MISO: Exploiting Multi-Instance GPU Capability on Multi-Tenant GPU Clusters

Baolin Li, Tirthak Patel, Siddharth Samsi, Vijay Gadepally, Devesh Tiwari
GPUs are everywhere in the cloud..
...but, they are severely underutilized

The state-of-the-art deep learning models utilize less 50% of the GPU resources on modern A100 GPUs and utilization varies significantly over run time.
...but, they are severely underutilized

Up to 50% of the GPU jobs may have less than 25% utilization on multi-tenant clusters
What is a potential solution?

GPU resource sharing allows better utilization
GPU Resource Sharing Allows Better Utilization

Multi-Process Service (MPS)  Multi-Instance GPU (MIG)

[4] https://docs.nvidia.com/deploy/mps/index.html
[5] https://docs.nvidia.com/datacenter/tesla/mig-user-guide/index.html
MPS and MIG Sharing Mode Trade-Offs

Software-based logical partition
Flexible
No perf isolation

Hardware-based physical partition
Limited granularity
Interference-free
Multi-Instance GPU (MIG) on NVIDIA GPUs

Different MIG slices on an A100 GPU

| Slice | Compute | Memory | Cache | Max Count |
|-------|---------|--------|-------|-----------|
| 7g.40gb | 7 GPC | 40 GB | Full | 1 |
| 4g.20gb | 4 GPC | 20 GB | 4/8 | 1 |
| 3g.20gb | 3 GPC | 20 GB | 4/8 | 2 |
| 2g.10gb | 2 GPC | 10 GB | 2/8 | 3 |
| 1g.5gb | 1 GPC | 5 GB | 1/8 | 7 |

[5] https://docs.nvidia.com/datacenter/tesla/mig-user-guide/index.html
Challenges in GPU Resource Partitioning

Brief experimental insights and motivation
Observation 1. Compared to MPS, MIG-based partitioning is more promising, but challenging

MIG’s interference-free partitioning provides an opportunity for higher performance than MPS’s interference-prone partitioning

Optimal GPU resource partitioning using MIG slices varies significantly across job mixes
Observation 1I. Determining effective MIG-based partitions incurs higher overhead

Determining the optimal MIG partition configuration for a job-mix, requires knowing individual job’s speedup on all different MIG slices.

But profiling the performance speedup for all jobs on every MIG slice in the MIG mode causes prohibitive checkpoint-restart overhead, unlike the MPS-mode.

A job-mix four jobs requires exploring multiple MIG configurations
MISO leverages the flexible but interference-prone MPS-based partitions to find the optimal MIG-based (interference-free) partitions to achieve higher performance for multi-tenant GPUs.

MISO leverages best of the both the worlds (MPS and MIG): MPS for profiling and performance estimation, MIG for interference-free resource partitioning.
Overview of the MISO Design

MISO uses lightweight MPS-mode run to quickly estimate jobs’ performance on different MIG configurations using a machine learning model, and then partition the GPU resources intelligently.
**MISO’s Job MIG Performance Estimator Using MPS mode**

Observation: Under MPS-mode, one can adjust GPU sharing levels for concurrently jobs in a job mix without frequently switching jobs in and out of the GPU.

MISO uses this flexibility to estimate performance on different MIG slices.

| MPS Level | Job1 | Job2 | ⋅⋅⋅ | Job7 |
|-----------|------|------|------|------|
| 100%      | \(A_{11}\) | \(A_{12}\) | ⋅⋅⋅   | \(A_{17}\) |
| 50%       | \(A_{21}\) | \(A_{22}\) | ⋅⋅⋅   | \(A_{27}\) |
| 14%       | \(A_{31}\) | \(A_{32}\) | ⋅⋅⋅   | \(A_{37}\) |

MPS Slice | Job1 | Job2 | ⋅⋅⋅ | Job7 |
|-----------|------|------|------|------|
| 7g        | \(B_{11}\) | \(B_{12}\) | ⋅⋅⋅   | \(B_{17}\) |
| 4g        | \(B_{21}\) | \(B_{22}\) | ⋅⋅⋅   | \(B_{27}\) |
| 3g        | \(B_{31}\) | \(B_{32}\) | ⋅⋅⋅   | \(B_{37}\) |
Train a U-Net variant to translate the MPS performance into MIG performance.

The 2g and 1g MIG slices can be extrapolated from 7g, 4g, 3g MIG performance.
MISO’s MIG Partition Optimizer

MISO quickly finds the optimal MIG partition without heuristics
Focuses on optimizing each GPU locally
Avoids overhead from the global NP problem
Avoids extra job checkpointing between GPU nodes
MISO: Evaluation and Insights
Experimental Methodology

**Metrics**
- Average Job Completion Time (JCT)
- Makespan
- System Throughput

**Setup**
- 4-node system
- 2 AMD EPYC 7542 CPUs each node
- 2 NVIDIA A100 GPUs each node

**Workloads**
- Helios Trace [6] (SC'21)
- Poisson distributed arrival
- Deep learning workloads including BERT, GNN, CycleGAN.

**Schemes**
- NoPart: no partition
- OptSta: optimal static MIG partitioning
- ORACLE: knows MIG speedup for every job

for $m$ jobs $J_1$ to $J_m$, suppose job $J_i$’s execution speed on an A100 GPU without co-location is $p_i$, and its current execution speed is $q_i$.

System Throughput (STP) = $\sum_{i=1}^{m} \frac{q_i}{p_i}$

[6] Hu, Q., Sun, P., Yan, S., Wen, Y. and Zhang, T., 2021, November. Characterization and prediction of deep learning workloads in large-scale gpu datacenters. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-15).
MISO offers significant improvements

Over 30% improvement in job completion time, makespan and system throughput

Where does MISO performance improvements come from?
MISO outperforms across different scenarios

Robust to different initial conditions

Robust to model prediction errors

Robust to different job-arrival rates
MISO is the first method for GPU resource partitioning on a MIG-enabled multi-tenant GPU cluster.

MISO combines the best of both worlds (MPS and MIG). MISO uses the lightweight MPS profiling to quickly estimate the optimal MIG partition without the excessive overhead to profile each job’s MIG slice performance.

MISO provides significant improvement over unpartitioned GPU cluster and close to oracle-partitioned GPU cluster.

Contact
Baolin Li
li.baol@northeastern.edu