Please **contact Tanvi (th2720@columbia.edu)** for questions & comments

Scheduling jobs on Multi Instance GPUs (MIG) using RL

Tanvi Hisaria, Devyani Vij

Context:

- NVIDIA's multi-instance GPUs (MIG) process AI/ML workloads and can be split into several (up to 7) slices with various configurations to run multiple jobs concurrently.
- We want to schedule jobs with varying characteristics on these GPUs in an online setting. We can generate jobs and job characteristics (such as runtime on different sized slices, deadlines) using simulations.

Objective:

- Optimize for a weighted measure of energy and tardiness as introduced in [1], a metric called ET, where:
  - Tardiness is a measure of how much past the deadline the job finishes running, and
  - Energy efficiency can be optimized by full GPU utilization, where power to use 4/7 is equivalent to using 7/7.
  - The key is the tradeoff here—we achieve energy efficiency by packing the GPU as much as possible, but putting more jobs on smaller slices on the same GPU can increase the time they take to run, and therefore affect tardiness.

[1] E. Lipe, N. Karia, C. Espenshade, C. Stein, A. Tantawi and O. Tardieu, "Energy Efficient Scheduling of AI/ML Workloads on Multi Instance Gpus with Dynamic Repartitioning," 2025 IEEE 25th International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Tromsø, Norway, 2025, pp. 53-62, doi: 10.1109/CCGRID64434.2025.00066.

RL Formulation

- State Space: Characteristics of the job being scheduled, the current queue of jobs waiting to be run with their characteristics, and the configuration of the GPUs and which slices are available.
- Action Space: Assign one job at a time
- Reward: ET (weighted measure of energy and tardiness) [1]
- Goal: Optimize ET better than deterministic greedy algorithms from previous research
- Environment: online setting and simulated (the code for simulating job queues is already written)

Algorithm design:

- RL paradigm: Start with a PPO algorithm on a discrete action space (choosing the next job) and test how reward weights shift the makespan–energy tradeoff.
- Complexity: Will start with above formulation, and then increase complexity in the state space by progressing to multiple GPUs and in the action space by assigning as many jobs in the ready queue as possible to all available slices.
- Evaluation Metrics: From the research done so far, we have deterministic greedy algorithms to schedule jobs which we can use to compare the performance of the model we build.

[1] E. Lipe, N. Karia, C. Espenshade, C. Stein, A. Tantawi and O. Tardieu, "Energy Efficient Scheduling of AI/ML Workloads on Multi Instance Gpus with Dynamic Repartitioning," 2025 IEEE 25th International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Tromsø, Norway, 2025, pp. 53-62, doi: 10.1109/CCGRID64434.2025.00066.