive shared resources can also bring resource contention, which undermines the advantage of resource sharing. Fig. 1 ③ gives an example of this case. The last case is fully sharing the GPU resource without partitioning, which leads to extreme competitions and results in fully sequential execution in certain cases. In Fig. 1 ④, *task-A* and *task-B*, when running large-volume operators, the GPU scheduler can alter the spatial sharing-based execution to the time slicing-based execution, in which one task will take all the resources in its duration.

Through identifying competitive resource sharing, we find controlling an appropriate degree of resource competition

and sharing is the key to achieve the optimal GPU performance.

When using competitive resource sharing to deploy complex multi-tasks in GPU, we propose a machine learning approach to estimate throughput based on a given tasks' combination of and the virtual resource allocation for each task (Yeung *et al.*, 2021).

2 INTER-DEVICE MULTI-TASK FEDERATED LEARNING COORDINATION

Based on the intra-device virtual resource management, we further bring it into the inter-device FL cluster and rethink the FL coordination from a GPU scheduling perspective.

Our goals are to make each device could be fully utilized during each synchronization cycle when multi-task models parallel in each device, and meanwhile maximize the overall GPU throughput to accelerate the overall FL training speed.

We achieve the first goal through adjusting mismatch between the ratio of different tasks' data volume D and the ratio of different tasks' workload O . For the second goal, we adjust the resource allocation according to the workload which is influenced by batch size to obtain the maximum throughput. And the goals can be formulated by the follow-

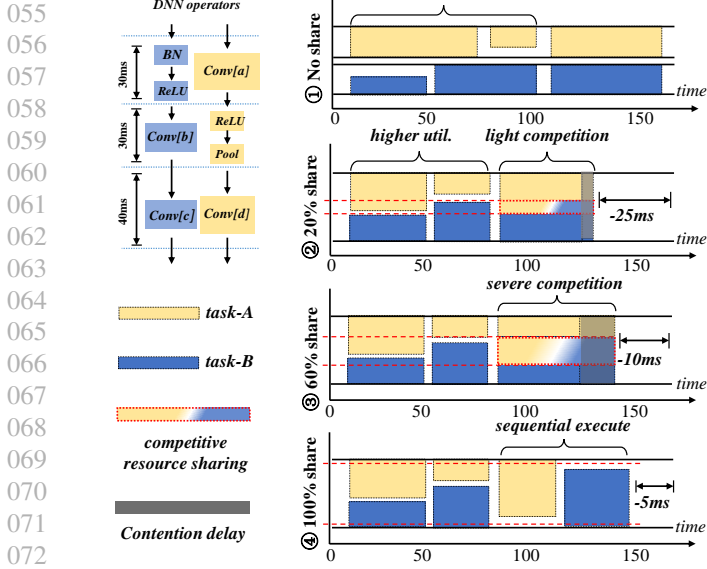


Figure 1. Competitive GPU Resource Sharing with Multi-Task DNN Training

ing objectives:

$$\begin{cases} \text{Objective 1: } \min \sum_i \sum_j \frac{|D_i|}{|D_j|} - \frac{o_i}{o_j}, \\ \text{Objective 2: } \max \sum_i P_1, \dots, P_i. \end{cases} \quad (1)$$

For the first objective, we can understand its principle using the relation between D and O . When a task model has larger data volume D , we improve its workload through increasing batch size to make this task model occupy more resource, so that this task model can consume correspondingly more amount of data during each synchronization cycle. Therefore, achieving co-scheduling of the multi-task FL coordination can be transformed into leveraging the workload adjustment and resource allocation to satisfy the above two objectives. We adopt a greed optimization method to find the optimal workload and resource allocation.

3 EXPERIMENTAL EVALUATION

Experimental Setup: We construct various multi-task scenarios with following DNN models: VGG16 (V16), ResNet18 (R18), ResNet50 (R50), ResNet101 (R101) MobileNet_v3 (M3), ShuffleNet_v2 (S2) DensNet121 (D121). We evaluate three multi-task settings on the CIFAR10 dataset and use the NVIDIA Titan V GPU.

We use raw throughput and fairness throughput to evaluate the performance. All methods' throughput are normalized to show the relative acceleration ratio.

Overall Speed-up: Our resource allocation method could consistently yield $2.16\times$ to $2.38\times$ speed-up. **Inter-Device**

Multi-task FL Coordination: We use a FL system with

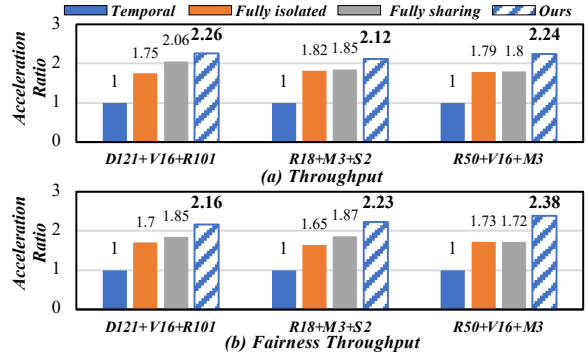


Figure 2. Throughput Advantages.

several devices, each devices have three tasks with different model structure and imbalance data volume. We give the the average throughput and fairness throughput in one synchronization cycle of all devices, and the results show in the Fig. 3. From the Fig. 3, we can find that we achieve $2.53\times$ to $2.80\times$ speed-up compared to the baseline methods.

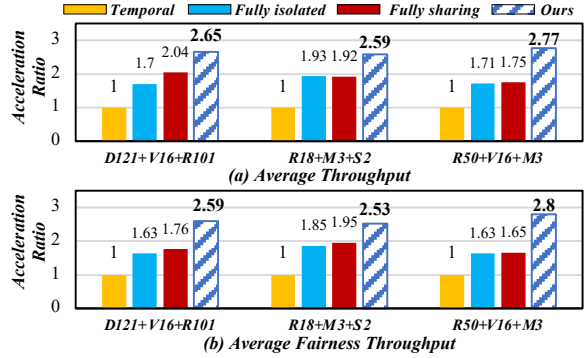


Figure 3. Inter-Device Average Throughput in a Federated Learning Synchronization Cycle

4 CONCLUSION

In this work, we propose a full-stack multi-task FL optimization scheme, which addresses both intra-device GPU scheduling with a novel competitive resource sharing scheme; and inter-device multi-task FL coordination with realistic GPU runtime synchronization. Experiments show that we could greatly enhance the GPU resource utilization, and improve the overall training throughput.

REFERENCES

- Nvidia. Multi-Process Service, 2021. URL https://docs.nvidia.com/deploy/pdf/CUDA_Multi_Process_Service_Overview.pdf.
- Yeung *et al.*, G. Horus: Interference-aware and prediction-based scheduling in deep learning systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(1):88–100, 2021.